SELECTIVE CONTEXT RETENTION FOR TABULAR FOUNDATION MODELS ON DATA STREAMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Tabular stream learning requires predictions on sequentially arriving examples under bounded memory and potential distribution shift. Standard stream learning methods operate by updating model state, such as split statistics or ensemble members. Recently, tabular foundation models (TFMs) that make predictions conditioned on a labeled context in an in-context manner have emerged as a natural alternative for stream learning. This turns stream adaptation via model updates into bounded context management, where the system must decide which examples to keep in the context. In this paper, we propose a future-information view that measures the usefulness of a context by the information it provides about near-future queries. This motivates preserving recent examples, retaining uncertain examples, and removing redundant examples. We instantiate these signals as RECAP (RETained Context for Adaptive Prediction), a simple context policy with entropy-gated admission and redundancy-aware eviction. RECAP achieves the highest prequential accuracy against six classical stream learners on seven streams. Under the same TFM backbone and context budget, it improves on DualFIFO, a prior context management approach. This advantage persists under noisy feedback, with gains of 0.27 to 1.95 percentage points over DualFIFO across three streams when 30% of feedback labels are corrupted. Code and datasets are available at link.

1 INTRODUCTION

Tabular stream learning studies supervised prediction from examples generated and observed sequentially as a data stream (Aggarwal, 2007). As examples arrive, predictions must be made before their labels become available, under bounded memory and real-time response requirements (Nguyen et al., 2015; Korycki & Krawczyk, 2022). This setting is challenging because the data distribution may drift over time (Gama et al., 2014), while storing the full history or repeatedly retraining on all past data is impractical (Babcock et al., 2002; Read & Zliobaite, 2025). Many existing approaches adapt to data streams by incrementally updating the model state. Hoeffding-tree methods update sufficient statistics and use statistical tests to grow decision trees (Domingos & Hulten, 2000; Manapragada et al., 2018; Hulten et al., 2001), while ensemble methods maintain multiple online learners through resampling, randomization, and replacement (Bifet et al., 2010a; Gomes et al., 2017; 2019).

Recently, tabular foundation models (TFMs) have emerged as a general-purpose interface for tabular tasks. Given a labeled context D and a query x, TFMs directly produce a posterior predictive distribution $q_\theta(\cdot \mid x, D)$ over the query's target y without requiring dataset-specific model updates (Hollmann et al., 2022). These models have demonstrated strong performance against well-trained and tuned baselines on tabular benchmarks (Hollmann et al., 2025; Zhang et al., 2025).

The ability to adapt to new datasets through context makes TFMs appealing for stream learning. Since a TFM can remain frozen throughout the stream, it adapts to evolving data by updating the retained context (Lourenço et al., 2025). The central challenge of stream learning therefore shifts from updating the model to managing its context (Figure 1). However, the context has a limited capacity, so a learner must decide which past examples to retain and which to discard, a problem we refer to as *bounded context management*. Recent work has shown that FIFO-based context policies can enable TFMs to outperform traditional stream learning approaches (Lourenço et al., 2026), but it remains unclear which past examples should be retained beyond the most recent window.

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.

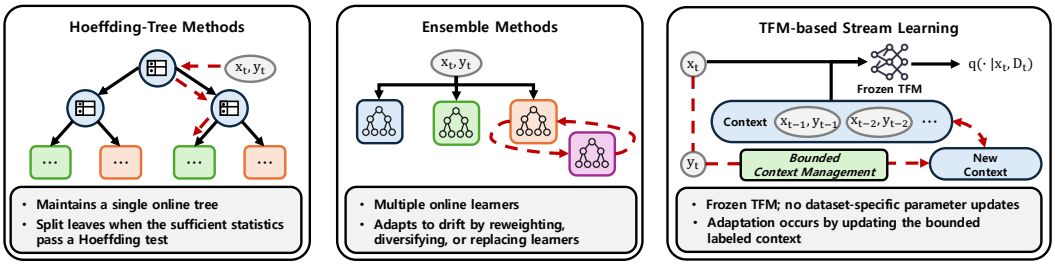

Figure 1: Adaptation mechanisms in stream learning. Classical tree and ensemble baselines adapt to new examples by updating sufficient statistics or ensemble states. TFM-based stream learning instead keeps the backbone frozen and adapts through bounded context management.

Motivated by this gap, we introduce a *future-information view* that quantifies the usefulness of a context through the information it provides about near-future queries. Under explicit assumptions, this view motivates preserving recent examples to approximate the near-future distribution, admitting uncertain examples with potentially high marginal information, and removing redundant examples with overlapping information. Based on this view, we propose RECAP (REtained Context for Adaptive Prediction), a context management policy with recency-aware short memory, entropy-gated long-memory admission, and same-class redundancy-aware eviction within the long bank.

Using a single fixed uncertainty threshold, RECAP achieves higher prequential accuracy than six classical stream learners on all seven streams. Selecting the threshold from an initial prefix also yields six wins and one loss against DualFIFO (Lourenço et al., 2026), a prior context policy based on two FIFO buffers. Together, these results support selective context retention as an effective mechanism for applying frozen TFMs to data streams.

2 Background and Problem Setup

2.1 Tabular Stream Learning

In this work, we consider stream classification under the test-then-train prequential protocol (Gama et al., 2009; 2013). At each step t, the learner predicts a label distribution for the incoming query x_t using its current model state. The model state is then updated using the true label y_t, if necessary, before the next query x_{t+1} arrives. Tree-based methods have been the central approaches. For example, VFDT (Domingos & Hulten, 2000) maintains node-level sufficient statistics and grows decision trees incrementally. Ensemble methods such as BOLE (de Barros et al., 2016), Adaptive Random Forests (Gomes et al., 2017), and Streaming Random Patches (Gomes et al., 2019) maintain multiple online learners and use mechanisms such as resampling, randomization, or drift-aware replacement. Recent works have extended toward dynamic ensemble diversification (Abadifard et al., 2023), online concept-drift detection (Wan et al., 2024), and neural classifiers such as EODL (Su et al., 2024). Appendix B.3 describes the six classical learners used in our experiments.

2.2 Tabular Foundation Models for Data Streams

Tabular foundation models (TFMs) are pretrained in-context predictors that use labeled examples to predict targets for new queries without dataset-specific parameter updates. They follow the concept of prior-data fitted networks, where a transformer is pretrained to approximate Bayesian posterior predictive inference from a labeled context (Müller et al., 2021). At stream step t, a TFM uses a bounded context of past labeled examples D_t to produce a predictive distribution $q_\theta(\cdot \mid x_t, D_t)$ over the class label of query x_t. Once y_t is observed, the labeled example $z_t = (x_t, y_t)$ becomes a candidate for future contexts. DualFIFO (Lourenço et al., 2026) is an existing context policy that uses a recency-aware short context and a class-coverage-aware long context. Appendix B.2 describes the TFM backbones, and Appendix B.4 details DualFIFO.

2.3 Problem Setup

Given a data stream, we formally define the task of updating the in-context training set of a frozen TFM as *bounded context management*. We consider numerical inputs $x_t \in \mathbb{R}^d$ and class labels $y_t \in \{1, \dots, C\}$ for a fixed $C \geq 2$, with immediate feedback after prediction.

Problem. Given a pretrained TFM, a prequential stream $\{(x_t, y_t)\}_{t \geq 1}$, and a context budget B, the problem is to design a context update policy $\mathcal{U}$ that decides whether the newly labeled example $z_t = (x_t, y_t)$ should be included in the next context, using only information available up to time t. Let M_t contain the context $D_t = D(M_t)$ and its stored metadata, and let $p_t = q_\theta(\cdot \mid x_t, D_t)$ be the prediction before observing y_t. The update must satisfy

$$M_{t+1} = \mathcal{U}(M_t, z_t, p_t), \qquad D_{t+1} = D(M_{t+1}), \qquad |D_{t+1}| \leq B. \tag{1}$$

The goal is to maximize the prequential performance of the TFM on future stream examples.

3 A Future-Information View of Bounded Context

A key question in context management is deciding which examples should be kept in the context and which should be removed when the budget is full. To answer this question, we formalize the usefulness of a context in terms of the near-future information that it provides. The assumptions and proofs for the lemmas and theorem in this section are provided in Appendix A.

3.1 Future-Information View

Let t be a stream step, and h be a short horizon length. We write $\mathcal{H}_t^+ = \{t+1, \dots, t+h\}$ for the corresponding near-future steps. For each step $s \in \mathcal{H}_t^+$, let $\mathcal{P}_{s,X}$ denote the feature marginal at that step. Then, the near-future feature distribution is the average feature marginal over this horizon

$$\mathcal{P}_{t,X}^+ = \frac{1}{h} \sum_{s \in \mathcal{H}_t^+} \mathcal{P}_{s,X}. \tag{2}$$

For a near-future query $x \sim \mathcal{P}_{t,X}^+$, let Y_x' denote a random variable corresponding to its label. For a candidate context $D_t = \{(x_j, y_j)\}_{j \in A}$, let A denote the indices of its examples and $Y_A = (Y_j)_{j \in A}$ their label random variables. We evaluate how much observing these labels would reduce uncertainty about a future query's label. For this calculation, we condition on a common background $\mathcal{F}$ that fixes the context and candidate features but leaves their labels random. We define the future usefulness of a context using the following information measure.

Definition 1 (Future usefulness of a bounded context)**.** *For a candidate context D_t with fixed example indices A, we define its future usefulness as*

$$\mathcal{J}_t(A) = \mathbb{E}_{x \sim \mathcal{P}_{t,X}^+} \left[I(Y_A; Y_x' \mid \mathcal{F}, x) \right]. \tag{3}$$

The term $I(Y_A; Y_x' \mid \mathcal{F}, x)$ measures how much the context labels inform the label of a future query x. Thus, a good context management policy should construct contexts with large future information.

However, Definition 1 is a measure for the entire context, not for an individual labeled example where the decision is actually made. Moreover, computing Equation 3 requires knowing $\mathcal{P}_{t,X}^+$, which is not available at time t. Our goal is thus to turn the measure into instance-level decisions on which new examples to admit and which stored examples to remove. We derive practical signals to instantiate the measure as a context update policy in Sections 3.2 to 3.4. Appendix A.1 details the conditional law underlying these results and the scope of the fixed-design analysis.

3.2 Recency for Future-Distribution Approximation

The first requirement lies in the outer expectation in Definition 1, which depends on the unavailable near-future feature distribution $\mathcal{P}_{t,X}^+$. In stream learning, recent past examples are commonly used to capture the current data-generating concept (Losing et al., 2017; Chen et al., 2020), and examples close in time are expected to be more related under concept drift (Read, 2018). This motivates using recent past data as an empirical proxy for near-future feature regions.

Definition 2 (Recent-window approximation). *Let $S_t = \{z_{t-|S_t|}, \ldots, z_{t-1}\}$ be the window of the most recently acquired labeled examples before time t, and let $\widehat{\mathcal{P}}_{S_t}$ be its empirical feature distribution. Replacing the outer distribution $\mathcal{P}^+_{t,X}$ with $\widehat{\mathcal{P}}_{S_t}$ gives the approximated future usefulness*

$$\widehat{\mathcal{J}}_{S_t}(A) = \mathbb{E}_{x\sim\widehat{\mathcal{P}}_{S_t}}\left[I(Y_A; Y'_x \mid \mathcal{F}, x)\right]. \tag{4}$$

This approximation is useful only if the replacement does not change the future-information objective too much. Lemma 1 shows that this gap is controlled by the distributional discrepancy between the recent window and the near future under the local stability assumption.

Lemma 1 (Recent-window approximation). *For a fixed context design A, suppose the query-wise information function $f_A(x) = I(Y_A; Y'_x \mid \mathcal{F}, x)$ is K-Lipschitz in the representation space. Under the recent-window local stability assumption, stated as Assumption 2 in Appendix A.2,*

$$\left|\mathcal{J}_t(A) - \widehat{\mathcal{J}}_{S_t}(A)\right| \le Kr_t, \tag{5}$$

where $r_t \ge 0$ bounds the Wasserstein-1 distance from $\widehat{\mathcal{P}}_{S_t}$ to $\mathcal{P}^+_{t,X}$.

Thus, when Kr_t is small, the recent-window objective is a reliable surrogate for the ideal objective. This provides a concrete design signal that a bounded context should reserve part of its budget for recent examples so that the context covers regions where near-future queries are likely to appear.

3.3 Uncertainty for High Marginal-Information Contribution

Recency indicates where future queries may emerge, but context admission requires assessing how much a new example would increase future information beyond the current context.

Definition 3 (Marginal future information). *For a newly observed candidate $z_i = (x_i, y_i)$ with $i \notin A$, let Y_i be its label random variable before its true label is observed, with x_i included in the common background $\mathcal{F}$. The marginal future information of adding i to A is*

$$\Delta_t(i \mid A) = \mathcal{J}_t(A \cup \{i\}) - \mathcal{J}_t(A). \tag{6}$$

The following lemma expresses this marginal contribution as the information that a candidate's label provides about near-future labels beyond the current context.

Lemma 2 (Item contribution as future-label information). *For a fixed candidate $i \notin A$, under the common conditional law and feature background $\mathcal{F}$,*

$$\Delta_t(i \mid A) = \mathbb{E}_{x\sim\mathcal{P}^+_{t,X}}\left[I(Y_i; Y'_x \mid Y_A, \mathcal{F}, x)\right]. \tag{7}$$

The conditional mutual information in Lemma 2 cannot be evaluated directly online. To obtain a practical signal, we assume local entropy consistency and local label coherence, meaning that uncertainty is consistent within a local feature region and nearby examples share label information (Assumptions 3 and 4 in Appendix A.4). Under these assumptions, Theorem 1 lower bounds a candidate's ideal local contribution in terms of its conditional label entropy.

Let $\mathcal{N}_i$ denote a fixed local feature region around x_i, called the candidate's effective future region, where its revealed label can provide evidence for near-future queries.

Theorem 1 (Entropy lower bound for local information). *For a candidate i and its effective future region $\mathcal{N}_i$ with near-future mass $\alpha_t(i) > 0$, under Assumptions 3 and 4, the expected local information gain satisfies*

$$\mathbb{E}_{x\sim\mathcal{P}^+_{t,X}}\left[\mathbb{I}\{x \in \mathcal{N}_i\} I(Y_i; Y'_x \mid Y_A, \mathcal{F}, x)\right] \quad \ge \alpha_t(i)\left[H(Y_i \mid Y_A, \mathcal{F}) - \delta - \epsilon\right], \tag{8}$$

where $\alpha_t(i) = \Pr_{x\sim\mathcal{P}^+_{t,X}}[x \in \mathcal{N}_i]$, and δ and ϵ are nonnegative slack variables.

Theorem 1 shows that higher conditional label entropy at x_i implies a larger lower bound on local future information when the neighborhood mass and slacks are held fixed. This matches the intuition that, when an uncertain example lies near a class boundary, its revealed label can help resolve nearby future predictions. We therefore use prediction-time entropy as a signal for admitting examples into the context. Appendix A.6 distinguishes the TFM's predictive entropy from the conditional label entropy in the theorem and discusses the quantities omitted by the deployed gate.

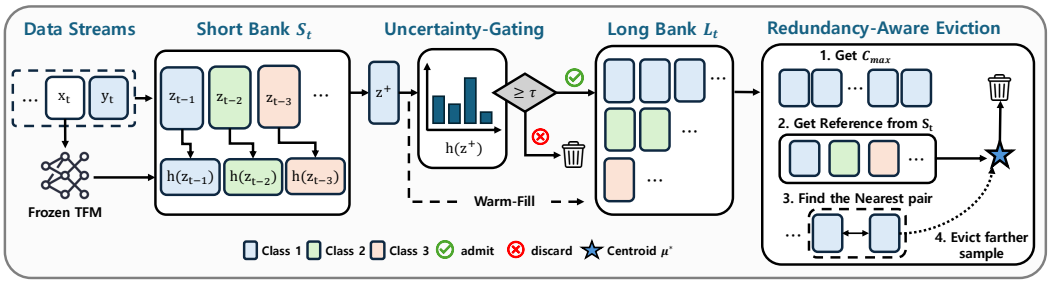

Figure 2: Overview of RECAP. A new labeled example $z_t = (x_t, y_t)$ first enters the short bank S to approximate the near-future distribution. When S overflows, the oldest item z^+ becomes a long-bank candidate and is admitted to the long bank L under warm fill or uncertainty gating. When L exceeds its budget, a locally redundant same-class example is removed.

3.4 Redundancy for Low-Information-Loss Removal

Uncertainty provides an admission signal for a new sample, but a bounded context also needs a removal signal. Under the same future-information view, the ideal removal rule deletes the stored item that causes the smallest loss of future information

$$j^* \in \arg\min_{j \in A} \ell_t(j \mid A), \qquad \ell_t(j \mid A) = \mathcal{J}_t(A) - \mathcal{J}_t(A \setminus \{j\}). \tag{9}$$

Like the admission criterion, this loss depends on unavailable future-label information. We therefore use redundancy as a tractable proxy for low contribution. Consider two stored context items $z_i = (x_i, y_i)$ and $z_j = (x_j, y_j)$ with $i, j \in A$, $i \neq j$, and $y_i = y_j$. If they are close in representation space, their evidence for future labels may overlap. We state this as an additional assumption for the specified conditional law and a fixed context design.

Assumption 1 (Same-class evidence redundancy)**.** *For two distinct context examples $i, j \in A$ with the same observed label, assume that the information lost by removing j is bounded by its distance to the retained example i,*

$$\mathcal{J}_t(A) - \mathcal{J}_t(A \setminus \{j\}) \leq \eta(\|\phi(x_i) - \phi(x_j)\|_2)\,, \tag{10}$$

where η is nondecreasing and $\eta(r) \to 0$ as $r \to 0$.

Assumption 1 says that if a stored example has a close same-class neighbor, then removing it causes only a small loss of future information under the specified law. This motivates removing close same-class examples from the context when the budget is full. Appendix A.5 explains the scope of this assumption and its relation to the class-based eviction rule.

4 Proposed Method

Motivated by the future-information view, we introduce RECAP, a simple context management policy for stream learning with TFMs. Figure 2 summarizes its overall process. A newly observed labeled example enters the short bank for recency. When the short bank overflows, its oldest item becomes a long-bank candidate and is filtered by uncertainty-gated admission. When the long bank is full, redundancy-aware eviction is conducted after admitting a candidate.

4.1 Dual Context with Short and Long Banks

RECAP maintains a dual context $D_t = S_t \cup L_t$ such that $|S_t| \leq B_S$, $|L_t| \leq B_L$, and $B_S + B_L = B$, where S_t is a FIFO-style short bank, L_t is a long bank, and B is the context budget. The short bank always contains the most recent labeled examples as a sliding window, to instantiate the recency-for-future-approximation view. In contrast, the long bank admits and evicts curated examples beyond the immediate recent window, reflecting the marginal information contribution via uncertainty-based addition and low-information-loss removal by redundancy. The banks have disjoint example identities, with capacities $B_S = \lfloor \rho B \rfloor$ and $B_L = B - B_S$. During an update, working banks S, L become S_{t+1}, L_{t+1} after the short-bank update, long-bank admission, and any required eviction.

4.2 Uncertainty-Gated Long-Bank Admission

The uncertainty bound in Theorem 1 motivates using entropy as a context admission signal. RECAP applies this signal only to the long bank. Therefore, long-bank capacity is reserved for labeled examples that are worth keeping beyond the immediate recent window.

When a stream example z_i is first predicted, the TFM outputs the posterior predictive distribution $p_i = q_\theta(\cdot \mid x_i, D_i)$ before observing y_i. We store its normalized predictive entropy

$$h(z_i) = h_i = \frac{-\sum_{c=1}^{C} p_i(c) \log p_i(c)}{\log C}, \tag{11}$$

where C is the number of classes. This score is computed on the fly and attached to the example. Therefore, if the example becomes a long-bank candidate, the stored entropy is used.

Let $z^+ = (x^+, y^+)$ be the item that overflows from the short bank. We use a *warm-fill* phase where candidates are inserted into the long bank unconditionally until $|L| = B_L$. This avoids applying the entropy gate before obtaining enough support. After this point, RECAP switches to selective admission with uncertainty gating, and a candidate is admitted only if its stored entropy meets the threshold τ

$$\mathrm{Admit}(z^+) = \begin{cases} 1, & |L| < B_L, \\ \mathbb{I}\{h(z^+) \geq \tau\}, & |L| \geq B_L. \end{cases} \tag{12}$$

Thus, the long bank preferentially retains examples that were uncertain when first observed, using predictive entropy as a practical proxy for potentially high marginal information.

4.3 Redundancy-Aware Long-Bank Removal

The redundancy assumption (Assumption 1) motivates removing close same-class examples when they provide overlapping evidence for future labels. When the long bank exceeds its budget, RECAP selects the most represented class $c_\star \in \arg\max_c |\{z \in L : y_z = c\}|$. This class-aware step reduces the risk of losing sparse class signals from the long bank. Within this class, RECAP finds the nearest same-class pair in representation space

$$(i_\star, j_\star) \in \arg \min_{\substack{i \neq j \\ z_i, z_j \in L \\ y_i = y_j = c_\star}} \|\phi(x_i) - \phi(x_j)\|_2. \tag{13}$$

Here, $\phi(x)$ is the standardized raw-feature representation used for distance computation.

After identifying the redundant pair, RECAP decides which endpoint to remove using the recent short bank as a reference for the current regime. Let $S(c_\star) = \{z_i \in S : y_i = c_\star\}$ be the examples of class $c_\star$ in the updated short bank. We define the reference centroid

$$\mu^* = \begin{cases} \frac{1}{|S(c_\star)|} \sum_{z_i \in S(c_\star)} \phi(x_i), & |S(c_\star)| > 0, \\ \frac{1}{|S|} \sum_{z_i \in S} \phi(x_i), & |S(c_\star)| = 0. \end{cases} \tag{14}$$

The policy evicts the endpoint farther from this centroid

$$u^* \in \arg \max_{u \in \{i_\star, j_\star\}} \|\phi(x_u) - \mu^*\|_2, \qquad L \leftarrow L \setminus \{z_{u^*}\}. \tag{15}$$

5 Experiments and Robustness

We first compare RECAP with classical stream learners to evaluate the predictive performance of the complete TFM-based system. We then isolate policy gains under a global threshold and examine robustness to context configuration.

5.1 Evaluation Protocol

We evaluate five real-world streams from the USP Data Stream repository[1] (NOAA, METER, RIALTO, POSTURE, and POKER), one additional real-world stream (NOMAO), and the synthetic

[1] https://sites.google.com/view/uspdsrepository

Table 1: **Prequential accuracy against six tuned classical stream learners.** Bold denotes the largest row accuracy or lowest mean rank. Ensembles report means and standard deviations over five seeds. EFDT, VFDT, and RECAP use deterministic fixed-order settings. The comparison includes differences in pretraining and inference resources.

Stream	RECAP	ARF	BOLE	LevBag	SRP	EFDT	VFDT
NOAA	**81.82**	79.42 ± 0.08	74.97 ± 0.19	79.40 ± 0.10	79.43 ± 0.10	75.56	75.66
METER	**90.68**	68.27 ± 0.09	68.06 ± 0.27	69.40 ± 0.25	71.49 ± 0.18	61.39	55.44
RIALTO	**92.00**	63.40 ± 0.12	53.62 ± 0.32	64.13 ± 0.18	72.45 ± 0.15	58.73	40.46
POSTURE-No8	**62.03**	59.44 ± 0.08	54.60 ± 0.38	60.52 ± 0.04	58.88 ± 0.15	55.34	53.11
POKER	**99.60**	84.05 ± 0.11	88.46 ± 0.19	97.53 ± 0.10	89.29 ± 0.16	82.75	89.09
NOMAO	**97.87**	97.19 ± 0.04	95.82 ± 0.12	97.27 ± 0.03	97.42 ± 0.04	94.55	93.98
AGR(A)	**90.52**	90.26 ± 0.08	88.29 ± 0.10	87.62 ± 0.29	90.13 ± 0.15	75.15	76.32
Mean rank	**1.00**	3.71	5.43	3.14	2.57	6.00	6.14

Table 2: **Prequential accuracy of RECAP, DualFIFO, and context-policy variants.** All policies use TabICL-v2 with $B = 1000$ and $\rho = 0.75$, with a global threshold $\tau = 0.3$ for gated admission. Bold indicates the highest accuracy or lowest mean rank in each row.

Stream	RECAP	DualFIFO	Entropy-only	Redundancy-only	RECAP-Margin	RECAP-GlobalNN
NOAA	**81.82**	81.42	81.61	81.64	81.73	81.76
METER	**90.68**	90.60	89.88	90.63	90.47	90.58
RIALTO	**92.00**	91.72	91.30	91.97	91.84	91.86
POSTURE-No8	**62.03**	61.85	61.83	62.01	61.93	61.79
POKER	**99.60**	98.87	98.79	98.71	99.05	99.14
NOMAO	**97.87**	97.47	97.65	97.64	97.81	97.75
AGR(A)	90.52	90.45	90.41	**90.54**	90.34	90.22
Mean rank	**1.14**	4.43	5.00	3.14	3.57	3.71

abrupt-drift stream AGR(A), generated using the Agrawal generator (Agrawal et al., 1993). For POSTURE, we remove class 8, the least frequent class, to obtain the ten-class variant POSTURE-No8. The primary backbone is TabICL-v2 (Qu et al., 2026), with one estimator, $B = 1000$, $\rho = 0.75$, and a single uncertainty threshold $\tau = 0.3$. We exclude the first 100 steps from cumulative prequential accuracy. With a fixed pretrained checkpoint and stream order, the single-estimator RECAP run is deterministic. Different backbone initialization seeds could introduce variation, but we did not repeat the computationally intensive stream evaluations with separately initialized backbones. Appendix B gives dataset statistics and the available implementation settings.

5.2 Experimental Results

Comparison with Classical Stream Learners. To evaluate TFMs with bounded context management as an alternative to model-state adaptation, we compare RECAP with six classical stream learners in Table 1. With a fixed uncertainty threshold $\tau = 0.3$, RECAP achieves the highest prequential accuracy on all seven streams and a mean rank of 1.00. Its gains over the strongest classical baseline range from 0.26 to 19.55 percentage points, with the largest improvements on RIALTO (19.55 points) and METER (19.19 points). The advantage also holds on NOMAO and AGR(A), where the strongest classical baselines are already competitive, with gains of 0.45 and 0.26 points, respectively. These results show that a frozen TFM equipped with bounded context management can outperform established stream learners across the evaluated streams, without online model-parameter updates or per-stream threshold tuning.

Comparison with Policy Variants. To isolate the contribution of selective retention, we compare admission and eviction variants with the backbone and bank capacities fixed. Entropy-only uses entropy-gated admission with FIFO eviction, while Redundancy-only uses unconditional admission with geometric eviction. RECAP-Margin substitutes top-two margin uncertainty for entropy, and RECAP-GlobalNN removes the same-class restriction on redundancy. See Appendix B.4 for details.

As shown in Table 2, RECAP outperforms DualFIFO on all seven streams and achieves the highest accuracy among the six policies on six streams at the global threshold $\tau = 0.3$. The mean and median gains over DualFIFO are 0.31 and 0.28 percentage points, respectively, with the largest gain of 0.73

Table 3: **Robustness to memory configuration and backbone choice.** (a) Accuracy and RECAP minus DualFIFO differences across budgets and bank splits, with the highest accuracy in each sweep in bold. (b) Accuracy of RECAP and DualFIFO with additional backbones under the original per-stream threshold selection, with the higher accuracy in each pair in bold.

(a) Memory budget and bank split

Setting	METER	NOMAO	AGR(A)
$B = 100$	82.54(−1.80)	96.99(+0.33)	89.08(+0.19)
$B = 500$	90.11(−0.77)	97.78(+0.55)	**91.02**(+0.13)
$B = 1000$	**90.68**(+0.08)	**97.87**(+0.40)	90.52(+0.07)
$B = 2000$	90.32(+0.36)	97.70(+0.06)	89.27(+0.40)
$\rho = 0.50$	90.24(−0.34)	97.78(+0.43)	90.47(+0.04)
$\rho = 0.75$	**90.68**(+0.08)	**97.87**(+0.40)	90.52(+0.07)
$\rho = 0.90$	90.45(−0.11)	97.75(+0.05)	**90.69**(+0.15)

(b) Additional backbones

	LimiX-v1		TabPFN-v2.5		TabDPT-v1	
Stream	RECAP	DualFIFO	RECAP	DualFIFO	RECAP	DualFIFO
NOAA	**81.43**	80.83	**81.82**	81.26	**81.94**	81.49
METER	**89.06**	88.69	**87.88**	87.64	**80.73**	79.65
RIALTO	**90.17**	89.59	**89.53**	88.93	**88.05**	87.64
POSTURE-No8	**60.50**	60.47	60.83	**60.99**	**61.67**	61.50
NOMAO	**97.83**	97.54	**97.85**	97.55	**97.34**	96.58
AGR(A)	**90.56**	90.54	**91.02**	90.85	**90.02**	89.93

points on POKER. Statistical tests yield $p = 0.015625$ for the two-sided exact Wilcoxon comparison with DualFIFO and $p = 0.00336$ for the Friedman comparison across all six policies (Appendix C.1).

5.3 Hyperparameter and Backbone Robustness

To assess how memory capacity and its allocation affect policy gains, we vary the total context budget B and the short-bank fraction ρ. We fix $\rho = 0.75$ when varying the budget and $B = 1000$ when varying the bank split. In Table 3(a), values are prequential accuracy, with percentage-point differences from DualFIFO in parentheses. At $B = 100$ and $B = 500$, RECAP trails DualFIFO on METER by 1.80 and 0.77 percentage points, respectively, while outperforming it on NOMAO and AGR(A). In the bank-split comparison, $\rho = 0.75$ yields positive gains on all three streams, whereas $\rho = 0.50$ and $\rho = 0.90$ yield losses on METER. Together, these results indicate that the benefit of RECAP depends on having sufficient context capacity and balancing recent and older examples.

Table 3(b) further tests whether these policy gains are specific to a particular TFM backbone, using LimiX (Zhang et al., 2025), TabPFN (Grinsztajn et al., 2025), and TabDPT (Ma et al., 2024). RECAP improves over DualFIFO in 17 of 18 comparisons, with the only loss being 0.16 points for TabPFN-v2.5 on POSTURE-No8. The gains across multiple backbones indicate that selective context retention is not tied to a single TFM architecture. These runs select thresholds separately for each stream, so they establish robustness across backbones under per-stream threshold selection.

6 Additional Analyses

To assess the benefits and robustness of RECAP, we compare it with alternative context selectors and evaluate its recovery from concept drift and tolerance to noisy feedback.

Retrieval and Replay. To test whether query-conditioned retrieval or representative replay can match the policy gains, we compare RECAP with five alternative selectors. kNN-FIFO retrieves the 1,000 nearest examples from the 5,000 most recent rows, whereas kNN-Reservoir retrieves from a 5,000-row uniform history sample; both use standardized Euclidean distance. The 1,000-row replay buffers sample history uniformly (Uniform reservoir), preserve class coverage (Class-balanced), or retain feature-space representatives through centroid herding. All methods therefore match the 1,000-example prediction context of RECAP, although the kNN variants retain five times more rows. Appendix B provides implementation details.

Table 4(a) shows that RECAP outperforms all tested selectors on METER, NOMAO, and AGR(A). The closest selector, kNN-FIFO, trails RECAP by 0.93, 0.44, and 8.04 percentage points on the three streams, respectively. The larger gaps to history-wide reservoirs are consistent with outdated concepts reducing context usefulness under drift.

Drift Recovery. Concept drift can make retained examples less relevant, requiring the context policy to respond as the predictive relationship changes. To assess this adaptation, we evaluate abrupt, gradual, and recurring Agrawal streams over three seeds, covering sudden changes, progressive transitions, and returning concepts. We measure post-drift regret area to capture the accumulated cost, and recovery steps to measure how quickly accuracy returns to 95% of its pre-drift level.

Table 4: **Additional context-selection and drift comparisons.** (a) Accuracy with 1,000 examples per prediction. Store denotes retained rows, and $\pm$ denotes standard deviation. (b) Three-seed mean drift metrics, averaging recovery steps over successful recoveries. Bold marks the highest accuracy in (a) and the lowest values in (b).

(a) Retrieval and replay

Policy	Store	METER	NOMAO	AGR(A)
RECAP	1000	**90.68**	**97.87**	**90.52**
kNN-FIFO	5000	89.75	97.43	82.48
kNN-Reservoir	5000	85.80	95.78	67.90
Uniform reservoir	1000	78.35 ± 0.20	94.50 ± 0.16	67.40 ± 0.82
Class-balanced	1000	78.75 ± 0.19	95.53 ± 0.28	62.72 ± 0.56
Centroid herding	1000	77.81	95.14	62.36

(b) Drift recovery

Drift	Metric	RECAP	DualFIFO	ARF
Abrupt	Regret area	**133.2**	144.0	217.3
	Recovery step	**924**	927	2,169
Gradual	Regret area	**61.5**	65.8	65.7
	Recovery step	**988**	1,073	1,095
Recurring	Regret area	**194.7**	196.0	393.6
	Recovery step	**953**	978	2,430

Table 4(b) shows that RECAP achieves the lowest mean regret area and recovery steps across all three drift types. Under abrupt drift, recovery times are nearly identical to DualFIFO, at 924 versus 927 steps, but RECAP reduces regret area from 144.0 to 133.2. Thus, the benefit appears primarily in the accumulated prediction cost rather than the time to reach the recovery threshold. Under gradual and recurring drift, RECAP recovers 85 and 25 steps earlier than DualFIFO, respectively. These results suggest that the context policy can reduce the cost of adapting to changing concepts. Appendix C.3.2 details the drift-generation settings and metric definitions.

Noisy Feedback. Predictive uncertainty can arise from both informative examples and label noise, raising the concern that entropy-gated retention may preserve corrupted feedback. To test whether RECAP remains effective in this setting, we symmetrically corrupt revealed labels and evaluate predictions against clean labels. In table 5 at 30% corruption, RECAP exceeds DualFIFO by 0.51, 0.27, and 1.95 percentage points on AGR(A), METER, and NOMAO, respectively. It also shows smaller accuracy declines from each method's clean baseline on all three streams, with the largest reduction on NOMAO, from 9.10 to 7.55 points. Appendix C.3.1 reports the clean and intermediate-noise results.

Table 5: **Accuracy at 30% label noise and decrease from each policy's clean baseline, in percentage points.** Bold marks higher accuracy and smaller decrease within each stream.

	Accuracy		Decrease	
Stream	RECAP	DualFIFO	RECAP	DualFIFO
AGR(A)	**88.25**	87.74	**2.27**	2.72
METER	**84.05**	83.78	**6.63**	6.82
NOMAO	**90.32**	88.37	**7.55**	9.10

7 Conclusion

Applying tabular foundation models to evolving streams requires deciding which past examples remain useful within a bounded context. We introduced a future-information view that motivates recency, uncertainty, and redundancy under explicit assumptions. RECAP implements these signals through a recent short bank, entropy-gated long-bank admission, and same-class redundancy-aware eviction. With a fixed uncertainty threshold, RECAP outperforms six classical stream learners on all seven evaluated streams. Comparisons with the same backbone and context budget show a consistent gain over DualFIFO, isolating the contribution of selective retention.

Beyond the main accuracy comparison, the additional analyses clarify where these gains arise. RECAP outperforms retrieval and replay baselines even when the kNN variants retain five times more examples, achieves the lowest mean regret area and recovery steps across abrupt, gradual, and recurring drift, and degrades less than DualFIFO under 30% label corruption. These results connect the gains of RECAP to context management itself, which combines recent coverage with selective retention, rather than to a particular retrieval mechanism or a stationary stream condition.

Limitations and Future Work. The future-information view theoretically motivates RECAP, but its gain over FIFO-based DualFIFO is modest in absolute terms. For real-time streams, however, a consistent gain at comparable prediction cost remains valuable (Appendix C.5). TFM inference remains the larger bottleneck. Reported means are 32.3 ms per example for RECAP on an H200 GPU and 0.41 ms for CPU ARF, despite the different hardware. Efficient context caching, reusable representations, and faster TFM inference are therefore important future directions.

AI USE STATEMENT

AI assistants were used for checking English grammar and LaTeX syntax, generating LaTeX table formats, and helping implement code for dataset loading and preprocessing. All text and code generated with AI assistance were checked by the authors and modified when necessary. The authors made the final decision on whether to use the generated text and code. All experimental results were obtained and verified by the authors.

REFERENCES

Soheil Abadifard, Sepehr Bakhshi, Sanaz Gheibuni, and Fazli Can. DynED: Dynamic ensemble diversification in data stream classification. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 3707–3711, 2023.

Charu C Aggarwal. *Data streams: models and algorithms*, volume 31. Springer Science & Business Media, 2007.

Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering*, 5(6):914–925, 1993. doi: 10.1109/69.250074.

Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 1–16, 2002.

Albert Bifet, Geoff Holmes, and Bernhard Pfahringer. Leveraging bagging for evolving data streams. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 135–150. Springer, 2010a.

Albert Bifet, Geoff Holmes, Bernhard Pfahringer, Philipp Kranen, Hardy Kremer, Timm Jansen, and Thomas Seidl. MOA: Massive online analysis, a framework for stream classification and clustering. In *Proceedings of the first workshop on applications of pattern analysis*, pp. 44–50. PMLR, 2010b.

Dongzi Chen, Qinli Yang, Jiaming Liu, and Zhu Zeng. Selective prototype-based learning on concept-drifting data streams. *Information Sciences*, 516:20–32, 2020.

Roberto Souto Maior de Barros, Silas Garrido T de Carvalho Santos, and Paulo Mauricio Gonçalves Júnior. A boosting-like online learning ensemble. In *2016 international joint conference on neural networks (IJCNN)*, pp. 1871–1878. IEEE, 2016.

Pedro Domingos and Geoff Hulten. Mining high-speed data streams. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 71–80, 2000.

Joao Gama, Raquel Sebastiao, and Pedro Pereira Rodrigues. Issues in evaluation of stream learning algorithms. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 329–338, 2009.

Joao Gama, Raquel Sebastiao, and Pedro Pereira Rodrigues. On evaluating stream learning algorithms. *Machine learning*, 90(3):317–346, 2013.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014.

Heitor M Gomes, Albert Bifet, Jesse Read, Jean Paul Barddal, Fabrício Enembreck, Bernhard Pfahringer, Geoff Holmes, and Talel Abdessalem. Adaptive random forests for evolving data stream classification. *Machine Learning*, 106(9):1469–1495, 2017.

Heitor Murilo Gomes, Jesse Read, and Albert Bifet. Streaming random patches for evolving data stream classification. In *2019 IEEE international conference on data mining (ICDM)*, pp. 240–249. IEEE, 2019.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, et al. TabPFN-2.5: Advancing the state of the art in tabular foundation models. *arXiv preprint arXiv:2511.08667*, 2025.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Geoff Hulten, Laurie Spencer, and Pedro Domingos. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 97–106, 2001.

Łukasz Korycki and Bartosz Krawczyk. Instance exploitation for learning temporary concepts from sparsely labeled drifting data streams. *Pattern Recognition*, 129:108749, 2022.

Viktor Losing, Barbara Hammer, and Heiko Wersing. Self-adjusting memory: How to deal with diverse drift types. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 4899–4903, 2017. doi: 10.24963/ijcai.2017/690.

Afonso Lourenço, João Gama, Eric P Xing, and Goreti Marreiros. Bridging streaming continual learning via in-context large tabular models. *arXiv preprint arXiv:2512.11668*, 2025.

Afonso Lourenço, João Gama, Eric P Xing, and Goreti Marreiros. In-context learning of evolving data streams with tabular foundational models. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 973–982, 2026.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Hamidreza Kamkari, Jesse C Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L Caterini, and Maksims Volkovs. TabDPT: Scaling tabular foundation models on real data. *arXiv preprint arXiv:2410.18164*, 2024.

Chaitanya Manapragada, Geoffrey I Webb, and Mahsa Salehi. Extremely fast decision tree. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1953–1962, 2018.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. *arXiv preprint arXiv:2112.10510*, 2021.

Hai-Long Nguyen, Yew-Kwong Woon, and Wee-Keong Ng. A survey on data stream clustering and classification. *Knowledge and information systems*, 45(3):535–569, 2015.

Jingang Qu, David Holzmuller, Gael Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. *arXiv preprint arXiv:2502.05564*, 2025.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. *arXiv preprint arXiv:2602.11139*, 2026.

Jesse Read. Concept-drifting data streams are time series; the case for continuous adaptation. *arXiv preprint arXiv:1810.02266*, 2018.

Jesse Read and Indre Zliobaite. Supervised learning from data streams: an overview and update. *ACM Computing Surveys*, 57(12):1–31, 2025.

Rui Su, Husheng Guo, and Wenjian Wang. Elastic online deep learning for dynamic streaming data. *Information Sciences*, 676:120799, 2024.

Ke Wan, Yi Liang, and Susik Yoon. Online drift detection with maximum concept discrepancy. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 2924–2935, 2024.

Xingxuan Zhang, Gang Ren, Han Yu, Hao Yuan, Hui Wang, Jiansheng Li, Jiayun Wu, Lang Mo, Li Mao, Mingchao Hao, et al. LimiX: Unleashing structured-data modeling capability for generalist intelligence. *arXiv preprint arXiv:2509.03505*, 2025.

Table 6: Notation used in the future-information view. Random labels in the analysis are distinguished from the observed examples retained by the policy.

Notation	Meaning
$t, h, \mathcal{H}_t^+$	Current stream step, short future horizon length, and corresponding future steps.
x_t, y_t, z_t	Feature vector, observed label, and labeled example (x_t, y_t).
D_t, S_t, L_t	Current context, recent short bank, and selective long bank.
A, Y_A	Fixed indices of the examples in D_t and their corresponding random label vector.
$\mathcal{F}$	Common background fixing context and candidate features while leaving the labels being measured random.
Y_i, Y'_x	Candidate label and future query label before realization.
$\mathcal{P}_{s,X}$	Feature marginal at future step s.
$\mathcal{P}^+_{t,X}$	Average near-future feature marginal over the horizon.
$\widehat{\mathcal{P}}_{S_t}$	Empirical feature distribution of the recent window.
$\mathcal{J}_t(A)$	Future information of the random context labels.
$\widehat{\mathcal{J}}_{S_t}(A)$	Future information with the recent-window feature distribution.
$\Delta_t(i \mid A)$	Marginal future information of candidate label Y_i.
$\mathcal{N}_i, \alpha_t(i)$	Effective future region of candidate i and its probability mass.
K, r_t	Lipschitz constant and upper bound on the recent-to-future Wasserstein-1 distance.
δ, ϵ	Local entropy consistency and label coherence slacks.
$\ell_t(j \mid A), \eta$	Removal loss and its assumed distance-dependent upper bound.

A Proofs for the Future-Information View

We provide the assumptions and proofs for the lemmas and theorem in Section 3. Notation used in this section can be found in Table 6, distinguishing random from observed labels.

A.1 Random Variables and Conditioning

We fix a stream step t and a candidate context $D_t = \{(x_j, y_j)\}_{j \in A}$, where A denotes its fixed example indices. Let $Y_A = (Y_j)_{j \in A}$ denote the context labels as random variables before their values $(y_j)_{j \in A}$ are observed. The objective measures the expected information supplied by observing these labels, although the deployed policy already knows the labels in its retained context. The label Y'_x denotes a future query label at feature value x under a specified joint law of context and future labels. This law can represent shared uncertainty about an evolving data-generating concept, but is not claimed to equal the true stream law or the frozen TFM's predictive distribution.

The common background $\mathcal{F}$ contains the fixed feature design, including the candidate feature x_i, but does not fix the context and candidate labels whose information is measured. All information terms in a comparison use the same conditional law and background. If the observed labels were fixed by the conditioning, the intended context mutual information would vanish. Likewise, adding a random candidate feature to only one side of a comparison could introduce information beyond its label. The marginal identity therefore concerns label information alone.

The near-future feature distribution $\mathcal{P}^+_{t,X}$ is the average feature marginal over the short horizon $\mathcal{H}_t^+$, as defined in Equation 2. Writing $\mathbb{E}_x$ for integration under this measure, the future usefulness of the context is $\mathcal{J}_t(A) = \mathbb{E}_x I(Y_A; Y'_x \mid \mathcal{F}, x)$. Query features are fixed and labels remain random under this law.

The information term measures how much the context labels inform the label of a future query. However, this is a measure for the entire context, not for an individual example where the decision is actually made. Moreover, computing it requires knowing $\mathcal{P}^+_{t,X}$, which is not available at time t. Our goal is thus to connect the measure to admission and removal decisions.

The results apply to fixed context designs under this law. An actual RECAP bank is selected adaptively from realized predictions and labels, and its class-based operations depend on those values. Extending the analysis to that process requires consistent treatment of selection events in the joint law. The following statements neither optimize adaptive retention nor bound its prequential regret.

A.2 Assumption and Proof for Lemma 1

Assumption 2 (Recent-window local stability)**.** *At time t, the empirical feature distribution of the recent window is close to the near-future feature distribution*

$$W_1(\widehat{\mathcal{P}}_{S_t}, \mathcal{P}^+_{t,X}) \le r_t, \tag{16}$$

where W_1 is the Wasserstein-1 distance in the representation space and $r_t \ge 0$ is an upper bound on the distance between the empirical recent-window and near-future feature distributions at time t. Both measures have finite first moments in that representation.

Proof of Lemma 1. Define the query-wise information function for the current context design as

$$f_A(x) = I(Y_A; Y'_x \mid \mathcal{F}, x).$$

By the Lipschitz condition in Lemma 1, f_A is K-Lipschitz in the representation space. The near-future objective and the recent-window objective can be written as

$$\mathcal{J}_t(A) = \mathbb{E}_{x\sim\mathcal{P}^+_{t,X}}[f_A(x)], \qquad \widehat{\mathcal{J}}_{S_t}(A) = \mathbb{E}_{x\sim\widehat{\mathcal{P}}_{S_t}}[f_A(x)].$$

By the Kantorovich–Rubinstein duality for Wasserstein-1 distance, for any two feature distributions P and Q with finite first moments,

$$|\mathbb{E}_{x\sim P}[f_A(x)] - \mathbb{E}_{x\sim Q}[f_A(x)]| \le K W_1(P, Q).$$

Taking $P = \mathcal{P}^+_{t,X}$ and $Q = \widehat{\mathcal{P}}_{S_t}$ gives

$$\left|\mathcal{J}_t(A) - \widehat{\mathcal{J}}_{S_t}(A)\right| \le K W_1(\widehat{\mathcal{P}}_{S_t}, \mathcal{P}^+_{t,X}).$$

By Assumption 2,

$$W_1(\widehat{\mathcal{P}}_{S_t}, \mathcal{P}^+_{t,X}) \le r_t.$$

Therefore,

$$\left|\mathcal{J}_t(A) - \widehat{\mathcal{J}}_{S_t}(A)\right| \le K r_t. \qquad \square$$

This approximation is useful only if replacing the feature distribution does not change the future-information objective too much. A large shift can make the bound uninformative, and identical feature marginals can coexist with different conditional label distributions. The result bounds the outer-distribution replacement for a fixed information function without selecting an optimal recent window or an item-level admission rule within the short bank.

A.3 Proof of Lemma 2

Proof of Lemma 2. For a candidate i and a context index set A not containing i, Definition 3 gives

$$\begin{aligned}\Delta_t(i \mid A) &= \mathcal{J}_t(A \cup \{i\}) - \mathcal{J}_t(A) \\ &= \mathbb{E}_x\left[I((Y_A, Y_i); Y'_x \mid \mathcal{F}, x) - I(Y_A; Y'_x \mid \mathcal{F}, x)\right].\end{aligned}$$

Since the candidate feature is held in the common background, the chain rule for mutual information gives

$$I((Y_A, Y_i); Y'_x \mid \mathcal{F}, x) = I(Y_A; Y'_x \mid \mathcal{F}, x) + I(Y_i; Y'_x \mid Y_A, \mathcal{F}, x).$$

Substituting this identity into the previous expression yields

$$\Delta_t(i \mid A) = \mathbb{E}_x\left[I(Y_i; Y'_x \mid Y_A, \mathcal{F}, x)\right],$$

which proves the claim under the specified conditional law. $\square$

The identity rewrites a candidate's marginal contribution as conditional future-label information by the chain rule, without additional local assumptions. It concerns expected information before the relevant labels are realized. It does not imply that every observed row improves a finite pretrained predictor or equate increased information with a particular change in accuracy.

A.4 Assumptions and Proof for Theorem 1

Let $\mathcal{N}_i$ denote the fixed effective future local region around candidate i, where its revealed label can provide local evidence for near-future queries. Let

$$\alpha_t(i) = \Pr_{x \sim \mathcal{P}^+_{t,X}} [x \in \mathcal{N}_i] > 0.$$

All following label entropies use the same conditional law, and expectations use the future feature measure. Nonnegative slacks δ and ϵ allow imperfect local entropy consistency and label coherence.

Assumption 3 (Local entropy consistency)**.** *For a candidate $i \notin A$ and nonnegative slack δ,*

$$\mathbb{E}_x[H(Y'_x \mid Y_A, \mathcal{F}, x) \mid x \in \mathcal{N}_i] \geq H(Y_i \mid Y_A, \mathcal{F}) - \delta. \tag{17}$$

Assumption 3 says that uncertainty at x_i is representative of uncertainty in the effective future local region of the candidate, up to slack δ. It allows imperfect locality, but for small δ rules out the case where the candidate is highly uncertain while nearby future queries are already easy.

Assumption 4 (Local label coherence)**.** *For the same candidate and neighborhood and nonnegative slack ϵ,*

$$\mathbb{E}_x[H(Y'_x \mid Y_i, Y_A, \mathcal{F}, x) \mid x \in \mathcal{N}_i] \leq \epsilon. \tag{18}$$

This assumption says that the candidate label is useful inside this local region. After Y_i is observed, the average residual uncertainty of nearby future labels is at most ϵ. A small ϵ therefore means that nearby labels are locally coherent and can be predicted from the candidate label.

Proof of Theorem 1. For any $x \in \mathcal{N}_i$, the definition of conditional mutual information gives

$$I(Y_i; Y'_x \mid Y_A, \mathcal{F}, x) = H(Y'_x \mid Y_A, \mathcal{F}, x) - H(Y'_x \mid Y_i, Y_A, \mathcal{F}, x).$$

Taking expectation over near-future feature values and restricting to the effective future local region,

$$\begin{aligned}
&\mathbb{E}_x[\mathbb{I}\{x \in \mathcal{N}_i\} I(Y_i; Y'_x \mid Y_A, \mathcal{F}, x)] \\
&= \alpha_t(i) \mathbb{E}_x[I(Y_i; Y'_x \mid Y_A, \mathcal{F}, x) \mid x \in \mathcal{N}_i] \\
&= \alpha_t(i) \mathbb{E}_x[H(Y'_x \mid Y_A, \mathcal{F}, x) - H(Y'_x \mid Y_i, Y_A, \mathcal{F}, x) \mid x \in \mathcal{N}_i] .
\end{aligned}$$

By Assumption 3,

$$\mathbb{E}_x[H(Y'_x \mid Y_A, \mathcal{F}, x) \mid x \in \mathcal{N}_i] \geq H(Y_i \mid Y_A, \mathcal{F}) - \delta.$$

By Assumption 4,

$$\mathbb{E}_x[H(Y'_x \mid Y_i, Y_A, \mathcal{F}, x) \mid x \in \mathcal{N}_i] \leq \epsilon.$$

Combining these two inequalities gives

$$\mathbb{E}_x[\mathbb{I}\{x \in \mathcal{N}_i\} I(Y_i; Y'_x \mid Y_A, \mathcal{F}, x)] \geq \alpha_t(i) \left[H(Y_i \mid Y_A, \mathcal{F}) - \delta - \epsilon\right],$$

which proves the stated lower bound on local future information. □

Higher uncertainty at x_i gives a larger lower bound on local future information when the neighborhood mass and slacks are held fixed. The result does not order actual candidate utilities, since their masses, slacks, and upper bounds can differ. If the bracket is negative, nonnegativity of mutual information supplies the stronger trivial lower bound of zero. If $\alpha_t(i) = 0$, the local integral is zero and the bound is interpreted trivially without a neighborhood-conditional expectation.

A.5 Scope of Assumption 1

The redundancy condition concerns removal from a fixed context design A. It formalizes the intuition that close same-class stored examples provide overlapping label evidence.

Assumption 1 says that if a stored example has a close same-class neighbor, then removing it causes only a small loss of future information under the specified law. This motivates removing close same-class examples from the context when the budget is full. The condition does not follow from equal observed labels and small Euclidean distance alone, nor does it establish the majority-class rule or the centroid-based endpoint choice. The ablations in Table 2 evaluate these policy choices.

A.6 Connection to the Deployed Signals

RECAP uses the stored normalized predictive entropy $h_i = H(p_i)/\log C$ as an admission signal, where $p_i = q_\theta(\cdot \mid x_i, D_i)$ is the prediction-time distribution. This predictive score serves as a practical proxy for the conditional label entropy $H(Y_i \mid Y_A, \mathcal{F})$ in the theoretical analysis. The score is recorded at arrival and reused after the short-bank delay, during which both the context and stream may change. The gate also omits the future mass $\alpha_t(i)$ and does not compute the information objective online.

Appendix C.4.3 maps each assumption to an observable diagnostic and reports the corresponding stream-level measurements in Table 14. The results show positive local entropy correlations and same-label lift, and favor recency on three of four streams, while providing weak evidence for distance-based redundancy ranking. Entropy rank correlation does not estimate δ, same-label lift does not estimate ϵ, and a sliced Wasserstein ratio does not upper-bound the W_1 term. The weak distance and removal correlations do not support a monotone raw-distance ranking of removal loss. These distinctions limit the guarantees for the deployed policy while preserving the information view as a design motivation.

B Experimental Details

This appendix provides additional details on datasets, TFM backbones, classical stream-learning baselines, context management policy variants, and evaluation settings. It also gives the context update procedure and the resource requirements of RECAP.

B.1 Datasets

We evaluate on seven datasets with six real-world data streams and one synthetic abrupt-drift stream. For real-world streams, we preserve the original row order and do not shuffle examples. Each file is loaded as a headerless CSV file, then the last column is treated as the class label while all preceding columns are used as input features. The METER and POSTURE loaders drop one malformed row before constructing the stream. POSTURE-No8 additionally removes the rarest encoded class, yielding the ten-class variant reported in Table 7.

A synthetic stream AGR(A) is generated from the Agrawal stream generator. We generated 30,000 examples with 9 features and 2 classes. The concept function changes abruptly at steps 7,500, 15,000, and 22,500 following the sequence $(0, 3, 6, 9)$, with 10% feature perturbation. Appendix C.3.2 reports these drift runs separately.

Table 7 summarizes the datasets used in our experiments. We categorize class balance using the imbalance ratio

$$\mathrm{IR} = \frac{\max_c n_c}{\min_c n_c},$$

where n_c denotes the number of examples in class c. A dataset is categorized as balanced if $\mathrm{IR} \leq 1.1$, imbalanced if $1.1 < \mathrm{IR} \leq 10$, highly imbalanced if $10 < \mathrm{IR} \leq 100$, and extremely imbalanced if $\mathrm{IR} > 100$.

B.2 Backbones

We evaluate RECAP with four frozen tabular foundation model backbones, TabICL-v2, TabPFN-v2.5, TabDPT-v1, and LimiX-v1, to test its use with different pretrained predictors.

TabICL-v2. TabICL was designed to scale tabular in-context learning beyond the expensive alternating row/column attention, using a two-stage column-then-row attention module to build row embeddings before efficient in-context prediction (Qu et al., 2025). TabICL-v2 further improves scalability and performance with a more diverse synthetic data engine and optimized pretraining protocols for both classification and regression (Qu et al., 2026). We use TabICL-v2 as the primary backbone for the global-threshold and prefix-selected experiments.

Table 7: Dataset statistics after preprocessing. POSTURE-No8 removes the rarest encoded class to obtain a ten-class stream for the benchmark evaluation.

Dataset	Rows	Features	Classes	Balance
NOAA	18,159	8	2	imbalanced
METER	22,948	96	10	balanced
RIALTO	82,250	27	10	balanced
POSTURE-No8	163,477	3	10	highly imbalanced
POKER	829,201	10	10	extremely imbalanced
NOMAO	34,465	118	2	imbalanced
AGR(A)	30,000	9	2	synthetic abrupt drift

TabPFN-v2.5. TabPFN-v1 instantiated Prior-Data Fitted Networks for tabular classification. TabPFN-v2 introduced a stronger synthetic task distribution and an alternating-attention architecture over samples and features (Hollmann et al., 2025). TabPFN-v2.5 follows this line and scales it toward larger tabular contexts (Grinsztajn et al., 2025).

TabDPT-v1. TabDPT targets the limitation of purely synthetic pretraining by training tabular in-context learning architectures on real data with self-supervised learning and retrieval (Ma et al., 2024). We use TabDPT-v1 as the real-data-pretrained backbone.

LimiX-v1. LimiX treats structured data as a joint distribution over variables and missingness, enabling classification, regression, imputation, and generation through query-based conditional prediction (Zhang et al., 2025). It is pretrained with masked joint-distribution modeling under an episodic context-conditional objective. We use LimiX-16M for classification without retrieval.

The main comparisons use one estimator and a fixed stream order, which produces deterministic outputs in this setup. Table 3(b) reports the additional-backbone accuracies on the six-stream subset under the original per-stream threshold-selection protocol.

B.3 Baselines

We compare RECAP against various classical online stream-learning baselines. These baselines are implemented using MOA 24.07 (Bifet et al., 2010b).

VFDT. The Very Fast Decision Tree (VFDT), also known as the Hoeffding Tree, is an incremental decision-tree algorithm for high-speed data streams. It uses Hoeffding bounds to decide when the best split attribute is statistically reliable, enabling incremental decision-tree induction without revisiting the full stream history (Domingos & Hulten, 2000).

EFDT. The Extremely Fast Decision Tree (EFDT), or Hoeffding Anytime Tree, modifies Hoeffding Tree by allowing splits to be made earlier and later revised when better split choices become statistically supported. Compared with the conservative split policy of VFDT, EFDT can improve statistical efficiency and prequential accuracy at modest additional cost (Manapragada et al., 2018).

ARF. Adaptive Random Forest (ARF) is an ensemble baseline for evolving data stream classification. It combines online resampling with multiple incremental tree learners and uses adaptive operators to replace underperforming trees under concept drift (Gomes et al., 2017).

SRP. Streaming Random Patches (SRP) is an ensemble method designed for evolving data streams that combines online bagging with random subspaces. It jointly exploits both instance resampling and feature-subspace diversity (Gomes et al., 2019).

Leveraging Bagging. Leveraging Bagging extends online bagging with stronger randomization to increase ensemble diversity. It was proposed for evolving data streams as a simple but effective bagging variant with additional randomness and drift-aware mechanisms (Bifet et al., 2010a).

BOLE. The Boosting-like Online Learning Ensemble (BOLE) adapts boosting-style ideas to concept-drifting data streams. It modifies online boosting mechanisms to better handle changing distributions and maintain ensemble performance under drift (de Barros et al., 2016).

B.4 Context Management Policy Variants

This section describes the context-policy variants used in the ablation study. The variants test whether uncertainty is needed for retaining informative candidates, whether redundancy is needed for low-loss removal, and how entropy and same-class nearest-neighbor removal instantiate them.

DualFIFO. DualFIFO is the FIFO-style dual-memory reference policy. It uses the same short-bank and long-bank budgets as RECAP. New examples enter the short bank, and short-bank overflow candidates are inserted into the long bank. When the long bank exceeds its budget, DualFIFO selects the most represented class

$$c_\star \in \arg\max_c |\{z \in L : y_z = c\}|$$

and removes the oldest long-bank item from that class. Thus, DualFIFO preserves recency and coarse class balance but does not use uncertainty or redundancy-aware removal.

Entropy-only. Entropy-only isolates uncertainty-based retention. It uses the same entropy-gated long-bank admission rule as RECAP, but keeps the DualFIFO class-aware FIFO removal rule when the long bank overflows. This tests whether uncertainty-based candidate filtering alone is sufficient without redundancy-aware management of the long-bank capacity.

Redundancy-only. Redundancy-only isolates redundancy-aware removal. It admits every short-bank overflow candidate into the long bank, but uses the same-class nearest-neighbor removal rule from RECAP when the long bank overflows. This tests whether removing duplicated same-class evidence is sufficient without uncertainty-based candidate filtering.

RECAP-Margin. RECAP-Margin tests the choice of uncertainty score. It keeps the same dual-memory structure and same-class redundancy-aware removal rule as RECAP, but replaces predictive entropy with a top-two margin score. For a prediction distribution $p_t = q_\theta(\cdot \mid x_t, D_t)$, let $p_{t,(1)}$ and $p_{t,(2)}$ be the largest and second-largest class probabilities. The margin-based uncertainty score is

$$h_{\text{margin}}(z_t) = 1 - \left(p_{t,(1)} - p_{t,(2)}\right).$$

A larger value indicates a smaller gap between the two most likely classes. This variant tests whether the full predictive entropy used in RECAP is more effective than a top-two ambiguity score. The common numerical threshold does not necessarily match admission rates between entropy and margin, since the scores have different distributions. The comparison therefore tests complete policies at the specified threshold rather than a rate-matched uncertainty ablation.

RECAP-GlobalNN. This variant tests the label-conditional nature of redundancy. It keeps entropy-gated admission but replaces same-class nearest-neighbor removal with class-agnostic nearest-neighbor removal. When the long bank overflows, it finds

$$(i_\star, j_\star) \in \arg \min_{\substack{i \neq j \\ z_i, z_j \in L}} \|\phi(x_i) - \phi(x_j)\|_2.$$

For a same-label pair, the endpoint rule uses the centroid of short-bank examples with that label, falling back to the full short bank when no such example exists. For a cross-label pair, we use the centroid of the full short bank. We remove the endpoint farther from the corresponding centroid, breaking ties by removing the older example. This variant tests whether redundancy should be defined purely geometrically or conditioned on the label.

B.5 Additional Selectors

kNN-FIFO retains the most recent $R = 5B$ labeled rows, and kNN-Reservoir maintains a uniform reservoir of $R = 5B$ rows from the observed history. Each selects up to B context examples for the

Table 8: Original selected classical configurations. The symbols s, g, and t denote ensemble size, grace period, and tie threshold. The revised RECAP comparison uses the shared threshold of 0.3.

Dataset	ARF	BOLE	LevBag	SRP	EFDT	VFDT
NOAA	$s90, g400, t0.1$	$s90, g100, t0.1$	$s90, g100, t0.1$	$s90, g400, t0.1$	$g400, t0.1$	$g400, t0.1$
METER	$s90, g400, t0.1$	$s90, g1000, t0.1$	$s90, g100, t0.1$	$s90, g100, t0.1$	$g1000, t0.01$	$g400, t0.1$
RIALTO	$s90, g100, t0.1$	$s90, g100, t0.1$	$s90, g100, t0.1$	$s90, g100, t0.1$	$g100, t0.1$	$g400, t0.1$
POSTURE-No8	$s90, g100, t0.1$	$s90, g100, t0.1$	$s90, g100, t0.1$	$s90, g100, t0.1$	$g400, t0.1$	$g100, t0.1$
POKER	$s90, g100, t0.1$	$s90, g100, t0.1$	$s90, g100, t0.1$	$s90, g100, t0.1$	$g100, t0.1$	$g400, t0.1$
NOMAO	$s90, g100, t0.1$	$s90, g1000, t0.01$	$s90, g400, t0.1$	$s90, g100, t0.1$	$g100, t0.05$	$g100, t0.1$
AGR(A)	$s90, g100, t0.1$	$s90, g100, t0.1$	$s90, g100, t0.1$	$s90, g100, t0.01$	$g100, t0.1$	$g400, t0.1$

current query using standardized raw-feature Euclidean distance. Prediction precedes insertion of the current labeled example into either store.

The additional bounded buffers use uniform reservoir sampling, class-balanced sampling, and centroid herding at the matched 1,000-row context setting. The reservoir results in Table 4(a) are reported as means and standard deviations.

B.6 Evaluation Details

Protocol. All methods are evaluated under the prequential protocol. At step t, the learner predicts the label of x_t using its current model or context, then receives the true label y_t, and finally updates its model or context using the labeled example $z_t = (x_t, y_t)$. We use a warm-up period of 100 stream steps and report prequential accuracy only after warm-up. The separate prefix-selection protocol also excludes its selection prefix, as specified in Appendix C.2.1.

The total context capacity is $B = 1000$, split into $B_S = 750$ and $B_L = 250$, except in the explicit sensitivity experiments. The first 100 observations are a scoring warm-up and are still available for subsequent context updates. This scoring warm-up differs from the unconditional long-bank fill, which continues until that bank reaches its own capacity. Geometric comparisons use standardized raw features.

Hardware. All TFM-based streaming experiments are run on a single NVIDIA H200 GPU. The classical methods use CPU execution, with timing scopes described in Appendix C.5.1.

MOA baseline hyperparameter grid. For all MOA baseline runs, we fix the split confidence to 10^{-7} and use `NBAdaptive` leaf prediction. For ensemble baselines, ARF, BOLE, LevBag, and SRP, we fix the ensemble size to 90 and grid search over grace period and tie threshold

$$g \in \{100, 400, 1000\}, \qquad t \in \{0.01, 0.05, 0.1\}.$$

Thus, each ensemble baseline is evaluated with $3 \times 3 = 9$ configurations per dataset. For tree baselines, EFDT and VFDT, we use the same grid over grace period and tie threshold. This also gives nine configurations per tree baseline for each evaluated stream.

Selected hyperparameters. Table 8 reports the classical hyperparameters selected for the results in Table 1. For the MOA baselines, s denotes the ensemble size, g denotes the grace period, and t denotes the tie threshold. The ensemble size is fixed to $s = 90$ for ARF, BOLE, LevBag, and SRP. EFDT and VFDT are single-tree methods and therefore do not use s. Ensemble accuracies are means and standard deviations over five seeds, while tree results use deterministic fixed-order runs.

B.7 Algorithmic Details

This appendix gives the procedural form of the context update used by RECAP. The short bank S implements the recency signal, while the long bank L stores older examples selected by entropy-gated admission and redundancy-aware eviction. The stored score h_i is the prediction-time entropy of z_i, computed when it was first observed and later used when the sample overflows from the short bank. The policy below uses $B_S \geq 1$ and $B_L \geq C$, as in the tested configurations, so an overflowing long bank contains a same-class pair. We use S and L as working copies of S_t and L_t during the update. After the update, these banks become S_{t+1} and L_{t+1}.

RECAP context update. The inputs are the short bank S, long bank L, new query x_t, capacities B_S, B_L, and threshold τ. The outputs are the updated banks S, L for the next prediction.

1. Predict $p_t = q_\theta(\cdot \mid x_t, S \cup L)$ and record h_t from Equation 11 before receiving y_t. After label revelation, append (x_t, y_t, h_t, t) to the short bank S.
2. If $|S| \le B_S$, finish the update. Otherwise remove its oldest example (x_i, y_i, h_i, i) and make that example the long-bank promotion candidate.
3. If either $|L| < B_L$ or $h_i \ge \tau$, insert z_i into L. Otherwise discard the candidate without changing the long bank and finish the update.
4. If $|L| \le B_L$, finish the update. Otherwise select the most represented class
$$c_\star \in \arg\max_c |\{z_j \in L : y_j = c\}|.$$
5. Find the nearest same-class pair $(i_\star, j_\star)$ in class $c_\star$ using the standardized raw-feature representation
$$(i_\star, j_\star) \in \arg \min_{\substack{i \ne j \\ z_i, z_j \in L \\ y_i = y_j = c_\star}} \|\phi(x_i) - \phi(x_j)\|_2.$$
6. Compute $S(c_\star) = \{z_j \in S : y_j = c_\star\}$ and its reference centroid
$$\mu^* = \begin{cases} |S(c_\star)|^{-1} \sum_{z_j \in S(c_\star)} \phi(x_j), & |S(c_\star)| > 0, \\ |S|^{-1} \sum_{z_j \in S} \phi(x_j), & |S(c_\star)| = 0. \end{cases}$$
7. Choose the endpoint farther from the reference centroid and remove that example from the long bank
$$u^* \in \arg\max_{u \in \{i_\star, j_\star\}} \|\phi(x_u) - \mu^*\|_2, \qquad L \leftarrow L \setminus \{z_{u^*}\}.$$

The nearest-pair search implements the redundancy signal. Among the overrepresented class in the long bank, the closest same-class pair is treated as the most redundant evidence, and the item farther from the recent reference centroid is removed.

At the end of each update, $|S| \le B_S$ and $|L| \le B_L$. The context therefore contains at most B examples regardless of stream length. This row-count invariant does not bound the pretrained model size or eliminate the distance-cache and inference-workspace costs discussed in Appendix B.8.

B.8 Resource Requirements

The retained feature rows, labels, arrival indices, and scores require $O(Bd + B)$ storage. A direct nearest-pair search within a class of size m costs $O(m^2 d)$ per admitted overflow, with $m \le B_L + 1$. Pair distances can be cached when the feature representation is fixed, at a cost of up to $O(B_L^2)$ additional storage. Maintaining new pair distances does not remove the cost of finding the next minimum after an eviction unless a suitable search structure is also maintained. These policy resources remain bounded in stream length, while the frozen backbone and its inference workspace incur separate costs. Appendix C.5 reports measured latency, throughput, and memory use.

C Additional Experiments and Analyses

We organize the supplementary evidence by the questions it addresses. Appendix C.1 gives the statistical analysis of the main policy comparison, Appendix C.2 examines threshold selection, and Appendix C.3 extends the noise and drift evaluations. Appendix C.4 examines score timing, representation choice, and the proposed information signals, followed by computational costs in Appendix C.5.

C.1 Paired Statistics for the Global-Threshold Comparison

For the primary global-threshold comparison in Table 2, the differences are 0.40, 0.08, 0.28, 0.18, 0.73, 0.40, and 0.07 points. The mean is 0.3057 and the median is 0.28 points. All seven signs are positive, yielding a two-sided exact Wilcoxon probability of $2/2^7 = 0.015625$ under the signed-rank randomization null and rank-biserial correlation 1.0. The stream-level block interpretation avoids treating serially dependent predictions as independent replications.

For six policies on seven streams, Friedman ranking uses within-stream ranks and the statistic

$$Q = \frac{12n}{k(k+1)} \sum_{j=1}^{k} \overline{R}_j^2 - 3n(k+1), \qquad n = 7, \quad k = 6. \tag{19}$$

The displayed global table yields $Q = 17.6939$ and the chi-squared reference with five degrees of freedom gives $p = 0.00336$. The omnibus result does not identify every significant pairwise difference, and no claim is made that RECAP significantly exceeds each ablation individually. Prior benchmark development and the seven-stream sample also limit broader statistical inference.

C.2 Threshold Selection and Sensitivity

The main comparisons fix one global threshold. The following experiments distinguish selection from an initial labeled prefix from retrospective sensitivity on completed streams.

C.2.1 Policy Variants with Prefix Selection

We evaluate a deployment setting in which an initial labeled prefix serves as validation data for selecting a stream-specific threshold. After 100 warm-up steps, we select $\tau \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ by prequential accuracy on steps 101 through 1100. We then freeze the selected threshold and evaluate from step 1101 onward, testing whether the policy gains persist when threshold selection uses only early labeled feedback.

Prefix selection preserves the average advantage without choosing thresholds from the scored suffix. In Table 9, RECAP beats DualFIFO on six suffixes and loses 0.11 points on METER, with mean and median gains of approximately 0.30 and 0.33 points. Absolute accuracies should be compared within each protocol.

Execution details. For each stream, we instantiate five independent policy states, one for each $\tau \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. All candidates receive the same examples in the same order and are updated independently under the test-then-train protocol during both the 100-step warm-up and the validation interval. We select the candidate with the highest prequential accuracy on steps 101–1100, breaking ties in favor of the smaller τ.

At step 1101, evaluation continues from the state accumulated by the selected candidate, and the four unselected states are discarded. The policy is not reinitialized, and its threshold is not changed retrospectively. The selected state therefore contains feedback observed through step 1100, while no suffix labels influence threshold selection.

Because the policies and stream order are deterministic, we implement this procedure using five independently evolved full-stream traces and apply the selection rule using only their validation-prefix predictions. This produces the same selected state and suffix predictions as maintaining five candidates concurrently through step 1100 and only the selected candidate thereafter. The deployed procedure requires five candidate-policy evaluations during selection and one thereafter, whereas our offline implementation computes all five full-stream traces.

C.2.2 Retrospective Threshold Analysis

Table 10 studies the sensitivity of RECAP to the entropy threshold τ. The $\tau = 0$ column disables uncertainty-gated admission and therefore corresponds to the Redundancy-only variant. The original sweep evaluates completed-stream accuracy and therefore characterizes sensitivity rather than providing a causal deployment selection rule. Across positive thresholds, the maximum within-stream spread is 0.44 points on AGR(A), from 90.49 to 90.93. Zero threshold exceeds the default on AGR(A).

Table 9: Prefix-selected accuracy from step 1101 onward. Threshold selection uses steps 101 through 1100 after 100 warm-up steps. Bold denotes the largest accuracy or lowest mean rank in each row. The scoring interval differs from the global protocol in Table 2.

Stream	RECAP	DualFIFO	Entropy only	Redundancy only	RECAP Margin	RECAP GlobalNN
NOAA	**81.96**	81.40	81.78	81.64	81.87	81.85
METER	90.74	**90.85**	90.15	90.50	90.62	90.72
RIALTO	**92.19**	91.86	91.86	91.69	91.95	92.05
POSTURE-No8	**62.01**	61.84	61.86	61.90	61.92	61.59
POKER	**99.55**	98.88	98.77	98.70	99.02	99.11
NOMAO	**97.79**	97.42	97.51	97.49	97.67	97.61
AGR(A)	90.49	90.41	90.35	90.43	**90.51**	90.40
Mean rank	**1.29**	4.36	4.79	4.71	2.43	3.43

Table 10: Retrospective TabICL-v2 threshold sweep. Zero disables gating, and bold marks maxima selected on completed streams. The sweep describes threshold sensitivity.

Dataset	$\tau = 0$	$\tau = 0.1$	$\tau = 0.2$	$\tau = 0.3$	$\tau = 0.4$	$\tau = 0.5$
NOAA	81.64	81.64	81.83	81.82	**81.94**	81.88
METER	90.63	**90.80**	90.62	90.68	90.62	90.48
RIALTO	91.97	**92.04**	91.98	92.00	92.02	91.96
POSTURE-No8	62.01	62.03	**62.10**	62.03	62.03	62.08
NOMAO	97.64	97.71	97.82	**97.87**	97.86	97.86
AGR(A)	90.54	90.49	90.61	90.52	**90.93**	90.61

C.3 Additional Noise and Drift Results

The main tables summarize noisy-feedback robustness and recovery from concept drift. We report additional noise levels and the drift-generation and evaluation settings.

C.3.1 Additional Label-Noise Results

The symmetric-noise experiment corrupts revealed feedback labels while preserving the clean labels for evaluation. Each method uses the same corruption mask for a given stream and noise level. Table 11 reports the clean and 10% noise conditions that complement the 30% results in Table 5. RECAP has higher reported accuracy in all six additional comparisons.

Clean-to-noisy accuracy decreases use each policy's clean baseline from Table 11.

C.3.2 Drift Generation and Evaluation

We evaluate abrupt, gradual, and recurring Agrawal streams using seeds 0, 1, and 2. For each run, we apply the same seed to the stream generator and the MOA learner. The abrupt stream follows the function sequence $0 \to 3 \to 6 \to 9$, with changes at steps 7,500, 15,000, and 22,500 and transition width 1. The gradual stream uses the same function sequence and transition centers, with transition width 5,000. The recurring stream follows $0 \to 3 \to 0$, with changes at steps 10,000 and 20,000 and transition width 1.

Metric definitions. Let c_s be the indicator of a correct prediction at step s. We smooth accuracy using the trailing 500 predictions, so $\bar{a}_s = \frac{1}{500} \sum_{u=s-499}^{s} c_u$. For a drift at step d, the pre-drift reference is the average correctness over the preceding 2,000 examples,

$$a_{\mathrm{pre}} = \frac{1}{2000} \sum_{u=d-2000}^{d-1} c_u.$$

Table 11: Additional symmetric label-noise results at 0% and 10% corruption. Table 5 reports the 30% condition. Corruption is applied to the feedback labels with shared masks across methods. Bold denotes the higher accuracy within each stream and noise level. Accuracy is measured against the clean target labels.

Stream	Noise (%)	RECAP	DualFIFO
AGR(A)	0	**90.52**	90.46
	10	**90.08**	89.87
METER	0	**90.68**	90.60
	10	**88.12**	87.98
NOMAO	0	**97.87**	97.47
	10	**96.29**	93.87

The post-drift interval runs from the drift point to the step before the next drift, or to the end of the stream for the final drift. Writing e for its final step, we measure regret area as

$$\sum_{s=d}^{e} \max\bigl(0, a_{\mathrm{pre}} - \bar{a}_s\bigr).$$

Recovery is the first offset from the drift point at which the trailing 500-step accuracy reaches 95% of a_{pre} within this interval. If no such step occurs, recovery is recorded as `None/NA` and excluded from the recovery mean. Table 4(b) reports the three-seed summaries, and the evaluation artifacts retain the individual seed- and drift-level measurements.

C.4 Context-Selection Mechanisms

These analyses test whether recomputing uncertainty or changing the eviction representation improves the policy, and whether observed stream relationships support the information-based motivation.

C.4.1 Stored versus Recomputed Entropy

To assess whether stored arrival scores become too stale for admission decisions, we compare storing arrival entropy with recomputing entropy at promotion. Recomputing entropy at promotion does not consistently improve accuracy despite its additional cost. Table 12 shows changes of $+0.36$, -0.11, and -0.03 points on AGR(A), METER, and NOMAO, with runtime multipliers of 7.62, 1.77, and 8.38. The tested runs favor storing arrival entropy, with selective re-scoring left for future evaluation.

The stored-score and re-scored runs take 50m 11s and 6h 22m 38s on AGR(A), 3h 58m 49s and 7h 2m 16s on METER, and 1h 3m 2s and 8h 48m 18s on NOMAO. Their ratios appear in Table 12. These totals have a different measurement scope from the policy latency table and the original component averages.

C.4.2 Representation Used for Eviction

To test whether a backbone-derived representation improves redundancy-aware eviction, Table 13 compares standardized raw-feature distance with a latent-space alternative. The alternative representation uses cosine similarity between TabICL-v2 final-layer Key vectors. Its accuracy trails standardized raw-feature distance by 0.59 points on METER, 1.17 on NOAA, and 1.50 on RIALTO at $\tau = 0.3$. The Key vectors depend on the context and therefore cannot be assumed stable across updates. These results concern one representation and similarity choice, leaving other latent-space eviction metrics open for evaluation.

C.4.3 Empirical Diagnostics of the Theoretical Assumptions

To examine whether the mechanisms underlying the future-information view are reflected in the evaluated streams, we assess observable proxies for its four assumptions. Table 14 reports measurements on sampled stream states from NOAA, METER, NOMAO, and AGR(A), using future horizons $h \in \{100, 500\}$. These retrospective diagnostics use future observations for analysis only, not for the online retention decisions.

Table 12: Accuracy with stored or recomputed entropy and recomputed-to-stored runtime ratios. Bold marks higher accuracy.

Stream	Stored	Re-scored	Runtime
AGR(A)	90.52	**90.88**	7.62×
METER	**90.68**	90.57	1.77×
NOMAO	**97.87**	97.84	8.38×

Table 13: Euclidean raw-feature distance versus cosine similarity of TabICL-v2 final-layer Key vectors at $\tau = 0.3$. Bold marks higher accuracy for each stream. The comparison jointly changes the representation and the associated similarity measure.

Representation	METER	NOAA	RIALTO
Raw features	**90.68**	**81.82**	**92.00**
Key vectors	90.09	80.65	90.50

Table 14: Empirical proxies for the theoretical assumptions. Paired entries give results for horizons $h = 100$ and $h = 500$, respectively, while distance–loss correlation has one reported value per stream. SWD is the sliced Wasserstein ratio of recent-to-future and older-to-future feature discrepancies, where values below one favor recency. These are qualitative proxies for the theoretical conditions.

Stream	Distance–loss correlation Assumption 1	Recent/older SWD Assumption 2	Entropy Spearman Assumption 3	Same-label lift Assumption 4
NOAA	-0.027	$0.938/0.826$	$0.617/0.756$	$+0.056/+0.105$
METER	$+0.024$	$0.814/0.799$	$0.315/0.424$	$+0.154/+0.316$
NOMAO	-0.043	$0.837/0.847$	$0.757/0.772$	$+0.043/+0.092$
AGR(A)	-0.126	$1.004/0.992$	$0.458/0.516$	$+0.043/+0.075$

Recent-window local stability (Assumption 2). We compare the sliced Wasserstein distance (SWD) from the recent feature window to the future window with that from an older window to the same future window. A recent-to-future divided by older-to-future distance ratio below one indicates that recent examples better approximate the future feature distribution. The ratios are below one at both horizons on NOAA, METER, and NOMAO, while AGR(A) is nearly neutral at 1.004 and 0.992. This supports recency as a feature-distribution proxy on three streams.

Local entropy consistency (Assumption 3). Entropy Spearman measures the rank correlation between a candidate's arrival entropy and the entropy of nearby future examples. Positive correlation indicates that uncertainty at the candidate is associated with uncertainty in its local future region. All four streams have positive correlations at both horizons, consistent with the local uncertainty relationship motivating entropy-gated admission.

Local label coherence (Assumption 4). Same-label lift measures the increase in label agreement among nearby future examples relative to agreement over the full future window. Positive lift indicates that the candidate's label is more representative of its local future neighborhood than of the future window as a whole. The lift is positive on every stream at both horizons, supporting the local label agreement that motivates the coherence assumption.

Same-class evidence redundancy (Assumption 1). Distance–removal-loss correlation measures whether farther same-class examples cause larger future loss when removed. A positive correlation would support distance-based ranking of redundancy, but the reported correlations range from -0.126 to 0.024 and are negative on three streams. They therefore do not support raw-feature distance as a reliable fine-grained ranking of removal loss. Appendix C.4.2 examines an alternative representation for the eviction rule.

Taken together, these measurements are consistent with local uncertainty and label agreement, and favor recency on three of the four streams, while providing weak evidence for distance-based

Table 15: Matched policy costs under TabICL-v2. Bold marks lower latency, lower memory use, and higher throughput, with GPU peak measured as allocated memory.

Metric	RECAP	DualFIFO
p50 latency (ms)	31.48	**31.43**
p95 latency (ms)	**44.28**	44.90
Throughput (steps/s)	**32.72**	31.58
CPU peak (GiB)	**1.74**	1.76
GPU peak (GiB)	**0.64**	0.71

Table 16: Original seven-stream mean component times in milliseconds. Fit prepares a context for the frozen predictor without updating its weights. Bold denotes the smaller time in each column. Total is the sum of context preparation, prediction, and eviction time.

Policy	Total	Fit	Predict	Evict
RECAP	28.3	4.4	22.6	1.3
DualFIFO	**25.9**	**4.0**	**21.4**	**0.5**

redundancy ranking. These diagnostics assess the qualitative mechanisms behind the assumptions, rather than estimating the constants in the theoretical bounds.

C.5 Computational Cost

Under the same TabICL-v2 backbone, hardware, context budget, and stream protocol, the measured costs of RECAP and DualFIFO are comparable. Table 15 reports p50 latency of 31.48 versus 31.43 ms and similar p95 latency, throughput, and peak memory. The 0.05 ms difference in the displayed medians is small relative to the approximately 31 ms shared inference path. Appendix C.5.1 distinguishes this measurement scope from the earlier component means of 28.3 and 25.9 ms.

Classical stream learners operate in a different computational regime. Over matched 5,000-step intervals on METER, NOMAO, and POKER, the reported mean latency is 32.3 ms per example for RECAP and 0.41 ms for ARF. The TFM runs use an NVIDIA H200 GPU, whereas the classical methods use CPU execution. The full accuracy comparison with six MOA learners appears in Section 5.2. Those accuracy gaps concern the full pretrained predictor, including the context policy and the shared backbone.

C.5.1 Timing Scopes

Table 16 reports the average per-step runtime of RECAP and DualFIFO, decomposed into context fitting, prediction, and eviction, over the original seven streams. The reported total sums the measured core components used by the streaming prediction loop. Its total increases by 2.4 ms, or approximately 9.3%, for RECAP relative to DualFIFO. The later matched-policy summary reports p50 and p95 latency, throughput, and peak memory, and shows a much smaller difference in the medians. These statements describe different measurement scopes and statistics.

The later RECAP and DualFIFO table uses the same backbone, budget, hardware, and stream protocol. The displayed throughput should not be interpreted as the reciprocal of an aggregated median latency.