LEARNING WHEN TO ADAPT TO CONCEPT DRIFT FROM A SINGLE PREDICTIVE TRAJECTORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Concept drift detectors typically trigger adaptation when the detector statistic crosses a predefined threshold. While the threshold controls detector sensitivity, the ultimate goal of data stream learning is to maintain predictive performance under drift. Accordingly, a threshold adjustment method has been proposed to adjust the detector threshold according to the performance of the predictor. However, the existing method determines whether to adapt and adjusts the detector threshold by comparing the predictive performance of different thresholding schemes using future observations. This can be impractical in many real-world streaming settings, where threshold adjustment and adaptation decisions must be made online as data arrive. To address this limitation, we propose a single-trajectory framework that jointly adjusts the detector threshold and determines whether to adapt using information observed along the realized predictive trajectory. This allows the threshold to be adjusted and adaptation decisions to be made online without waiting for future observations. We provide an optimization interpretation of our threshold adjustment rule, analyze its stability, and derive a dynamic regret bound. Extensive experiments across multiple classifiers, drift detectors, datasets, and adaptation settings show that the proposed framework achieves predictive performance comparable to the state of the art methods while reducing computation overhead.

1 INTRODUCTION

In many real-world applications, including medical diagnosis (Toor et al., 2020) and financial fraud detection (Dal Pozzolo et al., 2018), data arrive continuously in the form of data streams whose underlying distributions may evolve over time. Such distributional changes, known as concept drift, can progressively affect the predictive performance of models trained on historical observations (Lu et al., 2019). To detect concept drift, many drift detectors have been proposed to monitor drift statistics, and compare them against predefined thresholds (Gama et al., 2004; Frías-Blanco et al., 2015). Once drift is detected, an adaptation strategy is triggered to help the predictor adjust to the evolving data distribution.

However, choosing an appropriate detection threshold remains challenging (Liu et al., 2023). Detecting a distributional change does not necessarily mean that adapting the predictor at that moment will improve predictive performance (Baier et al., 2023). A threshold that is too sensitive may lead to false detections and unnecessary adaptations, whereas a threshold that is too conservative may delay the detection of meaningful changes and leave an outdated predictor in use (Ross et al., 2012). Threshold adjustment should therefore account not only for the strength of drift evidence, but also for the predictive performance after adaptation. Motivated by this observation, a recent threshold adjustment method incorporates predictive performance by comparing different adaptation timings to determine the detector threshold and whether adaptation should be performed (Lu et al., 2026).

Nevertheless, the existing method adjusts the detector threshold and determines whether to adapt by comparing the predictive performance of different thresholding schemes using future observations. Because these observations are not available when a potential drift is detected, both the threshold adjustment and adaptation decision must wait until subsequent data chunks arrive (Mi, 2025). This dependence on future observations can be impractical in many online streaming settings, where both decisions need to be made as data arrive. Moreover, comparing different thresholding schemes

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.

requires maintaining multiple predictive trajectories corresponding to different adaptation timings, requiring additional model training and evaluation, thereby increasing computational overhead.

To overcome these limitations, we propose a single-trajectory framework that uses the predictive performance observed along the deployed trajectory to guide both threshold adjustment and the decision of whether to adapt. We first adjust the detection threshold according to the discrepancy between drift evidence provided by the detector and the need for adaptation indicated by predictive performance: the threshold is increased when the detector appears more sensitive than warranted by the observed performance, and decreased when performance degradation indicates a greater need for adaptation. We then determine whether adaptation should be performed based on recent performance deterioration together with prior observations of performance improvement after adaptations, rather than relying solely on whether the detector statistic crosses the threshold. In this way, both threshold adjustment and adaptation decisions rely only on observations along a single trajectory, thus can be made online without waiting for future observations or maintaining multiple trajectories.

Based on this design, we propose the Single-Trajectory Threshold Learning and Adaptation Control (STAC) framework for concept drift. Our contributions are summarized as follows:

- We propose STAC, a single-trajectory framework that jointly adjusts the detector threshold and determines whether to adapt using predictive performance observed along the deployed trajectory, without relying on future observations.
- We provide an optimization interpretation of the threshold adjustment rule, analyze its stability, and derive a dynamic regret bound.
- We conduct extensive experiments across multiple classifiers, drift detectors, datasets, and adaptation settings. The results show that STAC achieves predictive performance comparable to the state of the art (SOTA) method while reducing computational overhead.

2 RELATED WORK

Concept drift detection methods can be broadly grouped into distribution-based methods and performance-based methods (Lu et al., 2019; Gama et al., 2014).

Distribution-based methods detect concept drift by identifying changes in the underlying data distribution. Early approaches used density estimation, statistical hypothesis testing, and distance-based measures to identify distributional changes (Song et al., 2007; Dries & Rückert, 2009; Ditzler & Polikar, 2011). Subsequent work explored multivariate statistical comparisons to identify recurring concepts (Gonçalves Jr & de Barros, 2013), competence-based representations to characterize distributional changes (Lu et al., 2014), and regional density measures to capture localized drift (Liu et al., 2017). For multivariate streams, QuantTree recursively partitions the feature space to obtain compact histogram representations (Boracchi et al., 2018), while KSWIN compares reference and recent samples within a sliding window using the non-parametric Kolmogorov–Smirnov test, without requiring explicit density estimation (Raab et al., 2020). Subsequent advances extended distribution comparison to changing contextual conditions (Cobb & Van Looveren, 2022) and learned representation spaces. More recently, contrastively learned concept embeddings have been used to detect complex distribution shifts (Wan et al., 2024), while DriftLens measures distances between distributions of deep representations for real-time unsupervised drift detection (Greco et al., 2025).

Performance-based methods detect concept drift from changes in predictive behavior. Classical methods such as Page–Hinkley monitor cumulative deviations in a statistic (Page, 1954), while DDM monitors the online classification error rate and its standard deviation to identify significant performance degradation (Gama et al., 2004). EDDM improves sensitivity to gradual drift by tracking the distance between consecutive errors (Baena-García et al., 2006). ADWIN takes a different approach by adjusting the observation window according to differences between the means of recent and past observations (Bifet & Gavaldà, 2007), whereas HDDM uses concentration bounds, with arithmetic and exponentially weighted variants designed for abrupt and gradual drift, respectively (Frías-Blanco et al., 2015). However, error-based statistics may show little change during the early stages of drift. PUDD addresses this limitation by monitoring prediction uncertainty, allowing drift to be detected even when the error rate remains stable (Lu et al., 2025). Related work has also explored a range of approaches to online learning and model adaptation, including aggregated Mondrian forests (Mourtada et al., 2021), classifier reweighting (Jiao et al., 2024), and gradient-boosting

adaptation (Wang et al., 2022; 2023). Nevertheless, most drift detectors still rely on predefined thresholds for alarms.

To address this limitation, DTD dynamically adjusts the detection threshold according to predictive performance (Lu et al., 2026). When a potential drift is detected, it compares the future predictive performance resulting from adaptations at nearby time points, including the option of not adapting, and uses these comparisons to determine both whether adaptation should be performed and whether the detector threshold should be adjusted to trigger adaptation earlier or later. Although effective, this design relies on future observations to determine the current decision, which limits its applicability in many real-world streaming settings.

3 PROBLEM FORMULATION

Consider a data stream $\mathcal{D} = \{D_t\}_{t=1}^{T}$, where $D_t = \{(x_{t,i}, y_{t,i})\}_{i=1}^{n_t}$ denotes the data chunk arriving at time t, with $x_{t,i} \in \mathcal{X}$ and $y_{t,i} \in \mathcal{Y}$. We assume that observations in D_t are drawn from a joint distribution P_t over $\mathcal{X} \times \mathcal{Y}$. The stream is stationary if P_t remains unchanged over time; otherwise, concept drift occurs when $P_t \neq P_{t+1}$ for some $t \in \{1, \ldots, T-1\}$.

To detect concept drift, a drift detector produces a statistic S_t from the observations available at time t, then compares it with a predefined detector threshold. For detectors where a larger S_t indicates stronger drift evidence, a drift signal is raised when S_t exceeds the threshold; for detectors with the opposite convention, the inequality is reversed. The detector threshold therefore controls the sensitivity of drift detection.

Let f_t denote the predictor deployed at time t. Upon receiving D_t, the predictor first makes predictions on the current data chunk, from which its predictive performance A_t is observed. We use $\alpha_t \in \{0, 1\}$ to denote the adaptation decision after observing D_t, where $\alpha_t = 1$ indicates that the predictor is adapted, and $\alpha_t = 0$ otherwise. A drift signal indicates evidence of a potential distributional change, but does not necessarily imply that adapting the predictor at that time will improve predictive performance. We therefore distinguish the detector threshold from the adaptation decision: the threshold controls when the detector raises a drift signal, whereas α_t determines whether the predictor is actually adapted.

We consider a single-trajectory setting with a deployed predictor trajectory $F_{1:T} = (f_t)_{t=1}^{T}$. At time t, only information generated along this realized trajectory is available. Let $S_{1:t} = (S_k)_{k=1}^{t}$, $A_{1:t} = (A_k)_{k=1}^{t}$, and $\alpha_{1:t-1} = (\alpha_k)_{k=1}^{t-1}$ denote the observed detector statistics, predictive performance, and previous adaptation decisions, respectively. We denote the information available at time t by

$$\mathcal{I}_t = (S_{1:t}, A_{1:t}, \alpha_{1:t-1}). \tag{1}$$

Only the predictive performance generated by the realized adaptation history is observed; future observations and the predictive outcomes under alternative adaptation decisions are unavailable.

Let θ_t denote the adjustable detector threshold parameter at time t. Under this information constraint, given $\mathcal{I}_t$, we define a threshold adjustment rule Φ to update the detector threshold, together with an adaptation policy π to determine whether the predictor should be adapted. The predictor is then updated according to the resulting adaptation decision:

$$\theta_{t+1} = \Phi(\theta_t, \mathcal{I}_t), \tag{2}$$
$$\alpha_t = \pi(\theta_{t+1}, \mathcal{I}_t), \tag{3}$$
$$f_{t+1} = \mathcal{T}(f_t, D_t, \alpha_t), \tag{4}$$

where Φ denotes the threshold adjustment rule, π the adaptation decision rule, and $\mathcal{T}$ the predictor update operator.

Our goal is therefore to design Φ and π so that predictive performance is maintained under concept drift using only information available along the realized trajectory, without relying on future observations or alternative predictive outcomes.

4 METHOD

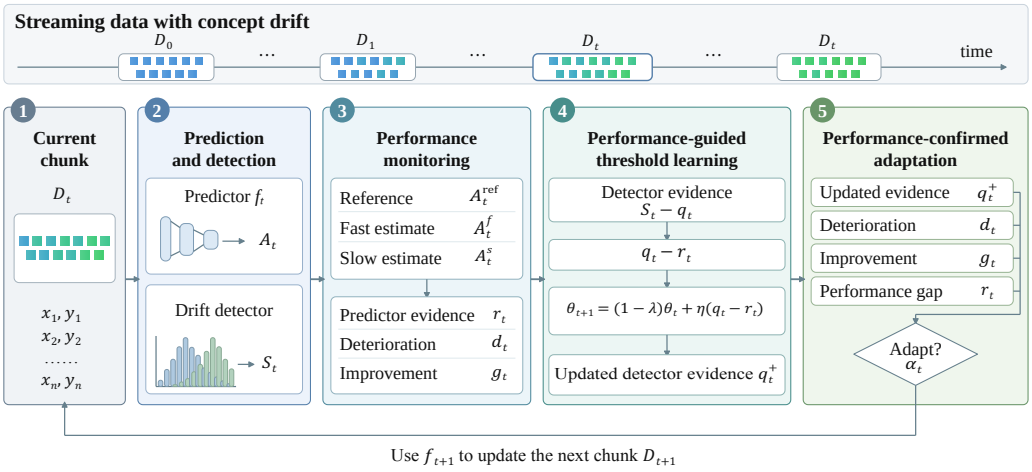


Figure 1: Overview of the STAC framework.

The complete pseudocode of the proposed method is provided in Appendix A.1.

4.1 PERFORMANCE-GUIDED THRESHOLD LEARNING

To instantiate the threshold adjustment rule Φ, we compare drift evidence (denoted by q_t) from the detector with performance evidence (denoted by r_t) from the observed performance. Since the two are measured on different scales, we map them to $[0, 1]$ for comparison, and the detector threshold is then adjusted according to their discrepancy.

Detector evidence. Since drift statistics differ in scale and direction across detectors, we first transform S_t into a common coordinate $\widetilde{S}_t$, where the detector's predefined threshold θ^{base} corresponds to zero, with larger values indicate stronger drift evidence. For detectors with monotonic statistics, we use a log-ratio transformation, typically $\widetilde{S}_t = \log(S_t/\theta^{\text{base}})$ or $\widetilde{S}_t = \log(\theta^{\text{base}}/S_t)$, with detector-specific mappings provided in Appendix A.3.5. We then define the detector evidence as

$$q_t = \sigma\left(\frac{\widetilde{S}_t - \theta_t}{\tau}\right), \tag{5}$$

where θ_t is the adjustable detector threshold parameter in the transformed coordinate, initialized with $\theta_1 = 0$, and $\tau > 0$ controls the smoothness of the mapping.

Predictor evidence. We measure predictor evidence as the need for adaptation using the degradation of recent performance relative to a recently achieved performance level. Let A_t denote the predictive performance observed at time t. To track the recently achieved performance level, we maintain a decaying reference $A_t^{\text{ref}} = \max\big(A_t, (1-\rho)A_{t-1}^{\text{ref}}\big)$, where $\rho \in (0, 1)$ controls how quickly the reference decays over time. Since the performance on a single data chunk can be noisy, we maintain a fast estimate of recent performance:

$$A_t^f = \begin{cases} A_t, & \alpha_{t-1} = 1, \\ (1-\beta_f)A_{t-1}^f + \beta_f A_t, & \alpha_{t-1} = 0, \end{cases} \tag{6}$$

where $\beta_f \in (0, 1]$ controls how quickly the estimate responds to recent observations. When adaptation occurs, the estimate is reset to A_t so that it reflects the performance of the adapted predictor.

We calculate performance degradation relative to the previous reference level as $[A_{t-1}^{\text{ref}} - A_t^f]_+$, where $[x]_+ = \max(x, 0)$. A larger value indicates a greater decline compared to the recently

achieved level. To compare the two sources of evidence on a common $[0, 1]$ scale, we map this degradation to the predictor evidence $r_t \in [0, 1]$ using

$$r_t = \frac{\sigma\left(\frac{[A_{t-1}^{\text{ref}} - A_t^f]_+ - \delta}{\gamma}\right) - \sigma(-\delta/\gamma)}{1 - \sigma(-\delta/\gamma)}, \tag{7}$$

where $\delta > 0$ and $\gamma > 0$ control the shift and steepness of the mapping. This normalization ensures that $r_t = 0$ when no performance degradation is observed.

Threshold update. Therefore, the discrepancy $q_t - r_t$ reflects the mismatch between drift evidence and performance evidence. We instantiate the threshold adjustment rule Φ as

$$\theta_{t+1} = (1 - \lambda)\theta_t + \eta(q_t - r_t), \tag{8}$$

where $\eta > 0$ controls the adjustment rate and $\lambda \in (0, 1)$ gradually pulls the threshold parameter back toward zero (the initial value). When $q_t > r_t$, the discrepancy term pushes the threshold upward, reducing detector sensitivity; when $q_t < r_t$, it pushes the threshold downward, reflecting a greater need for adaptation.

4.2 Performance-Confirmed Adaptation

After the detector threshold is adjusted, we recompute the detector evidence q_t^+ using the updated threshold θ_{t+1} following Eq. 5. When $q_t^+ > 0.5$, which indicates that the transformed drift statistic exceeds the updated threshold, the drift alarm does not necessarily imply that adaptation should be performed. We therefore further determine whether to adapt using the rule described below.

Ongoing performance deterioration. For the ongoing-deterioration case, adaptation requires evidence that predictive performance is declining. To characterize the trend, in addition to the fast estimate A_t^f, we maintain a slower estimate $A_t^s = (1 - \beta_s)A_{t-1}^s + \beta_s A_t$, where $\beta_s < \beta_f$, and compare the two. We define $d_t = [A_t^s - A_t^f]_+$ as the performance deterioration. Thus, $d_t > 0$ indicates that recent performance has fallen below the slower trend, with a larger d_t indicating stronger deterioration; otherwise, $d_t = 0$.

However, current performance deterioration alone is not sufficient to determine whether adaptation should be performed. If previous adaptations were followed by little performance improvement, we prefer to be more cautious about another adaptation. We therefore maintain a decayed estimate of the performance improvement following previous adaptations:

$$g_t = (1 - \lambda)g_{t-1} + \eta\alpha_{t-1}(A_t - A_{t-1}). \tag{9}$$

Only when adaptation was performed at $t - 1$ is the subsequent performance change $A_t - A_{t-1}$ incorporated into g_t. Thus, g_t summarizes the accumulated performance changes observed after previous adaptations, with more recent observations receiving greater weight. A positive value indicates improvement, whereas a negative value indicates deterioration.

We then combine the performance deterioration d_t with g_t to determine whether adaptation should be performed. For the ongoing-deterioration case, we confirm adaptation only when $d_t > 0$ and $d_t + g_t > b$, where $b > 0$ is a fixed confirmation level. A larger g_t indicates greater performance improvement observed after previous adaptations, so less current deterioration is required to support adaptation. In contrast, when g_t is small or negative, stronger current deterioration is required for confirmation.

Persistent performance deficit. After ongoing deterioration has triggered adaptation, the detector may continue to raise drift alarms. In this case, the recent predictive performance after adaptation may still remain below the previously achieved performance level. Since d_t may already be close to zero, the ongoing-deterioration condition may no longer support further adaptation. We therefore reuse the predictor evidence r_t to measure the remaining performance gap. To determine whether further adaptation is needed, we again combine it with the performance improvement following previous adaptations g_t, and confirm adaptation when

$$g_t > 0 \quad \text{and} \quad r_t - g_t > c, \tag{10}$$

where $c > 0$ is a confirmation level. A positive g_t indicates that previous adaptations were followed by overall improvement, while $r_t - g_t > c$ indicates that a substantial performance gap remains.

Adaptation rule. Combining the drift alarm with the two performance-based conditions, we instantiate the adaptation policy π as

$$\alpha_t = \mathbb{I}(q_t^+ > 0.5)\mathbb{I}\left([d_t > 0 \wedge d_t + g_t > b] \vee [g_t > 0 \wedge r_t - g_t > c]\right), \tag{11}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function.

5 THEORETICAL ANALYSIS

We analyze the threshold adjustment rule from an online optimization perspective. We first show that the update admits an exact online gradient descent interpretation, then characterize the discrepancy between threshold sequences under perturbations, and finally derive a dynamic regret bound as the surrogate-optimal threshold changes over time. Detailed proofs are provided in Appendix A.2.

Optimization interpretation. We define the following time-varying surrogate objective:

$$L_t(\theta) = \frac{\lambda}{2\eta}\theta^2 + r_t\theta + \tau \log\left(1 + \exp\left(\frac{\widetilde{S}_t - \theta}{\tau}\right)\right). \tag{12}$$

The objective combines the detector evidence and performance evidence used in the threshold update. The quadratic term pulls the threshold parameter toward zero, which corresponds to the predefined detector threshold in the transformed coordinate.

At each time step, r_t and $\widetilde{S}_t$ are held fixed when differentiating $L_t(\theta)$ with respect to θ. Differentiating gives $\nabla L_t(\theta) = \frac{\lambda}{\eta}\theta + r_t - \sigma((\widetilde{S}_t - \theta)/\tau)$. Since $q_t = \sigma((\widetilde{S}_t - \theta_t)/\tau)$,

$$\theta_t - \eta\nabla L_t(\theta_t) = (1-\lambda)\theta_t + \eta(q_t - r_t) = \theta_{t+1}. \tag{13}$$

Therefore, the threshold update is exactly one online gradient descent step on L_t with step size η.

Moreover, since $\lambda/\eta \le \nabla^2 L_t(\theta) \le \lambda/\eta + 1/(4\tau)$, L_t is λ/η-strongly convex and has a Lipschitz-continuous gradient. Therefore, L_t admits a unique minimizer θ_t^* at each time step.

Threshold stability under perturbations. We next consider two threshold sequences $\{\theta_t\}$ and $\{\theta_t'\}$ with initial values θ_1 and θ_1', driven by possibly different detector and predictor evidence. For $0 < \lambda < 1$,

$$|\theta_t - \theta_t'| \le (1-\lambda)^{t-1}|\theta_1 - \theta_1'| + \frac{2\eta}{\lambda}\left[1 - (1-\lambda)^{t-1}\right]. \tag{14}$$

Therefore, $\limsup_{t\to\infty} |\theta_t - \theta_t'| \le 2\eta/\lambda$. The contribution of the initial threshold discrepancy to the bound decays geometrically, while the overall threshold discrepancy remains bounded.

Dynamic regret. Let θ_t^* denote the unique minimizer of L_t, and define $\Delta_t^* = |\theta_{t+1}^* - \theta_t^*|$ and $V_T = \sum_{t=1}^{T-1}(\Delta_t^*)^2$ to characterize how the surrogate-optimal threshold changes over time. If $\lambda + \eta/(4\tau) < 1$, the dynamic regret $\mathrm{Regret}_T^{\mathrm{dyn}} = \sum_{t=1}^{T}[L_t(\theta_t) - L_t(\theta_t^*)]$ satisfies

$$\mathrm{Regret}_T^{\mathrm{dyn}} \le \left(\frac{\lambda}{\eta} + \frac{1}{4\tau}\right)\left[\frac{|\theta_1 - \theta_1^*|^2}{\lambda(2-\lambda)} + \frac{V_T}{\lambda^2}\right]. \tag{15}$$

Therefore, $\mathrm{Regret}_T^{\mathrm{dyn}} = O(1 + V_T)$. In particular, if $V_T = o(T)$, the learned threshold sequence achieves vanishing average regret relative to the sequence of surrogate-optimal thresholds $(\theta_t^*)_{t=1}^T$.

6 EXPERIMENTS

We evaluate STAC across multiple classifiers, drift detectors, datasets, and adaptation settings. We use three real-world data streams, Airline (Ikonomovska, 2011), Elec2 (Harries, 1999), and PowerSupply (Dau et al., 2019), together with synthetic streams including Sine and Mixed (Gama et al., 2004), CIFAR-10-CD (Lu et al., 2025), and the SEA variants (Bifet et al., 2010). We consider DDM (Gama et al., 2004), HDDM-A and HDDM-W (Frías-Blanco et al., 2015), KSWIN (Raab et al., 2020), Page–Hinkley (PH) (Page, 1954), and PUDD (Lu et al., 2025) as drift detectors. For PUDD, we follow its multiple initial-threshold settings and denote a detector initialized with 10^{-x} as PUDD-x.

For the baselines, we compare STAC with Fixed and the state-of-the-art threshold adjustment method DTD (Lu et al., 2026). Fixed retains the predefined detector threshold throughout the stream, whereas DTD adjusts the threshold by comparing the future predictive performance associated with different adaptation timings using multiple predictive trajectories. Detection thresholds are initialized according to the original methods where applicable.

We use GNB, VFDT, and DNN as base classifiers and consider continual and sporadic training. The classifier is updated at every time step under continual training and only when adaptation is triggered under sporadic training. For STAC, we set $\tau = 0.10$, $\eta = 0.25$, $\lambda = 0.005$, $\rho = 0.01$, $\beta_f = 0.60$, $\beta_s = 0.01$, $\delta = 0.005$, and $\gamma = 0.01$. Additional implementation details are provided in Appendix A.3.

6.1 Main Results

Table 1: Comparison of Fixed, DTD, and STAC (ours) in average accuracy (%) over eight datasets, for eight detector configurations and two training settings. Best in each setting shown in **bold**.

Detector	Training	GNB			VFDT			DNN		
		Fixed	DTD	STAC	Fixed	DTD	STAC	Fixed	DTD	STAC
KSWIN	continual	76.19±0.92	78.01±0.92	**78.96±0.45**	79.31±0.50	79.28±0.57	**80.21±0.42**	**82.58±0.36**	**82.58±0.37**	82.49±0.92
	sporadic	75.07±1.53	76.45±1.56	**77.33±0.90**	77.40±1.95	77.88±2.77	**79.58±0.51**	79.34±1.00	80.52±1.27	**81.67±0.48**
DDM	continual	77.42±0.57	**77.83±0.63**	77.64±0.44	79.76±0.50	79.15±1.45	**80.19±0.39**	**82.63±0.39**	82.47±1.04	82.57±0.38
	sporadic	76.96±0.77	76.67±0.88	**77.49±0.51**	79.08±0.57	78.38±0.96	**79.72±0.49**	79.93±2.91	80.08±2.88	**81.49±0.66**
PH	continual	78.17±0.42	78.22±0.46	**79.12±0.45**	79.97±0.42	79.97±0.44	**80.14±0.41**	82.57±0.52	**82.59±0.55**	82.50±0.81
	sporadic	77.17±0.79	77.13±1.04	**77.41±0.60**	79.05±0.85	79.21±0.92	**79.71±0.52**	81.09±0.51	81.37±1.85	**81.70±0.48**
HDDM-A	continual	77.56±0.45	77.90±0.44	**79.17±0.42**	79.85±0.42	79.81±0.49	**80.09±0.42**	**82.62±0.37**	**82.62±0.36**	82.58±0.49
	sporadic	**77.69±0.50**	77.25±1.56	77.11±0.96	79.26±0.75	79.19±1.13	**79.37±1.67**	81.18±0.62	81.34±2.05	**81.37±0.58**
HDDM-W	continual	76.57±0.54	78.20±0.64	**78.86±0.48**	79.66±0.43	79.82±0.47	**80.17±0.43**	82.54±0.75	81.37±4.07	**82.58±0.37**
	sporadic	76.10±1.00	77.11±0.73	**77.29±0.54**	78.94±0.56	78.92±0.70	**79.36±0.47**	80.57±1.09	79.02±4.60	**81.33±0.48**
PUDD-1	continual	78.18±0.48	78.08±0.53	**79.00±0.47**	79.67±0.47	75.98±2.54	**80.11±0.44**	82.46±1.07	82.28±2.55	**82.55±0.73**
	sporadic	75.49±2.59	76.35±1.19	**77.10±0.63**	78.23±1.66	79.17±0.72	**79.58±0.54**	80.63±1.72	81.34±0.86	**81.65±0.54**
PUDD-3	continual	78.16±0.45	78.19±0.45	**79.02±0.44**	79.62±0.48	75.56±2.79	**80.10±0.45**	**82.54±0.79**	82.14±2.69	**82.54±0.80**
	sporadic	75.88±0.94	76.83±0.76	**77.16±0.53**	78.88±0.74	79.27±0.45	**79.60±0.52**	80.92±0.60	81.33±0.60	**81.64±0.51**
PUDD-5	continual	78.08±0.43	77.95±0.43	**79.02±0.44**	79.60±0.50	75.60±2.76	**80.10±0.45**	**82.59±0.41**	82.16±2.55	82.52±0.68
	sporadic	76.13±0.54	76.81±0.77	**77.23±0.50**	78.97±0.52	79.18±0.47	**79.63±0.49**	80.65±0.72	81.10±0.71	**81.48±0.56**

Table 1 compares Fixed, DTD, and STAC across eight detector configurations, three base classifiers, and two training settings. Each entry reports the mean accuracy across eight datasets, with each dataset averaged over 20 random seeds. Using only a single predictive trajectory, STAC achieves the highest average accuracy in 41 of the 48 settings, including all 16 settings with VFDT. For GNB and DNN, STAC obtains the highest mean accuracy in 14 and 11 of the 16 settings, respectively.

Compared with the SOTA threshold adjustment method DTD, STAC achieves higher accuracy in 43 of the 48 settings, including 14/16 with GNB, 16/16 with VFDT, and 13/16 with DNN. Across all detectors, datasets, and training settings, STAC improves over DTD by 0.622, 1.331, and 0.524 percentage points for GNB, VFDT, and DNN, respectively. Detailed results are provided in the Appendix A.4.

6.2 Evaluation in Complex Scenarios

Table 2: Comparison of Fixed, DTD, and STAC variants in average accuracy (%) on CIFAR-10-CD across eight detector configurations. Best accuracy for each detector shown in **bold**. Numbers in parentheses indicate the average number of adaptations.

Detector	Fixed	DTD	STAC	STAC ($K = 2$)	STAC ($K = 3$)	STAC ($K = 4$)
KSWIN	75.79±0.89 (93.15)	76.80±0.93 (14.60)	76.55±0.88 (48.15)	77.84±0.90 (5.25)	**78.00±0.83** (5.40)	77.74±1.10 (4.15)
DDM	76.24±0.84 (78.85)	77.94±0.73 (4.40)	77.04±0.89 (45.45)	77.78±0.93 (5.25)	**77.97±0.85** (5.15)	77.81±1.05 (4.05)
PH	77.07±1.23 (32.25)	77.88±0.65 (9.90)	76.73±0.88 (44.30)	77.79±0.92 (5.45)	**77.97±0.78** (5.40)	77.77±1.06 (4.20)
HDDM-A	76.28±0.81 (88.45)	77.81±0.98 (6.20)	76.93±0.85 (50.70)	77.82±0.90 (4.80)	**78.02±0.79** (5.40)	77.73±1.08 (4.10)
HDDM-W	76.40±0.62 (88.25)	77.51±1.00 (5.05)	76.80±0.94 (56.30)	77.80±0.95 (5.45)	**77.96±0.77** (5.55)	77.73±1.10 (4.30)
PUDD-1	77.52±0.80 (19.50)	77.76±1.08 (7.70)	76.83±0.93 (43.30)	77.54±1.12 (5.10)	**77.94±0.82** (5.25)	77.72±1.10 (4.25)
PUDD-3	77.29±0.85 (17.65)	77.76±1.08 (7.70)	77.01±0.76 (43.45)	77.50±1.14 (4.90)	**77.94±0.82** (5.25)	77.72±1.11 (4.20)
PUDD-5	77.61±0.92 (16.35)	77.76±1.08 (7.70)	77.13±0.72 (42.90)	77.56±1.12 (4.90)	**77.94±0.82** (5.25)	77.72±1.11 (4.20)

We further evaluate STAC on CIFAR-10-CD in a more complex data stream. Following the original setting, three CIFAR-10 classes are initially positive and the remaining classes are negative, with their states evolving through a Markov process across 100 data chunks, with each chunk including 500 inputs. We use ResNet-18 as the base classifier with SGD at a learning rate of 0.01, following the original CIFAR-10-CD protocol (Lu et al., 2025). Fixed, DTD, and STAC are evaluated with eight detector configurations over 20 random seeds. The results are shown in Table 2.

Compared with the classic data streams, CIFAR-10-CD produces much more frequent adaptation decisions. Averaged over the eight detector configurations, the fixed-threshold baseline triggers 54.31 adaptations over the 100 chunks, while STAC reduces this number to 46.82. In contrast, DTD triggers only 7.91 adaptations on average, as it uses future predictive performance to guide adaptation decisions. This difference is also reflected in predictive performance, with STAC achieving an average accuracy of 76.88%, compared with 77.65% for DTD.

We observe that, in this more complex stream, predictive performance may remain unstable for several chunks after adaptation. As a result, the fast performance estimate can remain below the slow estimate, causing the deterioration condition to remain active across consecutive chunks and potentially trigger repeated adaptations within the same deterioration episode.

To reduce repeated adaptations caused by persistent performance deterioration, we introduce a K-step persistence rule. The condition $(d_t > 0 \wedge d_t + g_t > b)$ must hold for K consecutive chunks, and adaptation based on this condition can be triggered only at the K-th consecutive chunk. Once this point is reached, the same uninterrupted deterioration episode cannot trigger another adaptation until the condition becomes false. In contrast, the condition $(g_t > 0 \wedge r_t - g_t > c)$ is evaluated immediately without the persistence requirement.

To examine this modification, we vary K and report the results in Table 2. The persistence rule substantially reduces the number of adaptations, with the average decreasing from 46.82 to 5.14, 5.33, and 4.18 for $K = 2, 3$, and 4, respectively, while the corresponding average accuracies increase from 76.88% to 77.70%, 77.97%, and 77.74%. All three persistence settings achieve higher average accuracy than DTD. Among them, $K = 3$ performs best, achieving the highest accuracy for all eight detector configurations and exceeding DTD by 0.31 percentage points on average.

6.3 Efficiency Analysis

We then evaluate computational efficiency using algorithm runtime under matched configurations. Overall, STAC requires only $1.064\times$ the runtime of Fixed, whereas DTD requires $2.053\times$, corresponding to approximately $1.93\times$ the runtime of STAC. This advantage is consistent across classifier and training settings, where DTD requires $1.46\times$–$2.25\times$ the runtime of STAC. These results show that the single-trajectory design substantially reduces the computational overhead of performance-guided threshold adjustment and adaptation. Full runtime results are provided in Appendix A.4.3.

6.4 Ablation Study

Table 3: Ablation study of STAC components under continual and sporadic training. Results are reported as average accuracy (%) in mean ± standard deviation. Best in each setting shown in **bold**.

	GNB		VFDT		DNN		ResNet-18
Variant	Continual	Sporadic	Continual	Sporadic	Continual	Sporadic	Continual
Fixed	77.54±12.22	76.31±12.83	80.45±10.02	79.24±9.73	**83.56±11.06**	81.23±10.86	76.77±1.09
+ TL	77.48±13.00	76.75±13.03	80.76±9.93	80.06±9.70	83.52±11.11	**82.49±10.79**	76.50±0.81
+ DC	78.90±9.80	75.73±13.33	80.49±10.03	79.28±9.54	83.55±11.08	81.35±11.05	**78.08±0.79**
+ DC + RC	78.88±9.84	76.63±12.39	79.83±10.13	78.79±9.90	82.60±11.29	80.71±11.42	78.05±0.77
+ TL + DC	**79.03±10.31**	76.39±13.15	80.77±10.10	80.09±9.71	83.54±11.10	82.38±10.87	78.06±0.83
STAC	78.85±10.72	**77.27±12.25**	**80.89±9.97**	**80.14±9.70**	83.52±11.12	82.43±10.83	77.97±0.81

TL: Threshold Learning; DC: Deterioration Confirmation; RC: Re-adaptation Confirmation.

We also conduct an ablation study to examine the contributions of threshold learning (TL), deterioration confirmation (DC), and re-adaptation confirmation (RC). TL corresponds to the threshold adjustment rule, DC to the condition $(d_t > 0 \wedge d_t + g_t > b)$, and RC to the condition

$(g_t > 0 \wedge r_t - g_t > c)$. Starting from the Fixed baseline, we consider different combinations of these components under continual and sporadic training. Full results are provided in Appendix A.4.4.

As shown in Table 3, TL and DC contribute differently across training settings. TL is more beneficial under sporadic training, improving GNB, VFDT, and DNN over Fixed by 0.44, 0.82, and 1.26 percentage points, respectively. In contrast, DC provides its largest gains for GNB under continual training and for ResNet-18, improving accuracy by 1.36 and 1.31 percentage points, respectively. Combining TL and DC improves over Fixed in six of the seven reported settings. RC has a more selective contribution. Adding RC to TL and DC increases the mean accuracy in four of the seven settings, with the largest gain for GNB under sporadic training, from 76.39% to 77.27%. In the remaining settings, the changes are within 0.18 percentage points. The full STAC achieves the highest mean accuracy in three settings and remains within 0.18 percentage points of the best ablation variant in each of the other four settings.

Overall, the three components address distinct failure modes in the adaptation process. TL addresses inappropriate detector sensitivity by adjusting the threshold according to the discrepancy between drift evidence and performance evidence. DC reduces unnecessary adaptations by requiring a drift alarm to be supported by ongoing performance deterioration. RC addresses insufficient re-adaptation by allowing another adaptation when a substantial performance deficit remains after previous adaptations. Their combination allows STAC to maintain competitive performance across different classifiers and training settings.

6.5 Sensitivity Analysis

We examine the sensitivity of STAC to seven key hyperparameters. Overall, predictive performance remains relatively stable across the tested parameter ranges. On the classic data streams, the default value achieves the highest average accuracy for four of the seven parameters; for the remaining three, its gap from the best setting is below 0.02 percentage points. On CIFAR-10-CD, for each parameter, the difference between the highest and lowest average accuracy across five tested values is below 0.17 percentage points. Noticeable degradation occurs mainly at a few extreme values, suggesting that STAC does not require fine-grained hyperparameter tuning within the tested ranges.

Despite the stable predictive performance, the parameters still affect adaptation frequency. On the classic data streams, increasing τ generally reduces the number of adaptations with little change in average accuracy. A larger β_f makes the fast performance estimate more responsive to recent changes and generally leads to more frequent adaptations. Similarly, increasing the confirmation levels b and c makes adaptation more conservative, with the average number of adaptations across the three classifiers generally decreasing. These trends are consistent with the intended roles of the parameters, while the small variation in predictive performance indicates that STAC remains effective over a broad range of values. Complete results are provided in Appendix A.4.5.

7 Conclusion

In this work, we proposed STAC, a single-trajectory framework that jointly adjusts the detector threshold and determines whether to adapt using only information observed along the deployed predictive trajectory. Unlike the existing threshold adjustment method that compares future predictive performance across multiple alternative trajectories, STAC makes both decisions online without waiting for future observations or maintaining multiple predictive trajectories. We provided an online optimization interpretation of the threshold adjustment rule, characterized its stability under perturbations, and derived a dynamic regret bound for tracking the time-varying minimizer of the surrogate objective. Experiments across multiple classifiers, drift detectors, datasets, and training settings show that STAC achieves predictive performance comparable to the state of the art while substantially reducing computational overhead; the ablation and sensitivity analyses further show that its components contribute differently across training settings and that its performance remains stable across a broad range of hyperparameter values. These results show that performance-guided threshold adjustment and adaptation decisions can be carried out online from a single realized predictive trajectory. Future work could extend STAC to semi-supervised settings where performance feedback is delayed or limited.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing, identifying potentially relevant related work, checking mathematical claims, and debugging code. We have not used generative AI tools for research ideation or execution, drafting substantive sections of the paper, or generating synthetic datasets. We have reviewed all AI-assisted work. Specifically, we manually reviewed and revised all AI-assisted language edits, read and verified the original sources of related work suggested by generative AI, independently checked the mathematical arguments against the AI-generated verification reports, and reviewed and tested all AI-assisted code modifications for correctness. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To support the reproducibility of our research, we provide key materials in the supplementary file accompanying this submission. In particular, we include the core implementation of STAC. The complete source code will be released upon publication. We also provide detailed implementation information in Appendix A.3, including hyperparameter settings, training and adaptation protocols, datasets, and baseline configurations. Together, these materials provide the main algorithmic and experimental details needed to reproduce the reported results.

REFERENCES

Manuel Baena-García, José del Campo-Ávila, Raul Fidalgo, Albert Bifet, Ricard Gavaldà, and Rafael Morales-Bueno. Early drift detection method. In *Fourth International Workshop on Knowledge Discovery from Data Streams*, pp. 77–86, 2006.

Lucas Baier, Tim Schlör, Jakob Schoeffer, and Niklas Kühl. Detecting concept drift with neural network model uncertainty. In *Proceedings of the 56th Hawaii International Conference on System Sciences*, pp. 835–844, 2023.

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

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*, volume 11 of *Proceedings of Machine Learning Research*, pp. 44–50. PMLR, 2010. URL https://proceedings.mlr.press/v11/bifet10a.html.

Giacomo Boracchi, Diego Carrera, Cristiano Cervellera, and Danilo Macciò. QuantTree: Histograms for change detection in multivariate data streams. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 639–648. PMLR, 2018. URL https://proceedings.mlr.press/v80/boracchi18a.html.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Oliver Cobb and Arnaud Van Looveren. Context-aware drift detection. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 4087–4111. PMLR, 2022. URL https://proceedings.mlr.press/v162/cobb22a.html.

Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8):3784–3797, 2018. doi: 10.1109/TNNLS.2017.2736643.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The UCR time series archive.

IEEE/CAA Journal of Automatica Sinica, 6(6):1293–1305, 2019. doi: 10.1109/JAS.2019.1911747.

Gregory Ditzler and Robi Polikar. Hellinger distance based drift detection for nonstationary environments. In *2011 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*, pp. 41–48, 2011. doi: 10.1109/CIDUE.2011.5948491.

Anton Dries and Ulrich Rückert. Adaptive concept drift detection. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 2(5-6):311–327, 2009. doi: 10.1002/sam.10054.

Isvani Frías-Blanco, José del Campo-Ávila, Gonzalo Ramos-Jiménez, Rafael Morales-Bueno, Agustín Ortiz-Díaz, and Yailé Caballero-Mota. Online and non-parametric drift detection methods based on hoeffding's bounds. *IEEE Transactions on Knowledge and Data Engineering*, 27 (3):810–823, 2015. doi: 10.1109/TKDE.2014.2345382.

João Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with drift detection. In *Advances in Artificial Intelligence – SBIA 2004*, pp. 286–295. Springer Berlin Heidelberg, 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), 2014. doi: 10.1145/2523813.

Paulo Mauricio Gonçalves Jr and Roberto Souto Maior de Barros. RCD: A recurring concept drift framework. *Pattern Recognition Letters*, 34(9):1018–1025, 2013. doi: 10.1016/j.patrec.2013.02.005.

Salvatore Greco, Bartolomeo Vacchetti, Daniele Apiletti, and Tania Cerquitelli. Unsupervised concept drift detection from deep learning representations in real-time. *IEEE Transactions on Knowledge and Data Engineering*, 37(10):6232–6245, 2025. doi: 10.1109/TKDE.2025.3593123.

Michael Harries. Splice-2 comparative evaluation: Electricity pricing. Technical Report UNSW-CSE-TR-9905, University of New South Wales, School of Computer Science and Engineering, 1999. URL `https://cgi.cse.unsw.edu.au/~reports/papers/9905.pdf`.

Elena Ikonomovska. Airline dataset, 2011. URL `http://kt.ijs.si/elena_ikonomovska/data.html`. Accessed on 02/06/2020.

Botao Jiao, Yinan Guo, Cuie Yang, Jiayang Pu, Zhiji Zheng, and Dunwei Gong. Incremental weighted ensemble for data streams with concept drift. *IEEE Transactions on Artificial Intelligence*, 5(1):92–103, 2024. doi: 10.1109/TAI.2022.3224416.

Anjin Liu, Yiliao Song, Guangquan Zhang, and Jie Lu. Regional concept drift detection and density synchronized drift adaptation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 2280–2286, 2017. doi: 10.24963/ijcai.2017/317.

Anjin Liu, Jie Lu, Yiliao Song, Junyu Xuan, and Guangquan Zhang. Concept drift detection delay index. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4585–4597, 2023. doi: 10.1109/TKDE.2022.3153349.

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.

Ning Lu, Guangquan Zhang, and Jie Lu. Concept drift detection via competence models. *Artificial Intelligence*, 209:11–28, 2014. doi: 10.1016/j.artint.2014.01.001.

Pengqian Lu, Jie Lu, Anjin Liu, and Guangquan Zhang. Early concept drift detection via prediction uncertainty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19124–19132, 2025. doi: 10.1609/aaai.v39i18.34105.

Pengqian Lu, Jie Lu, Anjin Liu, En Yu, and Guangquan Zhang. Autonomous concept drift threshold determination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24079–24087, 2026. doi: 10.1609/aaai.v40i29.39586.

Yun-Long Mi. Concept neural network based on time-delay regret for dynamic stream learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3796–3814, 2025. doi: 10.1109/TPAMI.2025.3535636.

Jaouad Mourtada, Stéphane Gaïffas, and Erwan Scornet. AMF: Aggregated mondrian forests for online learning. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(3): 505–533, 2021. doi: 10.1111/rssb.12425.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Science & Business Media, 2013.

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

Christoph Raab, Moritz Heusinger, and Frank-Michael Schleif. Reactive soft prototype computing for concept drift streams. *Neurocomputing*, 416:340–351, 2020.

Gordon J. Ross, Niall M. Adams, Dimitris K. Tasoulis, and David J. Hand. Exponentially weighted moving average charts for detecting concept drift. *Pattern Recognition Letters*, 33(2):191–198, 2012. doi: 10.1016/j.patrec.2011.08.019.

Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. Statistical change detection for multi-dimensional data. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 667–676. Association for Computing Machinery, 2007. doi: 10.1145/1281192.1281264.

Ashraf Tahmasbi, Ellango Jothimurugesan, Srikanta Tirthapura, and Phillip B. Gibbons. Driftsurf: Stable-state / reactive-state learning under concept drift. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10054–10064. PMLR, 2021. URL https://proceedings.mlr.press/v139/tahmasbi21a.html.

Affan Ahmed Toor, Muhammad Usman, Farah Younas, Alvis Cheuk M. Fong, Sajid Ali Khan, and Simon Fong. Mining massive e-health data streams for IoMT enabled healthcare systems. *Sensors*, 20(7):2131, 2020. doi: 10.3390/s20072131.

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. Association for Computing Machinery, 2024. doi: 10.1145/3637528.3672016.

Kun Wang, Jie Lu, Anjin Liu, Yiliao Song, Li Xiong, and Guangquan Zhang. Elastic gradient boosting decision tree with adaptive iterations for concept drift adaptation. *Neurocomputing*, 491: 288–304, 2022. doi: 10.1016/j.neucom.2022.03.038.

Kun Wang, Jie Lu, Anjin Liu, Guangquan Zhang, and Li Xiong. Evolving gradient boost: A pruning scheme based on loss improvement ratio for learning under concept drift. *IEEE Transactions on Cybernetics*, 53(4):2110–2123, 2023. doi: 10.1109/TCYB.2021.3109796.

A Appendix

A.1 Algorithm

Algorithm 1 Single-Trajectory Threshold Learning and Adaptation Control

Input: data stream $\mathcal{D}$, predictor f_1, detector with its predefined threshold
Input: parameters $\tau, \eta, \lambda, \rho, \beta_f, \beta_s, \delta, \gamma, b, c$, and persistence length K when enabled
Initialize $\theta_1 \leftarrow 0$, $g_0 \leftarrow 0$, $\alpha_0 \leftarrow 0$, and $h_0 \leftarrow 0$ ▷ h_t: *persistence counter for the current deterioration episode*
for $t = 1, \ldots, T$ **do**
  Predict on D_t using f_t and observe A_t ▷ *performance observation*
  if $t = 1$ **then**
    $A_t^f \leftarrow A_t$, $A_t^s \leftarrow A_t$, $A_{t-1}^{\mathrm{ref}} \leftarrow A_t$
    $g_t \leftarrow 0$
  else
    if $\alpha_{t-1} = 1$ **then**
      $A_t^f \leftarrow A_t$
    else
      $A_t^f \leftarrow (1-\beta_f)A_{t-1}^f + \beta_f A_t$
    end if
    $A_t^s \leftarrow (1-\beta_s)A_{t-1}^s + \beta_s A_t$
    $g_t \leftarrow (1-\lambda)g_{t-1} + \eta\alpha_{t-1}(A_t - A_{t-1})$
  end if
  $d_t \leftarrow [A_t^s - A_t^f]_+$ ▷ *performance tracking*
  Obtain detector statistic S_t and transform it to $\widetilde{S}_t$ ▷ *detector normalization*
  Compute predictor evidence r_t using A_{t-1}^{ref} and A_t^f
  $q_t \leftarrow \sigma((\widetilde{S}_t - \theta_t)/\tau)$ ▷ *evidence construction*
  $\theta_{t+1} \leftarrow (1-\lambda)\theta_t + \eta(q_t - r_t)$ ▷ *threshold adjustment*
  $q_t^+ \leftarrow \sigma((\widetilde{S}_t - \theta_{t+1})/\tau)$
  $A_t^{\mathrm{ref}} \leftarrow \max\{A_t, (1-\rho)A_{t-1}^{\mathrm{ref}}\}$
  $C_t^d \leftarrow (d_t > 0 \wedge d_t + g_t > b)$ ▷ *deterioration confirmation*
  $C_t^r \leftarrow (g_t > 0 \wedge r_t - g_t > c)$ ▷ *re-adaptation confirmation*
  if K-step persistence is enabled **then**
    if C_t^d **then**
      $h_t \leftarrow h_{t-1} + 1$
    else
      $h_t \leftarrow 0$
    end if
    $\alpha_t \leftarrow \mathbb{I}(q_t^+ > 0.5)\mathbb{I}\big((C_t^d \wedge h_t = K) \vee (\neg C_t^d \wedge C_t^r)\big)$ ▷ *trigger only at K*
  else
    $\alpha_t \leftarrow \mathbb{I}(q_t^+ > 0.5)\mathbb{I}(C_t^d \vee C_t^r)$ ▷ *immediate confirmation*
  end if
  $f_{t+1} \leftarrow \mathcal{T}(f_t, D_t, \alpha_t)$ ▷ *predictor adaptation*
end for
Output: $F_{1:T} = (f_t)_{t=1}^T$, $\{\theta_t\}_{t=1}^{T+1}$, and $\{\alpha_t\}_{t=1}^T$

A.2 Proofs of Theorems

A.2.1 Proof of the Optimization Interpretation

We first explain the intuition behind the surrogate objective. The threshold update

$$\theta_{t+1} = (1-\lambda)\theta_t + \eta(q_t - r_t) \tag{16}$$

contains three contributions: $-\lambda\theta_t$ pulls the threshold toward the origin of the transformed coordinate, ηq_t pushes the threshold upward when the detector provides stronger drift evidence, and $-\eta r_t$ pushes the threshold downward when predictive performance indicates a greater need for adaptation. We now show that these three contributions arise from the gradient of a single surrogate objective.

For a fixed time step t, define

$$L_t(\theta) = \frac{\lambda}{2\eta}\theta^2 + r_t\theta + \tau \log\left(1 + \exp\left(\frac{\widetilde{S}_t - \theta}{\tau}\right)\right). \tag{17}$$

Here, r_t and $\widetilde{S}_t$ are treated as fixed when differentiating with respect to θ.

Differentiating Eq. (17) with respect to θ gives

$$\nabla L_t(\theta) = \frac{\lambda}{\eta}\theta + r_t - \sigma\left(\frac{\widetilde{S}_t - \theta}{\tau}\right). \tag{18}$$

By the definition of detector evidence, $q_t = \sigma((\widetilde{S}_t - \theta_t)/\tau)$. Therefore, evaluating Eq. (18) at the current threshold gives

$$\nabla L_t(\theta_t) = \frac{\lambda}{\eta}\theta_t + r_t - q_t. \tag{19}$$

Taking one gradient descent step with step size η, we obtain

$$\begin{aligned} \theta_t - \eta\nabla L_t(\theta_t) &= \theta_t - \eta\left(\frac{\lambda}{\eta}\theta_t + r_t - q_t\right) \\ &= (1-\lambda)\theta_t + \eta(q_t - r_t) \\ &= \theta_{t+1}. \end{aligned} \tag{20}$$

Hence, the STAC threshold adjustment is exactly a gradient descent step on $L_t(\theta)$ with step size η.

We next characterize the curvature of $L_t(\theta)$. Differentiating Eq. (18) once more gives

$$\nabla^2 L_t(\theta) = \frac{\lambda}{\eta} + \frac{1}{\tau}\sigma\left(\frac{\widetilde{S}_t - \theta}{\tau}\right)\left[1 - \sigma\left(\frac{\widetilde{S}_t - \theta}{\tau}\right)\right]. \tag{21}$$

Since $0 < \sigma(z)(1-\sigma(z)) \le 1/4$ for any finite z, Eq. (21) implies

$$\frac{\lambda}{\eta} \le \nabla^2 L_t(\theta) \le \frac{\lambda}{\eta} + \frac{1}{4\tau}. \tag{22}$$

The lower bound in Eq. (22) shows that $\nabla^2 L_t(\theta) \ge \lambda/\eta > 0$. Since L_t is twice continuously differentiable, the second-order characterization of strong convexity implies that L_t is (λ/η)-strongly convex (Boyd & Vandenberghe, 2004). Similarly, the upper bound implies that $\nabla L_t(\theta)$ is Lipschitz continuous with constant $\lambda/\eta + 1/(4\tau)$ (Boyd & Vandenberghe, 2004; Nesterov, 2013).

It remains to show that a minimizer exists. Since $\nabla^2 L_t(\theta) > 0$, the gradient $\nabla L_t(\theta)$ is strictly increasing in θ. Moreover, from Eq. (18) and $\lambda/\eta > 0$,

$$\lim_{\theta\to-\infty} \nabla L_t(\theta) = -\infty, \qquad \lim_{\theta\to+\infty} \nabla L_t(\theta) = +\infty. \tag{23}$$

Because $\nabla L_t(\theta)$ is continuous and strictly increasing, Eq. (23) implies that there exists a unique $\theta_t^* \in \mathbb{R}$ such that $\nabla L_t(\theta_t^*) = 0$. By the strong convexity of L_t, this stationary point is the unique global minimizer.

Finally, r_t and $\widetilde{S}_t$ are determined by the observations available at time t and may change as the stream evolves. Consequently, both the surrogate objective L_t and its minimizer θ_t^* may vary over time. Therefore, the STAC threshold sequence can be interpreted as online gradient descent on a sequence of time-varying strongly convex objectives. □

A.2.2 Proof of Threshold Stability under Perturbations

We first give the intuition behind the stability bound. Consider two threshold sequences driven by possibly different observed signals. Subtracting their updates gives a term equal to the current

threshold discrepancy multiplied by $1-\lambda$, together with an evidence discrepancy term. The latter is uniformly bounded because the detector and predictor evidence are both bounded between 0 and 1. Recursively applying this bound yields a geometrically decaying term involving the initial threshold discrepancy and a bounded accumulated evidence discrepancy term.

The two threshold recursions are

$$\begin{aligned}\theta_{t+1} &= (1-\lambda)\theta_t + \eta(q_t - r_t),\\ \theta'_{t+1} &= (1-\lambda)\theta'_t + \eta(q'_t - r'_t).\end{aligned} \tag{24}$$

Subtracting the two updates in Eq. (24) gives

$$\theta_{t+1} - \theta'_{t+1} = (1-\lambda)(\theta_t - \theta'_t) + \eta(q_t - q'_t) - \eta(r_t - r'_t). \tag{25}$$

Taking absolute values and applying the triangle inequality yields

$$|\theta_{t+1} - \theta'_{t+1}| \le (1-\lambda)|\theta_t - \theta'_t| + \eta|q_t - q'_t| + \eta|r_t - r'_t|. \tag{26}$$

Since $q_t, q'_t \in (0,1)$ and $r_t, r'_t \in [0,1]$, we have $|q_t - q'_t| \le 1$ and $|r_t - r'_t| \le 1$. Therefore,

$$|\theta_{t+1} - \theta'_{t+1}| \le (1-\lambda)|\theta_t - \theta'_t| + 2\eta. \tag{27}$$

Applying Eq. (27) recursively, for any $t \ge 2$ we obtain

$$|\theta_t - \theta'_t| \le (1-\lambda)^{t-1}|\theta_1 - \theta'_1| + 2\eta\sum_{k=0}^{t-2}(1-\lambda)^k. \tag{28}$$

The first term explicitly depends on the initial threshold discrepancy, whereas the second term accumulates the uniform bounds on the evidence discrepancies over time.

Since $0 < \lambda < 1$, the geometric sum satisfies $\sum_{k=0}^{t-2}(1-\lambda)^k = [1-(1-\lambda)^{t-1}]/\lambda$. Substituting this expression into Eq. (28) yields

$$|\theta_t - \theta'_t| \le (1-\lambda)^{t-1}|\theta_1 - \theta'_1| + \frac{2\eta}{\lambda}\left[1-(1-\lambda)^{t-1}\right]. \tag{29}$$

The same bound holds at $t = 1$ with equality. Moreover, since $0 < \lambda < 1$, $(1-\lambda)^{t-1} \to 0$ as $t \to \infty$. Therefore,

$$\limsup_{t\to\infty} |\theta_t - \theta'_t| \le \frac{2\eta}{\lambda}. \tag{30}$$

Thus, the term involving the initial threshold discrepancy in the bound decays geometrically, while the overall asymptotic threshold discrepancy is bounded above by $2\eta/\lambda$. □

A.2.3 Proof of the Dynamic Regret Bound

We first establish a one-step contraction toward the current minimizer. At time t, the threshold update uses the gradient of the current surrogate objective L_t, whose unique minimizer is θ_t^*. Therefore, before accounting for the movement of the minimizer over time, we compare both θ_t and the updated threshold θ_{t+1} with the same current minimizer θ_t^*.

Recall that

$$\theta_{t+1} = \theta_t - \eta\nabla L_t(\theta_t). \tag{31}$$

Since θ_t^* minimizes L_t, we have $\nabla L_t(\theta_t^*) = 0$. Hence,

$$\begin{aligned}|\theta_{t+1} - \theta_t^*| &= |\theta_t - \eta\nabla L_t(\theta_t) - \theta_t^*|\\ &= |\theta_t - \eta\nabla L_t(\theta_t) - (\theta_t^* - \eta\nabla L_t(\theta_t^*))|\\ &= |(\theta_t - \theta_t^*) - \eta\left[\nabla L_t(\theta_t) - \nabla L_t(\theta_t^*)\right]|.\end{aligned} \tag{32}$$

Equation (32) shows that the distance from θ_{t+1} to θ_t^* depends on the difference between the gradients at θ_t and θ_t^*. We next express this gradient difference in terms of the threshold difference.

If $\theta_t = \theta_t^*$, then $\theta_{t+1} = \theta_t^*$ by Eq. (31), and the contraction bound below holds trivially. Otherwise, by the mean value theorem applied to $\nabla L_t(\theta)$, there exists some ξ_t between θ_t and θ_t^* such that

$$\nabla L_t(\theta_t) - \nabla L_t(\theta_t^*) = \nabla^2 L_t(\xi_t)(\theta_t - \theta_t^*). \tag{33}$$

Substituting Eq. (33) into Eq. (32) gives

$$|\theta_{t+1} - \theta_t^*| = \left|1 - \eta \nabla^2 L_t(\xi_t)\right| |\theta_t - \theta_t^*|. \tag{34}$$

From the curvature bound established above, $\lambda/\eta \le \nabla^2 L_t(\xi_t) \le \lambda/\eta + 1/(4\tau)$. Multiplying by $\eta > 0$ gives $\lambda \le \eta \nabla^2 L_t(\xi_t) \le \lambda + \eta/(4\tau)$. Under the condition $\lambda + \eta/(4\tau) < 1$, we therefore have $0 < 1 - \eta \nabla^2 L_t(\xi_t) \le 1 - \lambda$. Combining this result with Eq. (34) yields

$$|\theta_{t+1} - \theta_t^*| \le (1 - \lambda)|\theta_t - \theta_t^*|. \tag{35}$$

Equation (35) has a direct interpretation: after the gradient update at time t, the distance from the updated threshold θ_{t+1} to the current minimizer θ_t^* is at most $1 - \lambda$ times the distance from the previous threshold θ_t to the same minimizer θ_t^*. Thus, before accounting for a change in the minimizer, the update reduces the current tracking error to at most a fraction $1 - \lambda$ of its previous value.

The minimizer, however, may change from θ_t^* to θ_{t+1}^* as the stream evolves. Let $\Delta_t^* = |\theta_{t+1}^* - \theta_t^*|$. Using the triangle inequality together with Eq. (35),

$$|\theta_{t+1} - \theta_{t+1}^*| \le |\theta_{t+1} - \theta_t^*| + |\theta_t^* - \theta_{t+1}^*| \le (1 - \lambda)|\theta_t - \theta_t^*| + \Delta_t^*. \tag{36}$$

Thus, the tracking error at time $t + 1$ is bounded by the contracted tracking error at time t together with the movement of the minimizer from time t to time $t + 1$.

Applying Eq. (36) recursively gives

$$|\theta_t - \theta_t^*| \le (1 - \lambda)^{t-1}|\theta_1 - \theta_1^*| + \sum_{k=1}^{t-1} (1 - \lambda)^{t-1-k} \Delta_k^*, \tag{37}$$

where the sum is interpreted as zero when $t = 1$.

Equation (37) bounds the tracking error by two terms. The first term depends on the initial tracking error and decays geometrically. The second term accumulates the movements of the minimizer from times 1 through $t - 1$, with earlier movements receiving smaller weights because their influence is repeatedly contracted.

To obtain a regret bound, we next control the square of the tracking error. Squaring Eq. (37) and using $(a + b)^2 \le 2a^2 + 2b^2$ gives

$$|\theta_t - \theta_t^*|^2 \le 2(1 - \lambda)^{2(t-1)}|\theta_1 - \theta_1^*|^2 + 2\left[\sum_{k=1}^{t-1} (1 - \lambda)^{t-1-k} \Delta_k^*\right]^2. \tag{38}$$

We now bound the square of the weighted sum of the minimizer movements. By the Cauchy–Schwarz inequality,

$$\left[\sum_{k=1}^{t-1} (1 - \lambda)^{t-1-k} \Delta_k^*\right]^2 \le \left[\sum_{k=1}^{t-1} (1 - \lambda)^{t-1-k}\right] \left[\sum_{k=1}^{t-1} (1 - \lambda)^{t-1-k} (\Delta_k^*)^2\right]. \tag{39}$$

Since $0 < \lambda < 1$, we have $\sum_{k=1}^{t-1} (1 - \lambda)^{t-1-k} \le 1/\lambda$. Therefore, Eq. (39) implies

$$\left[\sum_{k=1}^{t-1} (1 - \lambda)^{t-1-k} \Delta_k^*\right]^2 \le \frac{1}{\lambda} \sum_{k=1}^{t-1} (1 - \lambda)^{t-1-k} (\Delta_k^*)^2. \tag{40}$$

Substituting Eq. (40) into Eq. (38) gives

$$|\theta_t - \theta_t^*|^2 \le 2(1 - \lambda)^{2(t-1)}|\theta_1 - \theta_1^*|^2 + \frac{2}{\lambda} \sum_{k=1}^{t-1} (1 - \lambda)^{t-1-k} (\Delta_k^*)^2. \tag{41}$$

Summing Eq. (41) over $t = 1, \ldots, T$ gives

$$\sum_{t=1}^{T} |\theta_t - \theta_t^*|^2 \leq 2|\theta_1 - \theta_1^*|^2 \sum_{t=1}^{T} (1-\lambda)^{2(t-1)} + \frac{2}{\lambda} \sum_{t=2}^{T} \sum_{k=1}^{t-1} (1-\lambda)^{t-1-k} (\Delta_k^*)^2. \tag{42}$$

We bound the two terms separately. Since $0 < \lambda < 1$, the first geometric sum satisfies

$$\sum_{t=1}^{T} (1-\lambda)^{2(t-1)} \leq \frac{1}{1-(1-\lambda)^2} = \frac{1}{\lambda(2-\lambda)}. \tag{43}$$

For the second term, reversing the order of summation gives

$$\sum_{t=2}^{T} \sum_{k=1}^{t-1} (1-\lambda)^{t-1-k} (\Delta_k^*)^2 = \sum_{k=1}^{T-1} (\Delta_k^*)^2 \sum_{t=k+1}^{T} (1-\lambda)^{t-1-k} \leq \frac{1}{\lambda} \sum_{k=1}^{T-1} (\Delta_k^*)^2. \tag{44}$$

Let $V_T = \sum_{k=1}^{T-1} (\Delta_k^*)^2$. Substituting Eqs. (43) and (44) into Eq. (42) yields

$$\sum_{t=1}^{T} |\theta_t - \theta_t^*|^2 \leq \frac{2|\theta_1 - \theta_1^*|^2}{\lambda(2-\lambda)} + \frac{2V_T}{\lambda^2}. \tag{45}$$

Thus, the sum of the squared tracking errors over the stream is controlled by the squared initial tracking error and by V_T, which is the sum of the squared movements of the time-varying minimizer.

We now convert this tracking bound into a dynamic regret bound. Define

$$\mathrm{Regret}_T^{\mathrm{dyn}} = \sum_{t=1}^{T} \left[L_t(\theta_t) - L_t(\theta_t^*) \right]. \tag{46}$$

Since L_t is $(\lambda/\eta + 1/(4\tau))$-smooth and $\nabla L_t(\theta_t^*) = 0$, the smoothness inequality gives

$$L_t(\theta_t) - L_t(\theta_t^*) \leq \frac{1}{2} \left(\frac{\lambda}{\eta} + \frac{1}{4\tau} \right) |\theta_t - \theta_t^*|^2. \tag{47}$$

Summing Eq. (47) over $t = 1, \ldots, T$ and applying Eq. (45) yields

$$\mathrm{Regret}_T^{\mathrm{dyn}} \leq \left(\frac{\lambda}{\eta} + \frac{1}{4\tau} \right) \left[\frac{|\theta_1 - \theta_1^*|^2}{\lambda(2-\lambda)} + \frac{V_T}{\lambda^2} \right]. \tag{48}$$

Hence, for fixed parameters and initial tracking error, $\mathrm{Regret}_T^{\mathrm{dyn}} = O(1 + V_T)$. If $V_T = o(T)$, dividing Eq. (48) by T gives

$$\frac{\mathrm{Regret}_T^{\mathrm{dyn}}}{T} \to 0 \qquad \text{as } T \to \infty. \tag{49}$$

Thus, if V_T, the sum of the squared movements of the time-varying surrogate minimizer, grows sublinearly with the stream length, the average dynamic regret with respect to the surrogate objectives vanishes. □

A.3 Implementation Details

Our empirical evaluation consists of five main analyses: comparisons against existing methods, evaluation on the CIFAR-10-CD benchmark, an ablation study, an efficiency analysis, and a sensitivity analysis. All experiments were conducted on a server running Ubuntu 24.04.5 LTS, equipped with an AMD Ryzen 9 9950X CPU, an NVIDIA GeForce RTX 5080 GPU with 16 GB of VRAM, and 64 GB of system RAM. The sensitivity analysis was conducted with 5 random seeds, while all other experiments were repeated with 20 random seeds. We report the mean accuracy with standard deviation.

A.3.1 CLASSIFIERS AND ARCHITECTURES

We use three classifiers in the classic experiments: Gaussian Naive Bayes (GNB), a Very Fast Decision Tree (VFDT), and a Deep Neural Network (DNN). GNB is implemented using `GaussianNB` from scikit-learn (Pedregosa et al., 2011), while VFDT is implemented using `HoeffdingTreeClassifier` from scikit-multiflow. For the Airline dataset, VFDT uses majority-class leaf prediction (`leaf_prediction='mc'`), while the default configuration is used for the remaining datasets.

The DNN is implemented in PyTorch (Paszke et al., 2019) as a multilayer perceptron. We use two architectures according to dataset complexity:

- **Default Architecture:** For SEA, Sine, Mixed, Elec2, and PowerSupply, the network consists of two hidden layers, each containing 64 neurons followed by ReLU activation.
- **Airline Architecture:** For Airline, the network consists of three hidden layers containing 512, 256, and 64 neurons, respectively, with ReLU activation. The input layer receives the 679-dimensional one-hot encoded representation.

All tabular datasets in the classic experiments are binary-classification tasks. Therefore, the final fully connected layer contains two output units.

A.3.2 EVALUATION PROTOCOLS AND ADAPTATION

Stream Processing and Evaluation. All data streams are processed sequentially in chunks. The first chunk is used to initialize the classifier and is excluded from evaluation. For each subsequent chunk, the classifier first makes predictions on the incoming instances, after which drift detection and model updating are performed according to the corresponding adaptation scheme. We use online accuracy, a standard evaluation metric for online learning, computed as the average prediction accuracy over the evaluated chunks under the test-then-train protocol.

Drift Detection and Performance Feedback. For KSWIN (Raab et al., 2020), DDM (Gama et al., 2004), PH (Page, 1954), and HDDM-A/HDDM-W (Frías-Blanco et al., 2015), the per-instance binary prediction errors within each chunk are provided sequentially to the detector. PUDD (Lu et al., 2025) instead operates on the classifier's prediction uncertainty for the individual instances. Chunk-level accuracy is used both for evaluation and as the predictive performance feedback required by STAC. All drift detectors use the configurations reported in their original implementations.

Training Schemes. We evaluate all methods under two training schemes:

- **Continual Training:** Each chunk is first used for testing and is then used to update the classifier. Thus, the model continuously incorporates new stream information even when no adaptation is triggered.
- **Sporadic Training:** The classifier is trained on the initial chunk and is subsequently updated only when an adaptation event is triggered. If no adaptation is triggered, the classifier remains unchanged.

Both schemes follow the same test-then-train ordering and differ only in whether the classifier is updated when no adaptation event is triggered.

Adaptation Strategy. The adaptation procedure depends on the classifier. When an adaptation event is triggered, GNB and VFDT are reinitialized and retrained from scratch using the current chunk. For the DNN, the output layer is reinitialized while the hidden layers are preserved. A new Adam optimizer is then constructed, and the model is trained on the current chunk for 100 epochs with a learning rate of 1×10^{-2}. The same training budget is used for ordinary continual updates for all compared methods with the DNN classifier.

CIFAR-10-CD Evaluation. For the CIFAR-10-CD image stream, we use an ImageNet-pretrained ResNet-18 classifier and evaluate only the continual test-then-train setting. All batch-normalization layers are replaced with identity mappings, and the final classification layer retains 10 output units. The prediction task is binary: the time-varying interest label $y_t \in \{0, 1\}$ is used directly as the target index in the cross-entropy loss. Thus, target index 1 corresponds to the positive interest label and

target index 0 corresponds to the negative interest label, whereas output indices 2–9 are never used as training targets.

The first CIFAR-10-CD chunk is used to initialize the model for 10 epochs and is excluded from evaluation. For each subsequent chunk, prediction and drift detection are performed before model updating. When no adaptation is triggered, the pretrained model is fine-tuned on the current chunk for 10 epochs. When adaptation is triggered, this ordinary 10-epoch fine-tuning update is replaced by a shorter 5-epoch fine-tuning update on the same chunk. In both cases, training continues from the current model parameters and optimizer state; neither is reinitialized during the stream. All updates use the complete chunk of 500 images with stochastic gradient descent, a learning rate of 1×10^{-2}, and momentum 0.9.

Because an adaptation event replaces the ordinary 10-epoch update with a 5-epoch update, the adaptation decision also changes the amount of training performed on that chunk. This difference in training budget should therefore be considered when interpreting both predictive performance and computational efficiency on CIFAR-10-CD.

A.3.3 DATASETS

Our evaluation includes real-world tabular, synthetic, and image-based data streams, covering different data characteristics and concept-drift settings.

Real-World Tabular Data Streams. The real-world tabular datasets include Elec2, Airline, and PowerSupply.

- **Elec2** (Harries, 1999) is derived from the Australian electricity market. It contains 45,000 instances in their original temporal order, divided into 45 chunks of 1,000 instances. Each instance contains eight input features describing electricity-market conditions, and the binary target represents the direction of electricity-price change.
- **Airline** (Ikonomovska, 2011) contains 58,000 flight records divided into 58 chunks of 1,000 instances. The task is to predict whether a flight is delayed. Following the adopted preprocessing procedure (Tahmasbi et al., 2021), categorical variables are one-hot encoded, resulting in 679 input features.
- **PowerSupply** (Dau et al., 2019) contains 29,000 power-consumption records divided into 29 chunks of 1,000 instances. Each instance contains three input features and a binary target. The stream exhibits temporal variation associated with recurring power-consumption patterns.

Synthetic Data Streams. We additionally evaluate the methods on five synthetic data streams: SEA0, SEA10, SEA20, Sine, and Mixed. Each stream contains 100,000 instances divided into 100 chunks of 1,000 instances, with an abrupt concept drift introduced every 10 chunks.

- **SEA0, SEA10, and SEA20** (Bifet et al., 2010) generate concept drift by changing the threshold of the underlying classification function. The three variants differ in their label-noise rates: SEA0 contains no label noise, while SEA10 and SEA20 contain 10% and 20% label noise, respectively.
- **Sine** (Gama et al., 2004) contains two numerical attributes and generates concept drift by changing the sine-based function used to assign class labels.
- **Mixed** (Gama et al., 2004) contains both Boolean and numerical attributes and generates concept drift by alternating the underlying classification function.

For all synthetic data streams, the first chunk is used for classifier initialization and is excluded from evaluation. Predictive accuracy is therefore evaluated over the remaining 99 chunks.

Image-Based Data Stream. CIFAR-10-CD (Lu et al., 2025) is a concept-drift stream constructed from the 50,000 training images of CIFAR-10. The stream contains 100 chunks, each consisting of 500 images. The input images remain unchanged, while the prediction target is converted into a binary interest label indicating whether an image belongs to a time-varying set of classes of interest.

The set of classes of interest evolves according to a Markov process, causing the mapping from the original CIFAR-10 classes to the binary target to change over time. This construction introduces

concept drift through changes in the prediction target while preserving the underlying image stream. The first chunk is used for model initialization and is excluded from evaluation, and predictive accuracy is computed over chunks 2–100.

In our implementation, the final classification layer retains 10 output units, while the precomputed binary interest label $y_t \in \{0, 1\}$ is used directly as the target index in the cross-entropy loss. Thus, target index 1 corresponds to the positive interest label and target index 0 corresponds to the negative interest label, whereas output indices 2–9 are never used as training targets. At inference time, predictions are obtained by taking the argmax over all 10 outputs and comparing the resulting index directly with the binary target. Therefore, predictions with indices 2–9 are counted as incorrect.

A.3.4 STAC Hyperparameter Settings

The default numerical hyperparameters of STAC are fixed across datasets, classifiers, and drift detectors. The values used in the reported experiments are summarized in Table 4.

Table 4: Hyperparameter settings of STAC.

Parameter	Value	Description
τ	0.10	Temperature of the detector-evidence mapping
η	0.25	Update rate for threshold adjustment and response accumulation
λ	0.005	Decay rate for the threshold offset and performance-response state
ρ	0.01	Decay rate of the reference performance
β_f	0.60	Update rate of the fast performance estimate
β_s	0.01	Update rate of the slow performance estimate
δ	0.005	Shift of the predictor-evidence mapping
γ	0.01	Scale of the predictor-evidence mapping
b	0.01	Deterioration-confirmation level
c	0.70	Re-adaptation-confirmation level

The parameters have distinct roles in STAC. The temperature τ controls the mapping from the transformed detector statistic to detector evidence. The parameters η and λ determine the update and decay of the threshold offset and the accumulated post-adaptation performance response. The parameters ρ, β_f, β_s, δ, and γ control performance tracking and the construction of predictor evidence, while b and c determine the two adaptation-confirmation criteria described in Section 4.

For the classic data streams, adaptation confirmation is applied immediately at each chunk without an additional persistence requirement. For CIFAR-10-CD, we use $K = 3$ as the default persistence length, except in the experiments explicitly comparing different values of K. The deterioration condition must hold for K consecutive chunks, and adaptation based on this condition can be triggered only at the K-th consecutive chunk. If the condition remains true afterward, the same deterioration episode cannot trigger another adaptation until the condition becomes false. In contrast, the re-adaptation condition remains immediate when the deterioration condition is not active. The detector-evidence gate $q_t^+ > 0.5$ is evaluated at the current decision time and is not accumulated across the K chunks.

A.3.5 Normalization of Detector Evidence

Different drift detectors produce statistics with different scales and directions. STAC therefore transforms each detector statistic into a common transformed coordinate $\widetilde{S}_t$, where $\widetilde{S}_t = 0$ corresponds to the predefined detector threshold and larger values consistently indicate stronger evidence of drift. The learned threshold offset θ_t is defined in this transformed coordinate, with $\theta_1 = 0$ corresponding to the predefined detector threshold θ^{base}. The expressions below describe the transformations for finite statistics within their valid logarithmic domains.

Monotonic Statistics. For detectors whose statistics increase with drift evidence, including DDM and PH, we use the log-ratio transformation

$$\widetilde{S}_t = \log\left(\frac{S_t}{\theta^{\text{base}}}\right), \tag{50}$$

where S_t is the observed detector statistic and θ^{base} is the predefined detector threshold.

For detectors whose statistics decrease as drift evidence becomes stronger, including HDDM-A and HDDM-W, the direction is reversed:

$$\widetilde{S}_t = \log\left(\frac{\theta^{\text{base}}}{S_t}\right). \tag{51}$$

In both cases, the predefined detector threshold is mapped to zero, while statistics indicating stronger drift evidence are mapped to positive values.

Probability-Based Statistics. KSWIN and PUDD use probability-related statistics for which smaller values indicate stronger evidence of drift. For KSWIN, we use

$$\widetilde{S}_t = \frac{1}{2}\log\left(\frac{-\log(p_t)}{-\log(p_0)}\right), \tag{52}$$

where p_t is the observed p-value and p_0 is the predefined detection threshold. This transformation maps $p_t = p_0$ to $\widetilde{S}_t = 0$, while smaller p-values produce larger transformed values and therefore indicate stronger drift evidence.

For PUDD, different configurations use detection thresholds at different orders of magnitude. In our implementation, PUDD-1, PUDD-3, and PUDD-5 correspond to logarithmic thresholds of -1, -3, and -5, respectively. Directly comparing probability differences across these configurations would therefore produce values on different scales. We instead express both the observed statistic and its predefined threshold in logarithmic space and normalize their difference by the magnitude of the logarithmic threshold:

$$\widetilde{S}_t = \frac{\ell_0 - \ell_t}{\max(1, |\ell_0|)}, \tag{53}$$

where $\ell_t = \log_{10}(p_t)$ and ℓ_0 is the corresponding logarithmic detection threshold. The numerator measures how far the observed statistic lies beyond the predefined threshold in the direction of stronger drift evidence, while the denominator accounts for the different threshold scales across PUDD configurations. As a result, the predefined threshold is mapped to zero for every configuration, and positive values consistently indicate stronger drift evidence.

After transformation, all detector statistics use the same detector-evidence mapping:

$$q_t = \sigma\left(\frac{\widetilde{S}_t - \theta_t}{\tau}\right), \tag{54}$$

where θ_t is the learned threshold offset in the transformed coordinate and τ controls the temperature of the mapping.

A.4 FULL RESULTS

A.4.1 MAIN RESULTS

Table 5: Comparison of Fixed, DTD, and STAC with GNB in average accuracy (%) across eight datasets, eight detector configurations, and two training settings. Best in each setting shown in **bold**.

Dataset	Training	KSWIN			DDM			PH			HDDM-A		
		Fixed	DTD	STAC	Fixed	DTD	STAC	Fixed	DTD	STAC	Fixed	DTD	STAC
Airline	continual	49.80±1.98	**61.83±0.99**	59.77±0.00	52.18±0.00	**52.72±0.00**	48.85±0.00	53.87±0.00	52.68±0.00	**60.56±0.00**	52.85±0.00	52.90±0.00	**61.94±0.00**
	sporadic	49.02±1.71	**54.79±1.43**	53.67±0.33	52.47±0.00	51.77±0.00	**53.52±0.00**	50.00±0.00	50.49±0.00	**53.17±0.00**	**55.71±0.00**	52.20±0.00	53.21±0.00
Elec2	continual	68.19±0.04	69.60±0.82	**70.93±0.00**	69.17±0.00	**71.61±0.00**	70.93±0.00	70.46±0.00	**72.36±0.00**	70.93±0.00	68.13±0.00	70.34±0.00	**70.93±0.00**
	sporadic	67.97±0.12	**69.15±0.38**	66.95±1.97	**68.47±0.00**	66.46±0.00	67.49±0.00	**71.26±0.00**	69.59±0.00	67.44±0.00	68.13±0.00	**68.19±0.00**	67.40±0.00
PS	continual	**71.94±0.13**	71.78±0.07	71.80±0.04	71.75±0.00	71.81±0.00	**71.98±0.00**	71.69±0.00	71.67±0.00	**72.04±0.00**	71.61±0.00	**71.95±0.00**	71.78±0.00
	sporadic	69.26±1.83	70.22±0.80	**70.54±0.33**	70.89±0.00	70.79±0.00	**71.29±0.00**	69.30±0.00	70.30±0.00	**71.25±0.00**	72.00±0.00	**72.09±0.00**	72.03±0.00
SEA0	continual	91.66±1.08	91.66±1.08	**94.83±0.26**	94.68±0.40	94.73±0.53	**94.91±0.25**	94.87±0.27	**94.92±0.24**	94.91±0.26	94.72±0.32	94.77±0.22	**94.80±0.24**
	sporadic	89.61±2.99	89.54±2.92	**94.59±0.30**	94.25±0.43	94.33±0.50	**94.60±0.31**	94.53±0.31	**94.70±0.24**	94.61±0.34	94.36±0.34	**94.52±0.31**	94.41±0.39
SEA10	continual	84.78±0.52	85.25±1.11	**87.55±0.18**	86.87±0.55	87.12±0.45	**87.57±0.17**	87.61±0.14	87.61±0.16	**87.66±0.15**	87.26±0.32	87.28±0.30	**87.44±0.17**
	sporadic	83.59±1.05	85.36±1.19	**87.05±0.30**	86.07±0.81	86.38±0.71	**87.07±0.31**	86.97±0.32	**87.18±0.19**	87.06±0.41	86.53±0.40	**86.81±0.33**	86.76±0.34
SEA20	continual	76.59±0.53	76.92±0.56	**78.11±0.18**	77.05±0.57	77.28±0.78	**78.21±0.20**	78.25±0.19	**78.26±0.18**	78.24±0.17	77.85±0.33	77.81±0.40	**78.07±0.18**
	sporadic	75.39±0.71	76.44±0.99	**77.53±0.28**	76.30±0.67	76.61±0.73	**77.55±0.32**	77.31±0.49	**77.61±0.28**	77.58±0.34	76.72±0.56	77.22±0.40	**77.30±0.37**
Sine	continual	82.35±1.05	83.19±1.52	**84.44±1.20**	83.49±1.32	83.40±1.39	**84.47±1.18**	**84.40±1.14**	84.36±1.18	84.39±1.22	83.83±1.14	83.95±1.09	**84.17±1.13**
	sporadic	81.79±1.32	82.94±2.09	**84.49±1.32**	83.34±1.86	83.68±1.45	**84.51±1.30**	84.08±2.14	84.14±1.94	**84.43±1.38**	**84.10±1.16**	83.41±4.06	81.81±2.62
Mixed	continual	**84.23±0.11**	83.86±0.38	84.22±0.11	84.19±0.16	83.97±0.33	**84.22±0.11**	**84.23±0.11**	83.90±0.46	84.23±0.11	**84.23±0.11**	84.19±0.15	84.22±0.11
	sporadic	**83.92±0.20**	83.18±1.18	83.81±0.63	83.91±0.20	83.33±1.68	**83.94±0.21**	**83.92±0.20**	83.01±2.19	83.77±0.78	83.92±0.20	83.56±1.65	**83.93±0.20**
Dataset	Training	HDDM-W			PUDD-1			PUDD-3			PUDD-5		
		Fixed	DTD	STAC	Fixed	DTD	STAC	Fixed	DTD	STAC	Fixed	DTD	STAC
Airline	continual	48.64±0.00	58.26±0.00	**59.91±0.00**	53.64±0.09	52.84±0.66	**59.76±0.02**	53.03±0.48	53.38±0.34	**59.77±0.00**	53.00±0.41	51.80±0.47	**59.77±0.00**
	sporadic	48.60±0.00	**56.59±0.00**	54.61±0.00	50.11±0.02	50.69±0.01	**53.49±0.00**	49.77±1.60	52.78±0.07	**53.47±0.01**	53.09±0.08	53.24±0.54	**53.46±0.00**
Elec2	continual	68.13±0.00	69.90±0.00	**70.93±0.00**	**71.19±0.00**	70.83±0.00	70.93±0.00	**71.22±0.11**	70.87±0.12	70.93±0.00	**71.15±0.17**	70.97±0.18	70.93±0.00
	sporadic	**68.13±0.00**	65.84±0.00	67.40±0.00	62.92±0.00	63.73±1.64	**64.84±0.00**	59.29±0.00	63.73±1.64	**64.84±0.00**	59.41±0.37	63.73±1.64	**65.35±0.00**
PS	continual	71.81±0.00	**71.90±0.00**	71.84±0.00	**72.00±0.00**	71.94±0.02	71.99±0.00	**72.00±0.00**	71.94±0.02	71.99±0.00	71.80±0.00	71.76±0.00	**71.99±0.00**
	sporadic	**70.23±0.00**	70.14±0.00	70.16±0.00	71.13±0.22	70.65±0.00	**71.31±0.00**	71.20±0.17	70.65±0.00	**71.31±0.00**	70.40±0.17	70.65±0.00	**71.31±0.00**
SEA0	continual	92.53±0.92	92.72±1.14	**94.17±0.37**	94.78±0.25	**94.95±0.32**	94.89±0.21	94.89±0.29	**94.96±0.26**	94.91±0.24	94.91±0.24	**94.96±0.26**	94.91±0.24
	sporadic	91.81±1.05	92.32±1.22	**93.50±0.31**	94.14±1.02	**94.68±0.26**	94.54±0.28	94.60±0.26	**94.65±0.23**	94.60±0.29	94.56±0.30	**94.64±0.23**	94.61±0.30
SEA10	continual	86.06±0.48	86.77±0.65	**87.34±0.18**	87.60±0.20	**87.65±0.17**	87.63±0.19	87.62±0.16	**87.66±0.14**	87.65±0.15	87.54±0.17	87.61±0.17	**87.65±0.16**
	sporadic	85.70±0.45	86.77±0.38	**86.81±0.26**	86.80±0.55	**87.21±0.27**	87.06±0.30	86.98±0.33	**87.12±0.27**	87.11±0.25	86.78±0.37	86.95±0.31	**87.08±0.33**
SEA20	continual	78.07±0.19	77.92±0.28	**78.20±0.14**	78.26±0.19	**78.29±0.18**	78.22±0.19	78.23±0.21	**78.29±0.18**	78.27±0.20	78.09±0.28	78.21±0.21	**78.25±0.20**
	sporadic	77.28±0.29	77.42±0.24	**77.59±0.21**	77.29±0.54	**77.67±0.23**	77.53±0.32	77.22±0.70	77.44±0.44	**77.67±0.28**	76.81±0.64	77.19±0.57	**77.61±0.28**
Sine	continual	83.13±1.09	84.12±1.19	**84.26±1.28**	84.19±1.26	84.26±1.22	**84.40±1.28**	84.24±1.09	84.36±1.10	**84.45±1.19**	84.06±1.03	84.18±1.03	**84.47±1.18**
	sporadic	83.12±2.58	84.17±1.35	**84.33±1.45**	81.58±4.57	83.15±2.58	**84.44±1.38**	84.39±1.24	84.38±1.24	**84.45±1.35**	84.05±1.21	84.12±1.14	**84.50±1.29**
Mixed	continual	**84.23±0.11**	84.03±0.24	**84.23±0.11**	83.77±0.36	83.84±0.40	**84.18±0.17**	84.06±0.26	84.04±0.36	**84.22±0.11**	84.11±0.23	84.15±0.17	**84.22±0.11**
	sporadic	83.92±0.20	83.61±0.82	**83.92±0.20**	79.97±5.57	83.00±1.31	**83.60±0.97**	83.58±1.48	**83.89±0.26**	83.84±0.43	83.92±0.20	**83.93±0.19**	83.89±0.28

Table 6: Comparison of Fixed, DTD, and STAC with VFDT in average accuracy (%) across eight datasets, eight detector configurations, and two training settings. Best in each setting shown in **bold**.

Dataset	Training	KSWIN			DDM			PH			HDDM-A		
		Fixed	DTD	STAC	Fixed	DTD	STAC	Fixed	DTD	STAC	Fixed	DTD	STAC
Airline	continual	61.21±0.68	61.60±0.38	**61.89±0.13**	60.07±0.00	60.11±0.00	**61.44±0.00**	60.98±0.00	**61.53±0.00**	61.35±0.00	60.67±0.00	**60.87±0.00**	60.60±0.00
	sporadic	60.17±3.56	59.72±6.75	**62.01±0.08**	60.32±0.00	57.70±0.00	**62.25±0.00**	61.16±0.00	61.47±0.00	**62.05±0.00**	60.47±0.00	61.25±0.00	**61.37±0.00**
Elec2	continual	74.69±0.12	74.39±0.41	**75.70±0.00**	75.38±0.00	72.88±0.00	**75.70±0.00**	74.37±0.00	75.00±0.00	**75.54±0.00**	74.44±0.00	73.96±0.00	**75.72±0.00**
	sporadic	**74.64±0.07**	73.28±0.48	74.31±0.00	**75.31±0.00**	72.01±0.00	74.31±0.00	73.42±0.00	73.01±0.00	**74.57±0.00**	**74.38±0.00**	72.88±0.00	74.31±0.00
PS	continual	**72.09±0.29**	71.36±0.29	71.91±0.29	72.11±0.00	71.81±0.00	**72.19±0.00**	**72.02±0.00**	71.96±0.00	71.99±0.00	71.59±0.00	72.06±0.00	**72.22±0.00**
	sporadic	68.87±2.28	70.35±0.49	**70.43±0.13**	70.89±0.00	70.79±0.00	**71.29±0.00**	69.30±0.00	70.29±0.00	**71.25±0.00**	72.00±0.00	**72.09±0.00**	72.03±0.00
SEA0	continual	93.59±0.64	93.59±0.64	**95.52±0.31**	**95.57±0.30**	95.35±0.28	95.53±0.25	**95.80±0.25**	95.29±0.36	95.60±0.24	**95.78±0.24**	95.55±0.23	95.74±0.27
	sporadic	89.75±2.93	90.00±2.82	**94.55±0.29**	94.19±0.58	94.27±0.59	**94.62±0.30**	94.49±0.38	**94.67±0.27**	94.60±0.34	94.26±0.54	**94.54±0.27**	94.31±0.55
SEA10	continual	84.93±0.51	85.08±0.65	**86.59±0.19**	85.88±0.59	86.11±0.68	**86.57±0.16**	86.43±0.25	**86.80±0.23**	86.55±0.22	86.28±0.26	**86.48±0.27**	86.44±0.18
	sporadic	83.34±1.21	84.32±1.51	**87.05±0.38**	85.88±0.83	86.43±0.71	**87.06±0.32**	86.97±0.32	**87.17±0.21**	87.08±0.32	86.52±0.44	**86.85±0.37**	86.76±0.34
SEA20	continual	76.45±0.35	76.83±0.48	**77.24±0.23**	76.33±0.43	76.60±0.49	**77.27±0.18**	77.01±0.21	77.27±0.25	**77.28±0.24**	76.84±0.26	77.03±0.30	**77.08±0.19**
	sporadic	75.41±0.77	76.67±0.88	**77.57±0.30**	76.13±0.69	76.65±0.76	**77.50±0.34**	77.30±0.49	**77.62±0.28**	77.60±0.31	76.65±0.82	77.26±0.39	**77.30±0.37**
Sine	continual	86.81±0.80	87.44±1.00	**88.10±1.05**	**88.15±1.17**	87.82±1.28	88.10±1.04	**88.46±1.10**	88.09±1.06	88.11±1.06	**88.44±1.11**	88.03±1.19	88.16±1.11
	sporadic	83.10±1.34	85.60±1.36	**86.90±1.18**	86.03±1.02	85.89±1.78	**86.78±1.27**	85.86±2.31	86.46±1.33	**86.75±1.11**	**85.93±1.82**	85.08±2.67	84.94±4.65
Mixed	continual	**84.73±0.11**	83.94±0.36	84.70±0.13	84.60±0.19	82.50±3.79	**84.69±0.15**	**84.73±0.13**	83.81±0.45	84.71±0.14	**84.73±0.13**	84.49±0.51	84.72±0.13
	sporadic	**83.92±0.20**	83.10±1.65	83.81±0.63	83.92±0.20	83.29±1.67	**83.94±0.21**	**83.92±0.20**	83.01±2.19	83.77±0.78	83.92±0.20	83.56±1.65	**83.93±0.20**
Dataset	Training	HDDM-W			PUDD-1			PUDD-3			PUDD-5		
		Fixed	DTD	STAC	Fixed	DTD	STAC	Fixed	DTD	STAC	Fixed	DTD	STAC
Airline	continual	61.31±0.00	61.34±0.00	**62.03±0.00**	61.26±0.23	60.77±0.22	**61.69±0.00**	60.98±0.26	60.79±0.23	**61.60±0.19**	60.98±0.26	60.79±0.23	**61.60±0.19**
	sporadic	62.12±0.00	**62.20±0.00**	62.05±0.00	61.16±0.00	61.71±0.00	**62.25±0.00**	59.90±0.00	61.79±0.00	**61.99±0.00**	61.00±0.00	61.49±0.00	**61.99±0.00**
Elec2	continual	74.34±0.00	74.97±0.00	**75.70±0.00**	73.69±0.00	73.08±0.25	**75.21±0.06**	73.69±0.00	73.08±0.25	**75.21±0.06**	73.60±0.16	73.08±0.25	**75.23±0.08**
	sporadic	74.27±0.00	72.28±0.00	**74.31±0.00**	69.79±0.00	72.09±0.00	**73.75±0.00**	71.27±0.00	72.09±0.00	**73.75±0.00**	71.92±0.14	72.09±0.00	**73.94±0.02**
PS	continual	71.74±0.00	**72.27±0.00**	71.79±0.00	71.80±0.00	71.24±0.06	**72.08±0.00**	71.80±0.00	71.24±0.06	**72.08±0.00**	71.80±0.00	71.24±0.06	**72.08±0.00**
	sporadic	**70.24±0.00**	70.15±0.00	70.17±0.00	71.13±0.22	70.64±0.00	**71.31±0.00**	71.20±0.17	70.64±0.00	**71.31±0.00**	70.40±0.17	70.64±0.00	**71.16±0.35**
SEA0	continual	93.78±0.59	93.87±0.65	**95.18±0.34**	95.06±0.28	94.43±0.86	**95.31±0.33**	95.05±0.30	94.26±0.90	**95.35±0.38**	95.03±0.29	94.26±0.90	**95.36±0.36**
	sporadic	91.87±0.99	92.32±1.23	**93.50±0.31**	93.98±1.07	**94.64±0.24**	94.51±0.30	94.60±0.26	**94.64±0.23**	94.58±0.33	94.56±0.31	**94.64±0.23**	94.58±0.34
SEA10	continual	86.03±0.22	**86.47±0.29**	86.45±0.20	86.69±0.16	**86.72±0.36**	86.61±0.25	86.57±0.15	**86.77±0.30**	86.58±0.21	86.50±0.13	**86.77±0.30**	86.56±0.23
	sporadic	85.70±0.45	86.78±0.38	**86.80±0.26**	86.80±0.56	**87.18±0.23**	87.05±0.28	86.99±0.33	**87.14±0.24**	87.08±0.28	86.79±0.38	86.94±0.32	**87.09±0.34**
SEA20	continual	77.08±0.16	**77.39±0.21**	77.24±0.18	77.34±0.21	**77.37±0.23**	77.31±0.24	77.16±0.21	**77.35±0.22**	77.28±0.23	77.07±0.25	**77.35±0.22**	77.27±0.21
	sporadic	77.27±0.30	77.37±0.33	**77.58±0.23**	77.38±0.47	**77.64±0.22**	77.56±0.29	77.16±0.74	77.47±0.38	**77.64±0.28**	76.85±0.66	77.25±0.54	**77.65±0.27**
Sine	continual	**88.27±1.01**	88.11±1.06	88.21±1.12	87.88±1.19	87.39±5.31	**88.06±1.13**	87.99±1.22	86.02±6.97	**88.07±1.14**	**88.08±1.26**	86.31±6.85	88.08±1.14
	sporadic	86.13±1.09	**86.65±1.21**	86.55±1.22	84.87±2.58	86.09±1.66	**86.61±1.05**	86.36±1.18	86.46±1.14	**86.62±1.29**	86.36±1.18	86.48±1.13	**86.76±1.20**
Mixed	continual	**84.73±0.13**	84.15±0.31	**84.73±0.13**	83.64±0.37	56.86±4.71	**84.64±0.21**	83.75±0.39	55.00±3.59	**84.66±0.19**	83.74±0.39	55.00±3.59	**84.66±0.19**
	sporadic	83.92±0.20	83.61±0.82	**83.92±0.20**	80.75±3.69	83.35±1.13	**83.60±0.97**	83.58±1.48	**83.88±0.25**	83.85±0.43	83.92±0.20	**83.93±0.20**	83.89±0.28

Table 7: Comparison of Fixed, DTD, and STAC with DNN in average accuracy (%) across eight datasets, eight detector configurations, and two training settings. Best in each setting shown in **bold**.

Dataset	Training	KSWIN			DDM			PH			HDDM-A		
		Fixed	DTD	STAC	Fixed	DTD	STAC	Fixed	DTD	STAC	Fixed	DTD	STAC
Airline	continual	63.03±0.43	**63.07±0.29**	62.84±0.41	**63.08±0.39**	62.98±0.49	62.90±0.30	62.95±0.41	**63.10±0.29**	62.94±0.30	**62.98±0.38**	62.91±0.30	62.90±0.34
	sporadic	61.32±0.56	**62.45±0.81**	62.22±0.44	58.30±1.42	61.15±1.30	**62.11±0.65**	60.40±0.63	**62.35±0.53**	62.20±0.46	59.65±1.14	**61.83±1.01**	61.26±0.56
Elec2	continual	76.84±0.20	76.87±0.23	**76.94±0.25**	76.93±0.27	**77.04±0.23**	76.92±0.28	76.92±0.27	**76.99±0.23**	76.74±0.28	76.88±0.25	**76.96±0.21**	76.94±0.23
	sporadic	**76.80±0.30**	76.44±0.36	76.60±0.50	76.40±0.39	73.74±1.50	**76.55±0.55**	75.78±0.32	76.10±0.69	**76.60±0.34**	**76.87±0.24**	75.18±1.03	76.70±0.38
PS	continual	71.90±0.12	**71.95±0.14**	71.88±0.11	**71.94±0.17**	71.89±0.24	71.84±0.15	**71.96±0.17**	71.94±0.18	71.96±0.19	71.90±0.13	71.94±0.17	**71.96±0.15**
	sporadic	71.52±0.41	71.63±0.63	**71.72±0.26**	70.96±0.41	70.95±0.25	**71.68±0.23**	70.36±0.40	69.93±0.31	**71.56±0.26**	**72.06±0.26**	71.74±0.48	71.42±0.39
SEA0	continual	98.48±0.15	98.48±0.15	**98.54±0.14**	98.52±0.14	98.53±0.14	**98.54±0.14**	**98.55±0.13**	98.52±0.15	98.55±0.14	98.54±0.13	98.52±0.15	**98.55±0.13**
	sporadic	91.59±1.93	91.54±1.83	**97.89±0.17**	98.15±0.37	**98.22±0.31**	97.53±0.55	98.12±0.15	**98.33±0.17**	98.18±0.15	97.91±0.14	**98.23±0.17**	97.90±0.33
SEA10	continual	88.70±0.22	**88.70±0.22**	88.60±0.19	88.68±0.15	**88.72±0.19**	88.57±0.21	88.59±0.21	**88.64±0.21**	88.58±0.22	88.60±0.21	**88.72±0.20**	88.58±0.20
	sporadic	83.72±1.00	84.92±1.80	**87.38±0.34**	86.50±0.81	87.12±0.77	**87.42±0.26**	87.42±0.27	**87.78±0.27**	87.45±0.34	86.77±0.39	**87.61±0.43**	87.00±0.39
SEA20	continual	**78.32±0.33**	78.28±0.48	77.48±2.39	78.28±0.43	**78.33±0.42**	78.20±0.37	77.89±1.05	**78.14±1.02**	77.57±2.07	78.33±0.27	**78.42±0.30**	77.99±0.98
	sporadic	74.74±0.80	76.31±0.73	**76.69±0.47**	75.35±0.83	75.98±0.98	**76.72±0.49**	76.38±0.48	**76.89±0.39**	76.80±0.46	75.88±0.58	**76.71±0.62**	76.44±0.41
Sine	continual	93.39±0.79	93.32±0.81	**93.69±0.83**	93.65±0.86	92.34±2.85	**93.67±0.88**	**93.74±0.85**	93.44±1.07	**93.74±0.85**	**93.74±0.85**	93.53±0.83	**93.74±0.85**
	sporadic	86.37±1.37	91.60±2.13	**92.24±0.89**	86.15±6.82	86.36±7.04	**91.21±1.38**	91.65±0.94	**92.47±2.28**	92.15±0.96	**91.65±0.94**	91.00±5.23	91.56±1.21
Mixed	continual	**89.96±0.08**	89.96±0.07	**89.96±0.08**	**89.95±0.07**	89.92±0.09	89.93±0.08	**89.96±0.08**	89.93±0.09	**89.96±0.08**	**88.96±0.08**	89.95±0.09	**89.96±0.08**
	sporadic	88.65±0.46	**89.22±0.30**	88.65±0.46	87.59±4.16	87.15±3.30	**88.69±0.47**	**88.65±0.46**	87.09±4.60	**88.65±0.46**	**88.65±0.46**	88.37±1.78	**88.65±0.46**

Dataset	Training	HDDM-W			PUDD-1			PUDD-3			PUDD-5		
		Fixed	DTD	STAC	Fixed	DTD	STAC	Fixed	DTD	STAC	Fixed	DTD	STAC
Airline	continual	63.02±0.34	**63.21±0.45**	62.85±0.31	62.74±0.45	**63.15±0.29**	62.92±0.39	**62.99±0.28**	62.95±0.24	62.88±0.32	63.05±0.28	**63.07±0.25**	62.78±0.31
	sporadic	62.76±0.35	**62.96±0.36**	62.26±0.38	60.87±0.69	61.68±0.79	**62.16±0.30**	59.26±0.74	61.66±0.69	**62.30±0.39**	58.80±0.73	60.98±1.16	**62.10±0.50**
Elec2	continual	76.90±0.24	**76.98±0.27**	76.94±0.25	76.92±0.25	**77.11±0.21**	77.00±0.38	76.84±0.32	**77.04±0.23**	76.98±0.40	76.86±0.33	**77.04±0.23**	76.98±0.41
	sporadic	76.81±0.20	75.48±1.07	**76.84±0.31**	75.18±0.38	75.68±0.44	**76.47±0.52**	75.00±0.75	75.43±0.78	**76.26±0.54**	74.68±0.92	74.92±0.75	**76.33±0.65**
PS	continual	**71.94±0.19**	**71.94±0.15**	71.90±0.13	**72.13±0.15**	71.98±0.10	72.07±0.13	**72.04±0.18**	71.98±0.10	71.99±0.13	**72.00±0.17**	71.97±0.09	71.96±0.13
	sporadic	**71.69±0.24**	71.53±0.23	71.63±0.28	71.34±0.51	71.17±0.52	**71.38±0.56**	71.19±0.52	71.19±0.49	**71.38±0.56**	70.39±1.10	**70.83±0.67**	70.37±0.74
SEA0	continual	**98.52±0.13**	98.49±0.15	98.51±0.14	98.54±0.15	98.54±0.14	**98.55±0.15**	**98.55±0.15**	98.54±0.14	98.55±0.14	**98.55±0.14**	98.53±0.14	**98.55±0.14**
	sporadic	92.84±0.93	93.70±1.78	**95.79±0.38**	97.97±0.52	98.13±0.30	**98.22±0.17**	98.12±0.38	**98.21±0.17**	98.20±0.17	98.21±0.16	**98.21±0.17**	98.20±0.17
SEA10	continual	88.66±0.23	**88.72±0.18**	88.61±0.21	88.60±0.21	**88.71±0.23**	88.58±0.23	88.58±0.19	**88.70±0.21**	88.58±0.23	88.59±0.18	**88.72±0.17**	88.61±0.19
	sporadic	85.64±0.68	**87.28±0.96**	87.05±0.29	87.34±0.44	**87.52±0.44**	87.47±0.39	87.34±0.34	87.29±0.41	**87.42±0.31**	87.04±0.34	87.18±0.33	**87.43±0.33**
SEA20	continual	77.76±1.84	**78.38±0.42**	78.17±0.35	77.21±2.84	**78.10±1.07**	77.59±1.77	77.73±1.98	**78.09±1.07**	77.68±2.02	**78.09±0.55**	78.05±1.05	77.58±1.62
	sporadic	76.72±0.39	**76.97±0.44**	76.78±0.44	76.39±0.50	**76.87±0.45**	76.81±0.43	76.09±0.37	76.61±0.50	**76.84±0.46**	75.81±0.53	76.41±0.66	**76.66±0.42**
Sine	continual	93.56±0.87	83.62±11.47	**93.73±0.84**	93.63±0.89	90.68±7.11	**93.74±0.85**	93.65±0.86	89.84±7.52	**93.74±0.85**	93.68±0.86	90.00±7.13	**93.74±0.85**
	sporadic	89.46±2.76	77.00±12.14	**91.65±0.94**	90.39±2.50	91.75±1.51	**92.02±1.04**	91.67±0.96	91.63±0.91	**92.10±0.89**	91.65±0.94	91.64±0.91	**92.12±0.85**
Mixed	continual	89.95±0.07	89.62±0.75	**89.96±0.08**	89.93±0.08	89.95±0.08	**89.96±0.08**	89.94±0.08	89.94±0.08	**89.96±0.08**	89.94±0.07	89.95±0.08	**89.96±0.08**
	sporadic	**88.65±0.46**	87.21±4.03	**88.65±0.46**	85.57±3.98	87.89±1.42	**88.68±0.49**	**88.65±0.46**	88.63±0.55	**88.65±0.46**	**88.65±0.46**	88.63±0.55	**88.65±0.46**

A.4.2 CIFAR-10-CD RESULTS

Table 8: Comparison of Fixed, DTD, and STAC variants on CIFAR-10-CD across eight detector configurations. Accuracy and the number of adaptations are reported as mean ± SD over 20 random seeds. Best accuracy for each detector shown in **bold**.

Detector	Metric	Fixed	DTD	STAC	STAC ($K = 2$)	STAC ($K = 3$)	STAC ($K = 4$)
KSWIN	Acc.	75.79±0.89	76.80±0.93	76.55±0.88	77.84±0.90	**78.00±0.83**	77.74±1.10
	Adapt.	93.15±3.02	14.60±3.92	48.15±11.54	5.25±2.14	5.40±1.98	4.15±1.65
DDM	Acc.	76.24±0.84	77.94±0.73	77.04±0.89	77.78±0.93	**77.97±0.85**	77.81±1.05
	Adapt.	78.85±4.97	4.40±2.67	45.45±10.97	5.25±2.68	5.15±1.59	4.05±1.28
PH	Acc.	77.07±1.23	77.88±0.65	76.73±0.88	77.79±0.92	**77.97±0.78**	77.77±1.06
	Adapt.	32.25±5.18	9.90±3.28	44.30±10.81	5.45±2.62	5.40±1.93	4.20±1.44
HDDM-A	Acc.	76.28±0.81	77.81±0.98	76.93±0.85	77.82±0.90	**78.02±0.79**	77.73±1.08
	Adapt.	88.45±1.77	6.20±4.48	50.70±13.00	4.80±2.11	5.40±2.58	4.10±1.79
HDDM-W	Acc.	76.40±0.62	77.51±1.00	76.80±0.94	77.80±0.95	**77.96±0.77**	77.73±1.10
	Adapt.	88.25±1.64	5.05±4.90	56.30±13.59	5.45±2.36	5.55±1.83	4.30±1.68
PUDD-1	Acc.	77.52±0.80	77.76±1.08	76.83±0.93	77.54±1.12	**77.94±0.82**	77.72±1.10
	Adapt.	19.50±2.94	7.70±5.07	43.30±10.75	5.10±2.41	5.25±1.67	4.25±1.48
PUDD-3	Acc.	77.29±0.85	77.76±1.08	77.01±0.76	77.50±1.14	**77.94±0.82**	77.72±1.11
	Adapt.	17.65±2.52	7.70±5.07	43.45±10.79	4.90±1.95	5.25±1.67	4.20±1.50
PUDD-5	Acc.	77.61±0.92	77.76±1.08	77.13±0.72	77.56±1.12	**77.94±0.82**	77.72±1.11
	Adapt.	16.35±1.68	7.70±5.07	42.90±10.62	4.90±2.02	5.25±1.67	4.20±1.50

A.4.3 EFFICIENCY ANALYSIS

Table 9: Average algorithm runtime (seconds) of Fixed, DTD, and STAC with GNB across eight detector configurations and two training schemes. The last three columns report relative ratios.

Detector	Training	Fixed	DTD	STAC	STAC / Fixed	DTD / Fixed	DTD / STAC
KSWIN	continual	0.7955	1.1456	0.7890	0.992×	1.440×	1.452×
	sporadic	0.7733	1.2081	0.7646	0.989×	1.562×	1.580×
DDM	continual	0.1617	0.2444	0.1613	0.997×	1.512×	1.516×
	sporadic	0.1324	0.2126	0.1407	1.063×	1.606×	1.511×
PH	continual	0.1584	0.2647	0.1596	1.008×	1.671×	1.659×
	sporadic	0.1343	0.2668	0.1373	1.022×	1.986×	1.943×
HDDM-A	continual	0.3120	0.5850	0.3144	1.008×	1.875×	1.860×
	sporadic	0.2866	0.5595	0.2933	1.023×	1.952×	1.908×
HDDM-W	continual	0.3031	0.4804	0.2994	0.987×	1.585×	1.605×
	sporadic	0.2831	0.5331	0.2813	0.994×	1.883×	1.895×
PUDD-1	continual	1.0352	2.0400	1.5410	1.489×	1.971×	1.324×
	sporadic	1.3840	2.7047	1.6867	1.219×	1.954×	1.604×
PUDD-3	continual	1.6053	2.2514	1.5936	0.993×	1.403×	1.413×
	sporadic	1.8801	3.1063	1.7317	0.921×	1.652×	1.794×
PUDD-5	continual	1.7348	2.3783	1.5892	0.916×	1.371×	1.497×
	sporadic	2.1522	3.3594	1.7881	0.831×	1.561×	1.879×
Average	continual	0.7632	1.1737	0.8059	1.056×	1.538×	1.456×
	sporadic	0.8783	1.4938	0.8530	0.971×	1.701×	1.751×

Table 10: Average algorithm runtime (seconds) of Fixed, DTD, and STAC with VFDT across eight detector configurations and two training schemes. The last three columns report relative ratios.

Detector	Training	Fixed	DTD	STAC	STAC / Fixed	DTD / Fixed	DTD / STAC
KSWIN	continual	7.9767	14.4343	8.0136	1.005×	1.810×	1.801×
	sporadic	4.0816	7.7895	4.8840	1.197×	1.908×	1.595×
DDM	continual	7.3830	14.2426	7.3655	0.998×	1.929×	1.934×
	sporadic	3.0982	6.5332	3.7411	1.208×	2.109×	1.746×
PH	continual	7.3442	19.5455	7.3756	1.004×	2.661×	2.650×
	sporadic	2.9368	6.9408	3.2310	1.100×	2.363×	2.148×
HDDM-A	continual	7.5262	18.0962	7.5428	1.002×	2.404×	2.399×
	sporadic	3.3346	6.7703	4.0274	1.208×	2.030×	1.681×
HDDM-W	continual	7.4743	18.3502	7.4952	1.003×	2.455×	2.448×
	sporadic	4.8467	7.6490	3.3679	0.695×	1.578×	2.271×
PUDD-1	continual	8.4189	19.6199	8.5678	1.018×	2.330×	2.290×
	sporadic	4.4818	9.8238	5.5144	1.230×	2.192×	1.781×
PUDD-3	continual	8.4481	19.3727	8.5930	1.017×	2.293×	2.254×
	sporadic	5.0659	9.5079	6.2690	1.238×	1.877×	1.517×
PUDD-5	continual	8.4472	19.3797	8.5839	1.016×	2.294×	2.258×
	sporadic	5.1413	9.4333	6.3174	1.229×	1.835×	1.493×
Average	continual	7.8773	17.8801	7.9422	1.008×	2.270×	2.251×
	sporadic	4.1233	8.0560	4.6690	1.132×	1.954×	1.725×

Table 11: Average algorithm runtime (seconds) of Fixed, DTD, and STAC with DNN across eight detector configurations and two training schemes. The last three columns report relative ratios.

Detector	Training	Fixed	DTD	STAC	STAC / Fixed	DTD / Fixed	DTD / STAC
KSWIN	continual	3.5166	5.0225	3.3246	0.945×	1.428×	1.511×
	sporadic	1.2032	3.3190	1.3807	1.147×	2.758×	2.404×
DDM	continual	2.6616	4.1383	2.7613	1.037×	1.555×	1.499×
	sporadic	0.6965	2.1773	0.6951	0.998×	3.126×	3.133×
PH	continual	2.7160	5.1105	2.7085	0.997×	1.882×	1.887×
	sporadic	0.5705	2.5797	0.6851	1.201×	4.522×	3.765×
HDDM-A	continual	3.0412	5.2429	3.1750	1.044×	1.724×	1.651×
	sporadic	0.9550	3.0491	0.9099	0.953×	3.193×	3.351×
HDDM-W	continual	2.8929	5.0114	3.0274	1.046×	1.732×	1.655×
	sporadic	1.0310	2.8478	0.8539	0.828×	2.762×	3.335×
PUDD-1	continual	4.7568	11.3359	6.4161	1.349×	2.383×	1.767×
	sporadic	4.0236	7.7908	4.6368	1.152×	1.936×	1.680×
PUDD-3	continual	5.2939	10.8711	6.4554	1.219×	2.054×	1.684×
	sporadic	4.9401	8.2305	4.7176	0.955×	1.666×	1.745×
PUDD-5	continual	5.8241	11.0262	6.4893	1.114×	1.893×	1.699×
	sporadic	5.0376	8.2446	4.7924	0.951×	1.637×	1.720×
Average	continual	3.8379	7.2198	4.2947	1.119×	1.881×	1.681×
	sporadic	2.3072	4.7799	2.3339	1.012×	2.072×	2.048×

Table 12: Average algorithm runtime (seconds) of Fixed, DTD, and STAC with ResNet-18 on CIFAR-10-CD across eight detector configurations. The last three columns report relative ratios.

Detector	Fixed	DTD	STAC	STAC / Fixed	DTD / Fixed	DTD / STAC
KSWIN	3.2815	9.9645	5.6805	1.731×	3.037×	1.754×
DDM	3.4772	6.9422	5.4025	1.554×	1.996×	1.285×
PH	4.6431	9.2594	5.3776	1.158×	1.994×	1.722×
HDDM-A	3.4573	6.8770	5.4892	1.588×	1.989×	1.253×
HDDM-W	3.5575	6.1663	5.4899	1.543×	1.733×	1.123×
PUDD-1	5.3295	11.3530	6.2459	1.172×	2.130×	1.818×
PUDD-3	5.4198	10.8987	6.2506	1.153×	2.011×	1.744×
PUDD-5	5.5361	10.9066	6.3377	1.145×	1.970×	1.721×
Average	4.3377	9.0460	5.7842	1.333×	2.085×	1.564×

A.4.4 ABLATION STUDY

Table 13: Ablation results with **GNB** across eight datasets and two training settings, averaged over eight drift detector configurations. Results are reported as average online accuracy (%) in mean ± standard deviation. Best in each setting is shown in bold.

Dataset	Training	Fixed	+ TL	+ DC	+ DC + RC	+ TL + DC	STAC
SEA0	Continual	94.13±1.32	94.77±0.36	94.16±1.33	94.16±1.33	**94.79±0.35**	94.79±0.35
	Sporadic	93.48±2.08	94.41±0.48	93.56±1.96	93.56±1.93	94.42±0.47	**94.43±0.48**
SEA10	Continual	86.92±1.02	87.55±0.20	86.91±1.03	86.91±1.04	**87.56±0.20**	**87.56±0.20**
	Sporadic	86.18±1.22	86.98±0.35	86.12±1.46	86.20±1.24	**87.00±0.34**	**87.00±0.34**
SEA20	Continual	77.80±0.68	78.12±0.22	77.84±0.65	77.81±0.69	**78.21±0.19**	78.20±0.19
	Sporadic	76.79±0.86	77.48±0.32	76.83±0.88	76.85±0.83	**77.55±0.32**	77.55±0.32
Sine	Continual	83.71±1.31	84.33±1.21	83.73±1.33	83.72±1.34	84.37±1.21	**84.38±1.21**
	Sporadic	83.31±2.50	83.88±2.48	83.74±1.68	83.73±1.70	84.11±1.81	**84.12±1.80**
Mixed	Continual	84.13±0.25	84.04±0.33	**84.22±0.12**	**84.22±0.12**	84.22±0.13	84.22±0.12
	Sporadic	83.38±2.42	83.54±1.15	**83.84±0.74**	**83.84±0.74**	83.83±0.54	83.84±0.55
Elec2	Continual	69.70±1.36	69.43±0.54	**70.93±0.00**	**70.93±0.00**	**70.93±0.00**	**70.93±0.00**
	Sporadic	65.70±4.24	66.44±2.73	58.24±0.00	65.50±3.16	58.24±0.00	**66.47±1.34**
PowerSupply	Continual	71.83±0.14	71.61±0.17	71.83±0.14	71.81±0.13	71.92±0.11	**71.93±0.10**
	Sporadic	70.55±1.11	70.72±0.59	70.54±1.19	70.48±1.25	71.15±0.54	**71.15±0.54**
Airline	Continual	52.13±1.91	49.96±1.58	**61.59±0.59**	61.43±0.64	60.20±0.71	58.79±3.82
	Sporadic	51.10±2.43	50.57±1.04	52.95±1.18	52.89±1.01	**54.79±1.29**	53.58±0.44

TL: Threshold Learning; DC: Deterioration Confirmation; RC: Re-adaptation Confirmation.

Table 14: Ablation results with **VFDT** across eight datasets and two training settings, averaged over eight drift detector configurations. Results are reported as average online accuracy (%) in mean ± standard deviation. Best in each setting is shown in bold.

Dataset	Training	Fixed	+ TL	+ DC	+ DC + RC	+ TL + DC	STAC
SEA0	Continual	94.96±0.89	95.26±0.39	95.08±0.92	95.08±0.92	**95.45±0.35**	95.45±0.36
	Sporadic	93.46±2.03	94.38±0.50	93.51±2.01	93.53±1.89	94.40±0.50	**94.41±0.50**
SEA10	Continual	86.17±0.62	**86.62±0.20**	86.14±0.63	86.14±0.63	86.55±0.22	86.54±0.22
	Sporadic	86.12±1.31	86.99±0.34	86.13±1.37	86.17±1.24	86.99±0.33	**87.00±0.34**
SEA20	Continual	76.91±0.43	**77.33±0.21**	76.89±0.42	76.90±0.42	77.24±0.22	77.25±0.22
	Sporadic	76.77±0.91	77.48±0.33	76.80±0.89	76.83±0.85	**77.57±0.31**	77.55±0.32
Sine	Continual	88.01±1.22	87.83±1.12	88.08±1.20	88.08±1.21	88.11±1.09	**88.11±1.10**
	Sporadic	85.58±1.96	86.19±2.37	85.70±1.93	85.68±1.97	**86.50±2.07**	86.49±2.07
Mixed	Continual	84.33±0.55	84.10±0.62	**84.70±0.16**	**84.70±0.16**	84.69±0.17	84.69±0.16
	Sporadic	83.48±1.76	83.57±1.04	**83.84±0.74**	**83.84±0.74**	83.84±0.54	83.84±0.55
Elec2	Continual	74.28±0.57	74.47±0.71	74.38±0.90	74.92±0.88	74.91±0.25	**75.50±0.23**
	Sporadic	73.12±1.81	73.77±0.99	71.82±2.47	72.69±2.05	73.29±0.67	**74.16±0.28**
PowerSupply	Continual	71.87±0.20	71.56±0.37	**72.11±0.15**	71.91±0.20	72.09±0.10	72.04±0.17
	Sporadic	70.50±1.26	70.72±0.59	70.48±1.21	70.49±1.10	**71.14±0.54**	71.12±0.56
Airline	Continual	60.93±0.47	**61.67±0.27**	60.71±0.48	60.91±0.46	60.85±0.27	61.53±0.42
	Sporadic	60.79±1.43	61.90±0.23	61.61±0.82	61.09±1.16	**62.09±0.17**	61.99±0.26

TL: Threshold Learning; DC: Deterioration Confirmation; RC: Re-adaptation Confirmation.

Table 15: Ablation results with **DNN** across eight datasets and two training settings, averaged over eight drift detector configurations. Results are reported as average online accuracy (%) in mean ± standard deviation. Best in each setting is shown in bold.

Dataset	Training	Fixed	+ TL	+ DC	+ DC + RC	+ TL + DC	STAC
SEA0	Continual	98.53±0.14	**98.55±0.14**	98.53±0.14	98.53±0.14	98.55±0.14	98.54±0.14
	Sporadic	96.61±2.69	**97.81±0.80**	96.63±2.69	96.64±2.67	97.73±0.82	97.74±0.82
SEA10	Continual	**88.62±0.21**	88.56±0.23	88.61±0.21	88.61±0.21	88.59±0.21	88.59±0.21
	Sporadic	86.47±1.32	**87.40±0.36**	86.42±1.45	86.47±1.37	87.35±0.37	87.33±0.38
SEA20	Continual	77.95±1.51	77.91±1.41	77.81±1.78	**78.05±0.94**	77.79±1.82	77.78±1.64
	Sporadic	75.92±0.83	**76.77±0.42**	75.88±0.87	75.92±0.82	76.71±0.45	76.72±0.46
Sine	Continual	93.63±0.86	93.67±0.87	93.69±0.86	93.69±0.86	**93.73±0.86**	93.72±0.85
	Sporadic	89.87±3.63	91.73±1.77	91.06±1.89	90.99±2.12	**91.90±1.11**	91.88±1.09
Mixed	Continual	89.95±0.08	89.95±0.08	**89.96±0.08**	**89.96±0.08**	89.96±0.08	89.96±0.08
	Sporadic	88.13±2.32	88.64±0.71	**88.68±0.46**	**88.68±0.46**	88.67±0.46	88.66±0.47
Elec2	Continual	76.89±0.27	76.90±0.30	76.93±0.24	**76.97±0.26**	76.95±0.24	76.93±0.33
	Sporadic	75.94±0.98	76.54±0.36	75.58±1.02	75.91±1.10	76.22±0.56	**76.55±0.52**
PowerSupply	Continual	71.98±0.18	71.93±0.15	**72.00±0.17**	71.99±0.17	71.97±0.17	71.94±0.16
	Sporadic	71.19±0.78	**71.56±0.51**	71.02±0.69	71.02±0.73	71.42±0.58	71.39±0.60
Airline	Continual	62.98±0.39	62.86±0.39	**62.98±0.35**	62.97±0.35	62.95±0.35	62.88±0.34
	Sporadic	60.17±1.61	**62.30±0.69**	59.97±1.50	60.02±1.52	62.04±0.58	62.08±0.57

TL: Threshold Learning; DC: Deterioration Confirmation; RC: Re-adaptation Confirmation.

A.4.5 Sensitivity Analysis

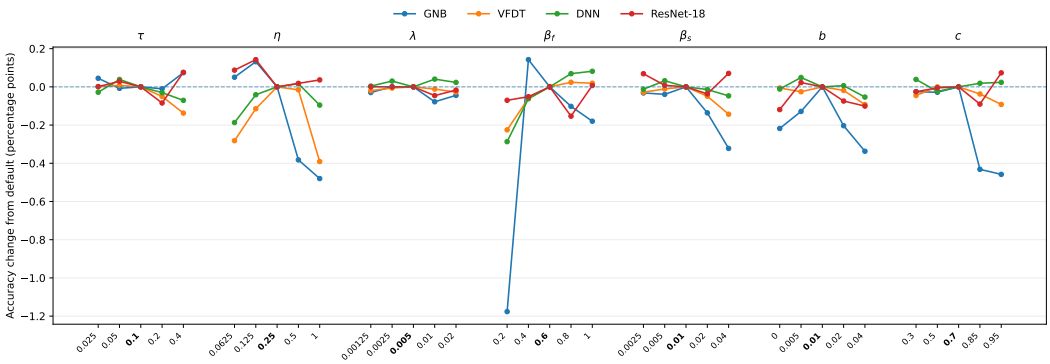


Figure 2: Sensitivity of predictive accuracy to STAC hyperparameters. Each curve shows the change in average accuracy relative to the default configuration when varying one hyperparameter at a time.