THE MEMORY HORIZON IS THE LEVER: BENCHMARKING ONLINE ACTIVE LEARNING UNDER CONCEPT DRIFT AND LABEL DELAY

Anonymous authors
Paper under double-blind review

ABSTRACT

Active learning (AL) is studied almost entirely as an acquisition problem, and benchmarks on stationary data rank uncertainty sampling as the strategy to beat. In streaming deployment, however, purchased labels arrive after a delay and may describe a concept that has changed by the time they arrive. We study which decision of an online learner determines accuracy under these conditions: which points to query, or how to use the delayed labels once they arrive. We build a factorial benchmark that crosses eleven acquisition strategies with ten update rules on 50 audited temporal streams at three label delays, and propose an online horizon selector that chooses how many recent labels to learn from using only labels already paid for. No tested acquisition strategy improves on random labeling by more than 0.8 points in any cell at delays up to 500 steps, and a positive control shows that the gains acquisition achieves on stationary image features vanish once the relation between features and labels drifts. The update rule moves accuracy by 9 points, and its decisive property is the memory horizon: a buffer of the 50 most recent labels scores 77.5% against 67.9% for an unbounded one, yet loses up to 9 points on real streams whose drift is slow. The selector matches a per-stream oracle at delays up to 500 steps and costs 0.5 points where nothing drifts. Code and results: https://anonymous.4open.science/r/active-learning-cd-iclr27-1342/.

1 INTRODUCTION

Active learning (AL) reduces the cost of supervision by querying labels only for the points a learner expects to be most informative (Settles, 2009). As acquisition scores have matured, from uncertainty (Houlsby et al., 2011; Gal et al., 2017) to optimal design (MacKay, 1992; Huang & Pinson, 2026b) and coverage (Sener & Savarese, 2018), choosing among them has become the field's central question, and a recent tabular benchmark (Lu et al., 2025) concludes that uncertainty sampling with a compatible model remains the strategy to beat. More concretely, benchmarks treat AL as a choice among acquisition strategies, with the model, the memory of past labels, and the timing of labels held fixed (Fig. 1, top left). The settings in which labels are genuinely expensive, such as fraud detection and industrial monitoring, break two of these assumptions. They are streaming: the learner must predict each arrival before any label exists, a purchased label returns only after an annotator has produced it, and the distribution moves in the meantime.

This setting creates a second decision that stationary benchmarks never face. Once labels arrive late and the data drift, a label bought before a change arrives describing a concept that no longer holds (Fig. 1, bottom left). The learner must therefore decide not only which arriving points to query, the *acquisition strategy*, but also how the labels that do arrive enter the model, the *update rule*: how many past labels are kept, whether old ones are down-weighted, and whether a change detector decides when to discard them. We refer to the number of most recent purchased labels that still shape the update as the learner's *memory horizon*, and to labels from a superseded concept as *stale*. The two decisions interact: a query rule that seeks the decision boundary fills the memory with points near a boundary that may have moved. A benchmark must therefore separate two questions. A stationary question asks which acquisition strategy is best with the memory policy fixed. A deployment question asks which of the two decisions a practitioner should spend effort on once labels are delayed and the data drift.

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

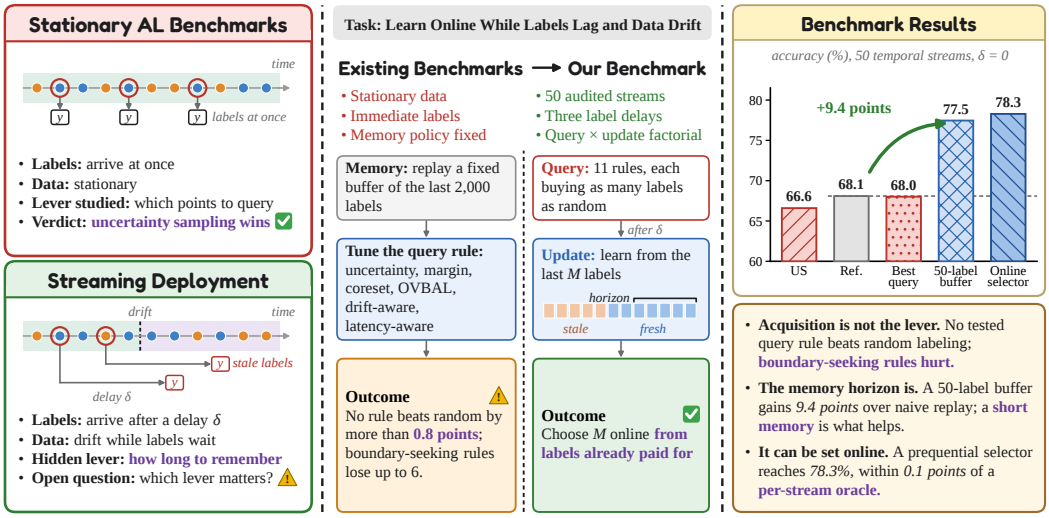

Figure 1: Stationary AL benchmarks tune which points to query, with labels on time and the memory fixed. Once labels lag and the data drift, how long the learner remembers decides accuracy: a 50-label buffer gains 9.4 points over naive replay while no query rule beats random, and the horizon can be chosen online from labels already paid for.

Prior work has approached this problem from several sides without settling it. Stream AL has produced acquisition rules for drifting streams (Žliobaitė et al., 2014) and for verification latency (Pham et al., 2022; Castellani et al., 2022), and these papers observe that latency erodes the value of a query. The update side has been developed by communities that rarely meet AL: concept-drift research supplies detectors (Gama et al., 2004; Bifet & Gavaldà, 2007; Page, 1954) and adaptive windows (Gama et al., 2014; Lu et al., 2019), and continual learning supplies replay methods built to retain old knowledge (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019). These lines target different objects: the value of a query under latency with the memory policy fixed, the timing of adaptation with the query rule fixed, or the retention of old tasks. We ask a different question. Specifically, when both decisions are free, which one determines prequential accuracy under drift and delay, and what property of it matters? No prior comparison crosses the two, so the answer cannot be read off the literature.

Motivated by the above questions, we build a factorial benchmark for cost- and delay-aware online AL (Fig. 1, middle). It crosses eleven acquisition strategies, six stationary, three built for drift, and two built for label latency, with ten update rules on 50 temporal streams, 22 real and 28 from drift generators in three realizations, at label delays of 0, 500, and 2,000 steps. Every real stream passes a label-order audit that removes files whose order is a class block rather than time, every feature is standardized causally, and each comparison is reported with average ranks, corrected signed-rank tests, and bootstrap intervals. Because the benchmark exposes the memory horizon as the decisive property, we also propose an online horizon selector that runs buffers of several lengths on the same purchased labels and predicts with the one whose error on those labels is lowest, so the horizon is chosen from labels the learner has already paid for.

We validate these findings with controlled studies and a measured decomposition. No tested acquisition strategy improves on random labeling by more than 0.8 points in any of the 132 cells of the factorial at delays up to 500 steps, and a positive control on image features shows that the gains acquisition does achieve on a stationary stream disappear once $p(y \mid x)$ changes. The memory horizon moves accuracy by 9 points (Fig. 1, right) and explains why short buffers, recency sampling, detector resets, and A-GEM all help, yet the same short horizons lose up to 9 points on real streams whose drift is slow, which the online selector avoids at delays up to 500 steps.

Key contributions.

- We formulate cost- and delay-aware online AL as two separable decisions, acquisition and update, and build an audited factorial benchmark that crosses them on 50 temporal streams at three label delays.
- We show that no tested acquisition strategy, including the latency-aware ones, improves on random labeling by more than 0.8 points at delays up to 500 steps, and use a positive control with a compatible learner to locate the loss of acquisition's advantage in changes of $p(y \mid x)$.
- We identify the memory horizon as the decisive property of the update rule, give a staleness decomposition whose alignment assumption we measure, and map where short horizons help and where they fail.
- We propose an online horizon selector driven by the prequential error on the purchased labels, which matches a per-stream oracle at delays up to 500 steps and costs 0.5 points on shuffled streams.

2 Problem formulation

We first define the learning protocol and the quantity we evaluate. A stream presents inputs $x_1, x_2, \ldots$ with labels $y_t \in \{1, \ldots, C\}$ drawn from a time-varying distribution P_t. At step t the learner emits a prediction $\hat{y}_t = f(x_t; W_t)$ before any label for x_t exists and may pay unit cost to query y_t. A queried label arrives only at step $t + \delta$, where $\delta \geq 0$ is the label delay, and the query decisions $q_t \in \{0, 1\}$ respect a budget $b \in (0, 1]$:

$$\tfrac{1}{t} \textstyle\sum_{s \leq t} q_s \leq b \quad \text{for all } t, \qquad \mathcal{L}_t = \{(x_s, y_s) : q_s = 1,\ s + \delta \leq t\}, \tag{1}$$

where $\mathcal{L}_t$ is the set of labels available at step t, each drawn from some P_s with $s \leq t - \delta$. Performance is the prequential (test-then-train) accuracy (Dawid, 1984)

$$\mathrm{Acc} = \frac{1}{N} \sum_{t=1}^{N} \mathbb{1}[\hat{y}_t = y_t], \tag{2}$$

which scores every prediction before its label is used and therefore rewards fast recovery after a change. A learner makes two separable decisions. The acquisition strategy α maps the current head and the arrivals of a micro-batch to the decisions q_t. The update rule u maps $\mathcal{L}_t$ to a buffer $\mathcal{B}_t \subseteq \mathcal{L}_t$ with nonnegative weights $w_{t,i}$ and moves the head along

$$g_t = \frac{\sum_{i \in \mathcal{B}_t} w_{t,i}\, \nabla \ell(W_t; x_i, y_i)}{\sum_{i \in \mathcal{B}_t} w_{t,i}}, \tag{3}$$

estimated on minibatches. The memory horizon of u is the number of most recent labels that carry non-negligible weight in (3): M for a first-in-first-out (FIFO) buffer of capacity M, of the order of h for exponential decay with half-life h labels, and the labels since the last detection for a detector-triggered reset. For a fixed learner, budget, and delay, our target is the accuracy surface $\mathrm{Acc}(\alpha, u)$ over the two decisions, and the question is which decision, and which property of it, moves this surface under drift and delay.

After a change at step τ, let $\varphi_t \in [0, 1]$ denote the share of the weight in (3) that falls on labels drawn before τ. Splitting the sum at τ gives

$$g_t = (1 - \varphi_t)\, g_t^{\mathrm{new}} + \varphi_t\, g_t^{\mathrm{old}}, \tag{4}$$

where g_t^{new} and g_t^{old} are the weighted mean gradients over fresh and stale labels. The delay fixes how long φ_t stays at 1, and the memory horizon fixes how fast it falls afterward (Section 3.4).

3 Benchmark and online horizon selection

Our benchmark evaluates the two decisions of Section 2 jointly on a shared learner. Every method uses a linear–softmax head on causally standardized features, trained by SGD with one global schedule: arrivals are processed in micro-batches of L=100, the acquisition strategy queries $\lceil bL \rceil$ labels per micro-batch, and a purchased label enters the buffer at the end of the micro-batch in which it becomes available, so the effective delay is δ rounded up to the next micro-batch boundary. The

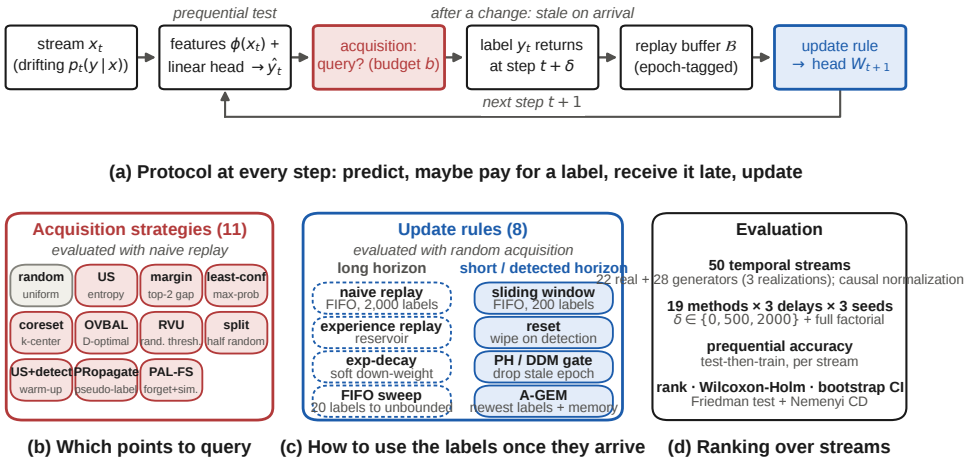

Figure 2: Overview of the benchmark. (a) Per-step protocol with delayed labels; (b) the eleven acquisition strategies; (c) the update rules, grouped by memory horizon; (d) evaluation over streams, delays, and seeds.

head then takes eight SGD steps (learning rate 0.3) on minibatches of 64 drawn from the buffer. Only the acquisition strategy and the update rule vary (Fig. 2); every strategy queries as many labels as random from the same micro-batches, and no method is tuned per stream. Appendix D.3 gives the loop of one benchmark cell (Fig. 9). Sections 3.1 and 3.2 describe the streams and methods, Section 3.3 the online horizon selector, and Section 3.4 why the memory horizon matters.

3.1 Streams and label-order audit

We start from 54 streams. The 26 real streams comprise elec2 (Harries, 1999), Covertype, six Insects variants with abrupt, gradual, incremental, and reoccurring drift (Souza et al., 2020), three streams with documented drift from independent sources (NOAA weather (Elwell & Polikar, 2011); Rialto (Losing et al., 2016); Outdoor (Losing et al., 2015)), the UCI gas-sensor drift array in batch order (Vergara et al., 2012), and fourteen further OpenML and UCI files. The 28 synthetic streams come from fourteen standard generators, each with abrupt and gradual drift and eight change points, in three realizations whose results are averaged (Appendix D.1). File order is not chronological order, so we audited the label sequence of every real stream. Four files (MAGIC, Skin, Spam, KDD99) are one or a few label blocks: a feature-free predictor that repeats the previous label is near perfect on them, and a reset rule scores 99.0% on MAGIC against 74.4% for a long buffer, so they measure block structure rather than drift. The *temporal set* of the remaining 50 streams carries all main-text results; a *strict set* that also drops three streams on which label persistence beats every learner, and a *deduplicated set* with one drift variant per generator and one Insects variant, serve as sensitivity checks. Every feature is standardized at step t by the running statistics of earlier arrivals, so no future covariate enters the stream.

3.2 Acquisition strategies and update rules

The eleven acquisition strategies and ten update rules fall into six families (Appendix D.2 gives every detail and hyperparameter, all global across streams):

- **Stationary acquisition**: random, uncertainty sampling (US, maximum entropy), margin, least-confidence, coreset (greedy k-center; Sener & Savarese, 2018), and OVBAL (D-optimal selection with forgetting; Huang & Pinson, 2026b).
- **Drift-aware acquisition**: RVU, randomized variable uncertainty (Žliobaitė et al., 2014); split, half random and half entropy; and US+detect, which queries at random for 1,000 steps after a Page–Hinkley detection.

- **Latency-aware acquisition**: PAL-FS, the forgetting-and-simulating scheme of Pham et al. (2022), and PRopagate, our adaptation of the label propagation of Castellani et al. (2022), both scoring candidates by the probabilistic active-learning gain (Kottke et al., 2016).
- **Fixed-horizon updates**: naive replay, a FIFO of the 2,000 most recent labels, which with random acquisition is the *reference* learner; the window and FIFO-50, the same FIFO with 200 and 50 labels; experience replay (ER, a reservoir); and exponential decay.
- **Detector-driven updates**: reset, which wipes buffer and head when a Page–Hinkley detector fires on the arriving-label error, and detector-gated replay (PH-gate, DDM-gate, and the recency-weighted hybrid), which drops labels from before the last detection.
- **Replay with a memory constraint**: A-GEM (Chaudhry et al., 2019), which takes its main gradient on the newest labels and projects it so as not to increase the loss on older memory.

3.3 Online horizon selection

The right memory horizon depends on how fast the stream drifts, which is unknown. The selector chooses the horizon from the purchased labels. It runs K FIFO sub-learners with capacities $M_1 < \cdots < M_K$ (we use 20, 50, 200, 2,000, and unbounded) that receive the same purchased labels and take the same SGD steps. When a point is queried, the prediction $\hat{y}_{k,i}$ of each sub-learner is stored; when its label arrives, each sub-learner's error estimate is updated as

$$\epsilon_k \leftarrow (1-\beta)\,\epsilon_k + \beta\,\mathbb{1}[\hat{y}_{k,i} \neq y_i], \qquad \beta = 1 - 2^{-1/h}, \tag{5}$$

with half-life h=50 labels, and every prediction on the stream is made by the sub-learner $k_t = \arg\min_k \epsilon_k$, ties resolved toward the longer horizon. Scoring each sub-learner with the prediction it made at query time makes ϵ_k a prequential error on the purchased labels, delayed by δ like the labels themselves. The selector buys no labels beyond those of the reference learner and costs about K times its compute; it extends the paired learners of Bach & Maloof (2008), which switch between one stable and one reactive learner, to a grid of horizons under label delay. Fig. 10 gives the procedure.

3.4 A staleness decomposition

Equation (4) explains why the horizon matters. The following statement is informal; the formal version and its proof are in Appendix B.

Proposition 1 (Staleness decomposition)**.** *Consider a FIFO buffer of capacity M filled at rate b and a single change at τ with no other change within $\delta + M/b$ steps. Then $\varphi_t = 1$ for $t \in [\tau, \tau+\delta]$, after which it falls linearly to 0 over M/b steps, whereas for an unbounded buffer it decays only as the ratio of old to all labels. If, at a fixed parameter, the stale gradient is less aligned with the current-risk gradient than the fresh one, the first-order decrease of the current risk under one step along (4) is nonincreasing in φ_t.*

Proposition 1 explains what the memory horizon controls: the delay fixes how long the update is fully stale, and the horizon fixes how long stale labels keep a share of it. The alignment condition concerns one step at one parameter and does not follow from the concepts having different minimizers, so we measured it on a stream with known change points (Appendix C.1.2). It held at 973 of 974 measurements, the stale gradient pointed uphill at 95% of them, and the mixture (4) still descended at 90%: stale labels mostly shorten the step of naive replay rather than reverse it. The decomposition explains the effect of the update rule; it says nothing about acquisition, which changes g_t^{new} and g_t^{old} themselves.

4 Experiments

When labels arrive late and the data drift, the decision that matters may not be the one AL benchmarks study. Our experiments evaluate whether acquisition still pays under these conditions, which property of the update rule determines accuracy, and whether that property can be set online. We organize these studies around the following four Research Questions (RQs):

(RQ1) Does any acquisition strategy improve on random labeling once labels are delayed and the data drift?

Table 1: Prequential accuracy (%) and difference Δ to the reference on the 50 temporal streams.

Decision	Method	δ=0 acc	δ=0 Δ	δ=500 acc	δ=500 Δ	δ=2,000 acc	δ=2,000 Δ
Acquisition	random*	68.1	–	65.9	–	61.0	–
with naive replay	US	66.6	-1.5✗	65.1	-0.7	60.8	-0.3
	margin	67.0	-1.1	65.5	-0.4	61.0	-0.1
	least-conf	66.9	-1.2	65.3	-0.6	60.9	-0.2
	coreset	67.1	-1.0✗	65.0	-0.9✗	60.2	-0.9✗
	OVBAL	65.0	-3.1✗†	63.1	-2.8✗†	58.5	-2.5✗†
	RVU	68.0	-0.1	65.9	0.0	61.2	$+0.2$
	split	68.0	-0.1	66.0	$+0.2$	61.4	$+0.3$
	US+detect	67.2	-0.9✗	65.2	-0.6	60.9	-0.2
	PRopagate	66.7	-1.4✗	65.2	-0.7✗	60.8	-0.2
	PAL-FS	66.6	-1.5✗	64.9	-1.0✗	60.4	-0.7
Best of 10		68.0	-0.1	66.0	$+0.2$	61.4	$+0.3$
Update rule	window	72.9	$+4.8$✓†	68.9	$+3.0$✓	59.9	-1.2
with random acq.	FIFO-50	**77.5**	$+9.4$✓†	**73.3**	$+7.5$✓†	63.4	$+2.4$✓
	reset	76.3	$+8.2$✓†	72.2	$+6.3$✓	62.5	$+1.4$
	PH-gate	75.4	$+7.3$✓	71.3	$+5.4$✓	62.0	$+1.0$
	DDM-gate	75.3	$+7.2$✓	71.3	$+5.5$✓	62.4	$+1.3$
	hybrid	75.5	$+7.4$✓	71.3	$+5.5$✓	62.1	$+1.0$
	A-GEM	77.3	$+9.2$✓†	73.1	$+7.2$✓†	**63.9**	$+2.9$✓
	exp-decay	68.5	$+0.4$✓	66.1	$+0.2$✓	61.0	-0.1
	ER	67.9	-0.2	65.7	-0.1	61.0	0.0
Best of 9		77.5	$+9.4$	73.3	$+7.5$	63.9	$+2.9$

(RQ2) Which property of the update rule determines accuracy?

(RQ3) Where does a short memory horizon help, and where does it fail?

(RQ4) Can the memory horizon be chosen online from the purchased labels?

Unless otherwise stated, every method runs at budget b=0.1 on real and 0.05 on synthetic streams, at $\delta \in \{0, 500, 2000\}$, with three seeds, and results are means over the 50 temporal streams; all runs are CPU-only NumPy (Appendix D.5). Following AL-benchmark practice (Lu et al., 2025) we report average ranks with the Friedman test and the Nemenyi critical difference (CD); because the CD is a conservative all-pairs bound, we add a Holm-corrected Wilcoxon signed-rank test against the reference over 19 comparisons and a 95% bootstrap interval of each method's mean difference over streams (Appendix C.3). Appendix C.1 provides controlled studies. Study A is a positive control for acquisition on frozen ResNet-18 (He et al., 2016) features of CIFAR-100 (Krizhevsky, 2009), where a linear head is a compatible learner. Study B measures the alignment condition of Proposition 1. Study C removes drift by shuffling each stream and counts detector false alarms on stationary streams. Appendix C.2 repeats the core comparison with a one-hidden-layer network and on the strict and deduplicated stream sets.

4.1 RQ1: Acquisition under drift and delay

We first answer RQ1 by comparing each acquisition strategy with random labeling, first under the reference update and then under every update rule. Table 1 reports the mean prequential accuracy and the difference Δ to the reference at the three delays. The ✓/✗ marker records whether a difference is significantly above or below the reference after Holm correction (no marker: not significant), and † a rank difference beyond the Nemenyi CD. Table 2 reports, under six update rules, the gain of each strategy over random acquisition with the same rule, with the same markers.

Tables 1 and 2 show that no tested acquisition strategy improves on random labeling. Under naive replay the best strategies, RVU and split, stay within 0.3 points of random at every delay, and the others lose 0.9 to 3.1 points at δ=0; six of the ten are significantly below random, including both latency-aware strategies. Across the 132 cells of the factorial at $\delta \leq 500$ the largest gain is 0.8 points (split under A-GEM), and at δ=2,000 it is 1.5, not significant (Appendix C.3). The factorial also shows that a learned query rule can cost a great deal: under the window, FIFO-50, reset, and the

Table 2: Gain (points) of each acquisition strategy over random acquisition under six update rules.

Acquisition	δ=0 naive	window	FIFO-50	reset	PH-gate	A-GEM	δ=500 naive	window	FIFO-50	reset	PH-gate	A-GEM
random (acc.)	68.1	72.9	77.5	76.3	75.4	77.3	65.9	68.9	73.3	72.2	71.3	73.1
US	−1.5✗	−4.4✗	−5.5✗	−3.8✗	−6.0✗	−0.9	−0.7	−2.7✗	−5.8✗	−4.4✗	−4.8✗	−2.6✗
margin	−1.1	−3.8✗	−5.0✗	−3.8✗	−5.7✗	−0.6	−0.4	−2.1✗	−5.2✗	−4.0✗	−4.5✗	−2.3✗
least-conf	−1.2	−4.0✗	−5.1✗	−3.9✗	−5.8✗	−0.7	−0.6	−2.3✗	−5.4✗	−4.4✗	−4.6✗	−2.4✗
coreset	−1.0✗	−1.1✗	−0.9	−0.8	−1.5✗	−1.5✗	−0.9✗	−1.0✗	−1.5✗	−1.1✗	−1.4✗	−1.5✗
OVBAL	−3.1✗	−3.1✗	−3.7✗	−3.4✗	−3.4✗	−4.2✗	−2.8✗	−2.6✗	−3.2✗	−3.0✗	−2.8✗	−3.5✗
RVU	−0.1	+0.1✓	+0.1✓	+0.3✓	−0.1	+0.3✓	0.0✓	−0.1	0.0✓	0.0	0.0	0.0✓
split	−0.1	−0.2	0.0	−0.3	−1.3	+0.8✓	+0.2	−0.8✗	−0.8✗	−1.2✗	−1.0✗	−0.2
US+detect	−0.9✗	−2.1✗	−1.9	−2.0✗	−1.9✗	0.0	−0.6	−2.1✗	−3.6✗	−3.4✗	−3.0✗	−1.3✗
PRopagate	−1.4✗	−1.2	−1.7✗	−1.3	−2.1✗	−2.5✗	−0.7✗	−1.4✗	−1.8✗	−1.5✗	−2.2✗	−2.3✗
PAL-FS	−1.5✗	−1.2	−1.6✗	−1.5	−2.3✗	−2.6✗	−1.0✗	−1.6✗	−2.2✗	−2.0✗	−2.2✗	−2.4✗
Best of 10	−0.1	+0.1	+0.1	+0.3	−0.1	+0.8	+0.2	−0.1	0.0	0.0	0.0	0.0

Table 3: Entropy-sampling gain over random on CIFAR-100 features (b=0.05, points).

Condition	δ=0 naive	FIFO-50	A-GEM	δ=500 naive	FIFO-50	A-GEM
Stationary	**+4.2**	**+2.9**	**+4.3**	–	–	–
Class-prior drift	**+4.1**	**+4.1**	**+5.0**	**+2.7**	**+2.9**	**+3.8**
Two-class swap	−1.3	−0.4	**+1.1**	−3.7	−2.9	−1.4
Gradual permutation	−4.0	−10.7	−9.1	−3.8	−12.4	−11.8
Abrupt permutation	−4.2	−11.9	−9.8	−3.9	−13.5	−12.4

gate, the boundary-seeking strategies fall 3.8 to 6.0 points below random at δ=0, because a short buffer filled with points near a boundary that has moved describes the current concept worse than a random one.

The loss of acquisition's advantage is caused by drift, not by the learner. On the tabular streams the tested strategies do not help even with the order shuffled or the budget cut to a fifth (Study A), which bounds these heuristics under this learner. On CIFAR-100 features, where they do help, Table 3 shows that entropy sampling gains 2.9 to 4.3 points over random on a stationary stream and keeps its gain under class-prior drift, which leaves $p(y \mid x)$ fixed. A swap of two class labels at each change removes most of it, and label permutations, abrupt or gradual, reverse it to −3.8 to −13.5 points, with no recovery at δ=500 or 2,000.

Takeaway 1

Under drift and delay, no tested acquisition strategy improves on random labeling, and the boundary-seeking ones cost up to 6 points under the update rules that work best; the gain acquisition shows on stationary features disappears once $p(y \mid x)$ changes.

4.2 RQ2: The memory horizon

We next answer RQ2 by holding acquisition at random and varying the update rule (Table 1, lower block). Eight of nine rules gain over naive replay, each significant with a bootstrap interval that excludes zero: FIFO-50 +9.4 points, A-GEM +9.2, reset +8.2, the three gates +7.2 to +7.4, the window +4.8, and exponential decay at its default timescale +0.4, while ER changes accuracy by −0.2. Naive replay, the window, and FIFO-50 are one FIFO policy at three horizons, so Fig. 3 sweeps the horizon directly.

Accuracy rises as the horizon shrinks, from 67.9% with an unbounded buffer to 72.9% at 200 labels, 77.5% at 50, and 77.6% at 20. Exponential decay follows the same curve once its minibatch is sampled in proportion to the weights (78.1% at a half-life of 10 labels), whereas the weighted mean gradient over a uniform minibatch trails by about two points, so hard removal is not the decisive property; the effective horizon is. The detector-based rules do not beat the best fixed horizon: reset ties the 100-label FIFO and the gates sit between the 100- and 200-label FIFOs. A-GEM ties FIFO-

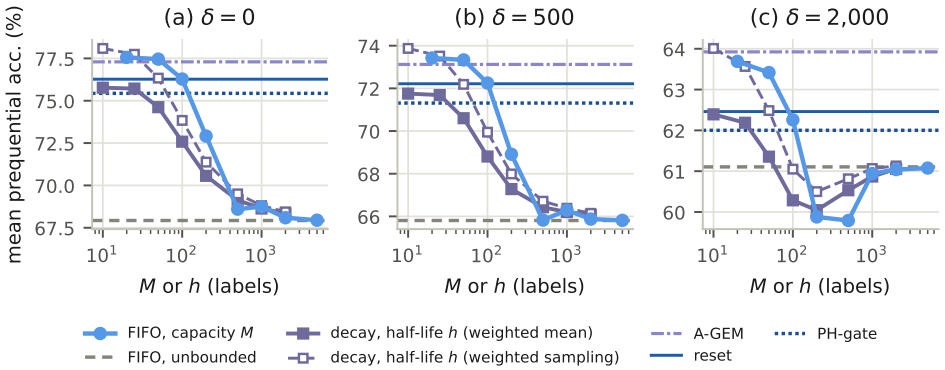

Figure 3: Accuracy against the memory horizon (log scale) at three delays, mean over the 50 temporal streams; FIFO capacity M, decay half-life h, and fixed rules as references.

Table 4: Accuracy (%) of fixed memory horizons, a per-stream oracle, and the online selector.

δ	Streams	Fixed FIFO capacity 20	50	200	2,000	∞	Oracle	Selector
δ=0	Insects, elec2, Covertype (8)	74.2	74.5	73.7	66.3	65.3	75.0	**75.4**
	NOAA, Rialto, Gas (3)	63.3	63.4	62.5	63.6	64.1	64.5	**64.9**
	Other real (11)	72.9	72.9	72.8	72.2	72.2	74.9	**74.2**
	Generators (28)	82.0	81.7	74.0	67.4	67.4	82.3	**82.2**
	All (50)	77.6	77.5	73.0	68.1	67.9	78.4	**78.3**
δ=500	Insects, elec2, Covertype (8)	70.5	70.9	70.2	64.1	63.5	71.4	**71.1**
	NOAA, Rialto, Gas (3)	55.0	56.2	59.0	61.5	**61.5**	61.6	60.7
	Other real (11)	66.8	67.4	68.2	**68.5**	68.3	68.9	68.4
	Generators (28)	**78.8**	78.1	70.0	65.8	65.8	78.9	78.1
	All (50)	73.4	73.3	69.0	65.9	65.7	74.5	**73.8**
δ=2,000	Insects, elec2, Covertype (8)	62.6	**63.0**	62.5	59.6	59.9	64.1	62.6
	NOAA, Rialto, Gas (3)	51.0	53.0	56.6	58.3	**58.3**	58.3	57.2
	Other real (11)	57.7	58.5	**60.2**	59.7	59.7	60.7	59.5
	Generators (28)	**67.7**	66.6	59.5	62.4	62.4	67.9	64.8
	All (50)	**63.7**	63.4	60.0	61.1	61.1	65.1	62.8
Shuffled	*All*, δ=0 (50)	63.9	65.1	67.1	68.4	**68.4**	68.4	67.9
	All, δ=500 (50)	63.1	64.3	66.2	67.5	**67.5**	67.6	66.9

50 because its main gradient is computed on the newest labels, whereas a version that draws both gradients from the same buffer ranks with random (Appendix C.2). A one-hidden-layer network reproduces the ordering (FIFO-50 79.4%, A-GEM 79.2%, reset 77.7%, window 74.1%, naive replay 69.3%). On the 28 deduplicated streams the gains keep their size (FIFO-50 +7.6 at δ=0, interval 3.9 to 11.5; +5.6 at δ=500), although corrected rank tests pass only at δ=0.

Takeaway 2

The update rule, not the query rule, moves accuracy (by up to 9 points), and its decisive property is the memory horizon: short FIFO buffers, recency-weighted sampling, detector resets, and A-GEM all help, and what they share is a short effective memory.

4.3 RQ3: Where a short horizon helps and where it fails

To answer RQ3, we split the streams by the source of their drift and vary the delay. Table 4 reports, for each stream group and delay, the accuracy of five fixed FIFO horizons, of a per-stream oracle that picks the best fixed horizon after the fact, and of the online selector of Section 3.3.

The advantage of a short horizon shrinks with the delay: FIFO-50 gains 9.4, 7.5, and 2.4 points over naive replay at the three delays (Table 1), and at δ=2,000 the 200-label window falls below the long buffer because its slice ends 2,000 steps in the past. The advantage also belongs to particular streams. On the Insects variants, elec2, and Covertype, FIFO-50 gains 8.2 points over the 2,000-label buffer at δ=0, and on the generators 14.3. On NOAA weather, Rialto, and the gas-sensor batches, whose documented drift is seasonal, day-to-day, or batch-to-batch, no fixed capacity beats the long buffer by more than 1.7 points, and short horizons lose: at δ=500 FIFO-50 loses 5.3 points on average (8.7 on Rialto) and a detector reset 10.3 (16.8 on the gas batches). With each stream's order destroyed, FIFO-50 loses 3.3 points and A-GEM 4.2, while no acquisition strategy gains more than 0.4 (Study C). The best fixed capacity at δ=0 is 20 or 50 labels on 35 of the 50 streams and 500 or more on 9.

Takeaway 3

A short horizon helps only where the concept changes faster than the buffer turns over; its advantage shrinks with delay and reverses where drift is slow or absent, so no single horizon suits every stream.

4.4 RQ4: Online horizon selection

Finally, we answer RQ4 with the selector (Table 4, last column). It scores 78.3% at δ=0 and 73.8% at δ=500, above every fixed horizon and within 0.1 and 0.7 points of the per-stream oracle. On NOAA weather, Rialto, and the gas batches it recovers the loss of the short horizons (64.9% against 63.4% for FIFO-50 at δ=0; 60.7% against 56.2% at δ=500), on the drifting groups it tracks the short horizons, and on shuffled streams it costs 0.5 points where FIFO-50 costs 3.3. At δ=2,000 the selection signal lags by the delay: with a half-life of 50 labels the selector falls 0.9 points below the best fixed horizon, and with 200 it ties it (Appendix C.3).

Takeaway 4

The memory horizon can be set online from the labels already paid for: choosing among a few buffers by their prequential error comes within 0.7 points of a per-stream oracle at delays up to 500 steps, which none of the detector-based rules does reliably.

5 Conclusion

Delayed labels and drift change which decision of an online active learner matters. With the memory policy free, no tested acquisition strategy improves on random labeling under drift and delay, and the gain acquisition shows on stationary data disappears once $p(y \mid x)$ changes. The memory horizon of the update rule moves accuracy by 9 points and explains why short buffers, recency sampling, detector resets, and A-GEM help; the same short horizons cost accuracy where drift is slow or absent, and an online selector driven by the prequential error on the purchased labels sets the horizon at delays up to 500 steps. Our evidence uses a linear head and a one-hidden-layer network with one global schedule, a positive control on one image stream with synthetic changes, and a real-stream advantage that rests on the Insects family, elec2, and Covertype; on the deduplicated set the gains hold as interval estimates rather than corrected rank tests under delay. Deep models, horizon selection at long delays, and stochastic delays remain future work.

Reproducibility statement

Code, per-cell results, and the scripts that produce every table and figure are available at https://anonymous.4open.science/r/active-learning-cd-iclr27-1342/. The benchmark is CPU-only NumPy with fixed seeds (three per cell); rerunning the drivers reproduces every per-cell accuracy of the benchmark exactly, and a smoke test checks five cells against the shipped results in about ten

seconds. A script fetches and checksums every public stream (the six Insects files are downloaded by hand, as the repository describes). Appendix D gives the streams, methods, protocol (Fig. 9), hyperparameters (Fig. 12), and compute.

ETHICS STATEMENT

This work benchmarks existing acquisition strategies and update rules for online active learning and proposes an online horizon selector. It collects no new data, involves no human participants, and trains only small linear and one-hidden-layer models; all data come from public datasets, used under their own terms, or from synthetic generators. Some streams describe people (bank marketing, human activity recognition, EEG eye state); we use them only as benchmark data and make no attempt to identify anyone. The main risk of misuse is a practitioner adopting a short memory horizon where drift is slow, which costs accuracy; we report where this happens (Section 4.3), and the online selector reduces the loss at delays up to 500 steps. We do not anticipate harms beyond those already present in the applications that deploy online learners.

USE OF LARGE LANGUAGE MODELS

During the preparation of this paper, we used a large language model to draft and revise text and to improve its grammar, word choice, and conciseness. We also used it to help identify relevant literature. The authors independently read all suggested papers and verified each citation against its original source before inclusion. We used Claude Code to assist with designing and running the experiments and analyzing the results. The authors reviewed the code and verified all reported numerical results against the saved outputs. All claims are the authors' own, and we take full responsibility for the content of this paper.

REFERENCES

Stephen H. Bach and Marcus A. Maloof. Paired learners for concept drift. In *IEEE International Conference on Data Mining (ICDM)*, pp. 23–32, 2008. doi: 10.1109/ICDM.2008.119.

Albert Bifet and Ricard Gavaldà. Learning from time-changing data with adaptive windowing. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pp. 443–448, 2007. doi: 10.1137/1.9781611972771.42.

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*, volume 13532 of *Lecture Notes in Computer Science*, pp. 260–272, 2022. doi: 10.1007/978-3-031-15937-4_22.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations (ICLR)*, 2019.

A. Philip Dawid. Present position and potential developments: Some personal views: Statistical theory: The prequential approach. *Journal of the Royal Statistical Society. Series A (General)*, 147(2):278–292, 1984. doi: 10.2307/2981683.

Ryan Elwell and Robi Polikar. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, 22(10):1517–1531, 2011. doi: 10.1109/TNN.2011.2160459.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In *International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1183–1192, 2017.

João Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with drift detection. In *Advances in Artificial Intelligence – SBIA 2004*, volume 3171 of *Lecture Notes in Computer Science*, pp. 286–295, 2004. doi: 10.1007/978-3-540-28645-5_29.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44:1–44:37, 2014. doi: 10.1145/2523813.

Michael Harries. Splice-2 comparative evaluation: Electricity pricing. Technical Report UNSW-CSE-TR-9905, School of Computer Science and Engineering, University of New South Wales, 1999.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Xiwen Huang and Pierre Pinson. How to purchase labels? A cost-effective approach using active learning markets. *INFORMS Journal on Data Science*, 2026a. doi: 10.1287/ijds.2025.0093.

Xiwen Huang and Pierre Pinson. QueryMarket: Cost-aware online active learning in data markets. *arXiv preprint arXiv:2606.17805*, 2026b.

Daniel Kottke, Georg Krempl, Dominik Lang, Johannes Teschner, and Myra Spiliopoulou. Multi-class probabilistic active learning. In *European Conference on Artificial Intelligence (ECAI)*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pp. 586–594, 2016. doi: 10.3233/978-1-61499-672-9-586.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pp. 6467–6476, 2017.

Viktor Losing, Barbara Hammer, and Heiko Wersing. Interactive online learning for obstacle classification on a mobile robot. In *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2015. doi: 10.1109/IJCNN.2015.7280610.

Viktor Losing, Barbara Hammer, and Heiko Wersing. KNN classifier with self adjusting memory for heterogeneous concept drift. In *IEEE International Conference on Data Mining (ICDM)*, pp. 291–300, 2016. doi: 10.1109/ICDM.2016.0040.

Jie Lu, Anjin Liu, Fan Dong, Feng Gu, João Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363, 2019. doi: 10.1109/TKDE.2018.2876857.

Po-Yi Lu, Yi-Jie Cheng, Chun-Liang Li, and Hsuan-Tien Lin. An expanded benchmark that rediscovers and affirms the edge of uncertainty sampling for active learning in tabular datasets. *Transactions on Machine Learning Research (TMLR)*, 2025.

David J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992. doi: 10.1162/neco.1992.4.4.590.

E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954. doi: 10.1093/biomet/41.1-2.100.

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.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations (ICLR)*, 2018.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

Vinicius M. A. Souza, Denis M. dos Reis, André G. Maletzke, and Gustavo E. A. P. A. Batista. Challenges in benchmarking stream learning algorithms with real-world data. *Data Mining and Knowledge Discovery*, 34(6):1805–1858, 2020. doi: 10.1007/s10618-020-00698-5.

Alexander Vergara, Shankar Vembu, Tuba Ayhan, Margaret A. Ryan, Margie L. Homer, and Ramón Huerta. Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical*, 166–167:320–329, 2012. doi: 10.1016/j.snb.2012.01.074.

Indrė Žliobaitė, Albert Bifet, Bernhard Pfahringer, and Geoffrey Holmes. Active learning with drifting streaming data. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1): 27–39, 2014. doi: 10.1109/TNNLS.2012.2236570.

A Related work

A first line of related work designs and benchmarks acquisition strategies. Uncertainty scores (Settles, 2009; Houlsby et al., 2011; Gal et al., 2017), optimal-design criteria (MacKay, 1992; Huang & Pinson, 2026b), and coverage criteria (Sener & Savarese, 2018) set the standards for label efficiency, and label markets price queries directly (Huang & Pinson, 2026a). The comprehensive tabular benchmark of Lu et al. (2025) compares acquisition strategies on pool-based, stationary data with immediate labels and a fixed memory policy, summarizes them by average rank and critical difference, and affirms uncertainty sampling with a compatible model. Our contribution is complementary: we keep their statistics and ask which decision matters once the conditions they hold fixed are released (Table 5). Under our linear learner the learned strategies gain at most 0.4 points on the tabular streams even without drift, consistent with their model-compatibility analysis, and 2 to 5 points on stationary image features with a compatible head, so we claim no reversal of their ranking; what we add is that a change in $p(y \mid x)$ removes that gain.

A second line studies acquisition under drift and verification latency. Žliobaitė et al. (2014) randomize an adaptive uncertainty threshold for drifting streams. Pham et al. (2022) forget observations that will have left the window when a candidate's label arrives and simulate the labels of pending queries, and Castellani et al. (2022) propagate labels to pending queries from their labeled neighbours. These papers observe that latency erodes the value of acquisition; we include the first as published and the second as an adaptation. Like stationary benchmarks, they hold the learner's memory policy fixed. Our results corroborate their observation and add that the memory horizon moves accuracy by an order of magnitude more than any query utility we tested.

A third line develops the update side. Concept-drift research supplies detectors that trigger adaptation, DDM (Gama et al., 2004), ADWIN (Bifet & Gavaldà, 2007), and Page–Hinkley (Page, 1954), and adaptive windows that forget stale data (Gama et al., 2014; Lu et al., 2019); paired learners (Bach & Maloof, 2008) switch between a stable and a reactive learner according to their recent error. Continual learning uses replay to retain old tasks (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019), the opposite of forgetting a superseded concept; A-GEM transfers because its main gradient is taken on the newest labels. Our selector extends paired learners to a grid of horizons under label delay, and to our knowledge no prior work crosses acquisition with the update rule under label delay or sweeps the memory horizon in an online AL benchmark.

Table 5: What the benchmark adds to prior AL benchmarks.

Aspect	Lu et al. (2025)	Ours
Acquisition strategies compared	✓	✓ (11)
Average rank and critical difference	✓	✓
Signed-rank tests and bootstrap intervals	–	✓
Online streaming (versus pool-based)	–	✓
Concept drift, synthetic and audited real	–	✓ (50 streams)
Label delay	–	✓ (three delays)
Update rule and memory horizon as axes	–	✓ (10 rules, 25 horizons)
Acquisition × update factorial	–	✓
Online horizon selection	–	✓

B PROOF OF THE STALENESS DECOMPOSITION

B.1 SETUP AND NOTATION

This appendix gives the formal version of Proposition 1. We use the protocol of Section 2; Table 6 collects the notation. The buffer is a FIFO of capacity M, minibatches are sampled uniformly from it, and $R_t(W)$ denotes the expected loss of the head W under the concept in force at step t.

Table 6: Notation.

Symbol	Meaning
$x_t, y_t, \hat{y}_t$	input, true label, and prediction at step t
δ, b	label delay; budget (fraction of the stream queried)
q_t, $\mathcal{L}_t$	query decision; labels available at step t
W_t, ℓ	linear–softmax head; cross-entropy loss
$\mathcal{B}_t$, M, $w_{t,i}$	buffer, its FIFO capacity, and the weights of the update
h	half-life of exponential decay or of the selector's error estimate, in labels
φ_t	share of the update weight on concept-stale labels
g_t^{new}, g_t^{old}, R_t	fresh and stale buffer gradients; current risk
λ, η	Page–Hinkley threshold and tolerance

B.2 STATEMENT AND PROOF

Proposition 1 (Staleness decomposition). *Let the buffer be a FIFO of capacity M, let labels arrive at rate b per step, and let a single change occur at τ with none within $\delta + M/b$ steps after it. Then:* (a) *the expected naive-replay gradient is the mixture $g_t^{\text{naive}} = (1-\varphi_t)g_t^{\text{new}} + \varphi_t g_t^{\text{old}}$;* (b) *$\varphi_t = 1$ for $t \in [\tau, \tau+\delta]$, then falls linearly to 0 over M/b steps, whereas with an unbounded buffer it decays only as the ratio of old to all labels;* (c) *at a fixed parameter W, if $\langle g_t^{\text{old}}, \nabla R_t(W)\rangle \le \langle g_t^{\text{new}}, \nabla R_t(W)\rangle$, then the first-order decrease of R_t under one step along $-g_t^{\text{naive}}$ is nonincreasing in φ_t.*

Proof. *(a)* For a minibatch sampled uniformly from $\mathcal{B}$, the expected gradient equals the buffer average $\frac{1}{|\mathcal{B}|}\sum_i \nabla\ell(x_i, y_i)$. Partition $\mathcal{B}$ into the fresh set F and stale set S with $\varphi_t = |S|/|\mathcal{B}|$; the average is $(1-\varphi_t)\,\overline{\nabla\ell}_F + \varphi_t\,\overline{\nabla\ell}_S = (1-\varphi_t)g_t^{\text{new}} + \varphi_t g_t^{\text{old}}$. This is an identity and needs no assumption. *(b)* By (1) no post-change label arrives before $\tau+\delta$, so throughout $[\tau, \tau+\delta]$ every buffer entry predates τ and $\varphi_t = 1$. Afterward fresh labels arrive at rate b per step; with a full FIFO of capacity M each arrival evicts one stale label, so after n arrivals $\varphi_t = 1 - n/M$ until $n = M$, which takes M/b steps, and the assumption of no further change within $\delta + M/b$ steps ensures that the fresh set stays fresh. With an unbounded buffer holding n_0 labels at τ, $\varphi_t = n_0/(n_0+n)$ after n fresh arrivals, which decays only as $1/n$. *(c)* Fix W and write $a_{\text{old}} = \langle g_t^{\text{old}}, \nabla R_t(W)\rangle$ and $a_{\text{new}} = \langle g_t^{\text{new}}, \nabla R_t(W)\rangle$. One step of size η along $-g_t^{\text{naive}}$ changes R_t to first order by $-\eta[(1-\varphi_t)a_{\text{new}} + \varphi_t a_{\text{old}}]$, whose derivative with respect to φ_t is $-\eta(a_{\text{old}} - a_{\text{new}}) \ge 0$ under the assumption $a_{\text{old}} \le a_{\text{new}}$. Hence the first-order decrease is nonincreasing in φ_t. □

Scope. The alignment assumption of (c) is not implied by the two concepts having different risk minimizers: for two logistic risks with opposite optimal predictions there are parameters at which the old-concept gradient still descends the current risk. Study B (Appendix C.1.2) therefore measures it rather than derives it, once per micro-batch before its SGD steps: $a_{\text{old}} \le a_{\text{new}}$ held on 973 of 974 measurements and $a_{\text{old}} < 0$ on 930. Nothing here bounds the excess risk along an SGD trajectory, and we make no such claim.

C ADDITIONAL EXPERIMENTS

This appendix gathers the supporting results in four parts: (i) controlled studies (Section C.1), namely a positive control for acquisition (Study A), the alignment condition of Proposition 1 (Study B), and stationary controls (Study C); (ii) robustness of the main comparison to the stream set, preprocessing, implementations, and learner (Section C.2); (iii) full benchmark results (Section C.3); and (iv) an auxiliary CIFAR study of detector-gated replay with two negative results (Section C.4).

C.1 CONTROLLED STUDIES

C.1.1 STUDY A: A POSITIVE CONTROL FOR ACQUISITION

Study A asks whether the absence of acquisition gains in the main benchmark reflects the learner and features or the conditions of drift and delay.

Setup. Frozen ResNet-18 (He et al., 2016) penultimate features of a ten-class CIFAR-100 (Krizhevsky, 2009) subset, PCA-whitened to 128 dimensions and causally standardized, are played as a 40,000-step stream. Conditions: stationary; abrupt label permutations at eight change points; class-prior drift, which draws each era's class prior from a Dirichlet(0.5) and leaves $p(y \mid x)$ fixed; a two-class swap, which exchanges the labels of two of the ten classes at each change; and gradual permutations mixed in linearly over a 2,000-step window around each change. Every acquisition strategy runs with naive replay, FIFO-50, and A-GEM at $b \in \{0.02, 0.05\}$, over three stream seeds (change points and permutations) and three learner seeds.

Main result: acquisition helps on stationary features. Table 7 shows that on the stationary stream entropy, margin, and least-confidence sampling gain 2 to 5 points over random under naive replay and FIFO-50 and 4 to 8 under A-GEM, while coreset and OVBAL lose and the latency-aware strategies stay within about 2 points of random. A linear head on these features is therefore a learner with which acquisition works.

Changes in $p(y \mid x)$ remove the gain. Under abrupt label permutations every gain reverses at δ=0 (entropy -2.8 to -4.2 under naive replay, -9 to -12 under FIFO-50) and stays reversed at δ=500 and 2,000. Table 8 shows that the erosion follows how much of $p(y \mid x)$ moves: class-prior drift keeps entropy sampling at $+3.5$ to $+7.7$ points at δ=0, a two-class swap cuts its gain under naive replay to -1.4 (margin keeps $+2.0$) and to -2.5 at δ=500, and gradual permutations reverse every gain as abrupt ones do.

The tabular streams bound the heuristics. At one fifth of the main budget (Table 9), where each label matters more, the tabular linear learner still shows no acquisition gain, with the original order or shuffled: the best strategy gains 0.8 points (RVU, shuffled), and the uncertainty-based strategies lose 1.4 to 2.7. Generator streams in this control use realization 0 only.

Takeaway. The tested heuristics do not help the tabular linear learner with or without drift, which bounds the heuristics rather than acquisition; on features where they do help, a change in $p(y \mid x)$ removes the gain and delay does not restore it.

Table 7: Study A: gain over random on CIFAR-100 features, stationary and under abrupt label permutations.

Condition	Update	b	random	Gain over random (points) US	margin	least-conf	coreset	OVBAL	RVU	split	US+detect	PRopagate	PAL-FS
Stationary	naive	0.02	76.3	**+3.4**	**+4.8**	**+4.5**	−11.5	−4.2	**+1.1**	**+2.1**	**+3.3**	−0.5	−0.2
		0.05	82.8	**+4.2**	**+5.4**	**+5.1**	−7.4	−3.3	**+1.2**	**+3.0**	**+4.2**	+0.1	+0.4
	FIFO-50	0.02	76.4	**+2.3**	**+3.6**	**+3.3**	−15.0	−3.9	**+1.1**	**+1.5**	**+2.3**	−2.1	−2.1
		0.05	82.3	**+2.9**	**+3.7**	**+3.6**	−10.6	−2.6	+0.9	**+1.8**	**+2.9**	−1.4	−1.4
	A-GEM	0.02	71.3	**+7.2**	**+8.1**	**+8.2**	−10.7	−4.9	**+1.2**	**+4.2**	**+7.2**	−1.1	−1.1
		0.05	80.7	**+4.3**	**+4.9**	**+4.9**	−8.5	−5.3	**+1.2**	**+3.2**	**+4.3**	−0.5	−0.4
Permutation, δ=0	naive	0.02	21.4	−2.8	−0.6	−1.6	−3.1	−2.0	+0.3	−1.3	−2.5	−0.7	−1.1
		0.05	23.9	−4.2	−0.7	−3.1	−2.0	−1.8	−0.4	−1.8	−3.0	−0.2	−0.3
	FIFO-50	0.02	36.8	−9.3	−4.7	−7.8	−5.4	−4.6	−0.7	−3.7	−8.5	−3.5	−3.5
		0.05	52.2	−11.9	−6.4	−9.3	−6.3	−4.0	−0.4	−4.6	−6.6	−3.8	−3.5
	A-GEM	0.02	39.1	−6.5	−2.9	−5.1	−8.4	−5.3	−0.3	−1.8	−5.7	−3.4	−3.6
		0.05	54.0	−9.8	−4.8	−7.1	−10.0	−6.3	−0.2	−3.1	−5.0	−2.7	−2.3
Permutation, δ=500	naive	0.02	19.8	−2.9	−1.3	−2.5	−2.9	−2.0	−0.4	−1.1	−2.4	−1.2	−1.5
		0.05	21.7	−3.9	−1.5	−3.2	−2.1	−1.6	−0.5	−1.5	−2.9	−0.4	−0.7
	FIFO-50	0.02	33.8	−10.2	−6.4	−9.2	−5.1	−5.0	−2.0	−4.5	−9.4	−4.6	−4.3
		0.05	47.3	−13.5	−8.5	−11.3	−6.5	−4.3	−1.4	−6.3	−8.3	−3.4	−4.5
	A-GEM	0.02	34.8	−7.5	−4.3	−6.7	−7.6	−4.3	−0.9	−2.2	−6.8	−3.7	−3.6
		0.05	48.8	−12.4	−7.9	−10.1	−9.2	−6.0	−1.4	−5.1	−7.0	−3.2	−3.3
Permutation, δ=2,000	naive	0.02	15.1	−2.2	−1.0	−1.9	−2.1	−1.0	+0.1	−0.9	−2.0	−0.5	−0.6
		0.05	16.2	−2.8	−1.4	−2.5	−1.4	−1.1	−0.1	−1.6	−1.9	−0.1	−0.4
	FIFO-50	0.02	24.1	−7.2	−4.3	−6.7	−3.9	−3.2	−1.6	−4.0	−6.8	−2.4	−2.7
		0.05	33.0	−10.6	−6.7	−8.8	−5.0	−2.6	−1.0	−4.8	−7.5	−0.9	−2.1
	A-GEM	0.02	25.0	−5.7	−3.2	−4.5	−5.0	−2.6	−0.3	−2.1	−5.1	−2.4	−1.6
		0.05	34.6	−10.4	−6.3	−8.8	−6.6	−3.9	−0.9	−4.9	−7.2	−1.3	−3.2

Table 8: Study A: gain over random on CIFAR-100 features under milder forms of drift.

Condition	Update	b	random	Gain over random (points) US	margin	least-conf	coreset	OVBAL	RVU	split	US+detect	PRopagate	PAL-FS
Class prior, δ=0	naive	0.02	74.6	**+3.5**	**+4.9**	**+5.1**	−6.4	−2.0	**+1.6**	**+2.7**	**+3.5**	−2.0	−2.0
		0.05	81.5	**+4.1**	**+5.6**	**+5.1**	−4.3	−1.3	+0.7	**+2.8**	**+4.2**	−0.2	−0.2
	FIFO-50	0.02	77.2	**+4.6**	**+4.9**	**+4.8**	−10.8	−3.2	**+1.1**	**+3.1**	**+4.6**	−3.1	−3.5
		0.05	84.5	**+4.1**	**+4.4**	**+4.4**	−7.6	−3.1	+0.6	**+3.0**	**+3.9**	−3.0	−3.4
	A-GEM	0.02	74.9	**+7.7**	**+8.0**	**+8.4**	−8.3	−4.5	+0.8	**+5.5**	**+7.7**	−2.1	−2.0
		0.05	83.7	**+5.0**	**+5.4**	**+5.3**	−7.2	−5.5	+0.9	**+3.7**	**+4.8**	−0.6	−0.3
Class prior, δ=500	naive	0.02	73.1	**+1.2**	**+3.0**	**+1.7**	−6.6	−2.0	+0.4	**+1.6**	**+1.2**	−3.8	−4.1
		0.05	79.0	**+2.7**	**+4.0**	**+3.9**	−4.6	−0.1	+0.9	**+2.4**	**+3.0**	−1.7	−2.0
	FIFO-50	0.02	75.2	**+1.9**	**+3.3**	**+2.0**	−8.7	−2.8	+0.8	**+2.0**	**+1.9**	−4.8	−4.6
		0.05	82.0	**+2.9**	**+3.4**	**+3.3**	−6.8	−2.1	+0.7	**+2.2**	**+2.5**	−2.3	−2.5
	A-GEM	0.02	72.6	**+4.8**	**+4.9**	**+5.0**	−9.3	−3.6	−0.4	**+3.2**	**+4.9**	−2.9	−3.4
		0.05	81.4	**+3.8**	**+3.9**	**+3.8**	−7.9	−4.6	+0.6	**+2.7**	**+3.5**	−0.9	−1.3
Two-class swap, δ=0	naive	0.02	49.7	−1.4	**+2.0**	+0.4	−8.7	−3.8	+0.7	0.0	−1.4	−1.6	−1.5
		0.05	54.2	−1.3	**+3.8**	**+2.0**	−4.7	−3.2	**+1.6**	+0.9	+0.8	+0.4	+0.2
	FIFO-50	0.02	62.7	−2.0	+1.0	−1.0	−11.3	−4.6	+0.4	−0.6	−2.0	−2.8	−2.8
		0.05	72.8	−0.4	**+1.3**	+0.2	−8.3	−2.7	+0.8	+1.0	+0.2	−2.9	−3.5
	A-GEM	0.02	60.1	**+2.9**	**+4.8**	**+4.3**	−10.3	−4.7	+0.7	**+2.8**	**+2.9**	−2.8	−2.8
		0.05	72.0	**+1.1**	**+2.7**	**+2.4**	−9.3	−4.9	**+1.1**	**+2.0**	**+2.0**	−1.2	−1.2
Two-class swap, δ=500	naive	0.02	48.6	−2.5	+0.3	−1.3	−9.9	−4.1	+0.5	+0.2	−2.0	−2.3	−2.1
		0.05	53.1	−3.7	+0.7	−1.0	−5.2	−3.5	+0.8	0.0	−0.5	−0.9	−1.9
	FIFO-50	0.02	60.9	−4.1	−1.2	−3.0	−11.1	−4.7	−0.2	−1.0	−3.8	−5.3	−5.1
		0.05	70.4	−2.9	−0.6	−1.8	−8.5	−2.4	+0.3	−0.6	−1.8	−2.1	−3.2
	A-GEM	0.02	58.4	−0.8	**+1.1**	+0.8	−13.0	−4.5	−1.1	+0.6	−0.2	−3.5	−4.2
		0.05	69.6	−1.4	+0.6	−0.2	−9.4	−4.5	+0.8	+0.7	+0.5	−1.8	−2.3
Gradual perm., δ=0	naive	0.02	21.3	−2.3	−0.7	−1.9	−3.3	−2.2	+0.3	−1.0	−2.0	−0.6	−0.5
		0.05	23.6	−4.0	−0.8	−2.4	−2.6	−1.6	−0.2	−1.8	−3.0	0.0	0.0
	FIFO-50	0.02	36.7	−8.9	−5.0	−8.0	−5.4	−5.3	−1.0	−3.5	−8.4	−3.0	−3.0
		0.05	51.3	−10.7	−6.2	−7.9	−6.8	−4.0	−0.7	−4.6	−6.2	−3.6	−4.0
	A-GEM	0.02	38.7	−6.3	−2.8	−5.1	−9.2	−6.1	+0.2	−1.4	−6.2	−3.4	−3.4
		0.05	52.5	−9.1	−4.2	−6.5	−9.9	−5.6	−0.6	−3.2	−4.1	−2.3	−2.7
Gradual perm., δ=500	naive	0.02	19.8	−2.8	−1.3	−2.6	−3.3	−2.2	+0.2	−1.5	−2.3	−1.2	−1.1
		0.05	21.6	−3.8	−1.6	−2.9	−2.5	−1.5	−0.7	−1.6	−2.7	−0.1	−0.6
	FIFO-50	0.02	33.4	−9.8	−5.9	−8.5	−4.2	−5.3	−1.5	−4.9	−8.4	−5.2	−4.0
		0.05	46.6	−12.4	−7.8	−10.3	−6.4	−3.8	−1.0	−5.2	−8.0	−3.3	−3.4
	A-GEM	0.02	35.5	−8.2	−5.2	−7.0	−8.5	−6.0	−1.4	−3.4	−7.7	−4.8	−3.9
		0.05	48.2	−11.8	−7.1	−9.4	−9.8	−5.4	−0.8	−4.8	−7.5	−3.3	−3.5

Table 9: Study A: acquisition at one fifth of the budget on the 50 tabular streams (δ=0, naive replay).

Order		US	margin	least-conf	coreset	OVBAL	RVU	split	US+detect	PRopagate	PAL-FS
Ordered	mean gain	−2.1	−1.4	−1.6	−2.2	−3.9	−0.1	−0.8	−1.8	−0.9	−0.7
	wins/losses	14/28	15/26	14/28	11/29	11/31	20/15	15/21	14/29	13/24	12/22
Shuffled	mean gain	−2.7	−1.8	−2.5	−2.9	−4.6	+0.8	−2.0	−2.5	−0.8	−0.6
	wins/losses	15/25	20/19	16/24	9/28	6/38	25/3	15/19	15/25	10/15	17/15

C.1.2 STUDY B: ALIGNMENT OF STALE GRADIENTS

Study B measures the alignment condition of Proposition 1 on a stream where the change points are known.

Setup. On the CIFAR-100 permutation stream (δ=500, b=0.02, naive replay, three seeds) we record, once per micro-batch after a change and before its SGD steps, the inner products a_{old}, a_{new}, and a_{mix} of the stale-label gradient, the fresh-label gradient, and their buffer-weighted mixture (4) with ∇R_t, estimated on 256 held-out current-concept arrivals from which every queried point is excluded. A positive inner product means that a step along the negative gradient lowers the current risk to first order.

Main result. Table 10 bins the measurements by φ_t. The condition $a_{\text{old}} \leq a_{\text{new}}$ held at 973 of 974 measurements, the fresh gradient descended at the same 973, and the stale gradient pointed uphill at 95%. The mixture still descended at 90%: at 94% of the measurements with $\varphi_t \geq 0.75$, right after a change, when the head still fits the old concept and the stale gradient is a tenth of the fresh one in norm, and at 70% once φ_t fell to a quarter to a half and the norms were nearly equal.

Takeaway. On this stream stale labels rarely reverse a step of naive replay; they shorten it, increasingly so as the head adapts, which is the slow recovery of Fig. 8. The check covers one artificial stream with a linear head, so it supports the decomposition as an intuition consistent with the measurements, not as a mechanism established across the benchmark.

Table 10: Study B: alignment of stale, fresh, and mixed gradients with the current-risk gradient.

		Share of measurements (%)				
φ_t	n	$a_{\text{old}} \leq a_{\text{new}}$	$a_{\text{old}} < 0$	$a_{\text{new}} > 0$	$a_{\text{mix}} > 0$	$\|g^{\text{old}}\| / \|g^{\text{new}}\|$
0.25–0.50	50	100	100	100	70	0.92
0.50–0.75	233	100	100	100	82	0.40
0.75–1.00	691	100	94	100	94	0.10
All	974	99.9	95	99.9	90	0.15

C.1.3 STUDY C: STATIONARY CONTROLS

Shuffled streams. Destroying each stream's order with a fixed permutation removes drift while keeping the learner, protocol, and streams (Table 11). The window and reset then lose 1.3 and 0.9 points, FIFO-50 and A-GEM 3.3 and 4.2, and no acquisition strategy gains more than 0.4, so the ordering of the update rules is a property of non-stationarity.

False alarms. On the fourteen generators run with no change point (Table 12), Page–Hinkley at the default threshold λ=10 fires 0.7 times per 40,000 steps and costs the gate 0.3 points and reset 0.2; the window costs 0.7, and at λ=5 the cost is 0.7 to 0.9 points with 1.8 to 3.4 false alarms.

Takeaway. A short horizon costs accuracy where nothing drifts, which is the reason a horizon has to be chosen per stream (Section 4.4).

Table 11: Study C: accuracy (%) and Δ to the reference with each stream's order destroyed.

Decision	Method	Ordered Δ	Shuffled, δ=0 acc	Shuffled, δ=0 Δ	Shuffled, δ=500 acc	Shuffled, δ=500 Δ
Acquisition	random	–	68.4	–	67.5	–
	US	−1.5	67.9	−0.5	67.5	0.0✓
	margin	−1.1	68.7	+0.3	68.1	+0.6✓
	least-conf	−1.2	68.2	−0.2	67.7	+0.2✓
	coreset	−1.0	67.0	−1.4✗	66.1	−1.4✗
	OVBAL	−3.1	64.7	−3.7✗	64.0	−3.6✗
	RVU	−0.1	68.7	+0.3✓	67.8	+0.3✓
	split	−0.1	68.8	+0.4✓	67.9	+0.4✓
	US+detect	−0.9	68.3	−0.1	67.7	+0.2✓
	PRopagate	−1.4	67.0	−1.4	67.1	−0.4
	PAL-FS	−1.5	67.3	−1.1	66.6	−0.9✗
Update rule	window	+4.8	67.1	−1.3✗	66.2	−1.3✗
	FIFO-50	+9.4	65.1	−3.3✗	64.3	−3.2✗
	reset	+8.2	67.5	−0.9✗	66.6	−0.9✗
	A-GEM	+9.2	64.2	−4.2✗	63.4	−4.1✗

Table 12: Study C: Page–Hinkley false alarms on fourteen stationary generator streams (δ=0).

Rule	λ	detections / stream	acc (%)
naive	–	0.00	88.5
window	–	0.00	87.8
reset	5	1.78	87.7
	10	0.56	88.2
	20	0.04	88.4
PH-gate	5	3.38	87.5
	10	0.71	88.2
	20	0.04	88.4

C.2 Robustness of the main comparison

Stream sets. Table 13 repeats the comparison on all 54 streams, the strict 47, and the deduplicated 28. On all and strict sets the best acquisition strategy gains at most 0.4 points and eight of nine update rules pass the corrected test at $\delta \leq 500$. On the deduplicated set the effect sizes are similar (FIFO-50 and A-GEM +7.6 at δ=0, +5.6 at δ=500, with intervals excluding zero; A-GEM +1.7 at δ=2,000 with an interval including zero), but only four of nine update rules pass the corrected test at δ=0 and none at the longer delays, and the Nemenyi threshold of 5.6 is never met.

Normalization and implementations. Whole-stream standardization changes no mean accuracy by more than 0.3 points (Table 14). Table 15 compares earlier and corrected implementations of four baselines: the earlier coreset scored candidates once against a fixed reference sample, the earlier A-GEM drew both gradients from the same buffer, the earlier PRopagate pseudo-labeled pending queries with the head and self-trained, and the earlier PAL-FS filtered observations by the step at which their label became available, which cancels the delay from its forgetting test. Only A-GEM changes materially (68.1% to 77.3% at δ=0).

Learner. A one-hidden-layer network with 128 ReLU units (learning rate 0.1, same schedule) on a smaller grid reproduces the update-rule ordering (Table 16); under naive replay every acquisition strategy is within 1.7 points of random, and under the window and reset uncertainty sampling loses 4 to 5.

Plasticity. Raising naive replay's learning rate up to ten-fold or its SGD steps ten-fold closes only a small part of its gap to the window and reset (Table 17; earlier protocol with whole-stream standardization), so the forgetting rules' advantage is not a learning-rate artifact.

Table 13: The main comparison on four stream sets.

Stream set	δ	Acquisition (10) best (Δ)	below / sig.	Update rule (9) best (Δ)	95% CI	sig. / CD	CD
All streams (54)	0	RVU (-0.1)	10/8	FIFO-50 ($+9.5$)	[$+7.1$, $+12.1$]	8/4	4.04
	500	split (0.0)	9/4	FIFO-50 ($+7.7$)	[$+5.4$, $+10.1$]	8/2	4.04
	2,000	split ($+0.3$)	8/2	A-GEM ($+3.2$)	[$+1.6$, $+4.7$]	2/0	4.04
Temporal (50)	0	split (-0.1)	10/6	FIFO-50 ($+9.4$)	[$+6.9$, $+12.1$]	8/4	4.19
	500	split ($+0.2$)	9/4	FIFO-50 ($+7.5$)	[$+5.2$, $+10.0$]	8/2	4.19
	2,000	split ($+0.3$)	8/2	A-GEM ($+2.9$)	[$+1.3$, $+4.4$]	2/0	4.19
Strict (47)	0	split ($+0.1$)	9/2	FIFO-50 ($+9.6$)	[$+7.1$, $+12.4$]	8/2	4.33
	500	split ($+0.3$)	9/4	FIFO-50 ($+7.9$)	[$+5.5$, $+10.4$]	8/2	4.33
	2,000	split ($+0.4$)	4/2	A-GEM ($+3.2$)	[$+1.5$, $+4.9$]	2/0	4.33
Deduplicated (28)	0	split ($+0.1$)	9/2	A-GEM ($+7.6$)	[$+3.9$, $+11.4$]	4/0	5.60
	500	split ($+0.3$)	8/3	FIFO-50 ($+5.6$)	[$+2.3$, $+9.2$]	0/0	5.60
	2,000	split ($+0.4$)	5/2	A-GEM ($+1.7$)	[-0.3, $+3.8$]	0/0	5.60

Table 14: Causal versus transductive standardization (accuracy, %, δ=0).

Decision	Method	causal	transductive	difference
Acquisition	random	68.1	68.1	0.0
	US	66.6	66.8	+0.2
	margin	67.0	67.2	+0.3
	least-conf	66.9	67.1	+0.2
	coreset	67.1	67.1	0.0
	OVBAL	65.0	64.8	−0.2
	RVU	68.0	68.0	0.0
	split	68.0	68.1	+0.1
	US+detect	67.2	67.2	+0.1
	PRopagate	66.7	67.0	+0.2
	PAL-FS	66.6	66.9	+0.2
Update rule	window	72.9	72.9	0.0
	FIFO-50	77.5	77.5	0.0
	reset	76.3	76.4	+0.1
	PH-gate	75.4	75.5	+0.1
	DDM-gate	75.3	75.5	+0.2
	hybrid	75.5	75.6	+0.1
	A-GEM	77.3	77.3	0.0
	exp-decay	68.5	68.4	0.0
	ER	67.9	67.9	0.0

Table 15: Earlier versus corrected implementations of four baselines (accuracy, %).

Method	δ=0 earlier	δ=0 corrected	δ=500 earlier	δ=500 corrected
coreset	66.7	67.1	64.9	65.0
A-GEM	68.1	77.3	66.0	73.1
PRopagate	66.6	66.7	65.1	65.2
PAL-FS	66.7	66.6	65.2	64.9
Reference	*68.1*		*65.9*	

Table 16: One-hidden-layer MLP (128 ReLU units): accuracy (%) on the 50 temporal streams.

Decision	Method	δ=0 naive	δ=0 window	δ=0 reset	δ=500 naive	δ=500 window	δ=500 reset
Acquisition	random	69.3	74.1	77.7	67.0	69.8	73.3
	US	67.6	69.9	73.0	65.9	67.4	68.2
	RVU	69.0	73.9	77.7	66.9	69.7	73.2
	PRopagate	68.4	73.4	76.7	66.5	68.9	71.7
	PAL-FS	68.4	73.4	76.6	66.0	68.5	71.3
	coreset	68.7	73.8	77.8	66.5	69.4	73.0
Update rule	random, FIFO-50	79.4			74.9		
	random, A-GEM	79.2			74.7		
	random, PH-gate	77.5			72.9		
	random, hybrid	77.6			73.0		
	random, exp-decay	69.6			67.2		
	random, ER	68.7			66.6		

Table 17: Plasticity control: naive replay with a larger learning rate or more SGD steps (ten streams, δ=0).

Update rule	lr	steps / micro-batch	acc (%)	beats window
naive replay	0.3	8	75.6	2/10
	1.0	8	75.2	2/10
	3.0	8	73.7	1/10
	0.3	80	76.4	2/10
	1.0	80	75.9	2/10
window	0.3	8	81.2	0/10
	0.3	80	82.4	6/10
reset	0.3	8	82.0	4/10

C.3 FULL BENCHMARK RESULTS

Table 18 gives the bootstrap intervals, average ranks, and wins/ties/losses behind Table 1; Table 19 the factorial at δ=2,000, where the largest gain over random is 1.5 points (margin under the window, not significant); Table 20 the full memory-horizon sweep; Table 21 the half-life sensitivity of the selector; Table 22 the ranks per drift regime; and Fig. 4 and Table 23 the per-stream results at δ=0.

Table 18: Bootstrap interval, average rank, and wins/ties/losses against the reference (Nemenyi CD 4.19; ✓/✗ significantly above/below the reference, Holm-corrected Wilcoxon; † beyond the CD).

Decision	Method	δ=0 95% CI of Δ	Average rank (of 20) δ=0	δ=500	δ=2,000	δ=0 W/T/L
Acquisition	random	–	11.0	10.8	9.8	0/50/0
	US	[−2.5, −0.5]	13.7✗	12.8	10.6	11/8/31
	margin	[−2.2, −0.1]	12.5	12.1	10.2	12/11/27
	least-conf	[−2.2, −0.3]	12.8	12.6	10.3	13/10/27
	coreset	[−1.5, −0.5]	14.0✗	13.6✗	12.8✗	1/29/20
	OVBAL	[−4.2, −2.1]	16.9✗†	16.1✗†	15.4✗†	1/16/33
	RVU	[−0.9, +0.4]	11.1	9.8	8.9	11/36/3
	split	[−0.4, +0.3]	10.5	10.1	9.2	9/26/15
	US+detect	[−1.6, −0.3]	12.5✗	12.6	10.5	10/15/25
	PRopagate	[−2.6, −0.4]	13.4✗	13.1✗	11.9	8/19/23
	PAL-FS	[−2.8, −0.4]	13.3✗	14.2✗	11.1	7/24/19
Update rule	window	[+3.5, +6.3]	6.7✓†	8.0✓	12.2	34/12/4
	FIFO-50	[+6.9, +12.1]	4.9✓†	5.7✓†	7.7✓	38/5/7
	reset	[+5.5, +11.0]	6.8✓†	7.1✓	10.0	33/8/9
	PH-gate	[+5.0, +9.6]	8.2✓	8.8✓	11.0	32/12/6
	DDM-gate	[+4.8, +9.7]	8.2✓	8.0✓	10.4	32/9/9
	hybrid	[+5.1, +9.9]	7.6✓	8.4✓	10.2	32/12/6
	A-GEM	[+6.5, +12.0]	5.4✓†	5.9✓†	7.7✓	37/6/7
	exp-decay	[+0.2, +0.5]	9.1✓	9.5✓	10.2	15/35/0
	ER	[−0.6, +0.1]	11.3	11.0	10.1	2/43/5

Table 19: Gain (points) over random acquisition under six update rules at δ=2,000.

Acquisition	δ=2,000 naive	window	FIFO-50	reset	PH-gate	A-GEM
random (acc.)	61.0	59.9	63.4	62.5	62.0	63.9
US	−0.3	+1.4	−2.4✗	−4.2✗	−1.6✗	−2.5✗
margin	−0.1	+1.5	−2.1✗	−3.8✗	−1.5✗	−2.2✗
least-conf	−0.2	+1.4	−2.2✗	−4.2✗	−1.6✗	−2.4✗
coreset	−0.9✗	−0.8✗	−1.9✗	−1.3✗	−1.3✗	−1.2✗
OVBAL	−2.5✗	−2.0✗	−2.1✗	−2.1✗	−2.2✗	−2.5✗
RVU	+0.2✓	−0.3	−0.3	−0.2	−0.2	0.0
split	+0.3	−0.1	−1.0✗	−1.2✗	−1.3✗	−0.5
US+detect	−0.2	−0.3	−2.8✗	−3.6✗	−2.4✗	−2.3✗
PRopagate	−0.2	−0.7	−1.3✗	−1.4✗	−1.6✗	−1.3✗
PAL-FS	−0.7	−1.1	−1.9✗	−1.6✗	−1.7✗	−1.5✗
Best of 10	+0.3	+1.5	−0.3	−0.2	−0.2	0.0

Table 20: Memory-horizon sweep with random acquisition: accuracy (%) and average rank (of 32).

Rule	Setting	δ=0 acc	δ=0 rank	δ=500 acc	δ=500 rank	δ=2,000 acc	δ=2,000 rank
FIFO	20	77.6	10.8	73.4	11.5	63.7	14.1
capacity M	50	77.5	10.0	73.3	10.8	63.4	12.8
	100	76.3	11.8	72.2	11.6	62.3	16.0
	200	72.9	14.8	68.9	16.1	59.9	20.9
	500	68.6	21.2	65.8	20.3	59.8	20.0
	1,000	68.7	19.9	66.3	19.1	60.9	16.5
	2,000	68.1	23.7	65.9	22.3	61.0	15.8
	5,000	67.9	24.1	65.8	22.5	61.1	16.0
	∞	67.9	24.1	65.8	22.5	61.1	15.6
Decay, weighted mean	10	75.8	16.9	71.8	16.8	62.4	18.4
half-life h	25	75.7	16.2	71.7	16.2	62.2	18.5
	50	74.6	16.5	70.6	16.9	61.4	19.6
	100	72.6	17.5	68.8	18.2	60.3	19.8
	200	70.6	18.2	67.3	18.6	60.0	20.9
	500	69.1	19.8	66.4	19.8	60.5	18.8
	1,000	68.6	21.4	66.2	20.6	60.9	17.5
	2,000	68.3	22.4	66.1	21.3	61.1	16.3
Decay, weighted sampling	10	78.1	8.4	73.9	9.5	64.0	13.0
half-life h	25	77.7	7.3	73.5	9.1	63.6	11.4
	50	76.3	9.5	72.2	10.2	62.5	12.9
	100	73.8	11.5	69.9	12.3	61.1	14.6
	200	71.4	14.4	68.0	14.9	60.5	17.1
	500	69.5	17.9	66.7	18.3	60.8	16.6
	1,000	68.8	20.4	66.4	19.4	61.1	15.5
	2,000	68.4	21.7	66.1	20.6	61.1	15.3
Reference rules	naive	68.1	23.8	65.9	22.2	61.0	15.8
	window	72.9	14.8	68.9	16.1	59.9	20.9
	reset	76.3	14.3	72.2	14.1	62.5	16.6
	PH-gate	75.4	16.9	71.3	17.1	62.0	17.9
	hybrid	75.5	16.2	71.3	16.7	62.1	17.4
	A-GEM	77.3	11.5	73.1	11.4	63.9	12.8

Table 21: Half-life sensitivity of the online selector (accuracy, %, 50 temporal streams).

δ	best fixed	oracle	Selector, half-life h h=20	h=50	h=200
δ=0	77.6	78.4	78.2	78.3	78.3
δ=500	73.4	74.5	73.4	73.8	74.0
δ=2,000	63.7	65.1	62.2	62.8	63.7

Table 22: Average rank (of 20) per drift regime at δ=0.

Decision	Method	Real abrupt (2)	Real gradual (7)	Real incr. (3)	Real none doc. (10)	Synthetic abrupt (14)	Synthetic gradual (14)	Excluded class-ord. (4)
Acquisition	OVBAL	20.0	18.0	19.7	17.3	15.9	16.1	17.5
	US	17.5	15.6	16.0	10.7	13.8	13.7	17.2
	PAL-FS	14.5	13.3	14.3	13.2	13.4	12.9	11.5
	RVU	10.0	10.9	10.7	7.2	12.7	12.6	14.0
	random	11.0	8.4	9.2	8.2	12.8	13.0	11.2
Update rule	window	5.0	5.0	3.0	7.7	7.4	7.1	5.2
	reset	1.0	7.7	9.3	10.3	5.4	5.5	2.8
	PH-gate	3.5	8.6	9.7	11.3	7.5	6.9	5.8
	A-GEM	6.0	8.3	1.7	11.2	2.6	3.3	4.2

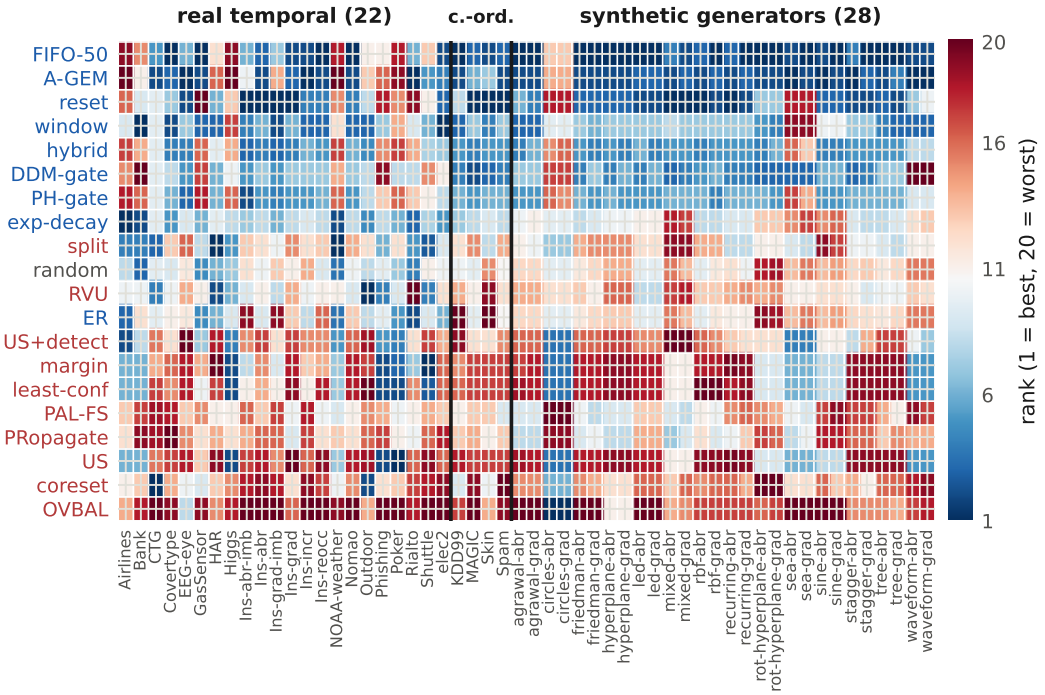

Figure 4: Per-stream rank of each method at δ=0 (rows sorted by mean rank; blue is rank 1, red rank 20); streams grouped as real temporal, class-ordered, and generators.

Table 23: Per-stream accuracy (%) at δ=0 on all 54 streams ([c] class-ordered, [P] label-persistent).

	Acquisition, with naive replay											Update rule, with random acquisition								
Stream	random	US	margin	least-conf	coreset	OVBAL	RVU	split	US+detect	PRopagate	PAL-FS	window	FIFO-50	reset	PH-gate	DDM-gate	hybrid	A-GEM	exp-decay	ER
Airlines	65.8	65.9	65.9	65.9	65.7	65.3	65.8	65.9	66.0	65.6	65.5	65.8	63.5	65.0	63.8	65.3	64.0	61.8	66.0	65.9
Bank	89.6	89.5	89.5	89.5	89.3	89.1	89.4	89.5	89.5	89.0	89.1	89.9	89.1	89.5	89.1	88.0	89.2	89.4	89.7	89.3
CTG	90.1	89.2	89.4	89.0	92.6	87.8	92.0	92.1	89.2	88.1	88.1	90.1	91.0	90.1	90.1	90.1	90.1	92.3	90.1	90.1
Covertype	80.5	78.0	78.4	78.4	79.2	77.4	80.2	79.7	79.9	77.3	77.5	83.0	84.0	80.6	80.8	80.5	80.9	82.9	80.9	79.9
EEG-eye[P]	52.0	49.2	49.2	49.2	52.1	55.0	51.6	51.1	48.6	51.4	52.0	58.3	65.9	70.7	69.0	68.4	68.9	81.7	52.6	52.0
GasSensor	80.6	78.2	78.2	78.7	79.0	72.3	78.6	80.1	80.2	78.4	77.9	80.7	81.2	71.2	75.4	75.5	75.7	81.2	80.5	80.6
HAR	85.1	78.2	73.6	79.9	83.7	81.1	85.5	86.8	78.9	83.6	84.6	85.1	84.3	85.1	85.1	85.1	85.1	80.9	85.1	85.1
Higgs	61.1	63.0	63.0	63.0	58.7	58.2	61.9	61.9	62.0	60.5	60.6	58.4	56.4	59.4	58.5	60.8	58.7	56.1	61.0	61.0
Ins-abr-imb	74.7	74.1	75.2	74.9	71.5	64.2	75.1	74.5	74.3	74.3	74.2	76.8	75.5	77.8	77.4	76.8	77.3	75.0	75.4	70.0
Ins-abrupt	56.4	54.0	55.4	55.5	52.3	50.0	56.4	56.0	55.2	55.3	55.8	66.8	68.9	70.3	67.6	65.8	67.9	69.0	57.7	57.3
Ins-grad-imb	75.0	73.9	75.3	75.1	72.9	67.4	75.6	75.1	75.0	73.9	73.8	77.3	75.6	77.4	76.9	76.8	77.1	75.0	75.7	69.3
Ins-gradual	55.9	48.5	51.8	49.8	55.7	54.7	55.6	55.1	54.1	57.3	57.0	70.2	72.3	72.6	71.6	69.5	71.7	71.8	57.8	55.6
Ins-incr	57.7	56.3	58.1	58.4	53.3	49.8	58.8	58.3	57.2	55.3	55.0	67.1	69.9	67.1	64.7	61.9	64.3	70.1	59.1	60.2
Ins-reocc	54.5	48.7	51.9	49.3	51.3	47.8	53.8	52.2	51.8	53.3	53.8	70.2	72.7	69.7	67.6	63.3	68.3	72.4	56.9	51.5
NOAA-weather	76.5	76.4	76.4	76.4	76.4	75.1	76.4	76.7	76.5	76.3	76.4	76.3	75.4	75.6	75.5	75.7	75.5	74.6	76.5	76.5
Nomao[P]	90.1	83.7	83.7	83.7	86.3	83.5	90.1	86.9	86.0	89.4	89.5	92.4	93.1	91.9	91.5	90.6	91.4	92.8	90.6	88.5
Outdoor[P]	31.2	20.7	20.8	20.6	31.7	26.9	34.6	28.6	20.7	23.8	24.2	31.3	29.7	31.2	31.2	31.2	31.3	28.2	31.3	31.2
Phishing	86.7	87.7	87.7	87.7	86.5	85.2	87.7	87.2	87.3	86.2	86.3	86.9	86.5	86.0	86.4	85.4	86.3	86.2	86.8	86.7
Poker	48.1	51.8	51.1	51.2	47.4	45.0	47.9	47.5	49.1	47.7	47.9	47.1	46.4	47.0	46.9	48.0	46.8	46.1	48.1	48.2
Rialto	34.7	28.6	33.2	29.3	26.6	26.0	18.1	34.7	30.4	31.2	30.8	31.3	34.1	25.7	29.3	32.3	29.3	38.8	34.8	35.1
Shuttle	95.5	91.4	96.9	96.8	93.0	90.3	96.5	96.8	93.9	95.2	95.3	95.5	95.3	95.5	95.5	95.2	95.5	95.6	95.5	95.5
elec2	74.9	74.5	74.5	74.5	73.7	72.3	74.7	74.9	74.5	74.4	74.5	77.5	76.8	76.1	75.0	74.7	75.1	76.0	75.5	75.3
KDD99[c]	96.9	96.3	96.6	96.6	96.9	96.4	96.7	96.8	96.3	96.8	96.8	98.2	98.6	97.0	97.4	97.5	97.4	98.4	97.1	96.2
MAGIC[c]	74.4	71.4	71.4	71.4	68.6	67.6	74.3	72.9	73.8	73.0	73.3	95.3	97.8	99.0	97.6	98.0	97.6	97.5	76.5	74.4
Skin[c]	96.8	96.7	96.7	96.7	97.3	96.9	96.6	97.4	96.9	97.0	96.9	99.4	99.6	99.7	99.3	99.6	99.4	98.8	97.3	91.2
Spam[c]	68.0	55.7	55.7	55.7	47.3	50.8	67.4	64.3	64.1	66.1	68.5	77.1	86.8	88.8	85.8	86.2	85.8	87.4	68.7	68.0
agrawal-abrupt	66.1	64.3	64.3	64.3	65.0	61.7	65.7	66.3	64.8	67.2	66.6	73.2	77.9	71.8	72.3	71.7	72.6	76.9	66.4	66.1
agrawal-gradual	65.9	64.4	64.4	64.4	65.1	61.8	65.8	66.3	64.9	67.1	66.7	72.9	77.3	71.4	71.3	70.2	71.1	74.8	66.2	65.9
circles-abrupt	85.7	85.8	85.8	85.8	85.8	85.8	85.6	85.8	85.8	68.4	66.1	85.7	85.6	85.0	85.3	85.1	85.3	85.6	85.7	85.7
circles-gradual	85.7	85.8	85.8	85.8	85.8	85.8	85.6	85.7	85.8	65.1	62.3	85.7	85.6	85.1	85.3	85.2	85.3	85.6	85.7	85.7
friedman-abrupt	67.4	66.5	66.5	66.5	67.2	66.3	67.4	67.3	66.7	67.5	67.4	71.4	76.4	73.2	72.3	72.7	72.7	75.0	67.5	67.4
friedman-gradual	67.4	66.4	66.4	66.4	67.3	66.4	67.5	67.3	66.7	67.5	67.7	71.5	75.6	72.5	71.6	70.6	71.7	73.3	67.6	67.4
hyperplane-abrupt	61.4	58.0	58.0	58.0	61.5	61.7	60.9	61.3	59.9	62.9	62.8	76.2	86.2	84.4	82.9	83.7	82.9	86.0	62.1	61.4
hyperplane-gradual	61.4	57.9	57.9	57.9	61.6	61.9	61.1	61.2	60.6	62.8	62.8	76.1	84.7	81.9	80.9	80.8	81.0	82.5	62.1	61.4
led-abrupt	73.1	68.7	68.7	68.7	69.3	59.7	74.7	70.7	69.3	69.7	70.4	80.4	87.3	86.3	84.8	85.5	84.8	86.2	72.9	73.1
led-gradual	73.1	68.9	68.9	68.9	69.3	60.7	74.6	70.8	69.3	69.9	70.6	80.5	86.6	84.8	83.9	82.8	84.0	84.2	73.0	73.1
mixed-abrupt	52.6	54.0	54.0	54.0	52.9	52.8	52.6	51.7	50.4	55.9	55.6	55.9	78.5	83.8	78.1	79.1	78.1	81.7	52.6	52.6
mixed-gradual	52.9	54.1	54.1	54.1	52.8	52.8	52.5	52.2	49.4	54.1	54.8	55.5	76.6	77.8	74.8	75.1	74.9	75.4	52.8	52.9
rbf-abrupt	49.6	47.3	47.4	46.8	48.4	47.8	49.1	49.0	48.4	49.1	49.3	60.5	71.3	72.2	68.6	68.7	68.8	71.6	50.0	49.6
rbf-gradual	49.6	47.5	47.8	47.3	48.7	48.0	49.2	49.1	48.9	49.9	49.9	60.4	69.9	68.5	66.1	64.2	66.0	68.1	50.1	49.6
recurring-abrupt	66.2	63.4	63.4	63.4	65.8	65.7	66.1	66.6	66.1	66.3	65.9	73.5	88.1	90.1	86.1	87.6	86.1	89.9	66.4	66.2
recurring-gradual	66.2	63.5	63.5	63.5	66.1	65.8	66.1	67.0	66.2	66.1	65.9	73.6	86.7	85.5	83.8	84.7	83.8	85.4	66.5	66.2
rot-hyperplane-abrupt	74.1	81.1	81.1	81.1	74.0	74.4	74.6	78.0	77.2	74.3	74.6	88.2	91.8	87.9	88.2	89.7	88.3	90.2	75.1	74.1
rot-hyperplane-gradual	74.0	80.5	80.5	80.5	74.0	74.4	74.6	77.8	77.1	74.6	74.6	88.1	91.4	87.0	87.6	89.1	87.6	89.5	75.0	74.0
sea-abrupt	92.9	93.0	93.0	93.0	93.0	92.1	93.0	93.0	93.2	93.0	93.0	92.4	95.4	92.5	92.7	94.8	92.8	95.8	92.9	92.9
sea-gradual	92.9	93.0	93.0	93.0	93.0	92.1	93.0	93.0	93.1	93.0	93.0	92.5	95.1	92.5	92.9	94.1	92.9	95.1	92.8	92.9
sine-abrupt	54.6	59.0	59.0	59.0	54.3	45.3	54.9	53.1	55.5	53.1	53.2	57.8	69.1	68.9	67.7	66.9	67.8	69.4	54.5	54.6
sine-gradual	54.6	59.1	59.1	59.1	54.7	45.4	55.0	53.4	55.3	52.7	52.3	57.6	68.1	66.1	65.0	63.3	65.3	66.2	54.6	54.6
stagger-abrupt	64.4	57.0	57.0	57.0	64.3	63.7	64.5	64.6	65.4	62.8	62.5	74.7	91.5	93.3	89.6	92.7	89.6	94.8	65.0	64.4
stagger-gradual	64.4	56.0	56.0	56.0	64.5	63.9	65.0	64.8	64.0	63.6	63.1	74.7	89.8	89.7	87.4	88.4	87.4	89.2	65.0	64.4
tree-abrupt	59.2	57.6	57.6	57.6	58.8	59.0	59.3	59.3	58.6	59.1	59.0	64.6	67.8	65.3	64.2	63.7	64.2	65.9	59.4	59.2
tree-gradual	59.1	57.3	57.3	57.3	58.9	59.1	59.3	59.4	58.6	59.0	59.1	64.6	67.0	64.9	63.9	62.0	64.0	64.5	59.4	59.1
waveform-abrupt	76.6	79.1	79.1	79.1	76.5	76.6	76.7	77.1	77.8	76.6	76.4	79.7	82.3	78.3	78.0	76.3	78.0	82.5	76.7	76.6
waveform-gradual	76.7	79.2	79.2	79.2	76.4	76.5	76.9	77.3	78.4	76.8	76.6	79.7	81.7	77.5	77.6	75.5	77.6	81.9	76.8	76.7

C.4 AUXILIARY CIFAR STUDY AND NEGATIVE RESULTS

The first version of this work studied detector-gated replay on the CIFAR-100 permutation stream with whole-stream standardization; we keep those results for completeness. Delay lowers every method but does not change the ordering (Fig. 5, Table 24); the gated rule's lead grows with the budget and with drift severity (Figs. 6 and 8, Table 25); and within the gated rule a hard gate and a Page–Hinkley threshold between 2 and 10 work best (Fig. 7, Table 26). The no-drift row of Table 25 is a single stationary run and is superseded by Table 12. Two ideas failed. Deep D-optimal selection on frozen features, with or without adaptive information-matrix forgetting, does not beat random under drift (Table 27), although entropy sampling beats random on the stationary control of the same table. Anticipatory budget timing from an unlabeled drift signal ties a trivial reactive controller (Table 28).

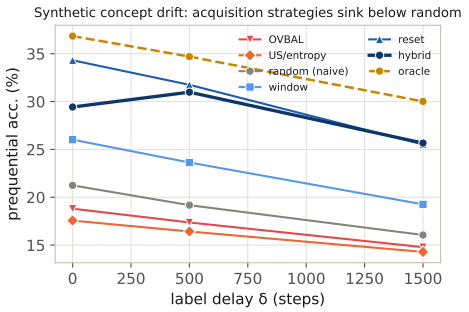

Figure 5: Auxiliary CIFAR study: accuracy against label delay (b=0.02, three seeds).

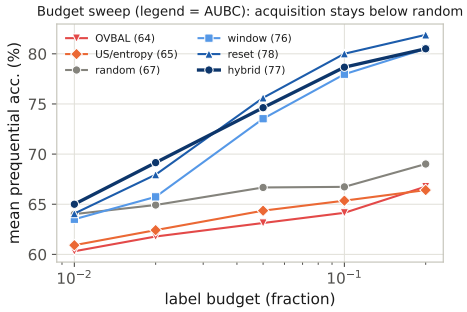

Figure 6: Budget sweep on six streams (earlier protocol, two seeds).

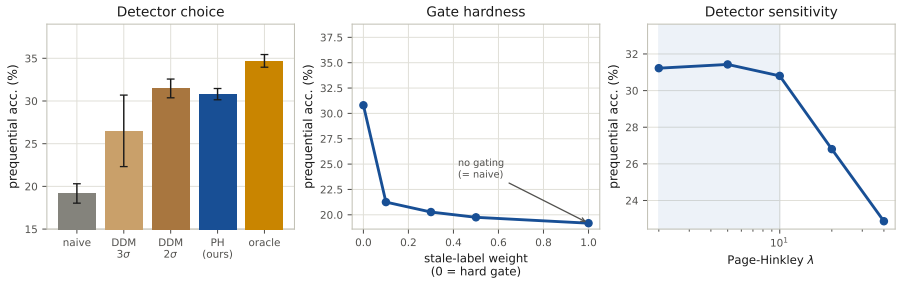

Figure 7: Auxiliary CIFAR study: detector choice (left), gate hardness (middle), and Page–Hinkley threshold (right) for detector-gated replay (δ=500, b=0.02).

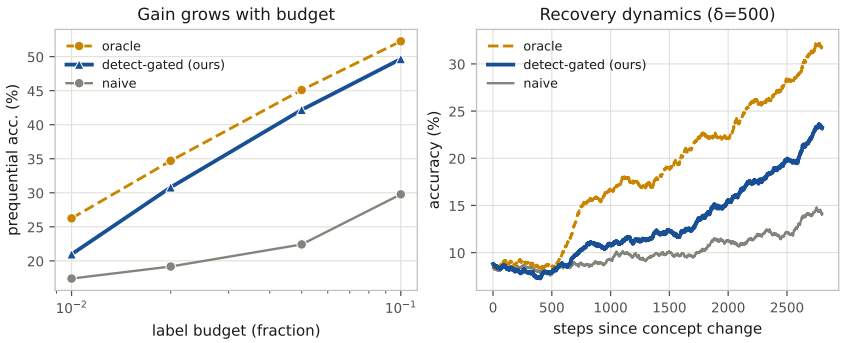

Figure 8: Auxiliary CIFAR study: accuracy against budget (left) and against steps since a change (right) for naive replay, gated replay, and an oracle gate (δ=500).

Table 24: Auxiliary CIFAR study: accuracy (%) against label delay under abrupt drift (mean $\pm$ s.e.m., three seeds).

δ	naive	age-discount	DGU (detect)	oracle
0	21.2 ± 1.5	25.4 ± 1.5	29.2 ± 1.6	36.8 ± 0.7
500	19.2 ± 1.1	22.9 ± 1.1	30.7 ± 1.3	34.7 ± 0.7
1500	16.1 ± 1.0	18.8 ± 1.1	25.1 ± 2.2	30.0 ± 0.8

Table 25: Auxiliary CIFAR study: accuracy (%) against drift severity at δ=500.

severity	naive	DGU (detect)	oracle	Δnaive	Δoracle
0.00	73.8	73.8	73.8	+0.00	+0.00
0.25	60.1	59.5	56.2	-0.54	-3.87
0.50	42.8	46.4	46.2	+3.60	+3.39
1.00	22.7	32.9	35.9	+10.14	+13.23

Table 26: Auxiliary CIFAR study: ablations of detector-gated replay (δ=500, b=0.02; "fires" counts detections for eight true changes).

(a) detector

detector	acc. (%)	fires
naive	19.2 ± 1.1	0.0
DDM (3σ)	26.5 ± 4.2	2.0
DDM (2σ)	31.5 ± 1.1	6.7
Page-Hinkley	30.8 ± 0.7	3.7
oracle	34.7 ± 0.7	0.0

(b) gate hardness

stale weight	acc. (%)
0.0	30.8 ± 0.7
0.1	21.3 ± 1.2
0.3	20.3 ± 1.1
0.5	19.8 ± 1.2
1.0	19.2 ± 1.1

(c) Page–Hinkley λ

λ	acc. (%)	fires
2	31.2 ± 0.8	12.3
5	31.4 ± 1.2	4.7
10	30.8 ± 0.7	3.7
20	26.8 ± 2.5	2.3
40	22.9 ± 3.2	1.0

(d) budget

b	naive	DGU	oracle
0.005	17.4	21.0	26.2
0.010	17.4	21.0	26.2
0.020	19.2	30.8	34.7
0.050	22.4	42.2	45.1
0.100	29.8	49.6	52.2

Table 27: Auxiliary CIFAR study: deep D-optimal point selection with and without forgetting (accuracy, %).

regime	b	random	entropy	D-opt	+fix.forget	+adapt.forget
drift	0.01	20.6	18.3	18.2	18.1	17.7
	0.02	27.2	24.2	25.7	26.4	25.7
	0.05	39.2	35.0	36.9	38.7	38.2
stationary	0.01	55.3	56.3	43.5	42.1	42.9
	0.02	62.7	66.1	56.7	55.1	56.1
	0.05	71.1	73.1	66.1	65.9	66.0

Table 28: Auxiliary CIFAR study: anticipatory versus reactive timing of the label budget (accuracy, %).

b	δ	uniform	reactive	anticipatory	oracle
0.01	0	18.2	20.9	21.6	20.0
	500	18.8	20.8	20.9	19.7
	1000	18.9	20.4	19.8	19.5
0.02	0	26.7	29.3	29.2	29.4
	500	25.8	27.5	27.5	29.6
	1000	25.1	26.6	26.7	28.3

D BENCHMARK DETAILS

D.1 STREAMS

Real streams. Of the 26 real streams, 22 are temporal: elec2 (45,312 × 7, binary), Covertype (120k×54, 7 classes), six Insects variants (Souza et al., 2020) (33 features, 6 classes: abrupt-balanced 52,848, gradual-balanced 24,150, incremental-reoccurring 79,986, incremental-abrupt 79,986, gradual-imbalanced 143,323, abrupt-imbalanced 355,275); NOAA weather (18,159 × 8, binary rain prediction from daily measurements at Offutt Air Force Base over 50 years, seasonal and long-term drift (Elwell & Polikar, 2011)); Rialto (82,250 × 27, 10 buildings seen from a webcam over 20 consecutive days under changing light, labels interleaved so that persistence is useless; Losing et al., 2016) and Outdoor (4,000 × 21, 40 objects photographed outdoors under changing light, frames grouped by object; Losing et al., 2015); Gas Sensor (13,910 × 128, 6 gases, the ten UCI batches over 36 months in batch order (Vergara et al., 2012); the OpenML copy used in an earlier version is not in batch order); and Airlines, Poker, Nomao, Bank marketing, EEG Eye State (14,980, binary), HAR (10,299, 6 classes), Shuttle (58,000, 7 classes), Phishing Websites (11,055, binary), Cardiotocography (2,126, 10 classes), and HIGGS (98,050, binary). The four class-ordered files excluded by the audit are KDD99, Spam, MAGIC (19,020, binary), and Skin Segmentation (100,000, binary); EEG Eye State, Nomao, and Outdoor are label-persistent and excluded from the strict set. All files are fetched from OpenML or UCI and played in file order; categorical features are label-encoded. Table 29 gives the provenance, label transitions, persistence-oracle accuracy, and audit flags, and Table 30 the feature-free references on the flagged streams. In Table 22 the two Insects abrupt variants form the real-abrupt group; elec2, Covertype, the two Insects gradual variants, NOAA weather, Rialto, and Outdoor the real-gradual group; the Insects incremental and reoccurring variants and Gas Sensor the real-incremental group; and the remaining ten temporal streams the group without a documented drift type.

Synthetic streams. Eight classical generators (SEA, Sine, STAGGER, Mixed, Agrawal, Hyperplane, RandomRBF, LED) and six further ones: Waveform (40-d, a binary task that thresholds a waveform-shaped combination of 21 signal features at its median, with 19 noise features; a custom generator, not the three-class UCI waveform), Circles (2-d, rotating circular boundary), Tree (10-d, random decision trees of depth 5), Rotating Hyperplane (10-d), Friedman (10-d, binarized, with shifting interaction terms), and Recurring (6-d, cyclic concepts). Each is streamed for 40k steps with eight evenly spaced change points in three realizations (generator seeds 0, 1, 2, which change the sampled features and labels, the gradual-mixing draws, and, for Hyperplane, RandomRBF, Tree, Rotating Hyperplane, and Recurring, the randomly drawn concepts; the concept order is cyclic in every realization). Abrupt variants switch instantly; in gradual variants each label within the 2k-step window centred on a change is drawn from the new concept with probability rising linearly from 0 to 1 and from the old concept otherwise (an earlier version switched the base concept at the window's midpoint). Every generator result in the main benchmark is the mean over the three realizations. The stationary control runs the same generators with no change point.

Table 29: Provenance and label structure of the real streams.

Stream	N	d	C	transitions	persistence (%)	source and order	flag
elec2	45,312	7	2	6,647	85.3	Harries (1999), NSW electricity, time-stamped	
Covertype	120,000	54	7	9,977	91.7	UCI, file order (standard drift benchmark)	
Ins-abrupt	52,848	33	6	37,534	29.0	Souza et al. (2020) Insects, sensor time order	
Ins-gradual	24,150	33	6	15,312	36.6	Souza et al. (2020) Insects, sensor time order	
Ins-reocc	79,986	33	6	47,629	40.5	Souza et al. (2020) Insects, sensor time order	
Ins-incr	79,986	33	6	46,089	42.4	Souza et al. (2020) Insects, sensor time order	
Ins-grad-imb	143,323	33	6	100,092	30.2	Souza et al. (2020) Insects, sensor time order	
Ins-abr-imb	355,275	33	6	251,682	29.2	Souza et al. (2020) Insects, sensor time order	
GasSensor	13,910	128	6	5,675	59.2	UCI gas-sensor drift (Vergara et al.), 10 batches, batch order	
Airlines	150,000	7	2	59,717	60.2	OpenML airlines, file order	
Poker	200,000	10	10	113,463	43.3	UCI poker-hand, file order	
KDD99	150,000	41	23	390	99.7	KDDCup99 (10%), file order	class-ordered
Nomao	34,465	118	2	1,781	94.8	OpenML nomao, file order	label-persistent
Bank	45,000	16	2	7,000	84.4	UCI bank-marketing, file order (2008–2010)	
Spam	4,601	57	2	1	100.0	UCI spambase, file order	class-ordered
CTG	2,126	35	10	833	60.8	UCI cardiotocography, file order	
EEG-eye	14,980	14	2	23	99.8	UCI EEG eye state, 117 s recording	label-persistent
HAR	10,299	561	6	399	96.1	UCI HAR, subject/activity order	
Higgs	98,050	28	2	48,906	50.1	UCI HIGGS, file order	
MAGIC	19,020	10	2	1	100.0	UCI MAGIC gamma telescope, file order	class-ordered
Phishing	11,055	30	2	5,372	51.4	UCI phishing websites, file order	
Shuttle	58,000	9	7	20,574	64.5	UCI statlog shuttle, file order	
Skin	100,000	3	2	1	100.0	UCI skin segmentation, file order	class-ordered
NOAA-weather	18,159	8	2	5,806	68.0	NOAA daily weather 1949–99 (Elwell & Polikar), time order	
Rialto	82,250	27	10	82,249	0.0	Rialto webcam, 20 days (Losing et al.), time order	
Outdoor	4,000	21	40	390	90.2	outdoor objects, changing light (Losing et al.), time order	label-persistent

Table 30: Feature-free references on the flagged real streams (accuracy, %, δ=0).

		Feature-free			Learner		
Flag	Stream	persistence	last label	majority-50	naive	window	reset
Class-ordered	MAGIC	100.0	99.6	98.6	74.4	95.3	99.0
	Skin	100.0	99.9	99.7	96.8	99.4	99.7
	Spam	100.0	95.2	91.6	68.0	77.1	88.8
	KDD99	99.7	95.5	93.9	96.9	98.2	97.0
Label-persistent	EEG-eye	99.8	88.3	71.4	52.0	58.3	70.7
	Nomao	94.8	88.0	88.5	90.1	92.4	91.9
	Outdoor	90.2	2.0	1.2	31.2	31.3	31.2
Temporal	HAR	96.1	10.1	13.1	85.1	85.1	85.1
	Covertype	91.7	58.6	70.9	80.5	83.0	80.6
	elec2	85.3	51.6	55.6	74.9	77.5	76.1

D.2 METHODS

Acquisition. US, margin, and least-confidence query the $\lceil bL \rceil$ points per micro-batch of highest predictive entropy, smallest top-two probability gap, and lowest maximum probability. Coreset runs greedy k-center: the centers are a random sample of up to 256 buffered points, the candidate farthest from the centers is picked and added to them, distances are updated, and the step repeats $\lceil bL \rceil$ times. OVBAL maintains A^{-1} with per-batch forgetting (γ=0.999), queries maximum $\phi^\top A^{-1}\phi$, and updates A^{-1} by Sherman–Morrison. RVU follows Žliobaitė et al. (2014): threshold θ initialized to 1, a point is queried when its margin is below $\theta\eta$ with $\eta \sim \mathcal{N}(1,1)$ truncated at 0, θ shrinks by 1% after a query and grows by 1% otherwise, and queries are allowed only while the spending is below the budget. Split draws half the micro-batch budget uniformly and the rest by entropy. US+detect runs the Page–Hinkley detector of the reset rule on the arriving-label error and queries uniformly for 1,000 steps after each detection. The latency-aware strategies share the multi-class probabilistic active-learning gain (Kottke et al., 2016): for a candidate x the kernel frequency vector $k(x) = \sum_i K(x, x_i)e_i$ is computed over a labeled set with label distributions e_i (Gaussian kernel, bandwidth the median squared distance in the micro-batch); with $n = \sum_c k_c$ and Dirichlet prior α=1, the utility is $n[\sum_c \hat{p}_c \max_j \hat{p}_j^{(+c)} - \max_j \hat{p}_j]$ with $\hat{p} = (k+\alpha)/(n+C\alpha)$. PRopagate uses the last 200 arrived labels plus every pending query, whose label distribution is a vote of its 10 nearest labeled neighbours weighted by $0.5^{\text{age}/100}$ (age in arrived labels); Castellani et al. (2022)

propagate a hard majority vote with weights that decay in wall-clock time, so ours is an adaptation. PAL-FS keeps an observation queried at t_i, or a pending query, only if $t_i \geq t + \delta - w$ with $w{=}200/b$ steps (at most 200 observations), and adds the surviving pending queries with the head's softmax as simulated labels; at $\delta{=}2{,}000$ and $b{=}0.1$ nothing survives and it falls back to random selection.

Update rules. ER keeps a reservoir sample of all history. Exponential decay weights labels by $\rho^{\mathrm{age}/\tau}$ ($\rho{=}0.9$, $\tau{=}300$), and the half-life sweep by $0.5^{\mathrm{age}/h}$ over an unbounded buffer; weighted updates use the self-normalized mean of (3) over a uniformly sampled minibatch, and the weighted-sampling variant draws the minibatch in proportion to the weights and takes an unweighted step. Reset and the gates run Page–Hinkley on the arriving-label error e_j: with running mean $\bar{e}_n$ and cumulative $m_n = \sum_{i \leq n}(e_i - \bar{e}_i - \eta)$, a detection fires when $m_n - \min_i m_i > \lambda$; reset then wipes buffer and head, while gating increments a concept-epoch counter and drops labels from earlier epochs (detector-gated replay is called DGU in the auxiliary tables). DDM-gate substitutes the DDM detector (Gama et al., 2004). A-GEM computes its main gradient on the labels that arrived since the previous micro-batch (at most 64), its reference gradient on a random sample of the remaining buffer, and projects the main gradient onto the half-space of nonnegative inner product with the reference gradient.

Horizon selection. The five sub-learners of Section 3.3 receive every purchased label and run the reference schedule. Error estimates start at 0.5, the unbounded buffer predicts until the first label arrives, and the half-life of (5) is 50 labels in the main setting, with 20 and 200 as sensitivity (Table 21).

D.3 Protocol, selector, and one benchmark cell

In this section, we give the online protocol that every benchmark cell runs (Fig. 9), the online horizon selector of Section 3.3 (Fig. 10), and one cell of the benchmark as the runner receives it and as the results file records it (Fig. 11). The listings follow the released code (https://anonymous.4open.science/r/active-learning-cd-iclr27-1342/); ASCII names replace the symbols of the main text (δ is `delta`, ϵ_k is `err_k`).

```
INPUT   a stream (x_1, y_1), ..., (x_N, y_N), played once in order; acquisition rule A; update rule U;
        budget b; label delay delta; micro-batch size L = 100; seed
STATE   linear-softmax head W = 0;   buffer B = empty (capacity M);   pending queue Q = empty

for each micro-batch S = (x_s, ..., x_e) of L consecutive arrivals:

  ## 1. Predict. Every arrival is scored before any label of S exists (test, then train).
  correct[t] = [ argmax_c (W x_t)_c == y_t ]                          for t in S

  ## 2. Acquire. A picks ceil(b * L) points of S using W, the buffer, and the features of the
  ##    queued points; it never sees a label that has not been released.
  Q = Q + {(t + delta, x_t, y_t)}                                      for each picked t

  ## 3. Release. Every queued label whose ready time is at most e enters the buffer, in query order.
  for each (r, x, y) in Q with r <= e:
      if U has a detector:  feed it the error of W on (x, y); on a detection,
                            reset: empty B and set W = 0;   gate: start a new concept epoch
      append (x, y) to B                (FIFO: the oldest label leaves a full buffer; ER: reservoir)

  ## 4. Update. Eight SGD steps with learning rate 0.3 on the cross-entropy of a minibatch of 64.
  repeat 8 times:
      draw the minibatch from B          (gate: labels of the current epoch only;
                                          exp-decay: weights rho^(age / tau);
                                          A-GEM: the newest labels, projected against a replay gradient)
      W = W - 0.3 * gradient

OUTPUT  prequential accuracy = mean of correct[t] over the stream; a cell reports its mean and standard
        error over seeds 0, 1, 2
```

Figure 9: Online protocol of one benchmark cell.

```
INPUT   capacities M_1, ..., M_5 = 20, 50, 200, 2000, unbounded;   half-life h = 50 labels
STATE   five heads W_k = 0, each with its own FIFO buffer B_k of capacity M_k;
        error estimates err_k = 0.5;   beta = 1 - 2^(-1/h);   current head k* = the longest horizon

for each micro-batch S:
  every head predicts every arrival of S; the learner outputs the prediction of head k*
  query ceil(b * L) arrivals of S uniformly at random; store the five predictions with each query
  for each label (x, y) released after its delay:
      err_k = (1 - beta) * err_k + beta * [ stored prediction of head k != y ]        for k = 1, ..., 5
      append (x, y) to every buffer B_k
  k* = argmin_k err_k                                                  (ties go to the longest horizon)
  every head takes eight SGD steps on minibatches of 64 from its own buffer

OUTPUT  prequential accuracy of the selected predictions; the accuracy of every fixed head in the same
        run gives the fixed horizons and the per-stream oracle
```

Figure 10: Online horizon selector.

```
JOB (built by scripts/run_benchmark.py and executed by task(job); shown as JSON):
{
  "arm": "factorial", "dataset": "sea-abrupt", "kind": "syn", "key": ["sea", "abrupt"],
  "type": "syn-abr", "bud": 0.05, "method": "random|FIFO-50",
  "spec": {"acq": "random", "upd": "naive", "cap": 50},
  "delay": 500, "norm": "causal", "rseed": 0, "acq": "random", "upd": "FIFO-50"
}

RESULT (one of the 40,706 rows of results/bench_v2.json):
{
  "arm": "factorial", "dataset": "sea-abrupt", "type": "syn-abr", "delay": 500, "budget": 0.05,
  "method": "random|FIFO-50", "acq": "random", "upd": "FIFO-50",
  "cap": null, "tau": null, "lam": null, "norm": "causal", "rseed": 0,
  "acc": 93.665, "sem": 0.09697508271028463, "fires": 0.0
}
```

Figure 11: One benchmark cell: the job and the logged result. The smoke test of the released code recomputes this row to within 10^{-9}.

D.4 HYPERPARAMETERS

Fig. 12 lists every hyperparameter. All are global across streams, and all results are emitted from JSON logs released with the code.

```
## learner and schedule
head            linear-softmax, SGD, learning rate 0.3
steps           8 minibatches of 64 per micro-batch
micro_batch     L = 100 arrivals
buffer          M = 2000 labels (window: 200; FIFO-50: 50; horizon sweep: 20 to unbounded)

## budget, delays, seeds
budget          b = 0.1 (real streams), 0.05 (synthetic streams)
delays          0, 500, 2000 steps
seeds           3 per cell; every generator stream also in 3 realizations

## detectors (reset, PH-gate, DDM-gate, hybrid, US+detect)
page_hinkley    lambda = 10, eta = 0.005
ddm             kappa = 2.5
warm_up         200 labels in the buffer before the first detection
cool_down       400 steps between detections

## update rules
recency         rho = 0.9, tau = 300 labels (exp-decay and the hybrid gate)
a_gem           main gradient on the newest labels (at most 64), reference on a buffer sample

## acquisition
ovbal           forgetting gamma = 0.999
coreset         256 buffered points as the initial centers
rvu             threshold step 1%, noise N(1, 1)
us_detect       random queries for 1000 steps after a detection
propagate       10 neighbours, vote half-life 100 labels, the last 200 labels plus the pending queries
pal_fs          window 200 labels, Dirichlet prior alpha = 1

## online horizon selector
capacities      20, 50, 200, 2000, unbounded
half_life       50 labels (20 and 200 as sensitivity); initial error 0.5
```

Figure 12: Hyperparameters of the benchmark.

D.5 COMPUTE

All benchmark experiments are pure NumPy with manual gradients, with no GPU and no automatic-differentiation framework in the online loop. The main grid ($40{,}706$ jobs of three seeds each, with three realizations of every generator stream) completes in about 8 minutes on a 200-worker CPU pool; the positive and low-budget controls ($1{,}892$ jobs), the milder-drift control ($1{,}188$ jobs), and the horizon-selection runs ($1{,}166$ jobs, five sub-learners each) take about a minute each, and the MLP grid ($5{,}280$ jobs) about 2 minutes. A PyTorch version of the MLP loop on A100 GPUs was slower per row than the NumPy version, because a 128-unit model is bounded by kernel-launch latency, and was abandoned. The only GPU use is a one-time frozen ResNet-18 feature extraction; no backbone is trained anywhere in the paper.