
Optimizing Retraining Schedules via Learning Curves

Jin Sima¹ Changlong Wu² Ananth Grama¹ Wojciech Szpankowski^{1,3}

¹Purdue University, West Lafayette, IN, USA

²University of Arizona, Tucson, AZ, USA

³Jagiellonian University, Kraków, Poland

simaj@purdue.edu clwu@arizona.edu {ayg,szpan}@purdue.edu

Abstract

Retraining is the primary mechanism by which deployed models adapt to new data, yet it is also among the most expensive operations in modern machine learning. How often must a model be retrained to remain near-optimal? We answer this question through a learning-curve based characterization of the retraining-frequency/risk trade-off in online learning. For the i.i.d. realizable setting, we observe that only $O(\log T)$ updates over a horizon of T rounds suffice to match the risk of *full retraining* within a factor of two whenever the learning curve is non-increasing. However, when the learning curve decays as a power law $t^{-\alpha}$ with $\alpha < 1$, as empirically observed in deep learning, the budget collapses further to $O(\log \log T)$ updates, yielding a *substantial asymptotic* improvement. We further design update schedules achieving these bounds, prove matching lower bounds, and present an adaptive algorithm that remains near-optimal when α is unknown. We then extend the analysis to piecewise-stationary and gradually drifting environments, and prove a no-free-lunch theorem: already with a single distribution change, triggered by the learner’s first update, every full-prefix update schedule that knows the environment but not the hypothesis class is polynomially suboptimal in some setting. Together, these results provide a sharp characterization of the frequency–accuracy trade-off in online retraining and bridge foundational learning theory with practical strategies for scalable deployment.

1 Introduction

Current AI systems have achieved remarkable success across a wide range of applications, enabled by training on large volumes of data to yield models with strong statistical concentration and effective representations. As these models are deployed over extended periods, they inevitably encounter a continual stream of new data. Incorporating such data can improve predictive accuracy, but doing so typically requires *retraining*, which updates the model’s internal parameters. Retraining, however, is among the most costly operations in modern AI—whether in terms of computational resources and/or stability. This raises a fundamental question: *as data accumulate, how often should retraining occur to maintain optimal predictive performance?*

We address this problem through the lens of *online learning* (Shalev-Shwartz & Ben-David, 2014), a setting in which data arrive sequentially over rounds $t = 1, 2, \dots, T$ and the learner must make predictions in each round while updating its hypothesis based on feedback. Formally, online learning can be abstracted as a game between a learner and nature. At each round t , the learner selects a predictor $\hat{h}_t$ on an input space $\mathcal{X}$ based on past data; upon receiving a new instance X_t , it outputs a prediction $\hat{Y}_t := \hat{h}_t(X_t)$. Nature then reveals the true label Y_t , and the learner incurs a loss $\ell(\hat{Y}_t, Y_t)$ for some predefined loss function ℓ . The performance of the learner over T rounds is measured by its

cumulative risk

$$\text{Risk}_T(\{(X_t, Y_t)\}_{t=1}^T) = \sum_{t=1}^T \ell(\hat{h}_t(X_t), Y_t).$$

While natural, this paradigm implicitly assumes that *retraining* (i.e., updating $\hat{h}_t$) at every step t is essential for maintaining optimal performance. A fundamental question that has remained largely unexplored is whether this assumption is truly necessary: can comparable accuracy be achieved with dramatically fewer updates? In this work, we show that the answer is affirmative, by providing a precise characterization of the trade-off between retraining frequency and predictive risk. Whereas common practice often relies on fixed-cadence or accuracy-triggered heuristics, we identify the exact update frequencies that suffice and prove that they are unimprovable in the corresponding regimes.

More precisely, we assume access to an *oracle* $\mathcal{O}$ that, given any dataset $\{(X_i, Y_i)\}_{i=1}^m$, returns a predictor $\hat{h} \in \mathcal{H}$ from a hypothesis class $\mathcal{H}$. Retraining corresponds to invoking the oracle. We allow at most S such retraining steps (i.e., oracle calls), which occur at chosen time points: $0 = \tau_0 < \tau_1 < \dots < \tau_S < T$. At each update time τ_s , the oracle produces a predictor $\hat{h}_s$ based on the data observed so far. This predictor is then deployed throughout the s -th epoch, i.e., $\forall t \in (\tau_s, \tau_{s+1}]$ we predict $\hat{Y}_t = \hat{h}_s(X_t)$.

Given a retraining budget S , our goal is to design the update schedule $\tau_1, \dots, \tau_S$ so as to minimize the cumulative risk $\text{Risk}_{S,T}$ in a specified learning environment (see Section 2 for definition). We also focus on the *minimal* budget S for which $\text{Risk}_{S,T}$ asymptotically *matches* the risk $\text{Risk}_{T,T}$ of the same oracle that would be attained if retraining were performed at *every* step (i.e., $S = T$).

Our primary focus in this paper is on the *frequency* of retraining, while we treat the (computational) cost of each retrain as a single oracle call. We emphasize that our framework is not constrained to any specific oracle; rather, our main theorems provide general characterizations of when to optimally invoke a *given* (black-box) oracle based on its predictive performance. Moreover, retraining *frequency* captures not only computational cost but also instability, catastrophic forgetting, and user adaptation overheads that arise from continuous model deployment. Since each retraining event can also require validation, redeployment, and monitoring, we count events rather than total FLOPs or examples processed.

Our Contributions. We start with the classical i.i.d. realizable setting, which already captures the fundamental trade-off between the retraining budget and the predictive risk. In this case, the data $\{(X_t, Y_t)\}_{t=1}^T$ are assumed to be drawn i.i.d. from an *unknown* distribution μ . Our analysis relies on the concept of a *learning curve* $\bar{L}(t)$ (see Eq. (3) for a precise definition), defined as: $\bar{L}(t) := \mathbb{E}_{\mathcal{D}_t} \mathbb{E}_{(X,Y) \sim \mu} [\ell(\hat{h}_{\mathcal{D}_t}(X), Y)]$, where $\mathcal{D}_t := \{(X_i, Y_i)\}_{i=1}^t \stackrel{\text{i.i.d.}}{\sim} \mu$, and $\hat{h}_{\mathcal{D}_t} := \mathcal{O}(\mathcal{D}_t)$ for some oracle $\mathcal{O}$ (e.g., an ERM rule), and (X, Y) denotes a fresh sample from μ .

As a baseline, Proposition 3.1 establishes that only $\log T$ updates suffice to achieve a risk within a factor of two of the optimal *full-update* risk, provided that the learning curve $\bar{L}(t)$ is *non-increasing*, regardless of its exact shape. The proof is reminiscent of the classical “doubling trick,” however, our result is more general in that it applies to *any* non-increasing learning curve, without assuming a specific functional form. We then establish the *precise* optimal risk when additional properties of the learning curve are satisfied. We show in Theorem 3.3 that if $\bar{L}(t) = \tilde{O}(1/t)$, e.g., when $\mathcal{H} \subset \{0, 1\}^{\mathcal{X}}$ has finite VC-dimension and the oracle $\mathcal{O}$ is ERM, near-optimal updates are scheduled at $\tau_i = T^{i/(S+1)}$ for $1 \leq i \leq S$, leading to the expected risk $\mathbb{E}[\text{Risk}_{S,T}] \leq \tilde{O}((S+1) \cdot T^{\frac{1}{S+1}})$. When the learning curve follows a *power law* $\bar{L}(t) = O(t^{-\alpha})$ for some $\alpha < 1$, Theorem 3.3 provides a weighted update schedule with risk $O(T^{\frac{1-\alpha}{1-\alpha S+1}})$, with a constant that does not depend on S or T . Perhaps surprisingly, this implies that when $\bar{L}(t) = \Theta(t^{-\alpha})$, only $S = O(\log \log T)$ updates suffice to match the *full-update* risk within a constant factor. We further show that these risks are essentially tight and design an update rule (Algorithm 1) that leverages the *empirical losses* incurred by the predictors to attain the optimal risk up to polylogarithmic factors (Theorem 3.6) even when α is unknown. Beyond the i.i.d. setting, our results also extend to *nonstationary* data distributions, including both *piecewise-i.i.d.* shifts and *gradual drift*. For the piecewise-i.i.d. regime, we propose an efficient update procedure (Algorithm 2) that achieves near-optimal risk, as shown in Theorem 3.8. For gradual drift of realizable distributions at total-variation rate at most $\bar{\gamma}$, the same algorithm attains risk $\tilde{O}(1 + T\bar{\gamma}^{1/2})$ with $\tilde{O}(1 + T\bar{\gamma}^{1/2})$ updates (Theorem 3.10). Finally, we show in Theorem 3.12

that, without any prior information about the learning environment, no universal update schedule can achieve competitive optimality (even up to polylogarithmic factors) relative to the case where the environment is known, already for a single change that reacts to the learner’s first update. Finally, Appendix B reports experiments on MNIST, Fashion-MNIST, and a multiscale synthetic model. With at most seven updates, the weighted schedule stays within a factor of 1.81 of full retraining on synthetic streams of up to 10^6 rounds, whereas fixed-cadence retraining with the same budget is up to 216 times worse. On MNIST, two updates keep it within a factor of 1.41 of full retraining up to $T = 50,000$. In summary, our results identify the learning curve as a key object governing the risk-update trade-off in retraining: sufficient information about its shape enables near-optimal scheduling (Theorems 3.3, 3.6, 3.8, and 3.10), whereas without such information no universal full-prefix rule can be near-optimal in general (Theorems 3.5 and 3.12).

Related Work. Online learning has been extensively studied in the literature, where updates are assumed at every time step (Rakhlin & Sridharan, 2015; Shalev-Shwartz & Ben-David, 2014; Foster et al., 2018; Wu et al., 2022, 2026). In contrast, literature that explicitly investigates the cost of retraining in online learning is scarce. Notable exceptions are Devulapalli & Hanneke (2024); Devulapalli et al. (2025), which consider a *continuous-time* online model that queries samples only at selected time steps and updates the model at those query times. However, their work does not treat the update schedule as a *design* objective and relies exclusively on the queried (sampled) *adversarial* examples, which can be overly pessimistic for practical scenarios. Recent efforts propose decision rules for triggering retraining based on uncertainty or cost considerations (Regol et al., 2025; Zanotti, 2025). These results provide useful heuristics and case studies, but they do not characterize the minimal update budget needed to match the predictive performance of continuous retraining. Other related areas include continual and incremental learning (He et al., 2020), as well as learning under nonstationarity and dynamic regret (Block et al., 2022; Haghtalab et al., 2020; Hazan & Seshadhri, 2009; Wu et al., 2023; Fahrback et al., 2023). Although these efforts study adaptation under distributional shift, their objectives differ from ours; we focus on the statistical limits of retraining under a fixed update budget. Finally, empirical studies of scaling laws suggest that learning curves often follow power-law behavior (Kaplan et al., 2020; Hestness et al., 2017; Hoffmann et al., 2022), which motivates the modeling assumptions adopted in this work. Closer in spirit, Altschuler & Talwar (2018) characterize the minimax regret of online learning under a switching budget, Hanneke (2014) develops the theory of label-efficient learning curves under disagreement-based active learning, and Ibrahim et al. (2024) empirically demonstrate that simple continual pre-training schedules can match full retraining of large language models at a fraction of the compute.

2 Problem Formulation

Let $\mathcal{X}$ be the context space, $\mathcal{Y}$ the label space, and $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ a hypothesis class. A *training oracle* (or learning rule) is a function $\mathcal{O} : (\mathcal{X} \times \mathcal{Y})^* \rightarrow \mathcal{H}$, that takes as input a sequence of context-label pairs and returns a predictor in $\mathcal{H}$. We consider the following online learning scenario with a limited update (or retraining) budget of S :

Protocol 1 Selective Update Online Learning Protocol

```

Select initial predictor  $\hat{h}_0 \in \mathcal{H}$  and set update index  $s = 0$ 
for  $t = 1, \dots, T$  do
  Use predictor  $\hat{h}_s$  to make a prediction on  $X_t$ 
  Observe true label  $Y_t$ 
  Use update rule  $\Phi$  with  $\{(X_i, Y_i)\}_{i=1}^t$  and predictor  $\hat{h}_s$  to determine whether to update
  if  $\Phi$  triggers an update and  $s < S$  then
    Select data subset  $\mathcal{D}_{s+1} \subset \{(X_i, Y_i)\}_{i=1}^t$  and update predictor  $\hat{h}_{s+1} = \mathcal{O}(\mathcal{D}_{s+1})$ 
    Set  $s \leftarrow s + 1$ 
  end if
end for
```

We denote $\tau_1, \dots, \tau_S$ as the update time steps, and define $s(t) := \max\{s \in \{0, \dots, S\} : \tau_s < t\}$ as the index of the most recent update strictly before prediction at time $t \in \{1, \dots, T\}$. Our goal is to design an update rule Φ that, given access to a retraining oracle $\mathcal{O}$ and a fixed update budget

S , minimizes the total cumulative risk incurred by the learner over a time horizon of T prediction rounds. Specifically, the empirical cumulative risk (or total loss) incurred by a given *update rule* Φ (which at time t can take value 1, if update happens at this time and 0 otherwise) is defined as:

$$\text{Risk}_{S,T}(\Phi, \{(X_t, Y_t)\}_{t=1}^T) := \sum_{t=1}^T \ell(\hat{h}_{s(t)}(X_t), Y_t) = \sum_{s=0}^S \sum_{t=\tau_s+1}^{\tau_{s+1}} \ell(\hat{h}_s(X_t), Y_t),$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_{\geq 0}$ is a loss function and we assume $\tau_0 = 0$ and $\tau_{S+1} = T$.

In many practical settings, retraining a model is significantly more expensive than evaluating it only on new data. Therefore, it is desirable to limit the number of retraining events, while allowing the learner to freely test the current predictor on incoming examples. This motivates a class of update rules that rely solely on observable predictive performance.

Risk-driven update. An update rule Φ is said to be *risk-driven* if, at any time step t , it only treats the current predictor $\hat{h}_{s(t)}$ as a black box and bases its update decision solely on the *observed losses* $\{\ell(\hat{h}_{s(t)}(X_i), Y_i)\}_{i=1}^t$ over the available data. That is, Φ must determine whether to update exclusively based on the predictive performance of the current predictor on the historical data.

This paper focuses primarily on designing *risk-driven* update rules that achieve optimal cumulative risk. To proceed, we assume that the data is generated by an unknown *random process* belonging to a known class $\mathcal{P}$. We refer to the tuple $(\mathcal{P}, \mathcal{H}, \mathcal{O})$ as a *learning environment*.

For any given learning environment, we first define the expected risk, for $\mu \in \mathcal{P}$, as follows:

$$\text{Risk}_{S,T}(\Phi, \mu) := \mathbb{E}_{\{(X_t, Y_t)\}_{t=1}^T \sim \mu} [\text{Risk}_{S,T}(\Phi, \{(X_t, Y_t)\}_{t=1}^T)], \quad (1)$$

Our goal is to characterize the following minimax risk:

$$\text{Risk}_{S,T}(\mathcal{P}) := \inf_{\Phi} \sup_{\mu \in \mathcal{P}} \mathbb{E} [\text{Risk}_{S,T}(\Phi, \{(X_t, Y_t)\}_{t=1}^T)], \quad (2)$$

where the infimum is taken over all *risk-driven* update rules Φ that satisfy the update budget constraint S , and the expectation is over the randomness of the data sequence $\{(X_t, Y_t)\}_{t=1}^T \sim \mu$ as well as any internal randomness of Φ and the retraining oracle $\mathcal{O}$.

Realizable i.i.d. Learning Environment. In this setting, the data points $\{(X_t, Y_t)\}_{t=1}^T$ are drawn independently and identically from an unknown distribution μ over $\mathcal{X} \times \mathcal{Y}$. We assume there exists an unknown hypothesis $h \in \mathcal{H}$ such that $h(X) = Y$ for $(X, Y) \sim \mu$ almost surely (i.e., μ is realizable w.r.t. $\mathcal{H}$). The hypothesis class $\mathcal{H}$ is arbitrary, and the oracle $\mathcal{O}$ is assumed to return an empirical risk minimizer (ERM) over a given dataset. Let $\mathcal{D}_t := \{(X_i, Y_i)\}_{i=1}^t \stackrel{i.i.d.}{\sim} \mu$, and define $\hat{h}_t = \mathcal{O}(\mathcal{D}_t)$. We define the learning curve $L(t)$ and the expected learning curve $\bar{L}(t)$ as follows

$$L(t) = \mathbb{E}_{(X, Y) \sim \mu} [\ell(\hat{h}_t(X), Y)] \quad \text{and} \quad \bar{L}(t) := \mathbb{E}_{\mathcal{D}_t} \mathbb{E}_{(X, Y) \sim \mu} [\ell(\hat{h}_t(X), Y)] \quad (3)$$

where (X, Y) is a fresh sample independent of $\mathcal{D}_t$.

Remark 2.1. The fixed-design upper bounds hold for any oracle: they depend only on its expected learning curve $\bar{L}(t)$ on an i.i.d. prefix and require neither realizability nor exact ERM. The comparison is to full retraining with that same oracle. Protocol 1 permits arbitrary training subsets, but the fixed-design results use all previous samples. Algorithm 2 instead uses data since the detected change. Keeping only data since the previous update requires a local-buffer learning curve, and can change the optimal schedule.

It is known that for a hypothesis class $\mathcal{H}$ with finite VC-dimension and for the 0-1 loss, the learning curve satisfies, for any $\mu \in \mathcal{P}$ $\bar{L}(t) \leq O(\text{VC}(\mathcal{H})t^{-1} \log t) = \tilde{O}(t^{-1})$, where $\tilde{O}$ hides poly-logarithmic factors in t . It is also known that the learning curve behaves like $L(t) = O\left(\frac{\text{VC}(\mathcal{H}) \log t + \log(1/\delta)}{t}\right)$ with probability at least $1 - \delta$ (Shalev-Shwartz & Ben-David, 2014).

In the case of real-valued hypothesis classes, when the loss function is Lipschitz (e.g., absolute loss) and the fat-shattering dimension scales as γ^{-p} for some $p > 0$ (for example, for $\mathcal{H} = \{h_{\mathbf{w}}(\mathbf{x}) := \langle \mathbf{w}, \mathbf{x} \rangle : \|\mathbf{w}\|_p \leq 1, \|\mathbf{x}\|_{\frac{p}{p-1}} \leq 1\}$), it can be shown (see (Attias et al., 2023, Theorem 2) and the discussion that follows) that, for any $\mu \in \mathcal{P}$,

$$\bar{L}(t) \leq \tilde{O}\left(t^{-\frac{1}{p+1}}\right). \quad (4)$$

We refer to learning curves of the form $\bar{L}(t) = \tilde{O}(t^{-\alpha})$ with $\alpha < 1$ as exhibiting *power-law* scaling, a phenomenon that is also observed empirically in many practical settings (Kaplan et al., 2020; Hestness et al., 2017; Hoffmann et al., 2022). A concrete example where the learning curve provably follows a power law is provided in Appendix D. Beyond i.i.d. processes, we consider more general nonstationary data-generating processes, including:

Piecewise-Stationary Environment. The data are generated independently according to a sequence of (unknown) distributions $\{\mu_t\}_{t=1}^T$, where μ_t remains fixed within each contiguous segment but may *change* between segments. Here, the class $\mathcal{P}$ consists of all such piecewise-stationary processes.

Gradual Distribution Drift. The data are generated independently according to $\{\mu_t\}_{t=1}^T$ with bounded stepwise variation: $\|\mu_t - \mu_{t-1}\|_{\text{TV}} \leq \gamma$ for $t \in \{2, \dots, T\}$. Here, $\mathcal{P}$ consists of all such processes with drift rate at most γ . The learner does not know γ but is given an upper bound $\bar{\gamma} \in (0, 1]$ on it.

3 Main Results

In this section, we present our main theoretical and algorithmic results, focusing primarily on the i.i.d. setting, with extensions to the piecewise-stationary case in Section 3.2.1. Experiments are reported in Appendix B. Throughout, we assume that the loss function $\ell(\cdot, \cdot)$ is bounded.

3.1 Fixed-Design Schedule

We begin with a general result showing that only $\log T$ updates are sufficient to be *competitively* optimal when the learning curve $\bar{L}(t)$ is *non-increasing*. Observe that $\text{Risk}_{T,T}$ corresponds to the area under the learning curve $\bar{L}(t)$, while the risk $\text{Risk}_{S,T}$ of the update schedule at $\tau_1, \dots, \tau_S$ can be viewed as the area under a step function.

We begin with the following result, whose proof is given in Appendix A.

Proposition 3.1. *Consider any i.i.d. learning environment $(\mathcal{P}, \mathcal{H}, \mathcal{O})$ with a fixed full-prefix oracle whose learning curve $\bar{L}(t)$ is non-increasing. If the update times are chosen as $\tau_i = 2^{i-1}$ for $i = 1, \dots, \lceil \log_2 T \rceil$, then, with $S = \lceil \log_2 T \rceil$,*

$$\text{Risk}_{S,T}(\mathcal{P}) \leq 2 \sup_{\mu \in \mathcal{P}} R_{\text{full}}(\mu).$$

Here $R_{\text{full}}(\mu) = \sum_{t=0}^{T-1} \bar{L}_\mu(t)$ includes the initial predictor. The doubling schedule matches this same-oracle full-update risk within factor 2, using only $O(\log T)$ updates.

Remark 3.2. Proposition 3.1 is, at its core, the classical doubling trick (Cesa-Bianchi & Lugosi, 2006) repurposed for our setting. We claim no novelty for the technique itself. However, the level of generality presented here is, to our knowledge, novel. In particular, Proposition 3.1 is conceptually significant: the update rule requires no knowledge of the exact curve, yet with only exponentially fewer updates, it achieves risk that is competitively optimal w.r.t. the full-update benchmark.

While Proposition 3.1 establishes a universal update rule with only $\log T$ updates, it leaves open the more challenging regime where the update budget is substantially smaller provided we have some extra knowledge about the learning curve. Quite surprisingly, when the learning curve follows a power law $\bar{L}(t) \leq 1/t^\alpha$ with $\alpha < 1$, the structure of optimal scheduling becomes fundamentally different, as shown in the theorem below, with full proof in Appendix C.

Theorem 3.3. *Consider an i.i.d. environment and a fixed oracle trained on the full prefix, with loss in $[0, B]$ and expected learning curve $\bar{L}(t)$. (i) **Power-law regime.** Suppose $\bar{L}(t) \leq Ct^{-\alpha}$ for $t \geq 1$, uniformly over $\mu \in \mathcal{P}$, where $0 < \alpha < 1$. For a budget S , put $n = S + 1$, $w_i = \alpha^{n-i}$, $W = \sum_{i=1}^n w_i$, and*

$$\beta = \frac{1}{W} = \frac{1 - \alpha}{1 - \alpha^{S+1}}, \quad \lambda = \left(\prod_{i=1}^n w_i^{w_i} \right)^{-1/W}, \quad r_0 = 1, \quad r_i = \lambda w_i T^\beta r_{i-1}^\alpha. \quad (5)$$

Then $r_n = T$. The integer schedule obtained from $r_1, \dots, r_S$ as specified in Appendix C uses at most S updates and satisfies

$$\text{Risk}_{S,T}(\mathcal{P}) \leq B + 2 \max\{B, C\} \lambda W T^\beta = O_{\alpha, B, C}(T^\beta), \quad \lambda W \leq \frac{\alpha^{-\alpha/(1-\alpha)}}{1 - \alpha}. \quad (6)$$

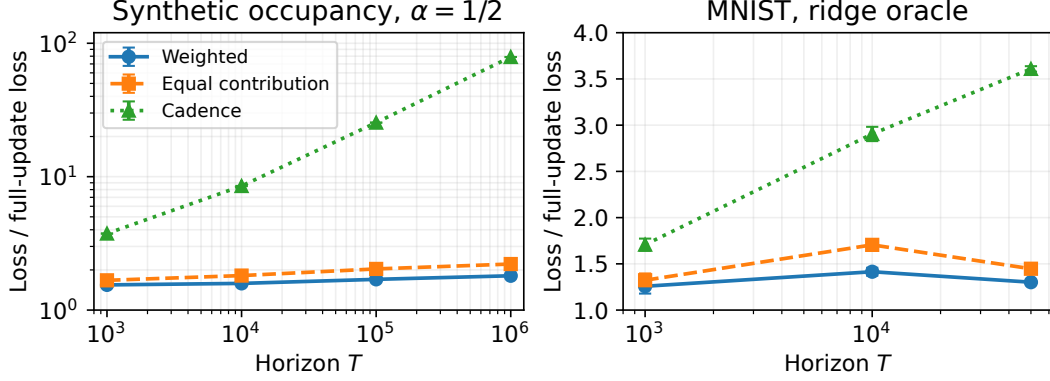


Figure 1: Cumulative loss relative to full update as the horizon grows. Left: synthetic streams from the multiscale occupancy model of Example 1 with $\alpha = 1/2$, whose learning curve is $\Theta(t^{-1/2})$ (Lemma D.1); $S = \lceil \log \log T / \log 2 \rceil$ updates (three at $T = 10^3$, four for larger T), three streams. Right: MNIST, exact ridge oracle, fitted exponent $\hat{\alpha} = 0.30$ (Table 1); $S = 2$ updates, five streams. Every schedule makes the same S updates and retrains the same oracle on all past data: Weighted is the schedule of Theorem 3.3(i), Equal contribution updates at $\lfloor T^{(1-\alpha^i)/(1-\alpha^{S+1})} \rfloor$, and Cadence at the nearest integers to $iT/(S+1)$; full update retrains after every round. Points show means and sample standard deviations. Weighted stays within a factor of 1.81 of full update on synthetic streams up to $T = 10^6$ and 1.41 on MNIST up to $T = 50,000$, whereas Cadence reaches 79 and 3.6. Setup, other exponents, and Fashion-MNIST: Appendix B.4, Tables 3 and 4.

In particular, $S = \max\{0, \lceil \log \log T / \log(1/\alpha) \rceil - 1\}$ suffices for $O_{\alpha,B,C}(T^{1-\alpha})$ risk, for $T \geq e^2$.

(ii) **Logarithmic regime.** If $\bar{L}(t) \leq C/t$, the rounded geometric schedule $\tau_i = T^{i/(S+1)}$ attains $O_{B,C}((S+1)T^{1/(S+1)})$ risk. Thus $S = \lceil \log T \rceil$ gives $O_{B,C}(\log T)$ risk. Finite-VC ERM curves satisfy the corresponding statement up to their additional logarithmic factors.

(iii) **Lower bounds.** If, for some μ_* , $\bar{L}_{\mu_*}(0) \geq c$ and $\bar{L}_{\mu_*}(t) \geq ct^{-\alpha}$, every fixed full-prefix schedule incurs at least $(c/2)T^\beta$ risk. Thus part (i) is tight with constants independent of S, T . For $\alpha = 1$ the analogous lower bound is $\Omega_c(T^{1/(S+1)})$. These lower bounds also cover adaptive rules if the lower envelope holds for their conditional predictive risks at every selected update.

Sketch of Proof. Unrolling (5) gives $r_n = T^{\beta W} \lambda^W \prod_i w_i^{w_i} = T$. Before rounding, for $r_1 \geq 1$, the stepwise risk is at most

$$Br_1 + C \sum_{i=2}^n (r_i - r_{i-1}) r_{i-1}^{-\alpha} \leq \lambda[Bw_1 + C(W - w_1)]T^\beta.$$

The epoch contributions now have geometric weights w_i , removing the $S+1$ prefactor of the equal-contribution schedule $\tau_i = T^{(1-\alpha^i)/(1-\alpha^{S+1})}$. Moreover, $\log \lambda \leq \alpha \log(1/\alpha)/(1-\alpha)$. Appendix C gives the integer construction and lower bound. $\square$

Remark 3.4. Note that, for the power-law curve $\bar{L}(t) = \Theta(1/t^\alpha)$, the optimal *full-update* risk is of order $\Theta(T^{1-\alpha})$. Theorem 3.3 shows that with only $S = O(\log \log T)$ updates, doubly exponentially fewer than full retraining, the weighted schedule *matches* this risk within a constant factor depending only on α, B, C, c . For $B = 1$ and $\bar{L}(t) = t^{-1/2}$, the explicit asymptotic ratio is at most $2\sqrt{e} + o(1)$ (Appendix C). Since the budget permits *at most* S updates, optimal risk cannot increase with S .

Moreover, Theorem 3.3 remains near optimal (up to polylogarithmic factors in the risk) even when the learning curve fluctuates within a polylogarithmic factor, i.e., when $\bar{L}(t) = \tilde{\Theta}(1/t)$ or $\bar{L}(t) = \tilde{\Theta}(1/t^\alpha)$. In fact, for $\bar{L}(t) = \tilde{\Theta}(1/t^\alpha)$ with $\alpha < 1$, Theorem 3.3 with $S = O(\log \log T)$ remains *competitively* (near) optimal w.r.t. optimal full-update risk. Figure 1 confirms this on synthetic streams up to $T = 10^6$ and on MNIST; Appendix B reports the full comparisons.

Unfortunately, the doubly-exponential improvement in the update budget does not extend to the case where the fluctuations are of polynomial size, as shown below. The proof is deferred to Appendix E.

Theorem 3.5. Fix constants $0 < \alpha_1 < \alpha_2 < 1$. There is a constant $c > 0$ such that, for all sufficiently large T and every $1 \leq S \leq c \log T / \log \log T$, there exists a non-increasing function $\bar{L} = \bar{L}_{T,S}$ with $t^{-\alpha_2} \leq \bar{L}(t) \leq t^{-\alpha_1}$ for $t \geq 1$ and the following property: for any realizable i.i.d. learning environment with learning curve $\bar{L}(t)$, every full-prefix update schedule fixed in advance with at most S updates incurs risk at least $T^{\Omega(1/S)} \cdot \text{Risk}_{T,T}(\mathbf{P})$. In particular, for $S \leq o(\frac{\log T}{\log \log T})$, the competitive ratio must be super-polylogarithmic. The same holds for adaptive full-prefix rules under the conditional lower envelope of Theorem 3.3(iii).

3.2 Risk-Driven Update Schedule

We have established several optimal update schedules that rely only on the shape of the learning curve. In practice, however, the learning curve is rarely known in advance, and the learning environment may deviate substantially from the i.i.d. assumption. In this section, we leverage our theoretical schedules as *primitives*, which can be composed and adapted to design update rules for more realistic settings, thereby bridging the gap between idealized theory and practical applicability.

We begin with the case where the (realized) learning curve $L(t)$ follows a power law with high probability, but with an *unknown* exponent α (e.g., hypothesis classes with finite fat-shattering dimension under ERM retraining). Specifically, we assume that $L(t) = \tilde{\Theta}(\frac{1}{t^\alpha})$ holds simultaneously for all t with probability at least $1 - \frac{1}{T}$, for some (unknown) $\alpha > 0$.

Without knowledge of the exponent α , the update schedule proposed in Theorem 3.3 no longer applies. To address this, we introduce Algorithm 1, which monitors the *empirical loss* $\hat{L}(t)$ (defined in Algorithm 1) to guide updates. The algorithm begins with an initial estimate $\hat{\alpha}$ and keeps it fixed as long as the empirical learning curve $\hat{L}(t)$ behaves typically, as guaranteed by Lemma F.1. When $\hat{L}(t)$ deviates from this typical behavior, $\hat{\alpha}$ is updated according to lines 7–9 of Algorithm 1.

Our next main theorem shows that Algorithm 1 is *agnostic* to α and matches the full-update risk up to polylogarithmic factors. Unlike the fixed schedules above, it chooses its own number of updates, which the theorem bounds. We provide a proof sketch here and defer the detailed proof to Appendix F. Note that while the fluctuations of $L(t)$ are logarithmic in the theorem and in Algorithm 1, both the algorithm and its proof can be easily extended to handle polylogarithmic fluctuations.

Algorithm 1 Adaptive schedule for unknown $\alpha < 1$

- 1: **Input:** Time horizon T , constant C_1 .
 - 2: **Output:** Update rule $\Phi = (\Phi[1], \dots, \Phi[T])$.
 - 3: **Initialize:** Let $\hat{\alpha} = \hat{\alpha}_0 := 1 - \frac{\log \log T}{\log T}$, $\hat{S} = \frac{\log \log T}{\log(1/\hat{\alpha})}$, $\hat{\beta} = \frac{1-\hat{\alpha}}{1-\hat{\alpha}\hat{S}}$, $i = 0$, $\tau_0 = 0$, $\tau_1 = \lfloor T^{\hat{\beta}} \rfloor$,
and randomly pick $\hat{h}_{\tau_0} = \hat{h}_0 \in \mathcal{H}$.
 - 4: **for** $t = 1, \dots, T$ **do**
 - 5: Compute $\hat{L}(t) = \frac{\sum_{j=\tau_i+1}^t \ell(\hat{h}_{\tau_i}(X_j), Y_j)}{t - \tau_i}$ and set $\Phi[t] = 0$.
 - 6: **if** $t = \tau_{i+1}$ **then**
 - 7: **if** $i \geq 1$ and $\frac{(t-\tau_i)\hat{L}(t)}{14} \geq 2 \log T$ **then**
 - 8: Compute $\alpha_{\max} = \max \left\{ \alpha : \frac{2\hat{L}(t)}{3} \leq \frac{C_1 \log \tau_i}{\tau_i^\alpha} \right\}$.
 - 9: Set $\hat{\alpha} = \min\{\max\{\alpha_{\max}, \frac{1}{\log T}\}, \hat{\alpha}_0\}$, $\hat{S} = \frac{\log \log T}{\log(1/\hat{\alpha})}$, and $\hat{\beta} = \frac{1-\hat{\alpha}}{1-\hat{\alpha}\hat{S}}$.
 - 10: **end if**
 - 11: Set $\Phi[t] = 1$ and $i \leftarrow i + 1$.
 - 12: Set $\hat{h}_{\tau_i} = \mathcal{O}(\{(X_j, Y_j)\}_{j=1}^{\tau_i})$ and $\tau_{i+1} = \left\lfloor \max \left\{ 2^{-i} \tau_i^{\hat{\alpha}} T^{\hat{\beta}}, 2\tau_i \right\} \right\rfloor$.
 - 13: **end if**
 - 14: **end for**
-

Theorem 3.6. Assume losses take values in $[0, 1]$ and there exist constants $C_1 \geq C_2 > 0$ and an unknown exponent $\alpha \in [\Omega(\frac{\log \log \log T}{\log \log T}), 1 - \frac{\log \log T}{\log T}]$ such that, with probability at least $1 - \frac{1}{T}$, the realized learning curve $L(t)$ of the oracle satisfies $C_2 t^{-\alpha} \leq L(t) \leq C_1 \log t \cdot t^{-\alpha}$ simultaneously for all $2 \leq t \leq T$. Then, with probability at least $1 - \frac{2}{T}$, Algorithm 1 makes at most $S =$

$O(\min\{\log T, \frac{\log \log T}{(1-\alpha) \log(1/\alpha)}\})$ updates, which is $O(\frac{\log \log T}{\log(1/\alpha)})$ when α is bounded away from one, and its expected cumulative risk is $\tilde{O}(T^{1-\alpha})$.

Sketch of Proof. At a high level, Algorithm 1 estimates the true exponent α from the empirical loss $\hat{L}(t)$, and then uses the current estimate $\hat{\alpha}$ to set the update schedule as in Theorem 3.3. Let $\tau_{i_1}, \dots, \tau_{i_J}$ be the times when the **if** condition in line 7 of Algorithm 1 is active, i.e., when $\hat{\alpha}$ is updated. We show that, with probability at least $1 - \frac{2}{T}$, for all $t \geq \tau_{i_j}$ and all $j = 1, \dots, J$, $\hat{\alpha} \geq \alpha \geq \hat{\alpha} - \frac{\log \log \tau_{i_j-1} + \log(3C_1/C_2)}{\log \tau_{i_j-1}}$. Thus $\hat{\alpha}$ never underestimates α , and the gap shrinks as τ_{i_j} grows. We then show that the *upper estimate* yields a risk bound $\tilde{O}(T^{1-\alpha})$ by our schedule design, while the *lower bracket* on α eventually keeps $\hat{\alpha} \leq \sqrt{\alpha}$, after which the distance $\log(T/\tau_i)$ contracts geometrically; this bounds the number of updates. $\square$

3.2.1 Shifting Distributions

We now turn to *nonstationary* environments, where the data distribution may *change* over time. We consider two types of distribution shift: (i) a piecewise i.i.d. shift, and (ii) a gradual distribution drift with bounded total variation.

Piece-wise i.i.d. environments. We assume that there are K change points at *unknown locations* $1 < \kappa_1 < \dots < \kappa_K \leq T$, with $\kappa_0 = 1$ and $\kappa_{K+1} = T + 1$; the learner is given K (or an upper bound on it). Within each segment $t \in [\kappa_{i-1}, \kappa_i)$ the samples $\{(X_t, Y_t)\}$ are i.i.d. from some (unknown) distribution $\mu^{(i)}$, while they may differ across segments.

For simplicity, we focus on binary classification with a hypothesis class $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$ of finite VC dimension and an ERM oracle. We further assume *segment-wise realizability*: for all $i \leq K + 1$, there exists $h_i^* \in \mathcal{H}$ such that $h_i^*(X_t) = Y_t$ almost surely for all $t \in [\kappa_{i-1}, \kappa_i)$.

We begin with the simpler benchmark setting in which the change points are *known* to the scheduler.

Proposition 3.7. *Assume that the change points $\kappa_1, \dots, \kappa_K$ are known. Allocate $m := \lfloor S/(K+1) \rfloor$ updates per segment and apply the schedule from Theorem 3.3 (ii) within each segment. Then the cumulative risk satisfies: $\text{Risk}_{S,T}(\mathcal{P}) \leq \tilde{O}((S+K+1) \left(\frac{T}{K+1}\right)^{\frac{1}{\lfloor S/(K+1) \rfloor + 1}})$.*

Sketch of Proof. Within segment k of length n_k , Theorem 3.3 (ii) (using data *only* from that segment) gives a per-segment risk of order $(m+1) n_k^{1/(m+1)} \log n_k$. The result follows by summing over all $K+1$ segments and applying Jensen's inequality, noting that $x^{1/(m+1)}$ is concave and $\sum n_k = T$. $\square$

Algorithm 2 Adaptive schedule for piecewise-stationary environment

- 1: **Input:** Horizon T , budget S , known bound K , constant C_0
 - 2: **Initialize:** Let $m = \lfloor S/(K+1) \rfloor$, $N = T/(K+1)$, $j = 0$, $i = 1$, $\tau_0 = 0$, $\tau_1 = \lceil N^{\frac{1}{m+1}} \rceil$, $d = 0$, $\hat{\kappa} = 0$, and randomly pick $\hat{h}_{\tau_0} = \hat{h}_0 \in \mathcal{H}$;
 - 3: /* j is the number of updates since the last distribution change detected by the **if** condition 6. τ_{i-1} and τ_i are the last and next time for model update, respectively. d is used to guarantee that τ_i cannot be changed once given. */
 - 4: **for** $t = 1, \dots, T$ **do**
 - 5: Compute $\widehat{CL}(t) = \sum_{u=\tau_{i-1}+1}^t \ell(\hat{h}_{\tau_{i-1}}(X_u), Y_u)$ and $\epsilon(t) = \widehat{CL}(t) - \frac{C_0 \log T (t-\tau_{i-1})}{N^{j/(m+1)}}$;
 - 6: **if** $\epsilon(t) > 0$ and $\frac{\epsilon^2(t)}{2C_0 \log T (t-\tau_{i-1})/N^{j/(m+1)} + 2\epsilon(t)/3} \geq 2 \ln T$ and $d = 0$ **then**
 - 7: Let $\hat{\kappa} = t$, $\tau_i = t + \lceil N^{\frac{1}{m+1}} \rceil$, $d = 1$, and $j = 0$;
 - 8: **else if** $t = \tau_i$, $i \leq S$, and $j < m$ **then**
 - 9: Update model $\hat{h}_t = \mathcal{O}(\{(X_u, Y_u)\}_{u=\hat{\kappa}+1}^t)$ and let $j = j + 1$, $i = i + 1$, $\tau_i = t + \lceil N^{\frac{j+1}{m+1}} \rceil$, and $d = 0$;
 - 10: **end if**
 - 11: **end for**
-

The more challenging regime is when the change points are *unknown*: the scheduler must choose both the update times and which samples to pass to the retraining oracle. Algorithm 2 addresses this issue by monitoring the empirical losses to detect shifts. In Theorem 3.8, we show that the algorithm achieves the same risk (up to polylog factors) as the known-change-point benchmark. The proof is deferred to Appendix G.

Theorem 3.8. *Let $\mathcal{P}$ be piecewise-stationary i.i.d. processes with K change points at unknown locations, with K known. Assume binary classification with a finite-VC class $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$ and an ERM oracle such that, with probability at least $1 - \frac{1}{T}$, every predictor trained on $n \geq 1$ samples from a single segment has realized loss at most $C_0 \log T/n$ on that segment. Then Algorithm 2 uses at most S updates and achieves*

$$\text{Risk}_{S,T}(\mathcal{P}) \leq \tilde{O}\left((S + K + 1) \left(\frac{T}{K + 1}\right)^{\frac{1}{\lfloor S/(K+1) \rfloor + 1}}\right).$$

Sketch of Proof. Let $m := \lfloor S/(K + 1) \rfloor$. The proof controls two possible errors of the change detector: false detections and missed changes. If no distributional change occurs in the current epoch, then the data remain i.i.d. within the same stationary segment. Bernstein’s inequality, with the threshold calibrated in line 6, ensures that a false detection does not occur with high probability, and the risk is bounded by applying the schedule of Theorem 3.3 within that segment.

If a distributional change occurs, there are two cases. If the change is detected, the algorithm resets $\hat{\kappa}$ and restarts the schedule of Theorem 3.3 using only post-detection samples. If the change is missed before the next scheduled update, the loss is still controlled: the detection statistic is the cumulative empirical loss itself, so non-detection implies that the loss accumulated since the last update remains below the threshold. Thus each stationary segment contributes at most $\tilde{O}((m + 1)(T/(K + 1))^{1/(m+1)})$, and summing over the $K + 1$ segments gives the claimed bound. $\square$

Remark 3.9. Note that the restriction to learning curves of the form $\bar{L}(t) = \tilde{O}(1/t)$ is not essential. The same proof technique also extends to power-law learning curves $\bar{L}(t) = \tilde{O}(t^{-\alpha})$ with $\alpha < 1$, by replacing the within-segment schedule with that of Theorem 3.3(i) and calibrating the detection threshold to the corresponding epoch-wise risk scale. We omit the details for clarity of exposition.

Gradual distribution drift environments. We now consider the *gradual distribution shift* setting, where the data (X_t, Y_t) follow a distribution μ_t such that the total variation distance between μ_t and μ_{t-1} is bounded by a parameter γ , i.e., $\|\mu_t - \mu_{t-1}\|_{\text{TV}} \leq \gamma$ for $t \in \{2, \dots, T\}$. The learner does not know γ but is given an upper bound $\bar{\gamma} \in (0, 1]$ on it. Learning with such slowly changing distributions was studied by Bartlett (1992). Assuming that every μ_t is realizable by $\mathcal{H}$, the following theorem, proven in Appendix H, shows that Algorithm 2 attains risk $\tilde{O}(1 + T\bar{\gamma}^{1/2})$ using $\tilde{O}(1 + T\bar{\gamma}^{1/2})$ updates.

Theorem 3.10. *Consider a learning environment where the data distributions μ_t , $t \in \{1, \dots, T\}$, satisfy $\|\mu_{t+1} - \mu_t\|_{\text{TV}} \leq \gamma$ and each μ_t is realizable by $\mathcal{H}$, which has VC dimension d . The loss is 0–1 valued, γ is unknown, and the learner is given d and $\bar{\gamma} \in (0, 1]$ with $\gamma \leq \bar{\gamma}$. Then Algorithm 2, with K , S , and C_0 chosen as in Appendix H, achieves risk $\text{Risk}_{S,T}(\mathcal{P}) \leq \tilde{O}(1 + T\bar{\gamma}^{1/2})$ with at most $S = \tilde{O}(1 + T\bar{\gamma}^{1/2})$ updates.*

Remark 3.11. The bound has a natural learning-curve interpretation. Under realizability, Lemma H.1 shows that an empirical risk minimizer trained on n consecutive samples has realized risk $O(\log T/n)$ plus 8γ times the span of the window. Windows of length of order $(\log T/\bar{\gamma})^{1/2}$ balance these two terms, so within such a window the predictive error behaves as in an i.i.d. environment with the standard $1/t$ learning curve. Algorithm 2 thus partitions the stream into $\tilde{O}(1 + T\bar{\gamma}^{1/2})$ effective segments and applies the i.i.d. schedule within each. The same learning-curve abstraction that drove our i.i.d. results continues to govern the drift case—only now the relevant curve is windowed.

3.2.2 A No Free Lunch Theorem

Finally, we present a *no-free-lunch*-type negative theorem: in the absence of *any* prior information about the learning environment, no risk-driven (empirical-loss-based) universal update rule can achieve competitively optimal risk to the known environment setting for a given number of model

updates. A single *reactive* change already suffices: it occurs at the learner’s first update, and the post-change distribution depends on the samples observed up to that time. The rules choose their update times and retrain on the full prefix, starting from the oracle’s output on the empty sample; optimality is measured against the best update rule in the same environment. The full proof is deferred to Appendix I.

Theorem 3.12. *There exist a reactive environment E with one change, which occurs at the learner’s first update, and two hypothesis classes $\mathcal{H}_1$ and $\mathcal{H}_2$ such that any update rule Φ with $S = 2$ that retrains on the full prefix, starts from the oracle’s output on the empty sample, and lacks knowledge of the hypothesis class (while being allowed knowledge of E) incurs $T^{\Omega(1)}$ times the optimal expected risk against E in at least one of the settings $(E, \mathcal{H}_1, \mathcal{O}^{\mathcal{H}_1})$ or $(E, \mathcal{H}_2, \mathcal{O}^{\mathcal{H}_2})$, where $\mathcal{O}^{\mathcal{H}_1}$ and $\mathcal{O}^{\mathcal{H}_2}$ denote ERM oracles for $\mathcal{H}_1$ and $\mathcal{H}_2$.*

Sketch of Proof. The proof constructs two environments that are indistinguishable to any risk-driven scheduler but have incompatible near-optimal update windows. Since a risk-driven scheduler observes only the losses of its current predictors, we build $\mathcal{H}_1, \mathcal{H}_2$, target functions, and a reactive one-change environment so that the observed loss sequence before the second update is identical in both environments; hence the scheduler must choose the same update times τ_1, τ_2 . However, near-optimality imposes opposite requirements: under $\mathcal{H}_1$, the second update must occur in an early window, while under $\mathcal{H}_2$, any such early update leaves a polynomially large loss, so the second update must occur later. Thus the same loss history cannot support a schedule that is near-optimal in both environments. The construction can also be extended to any constant $S \geq 2$ by nesting the same gadget. $\square$

Remark 3.13. Theorem 3.12 can be viewed as a converse to our positive results. Theorems 3.6 and 3.8 show that limited information about the learning curve enables near-optimal scheduling relative to the best environment-aware schedule, while Theorem 3.12 shows that, without *any* such information, no full-prefix risk-driven rule can be near-optimal in general. Together, these results identify information about the learning curve as the key prior for competitive-optimal retraining.

4 Conclusion and Discussion

In this paper, we introduced a general framework for analyzing the *optimal scheduling* of retraining in online models. Our results provide a precise characterization of when and how often retraining is needed: we designed schedule rules tailored to the shape of the learning curve, proved they achieve fundamental performance limits, and extended them to more realistic regimes by leveraging empirical losses to guide updates. Together, these results move beyond heuristic criteria and establish a principled theory of retraining frequency. For clarity of exposition, we adopted several simplifying assumptions. For instance, we treated the retraining budget as a single oracle call, independent of the size of the training set. In practice, the computational cost of retraining often depends on the volume of data processed. A natural future direction is to investigate optimal update schedules under joint constraints on update frequency and data cost (e.g., in Algorithm 2 we *jointly* design the update schedule and the dataset). Another promising extension is to adapt our framework to broader learning environments, such as the smoothed adversarial processes (Haghtalab et al., 2020), where nonstationarity is more structured yet still adversarial in spirit.

Acknowledgments and Disclosure of Funding

This work is partially supported by the NSF Center for Science of Information (CSoI) Grant CCF-0939370, and by NSF Grants CCF-2006440 and CCF-2211423.

References

Jason Altschuler and Kunal Talwar. Online learning over a finite action set with limited switching. In *Proceedings of the 31st Conference on Learning Theory (COLT)*, volume 75 of *Proceedings of Machine Learning Research*, pp. 1569–1573, 2018. URL <http://proceedings.mlr.press/v75/altschuler18a/altschuler18a.pdf>. Extended version: arXiv:1803.01548.

- Idan Attias, Steve Hanneke, Alkis Kalavasis, Amin Karbasi, and Grigoris Velezgas. Optimal learners for realizable regression: Pac learning and online learning. *Advances in Neural Information Processing Systems*, 36:44707–44739, 2023.
- Peter L Bartlett. Learning with a slowly changing distribution. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 243–252, 1992.
- Adam Block, Yuval Dagan, Noah Golowich, and Alexander Rakhlin. Smoothed online learning is as easy as statistical learning. *arXiv preprint arXiv:2202.04690*, 2022.
- Stéphane Boucheron, Olivier Bousquet, and Gábor Lugosi. Theory of classification: a survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.
- N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning and Games*. Cambridge University Press, 2006.
- Pramith Devulapalli and Steve Hanneke. Learning from snapshots of discrete and continuous data streams. *arXiv preprint arXiv:2412.06079*, 2024.
- Pramith Devulapalli, Changlong Wu, Ananth Grama, and Wojciech Szpankowski. Agnostic continuous-time online learning. *NeurIPS*, 2025.
- Matthew Fahrback, Adel Javanmard, Vahab Mirrokni, and Pratik Worah. Learning rate schedules in the presence of distribution shift. In *International Conference on Machine Learning*, pp. 9523–9546. PMLR, 2023.
- Dylan J Foster, Satyen Kale, Haipeng Luo, Mehryar Mohri, and Karthik Sridharan. Logistic regression: The importance of being improper. In *COLT - Conference on Learning Theory*, 2018.
- Nika Haghtalab, Tim Roughgarden, and Abhishek Shetty. Smoothed analysis of online and differentially private learning. *Advances in Neural Information Processing Systems*, 33:9203–9215, 2020.
- Steve Hanneke. Theory of disagreement-based active learning. *Foundations and Trends in Machine Learning*, 7(2–3):131–309, 2014. doi: 10.1561/22000000037.
- Elad Hazan and Comandur Seshadhri. Efficient learning algorithms for changing environments. In *Proceedings of the 26th annual international conference on machine learning*, pp. 393–400, 2009.
- Xudong He et al. Incremental learning in online scenario. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/papers/He_Incremental_Learning_in_Online_Scenario_CVPR_2020_paper.pdf.
- Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable. *Empirically*. *arXiv*, 1712(2), 2017.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.
- Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L. Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *Transactions on Machine Learning Research (TMLR)*, 2024. URL <https://arxiv.org/abs/2403.08763>. arXiv:2403.08763.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- Alexander Rakhlin and Karthik Sridharan. Sequential probability assignment with binary alphabets and large classes of experts. *arXiv preprint arXiv:1501.07340*, 2015.

- Florence Regol, Leo Schwinn, Kyle Sprague, Mark Coates, and Thomas Markovich. When to retrain a machine learning model. *ICLR 2025 (OpenReview)*, 2025. Submitted to ICLR 2025; arXiv preprint arXiv:2505.14903 (May 2025).
- Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- Changlong Wu, Mohsen Heidari, Ananth Grama, and Wojciech Szpankowski. Precise regret bounds for log-loss via a truncated bayesian algorithm. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26903–26914, 2022.
- Changlong Wu, Ananth Grama, and Wojciech Szpankowski. Online learning in dynamically changing environments. In *Conference on Learning Theory*, pp. 325–358. PMLR 195, 2023.
- Changlong Wu, Ananth Grama, and Wojciech Szpankowski. Online universal learning from information-theoretic perspective. *Foundation and Trends in Communication and Information Theory*, 2026. URL <https://www.cs.purdue.edu/homes/spa/papers/fnt25.pdf>.
- Marco Zanotti. On the retraining frequency of global forecasting models. *arXiv preprint arXiv:2505.00356*, 2025.

A Proof of Proposition 3.1

Proof. Assume $T \geq 2$; the case $T = 1$ is trivial. Let $M := \lceil \log_2 T \rceil$, set $\tau_0 = 0$, $\tau_i = 2^{i-1}$ for $1 \leq i \leq M$, and $\tau_{M+1} = T$. Fix any $\mu \in \mathcal{P}$, and write $a_t := \bar{L}(t)$. Since $\bar{L}(t)$ is non-increasing, the same-oracle full-update benchmark is

$$R_{\text{full}}(\mu) = \sum_{t=0}^{T-1} a_t.$$

For the dyadic schedule Φ_{dyad} ,

$$\text{Risk}_{M,T}(\Phi_{\text{dyad}}, \mu) = \sum_{i=0}^M (\tau_{i+1} - \tau_i) a_{\tau_i}.$$

The first two terms are bounded by

$$(\tau_1 - \tau_0) a_{\tau_0} + (\tau_2 - \tau_1) a_{\tau_1} \leq 2a_0,$$

where we use monotonicity. For $i \geq 2$, the choice $M = \lceil \log_2 T \rceil$ ensures $\tau_{i+1} - \tau_i \leq 2(\tau_i - \tau_{i-1})$, including the final interval. Thus, again by monotonicity,

$$(\tau_{i+1} - \tau_i) a_{\tau_i} \leq 2(\tau_i - \tau_{i-1}) a_{\tau_i} \leq 2 \sum_{t=\tau_{i-1}}^{\tau_i-1} a_t.$$

Summing the above bounds gives

$$\text{Risk}_{M,T}(\Phi_{\text{dyad}}, \mu) \leq 2 \sum_{t=0}^{T-1} a_t = 2R_{\text{full}}(\mu).$$

Taking the supremum over $\mu \in \mathcal{P}$ gives the stated upper bound. $\square$

B Empirical evaluation

We evaluate the frequency–risk trade-off on MNIST, Fashion-MNIST, and a synthetic multiscale model: learning-curve fits, matched-budget comparisons, longer horizons, adaptive schedules, and change-point detection. All tables and figures are generated by the accompanying code from per-seed result files.

B.1 Protocol, data, and oracles

At each round, prediction and loss evaluation precede any training on that example. Each real-data stream is a random permutation of the 60,000-example training split, with identical prefixes across schedules and horizons. Pixels are divided by 255, without augmentation or further normalization. Sampling without replacement approximates the i.i.d. setting. Classification loss is zero–one error.

Exact ridge oracle. We use 512 random ReLU features $\phi(x) = \max\{xA + b, 0\}$, with $A_{ij} \sim N(0, 1/784)$ and $b_j \sim U[-0.5, 0.5]$, fixed within each seed. On a training prefix the oracle returns

$$\widehat{W} = (\Phi^\top \Phi + 0.1I)^{-1} \Phi^\top Y,$$

where Y contains one-hot labels; prediction is the largest output coordinate. The oracle exactly minimizes a regularized squared loss; we report zero–one error. The full-update reference uses the identical oracle after every observation, computed by rank-one inverse updates; on a small test problem, its predictions coincide with those of independent batch solves. No model update after the final prediction is charged; “full update” denotes $T - 1$ effective refits, consistent with the budget convention $S = T$.

MLP oracle and reference. The second oracle is a 784–128–10 ReLU MLP trained from the same seed-specific initialization at each checkpoint for three shuffled passes of Adam (learning rate 10^{-3} , batch size 64, moment parameters 0.9 and 0.999, numerical stabilizer 10^{-8} , no weight decay). The reference takes one Adam minibatch step per round on samples drawn with replacement from the current prefix. Because this is a different training rule from the three-pass checkpoint oracle, MLP ratios compare against online Adam rather than against the same oracle; the ridge experiments provide the same-oracle comparison.

Synthetic model. The synthetic streams implement Example 1 with ground truth $\sigma \equiv 1$. For $x_t \stackrel{\text{i.i.d.}}{\sim} U[0, 1)$, level ℓ has $2^{q\ell}$ cells with weight $4 \cdot 2^{-p\ell}$, and the loss of the ERM oracle at a fresh point is the total weight of the cells containing it that contain no training sample (Proposition D.2). Thus

$$\bar{L}(t) = \sum_{\ell \geq 1} 4 \cdot 2^{-p\ell} (1 - 2^{-q\ell})^t = \Theta(t^{-p/q})$$

by Lemma D.1. We use $(p, q) = (1, 2), (1, 3), (2, 3)$. Levels are truncated when the next weight is at most 10^{-6} or the next cell index would require more than 45 bits. The omitted weight is added as a constant to every per-round loss under every schedule, which bounds the truncation error per round.

Schedules and statistics. Every schedule calls the same oracle on the full prefix at its checkpoints $\tau_1 < \dots < \tau_S$. Weighted denotes (5), rounded as in Appendix C; Equal denotes the equal-contribution schedule $\tau_i = \lfloor T^{(1-\alpha^i)/(1-\alpha^{S+1})} \rfloor$. Both are run with the fitted exponent of Table 1 (“fit”); for the occupancy model, the true exponent p/q and with $\alpha = 0.5$ (“0.5”). We also evaluate the geometric schedule $\tau_i = \lfloor T^{i/(S+1)} \rfloor$; fixed cadence (Cadence), with τ_i the nearest integer to $iT/(S+1)$; doubling truncated to its first S checkpoints $\tau_i = 2^{i-1}$; and S distinct checkpoints drawn uniformly from $\{1, \dots, T-1\}$. Duplicate integer checkpoints are removed, so a schedule may use fewer than S updates; for example, with $S = 8$, Weighted with the fitted ridge exponent uses six updates on MNIST and five on Fashion-MNIST. The complete budget sweep is $S \in \{1, 2, 3, 4, 6, 8\}$. Ratios are calculated within each seed before aggregation. Tables report means and sample standard deviations (divisor $n - 1$) over five seeds (0–4), or three seeds (0–2) for MLP and synthetic horizon scaling. Seeds vary stream order and oracle initialization; random schedule seeds are paired across comparisons.

B.2 Learning-curve fits

For each oracle and seed, an independent shuffle (seed 1000 + s) supplies training prefixes of sizes given by 14 logarithmically spaced integers from 20 to 10,000. We evaluate on 4,000 disjoint examples at shuffled positions 50,000–53,999 and fit a straight line to log error versus log training size for sizes at least 50. These calibration runs use separate random permutations of the same training split.

Table 1: Power-law fits over $t \geq 50$. The exponent is fitted to the mean curve; its reported dispersion is the sample standard deviation of per-seed fits.

Dataset	Oracle	$\hat{\alpha}$	$\hat{C}$	R^2
MNIST	Ridge	0.302 ± 0.020	1.45	0.964
MNIST	MLP	0.489 ± 0.035	5.51	0.977
Fashion-MNIST	Ridge	0.181 ± 0.017	0.83	0.963
Fashion-MNIST	MLP	0.268 ± 0.002	1.78	0.973

B.3 Matched budgets and sensitivity to the exponent

Table 2 reports representative budgets from all six dataset/oracle environments; the accompanying CSV and JSON files contain every schedule and budget, including geometric, truncated doubling, and random updates. We report both the fitted exponents and the fixed choice $\alpha = 0.5$. With the fitted exponent, Weighted has a lower ratio than Equal in 17 of the 18 settings of Table 2 (the exception is MNIST/MLP with $S = 2$) and a lower ratio than fixed cadence in all 18. With the fixed choice 0.5, Equal is lower than Weighted in 12 of the 18 settings. Because the weighted construction optimizes the upper bound rather than the measured curve, Equal can also win against a fitted exponent: with four updates on MNIST/ridge, Equal with exponent 0.5 attains 1.21, below both weighted variants (1.31 and 1.33). An additional sweep over $\alpha = 0.05, 0.10, \dots, 0.95$ at budgets $S = 2, 4, 6$ for the Equal schedule is provided in `e2c_sens.json`; it was not used to select any reported schedule.

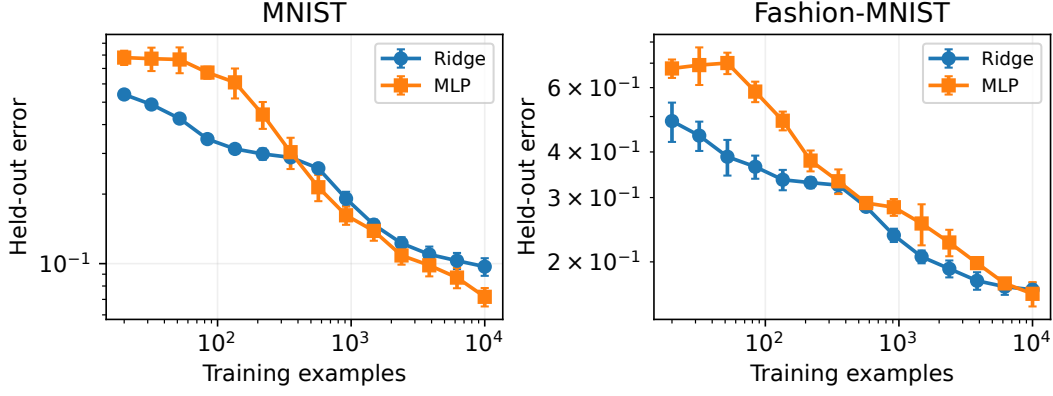


Figure 2: Measured learning curves. Error bars show sample standard deviations over five ridge or three MLP runs.

Table 2: Matched-budget loss ratios at $T = 10^4$; “fit” uses the fitted exponent, or the true exponent for occupancy, and “.5” uses $\alpha = 0.5$. Ridge and occupancy denominators use the identical oracle every round; MLP denominators are the online-Adam reference, a different training rule. Five seeds, or three for MLP.

Environment	S	Weighted fit	Weighted .5	Equal fit	Equal .5	Cadence
MNIST / ridge	2	1.41 ± 0.04	1.47 ± 0.03	1.71 ± 0.03	1.41 ± 0.03	2.90 ± 0.08
MNIST / ridge	4	1.33 ± 0.02	1.31 ± 0.02	1.63 ± 0.04	1.21 ± 0.02	2.10 ± 0.06
MNIST / ridge	8	1.33 ± 0.02	1.24 ± 0.02	1.62 ± 0.04	1.19 ± 0.03	1.58 ± 0.03
Fashion / ridge	2	1.26 ± 0.02	1.25 ± 0.02	1.72 ± 0.02	1.23 ± 0.02	2.19 ± 0.03
Fashion / ridge	4	1.23 ± 0.01	1.16 ± 0.02	1.69 ± 0.02	1.12 ± 0.01	1.69 ± 0.02
Fashion / ridge	8	1.23 ± 0.02	1.13 ± 0.01	1.69 ± 0.02	1.10 ± 0.01	1.37 ± 0.02
MNIST / MLP	2	2.74 ± 0.06	2.80 ± 0.08	2.63 ± 0.16	2.73 ± 0.19	4.65 ± 0.19
MNIST / MLP	4	2.13 ± 0.03	2.12 ± 0.04	2.31 ± 0.04	2.22 ± 0.07	3.30 ± 0.11
MNIST / MLP	8	2.03 ± 0.08	1.99 ± 0.04	2.11 ± 0.06	2.07 ± 0.07	2.42 ± 0.08
Fashion / MLP	2	1.72 ± 0.04	1.78 ± 0.02	1.81 ± 0.03	1.73 ± 0.08	2.55 ± 0.01
Fashion / MLP	4	1.65 ± 0.10	1.47 ± 0.02	1.74 ± 0.05	1.53 ± 0.09	2.02 ± 0.04
Fashion / MLP	8	1.60 ± 0.05	1.50 ± 0.03	1.70 ± 0.03	1.44 ± 0.01	1.66 ± 0.05
Occupancy 1/2	2	2.24 ± 0.06	2.24 ± 0.06	2.39 ± 0.04	2.39 ± 0.04	13.63 ± 0.02
Occupancy 1/2	4	1.59 ± 0.02	1.59 ± 0.02	1.82 ± 0.03	1.82 ± 0.03	8.51 ± 0.01
Occupancy 1/2	8	1.49 ± 0.03	1.49 ± 0.03	1.69 ± 0.03	1.69 ± 0.03	5.11 ± 0.01
Occupancy 1/3	2	1.62 ± 0.03	1.66 ± 0.03	1.84 ± 0.01	1.60 ± 0.02	4.30 ± 0.01
Occupancy 1/3	4	1.40 ± 0.01	1.36 ± 0.01	1.67 ± 0.00	1.34 ± 0.02	2.94 ± 0.00
Occupancy 1/3	8	1.39 ± 0.01	1.31 ± 0.01	1.65 ± 0.00	1.28 ± 0.01	2.05 ± 0.00

B.4 Broader horizons

Table 3 covers every decade from 10^3 to 10^6 and all three exponents, using $S = \lceil \log \log T / \log(1/\alpha) \rceil$, one more update than the minimal budget in Theorem 3.3(i). The doubling schedule uses its own budget $\lceil \log_2 T \rceil$ and is shown for reference. Weighted has a lower ratio than Equal in 11 of the 12 settings, and its ratio stays at most 1.81, whereas the ratio of fixed cadence grows with T . We additionally extend the exact ridge comparison on both real datasets to $T = 50,000$, holding stream prefixes and features fixed across horizons (Table 4; the MNIST rows are plotted in Figure 1). The budget in each row is determined by the offline fitted exponent, and every compared schedule receives that same budget. The $T = 10,000$ rows coincide with the $S = 2$ rows of Table 2.

B.5 Adaptive updates and change-point detection

We run Algorithm 1 as stated, with $C_1 = 1$. The fixed schedules in Table 5 receive the number of updates chosen by the algorithm. At $T = 10^4$, the doubling floor $2\tau_i$ determines every checkpoint after the second, so all 25 runs use the same ten update times 12, 39, 78, $\dots$, 9984. The exponent

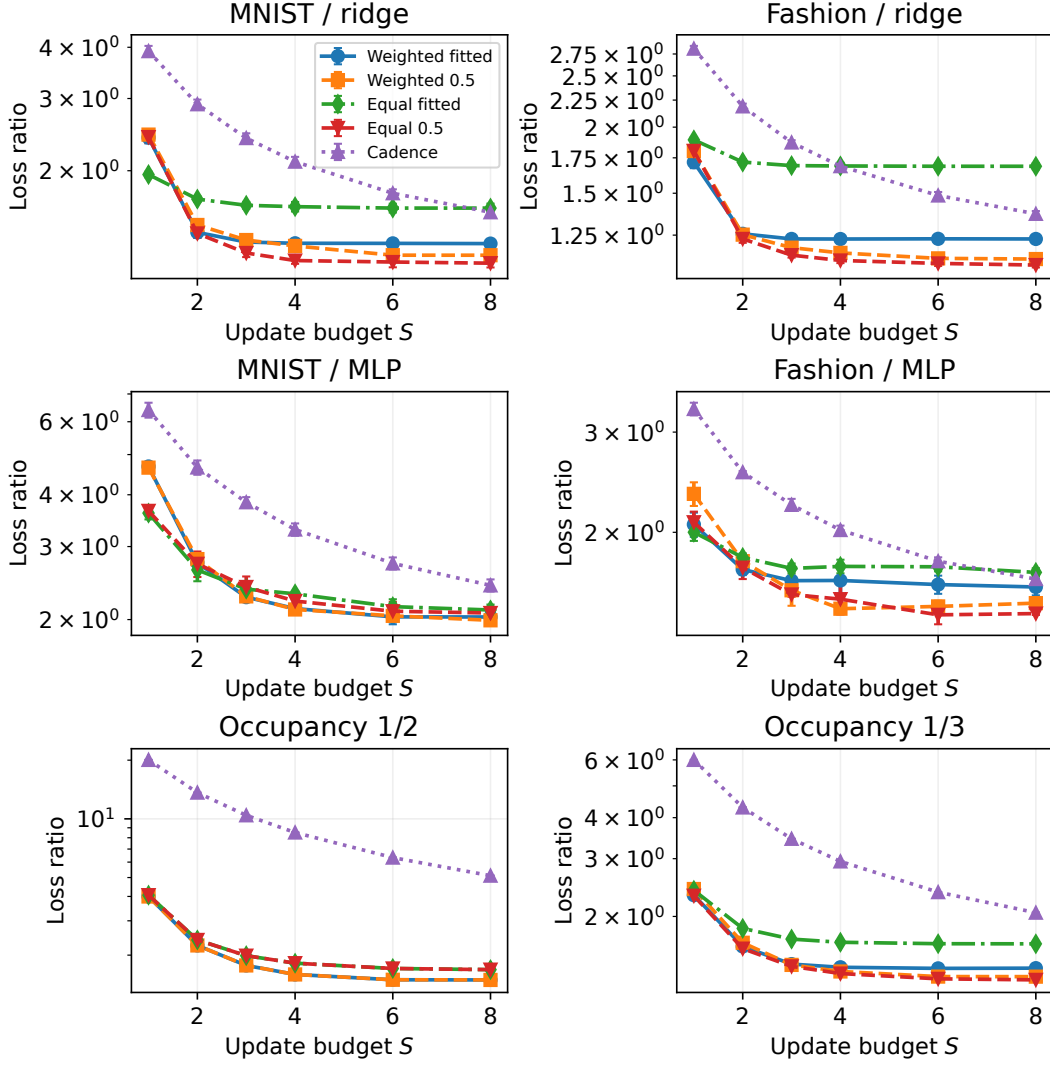


Figure 3: Budget sweeps in six environments (mean and sample standard deviation). The MLP reference is online Adam; the other references use the same oracle.

Table 3: Synthetic horizon scaling; loss ratios (mean $\pm$ sample standard deviation, three seeds). Doubling uses its larger natural budget.

α	T	S	Weighted	Equal	Cadence	Doubling	S_{dbl}
1/3	10^3	2	1.50 ± 0.05	1.60 ± 0.03	2.31 ± 0.01	1.13 ± 0.02	10
1/3	10^4	3	1.45 ± 0.01	1.71 ± 0.01	3.45 ± 0.00	1.12 ± 0.00	14
1/3	10^5	3	1.50 ± 0.00	1.86 ± 0.00	6.67 ± 0.00	1.12 ± 0.00	17
1/3	10^6	3	1.56 ± 0.00	1.99 ± 0.00	13.63 ± 0.00	1.13 ± 0.00	20
1/2	10^3	3	1.54 ± 0.05	1.67 ± 0.08	3.75 ± 0.00	1.21 ± 0.01	10
1/2	10^4	4	1.58 ± 0.01	1.81 ± 0.04	8.51 ± 0.02	1.19 ± 0.01	14
1/2	10^5	4	1.70 ± 0.01	2.03 ± 0.01	25.41 ± 0.02	1.19 ± 0.00	17
1/2	10^6	4	1.81 ± 0.00	2.21 ± 0.00	78.91 ± 0.01	1.20 ± 0.00	20
2/3	10^3	5	1.49 ± 0.09	1.47 ± 0.08	3.74 ± 0.01	1.24 ± 0.03	10
2/3	10^4	6	1.60 ± 0.03	1.70 ± 0.17	12.42 ± 0.03	1.25 ± 0.01	14
2/3	10^5	7	1.65 ± 0.01	1.78 ± 0.02	47.51 ± 0.01	1.26 ± 0.00	17
2/3	10^6	7	1.76 ± 0.01	1.96 ± 0.03	216.47 ± 0.02	1.28 ± 0.00	20

Table 4: Longer real-data horizons: loss ratios to full update with the exact ridge oracle, five seeds. The same budget $S = \lceil \log \log T / \log(1/\hat{\alpha}) \rceil$ is used for every schedule in a row. W denotes Weighted and E denotes Equal.

Dataset	T	S	W fit	W .5	E fit	E .5	Cadence
MNIST	1000	2	1.26 ± 0.08	1.27 ± 0.09	1.32 ± 0.07	1.31 ± 0.11	1.71 ± 0.07
MNIST	10000	2	1.41 ± 0.04	1.47 ± 0.03	1.71 ± 0.03	1.41 ± 0.03	2.90 ± 0.08
MNIST	50000	2	1.30 ± 0.02	1.27 ± 0.02	1.45 ± 0.02	1.42 ± 0.02	3.61 ± 0.03
Fashion-MNIST	1000	2	1.21 ± 0.08	1.21 ± 0.07	1.60 ± 0.07	1.21 ± 0.03	1.64 ± 0.07
Fashion-MNIST	10000	2	1.26 ± 0.02	1.25 ± 0.02	1.72 ± 0.02	1.23 ± 0.02	2.19 ± 0.03
Fashion-MNIST	50000	2	1.14 ± 0.00	1.13 ± 0.01	1.63 ± 0.01	1.20 ± 0.01	2.41 ± 0.02

estimate is first revised at the seventh to tenth update, and never for $\alpha = 2/3$, where it keeps its initial value 0.76. These revisions do not change any update time, so the Adaptive column reflects this checkpoint sequence rather than adaptation of the exponent.

Table 5: Adaptive exponent experiment: loss ratios to full update, five streams at $T = 10^4$. Fixed schedules receive the same number of updates as Algorithm 1. Equal uses the true synthetic or fitted real-data exponent.

Environment	Final $\hat{\alpha}$	Updates	Adaptive	Equal	Cadence
Occupancy 1/3	0.44 ± 0.00	10 ± 0	1.14 ± 0.00	1.65 ± 0.00	1.85 ± 0.00
Occupancy 1/2	0.61 ± 0.00	10 ± 0	1.22 ± 0.01	1.68 ± 0.02	4.35 ± 0.01
Occupancy 2/3	0.76 ± 0.00	10 ± 0	1.30 ± 0.04	1.43 ± 0.02	8.18 ± 0.01
MNIST / ridge	0.57 ± 0.01	10 ± 0	1.09 ± 0.02	1.62 ± 0.04	1.48 ± 0.02
Fashion / ridge	0.50 ± 0.00	10 ± 0	1.05 ± 0.02	1.69 ± 0.02	1.30 ± 0.01

Piecewise-stationary comparison. We use $T = 12,000$, budget $S = 8$, and $K = 3$ label-permutation changes after rounds 3,000, 6,000, and 9,000. The detector is given K but not these locations. Because the ridge oracle and ten-class task do not satisfy the binary realizable assumptions of Theorem 3.8, we evaluate the power-law variant described after that theorem, with $C_0 = 2$ and the fitted $(\hat{C}, \hat{\alpha})$. For an epoch of n observations whose predictor was fitted on N examples, the expected-loss threshold is $n \min\{1, C_0 \hat{C} N^{-\hat{\alpha}}\}$, with N frozen until the next refit, and the positive loss excess is tested with the Bernstein statistic in Algorithm 2. After a detection the buffer starts with the next observation. Integer epoch lengths are rounded down, whereas Algorithm 2 rounds them up, and are at least one. Known CPs is told the change points; within each segment it applies the geometric schedule with $m = \lfloor S/(K+1) \rfloor = 2$ updates and trains on that segment’s data (Proposition 3.7). Window updates at fixed cadence and trains on the last $\lfloor T/(S+1) \rfloor$ examples. i.i.d. geom. ignores the changes and applies the geometric schedule $\lfloor T^{i/(S+1)} \rfloor$ to the full prefix. Full prefix refits after every round on all past examples, including those labeled under earlier permutations, and has no budget. At this calibration, the detector has higher mean loss than the fixed-cadence

Table 6: Cumulative classification errors on label-permuted streams ($T = 12000$, $K = 3$, budget $S = 8$ except full-prefix retraining), five seeds. The detector is the empirical power-law variant described in the text.

Dataset	Known CPs	Detector	Window	Full prefix	i.i.d. geom.
MNIST	3965 ± 95	7329 ± 377	5862 ± 329	7392 ± 716	8454 ± 761
Fashion-MNIST	4099 ± 57	7832 ± 397	6214 ± 278	7590 ± 593	8626 ± 706

window baseline on both datasets and, on Fashion-MNIST, than full-prefix retraining. Because the detection statistic accumulates from the last update, rounds with below-threshold loss before a change delay its detection. Across the ten runs, 18 of the 30 changes are detected, with a median delay of about 1,900 rounds; none of the first changes on Fashion-MNIST is detected. No detection occurs before the first change. The detector makes 4 to 8 updates (means 6.2 on MNIST and 4.8 on

Fashion-MNIST), whereas the other budgeted policies make 8. The supplementary calibration sweep is exploratory, and we do not select a constant on these evaluation streams.

B.6 Compute, reproducibility, and limitations

All experiments use NumPy on an Intel Core Ultra 9 285 CPU, with one BLAS thread per experiment process and 23 GiB of system memory; no GPU is used. Per-group elapsed times are recorded in the logs. The 50,000-round runs took 195 seconds in total, with peak resident memory of 1.2 GiB.

Besides update counts, the code records operation-count proxies, $td^2 + d^3/3$ for a batch ridge fit and $6Et(d_{\text{in}}h + 10h)$ for E MLP passes, which omit feature extraction, prediction, and memory movement. The incremental full-update ridge reference costs $O(Td^2)$ in total rather than that of T batch refits. On the hardware above, it takes 15–19 seconds per 50,000-round stream, whereas fitting and predicting with each $S = 2$ schedule takes at most 0.31 seconds.

The supplementary code contains the download, training, evaluation, verification, and rendering scripts, together with per-seed outputs, logs, commands, and package versions. MNIST is credited to LeCun, Cortes, and Burges, with files obtained from <https://github.com/fgmt/mnist>. Fashion-MNIST is credited to Xiao, Rasul, and Vollgraf and obtained from its MIT-licensed repository, <https://github.com/zalandoresearch/fashion-mnist>. The scripts download the datasets rather than redistributing them.

The experiments use small models, shuffled benchmark images, and synthetic label shifts; they do not cover larger architectures, naturally occurring drift, or the gradual-drift setting of Theorem 3.10. Reduced update frequency may lower validation and deployment overhead, but delayed responses to distribution changes can also prolong errors, so application-specific monitoring remains necessary.

C Proof of Theorem 3.3

Benchmark and scope. For deterministic full-prefix update times, independence of the next example gives the exact expected-risk objective

$$R(\tau) = \tau_1 \bar{L}(0) + \sum_{i=1}^S (\tau_{i+1} - \tau_i) \bar{L}(\tau_i), \quad \tau_{S+1} = T. \quad (7)$$

The full-retraining benchmark is $R_{\text{full}} = \sum_{t=0}^{T-1} \bar{L}(t)$ for the same oracle and initial predictor. A non-increasing expected curve makes this optimal among deterministic full-prefix schedules. It need not make full retraining optimal among all data-dependent choices of predictors or training subsets. Upper bounds for a fixed schedule remain upper bounds on the minimax risk in (2); the lower-bound scope is stated separately in part (iii).

Weighted schedule. Write $a = \alpha$ and use the definitions in (5). Since $\beta W = 1$ and $w_i = a^{n-i}$,

$$r_n = \prod_{i=1}^n (\lambda w_i T^\beta)^{a^{n-i}} = \lambda^W T^{\beta W} \prod_i w_i^{w_i} = T.$$

Also $W \leq (1-a)^{-1}$ and

$$\log \lambda = \log(1/a) \frac{\sum_{k=0}^{n-1} k a^k}{\sum_{k=0}^{n-1} a^k} \leq \frac{a \log(1/a)}{1-a}.$$

For $r_1 \geq 1$, the continuous stepwise objective is bounded by

$$Br_1 + C \sum_{i=2}^n (r_i - r_{i-1}) r_{i-1}^{-a} \leq \lambda [Bw_1 + C(W - w_1)] T^\beta. \quad (8)$$

This is the sharpened construction; the equal-contribution schedule remains a valid comparison but has the weaker $(S+1)T^\beta$ upper bound.

Integer checkpoints, including large budgets. Take the distinct values of $\lceil \max\{1, r_i\} \rceil$, $1 \leq i \leq S$, that are strictly below T , in increasing order. For $S = 0$ make no update. To verify this prescription, let $d_i = \log r_i - \log r_{i-1}$. Then $d_i = a d_{i-1} + \log(1/a)$ for $i \geq 2$. If $d_1 \geq 0$, all checkpoints increase. Otherwise they first decrease below one and then increase; after their first crossing of one they increase strictly. Thus clipping only produces a block of repeated checkpoints at one.

Let $M = \max\{B, C\}$. If $r_1 \geq 1$, the first epoch costs at most $B + M\lambda w_1 T^\beta$. Every subsequent nonempty rounded epoch ending at $\lceil r_i \rceil$ costs at most $2Mr_i r_{i-1}^{-a} = 2M\lambda w_i T^\beta$, since $\lceil r_i \rceil \leq 2r_i$ and the preceding checkpoint is at least r_{i-1} . Repeated rounded checkpoints have empty epochs. If $r_1 < 1$, the first epoch ends at one and costs at most B . Let j be the first index with $r_j \geq 1$. The epoch from one to $\lceil r_j \rceil$ costs at most $2Mr_j \leq 2M\lambda w_j T^\beta$, because $r_{j-1} < 1$. The same bound handles subsequent epochs. The endpoint $r_n = T$ is not an update. If $\lceil r_S \rceil \geq T$, that checkpoint is dropped; the final epoch then ends at $T < 2r_S$ and costs at most $2M\lambda w_S T^\beta$, while the term for $i = n$ is not used. In both cases the total is at most $B + 2M\lambda W T^\beta$, uniformly in S, T .

For $n \geq \lceil \log \log T / \log(1/a) \rceil$ and $T \geq e^2$, $a^n \leq 1/\log T$, so

$$T^{\beta-(1-a)} = \exp\left(\frac{(1-a)a^n \log T}{1-a^n}\right) \leq e^{2(1-a)}.$$

When $\bar{L}(t) \geq ct^{-a}$, the same-oracle full-update benchmark is $\Theta(T^{1-a})$, proving constant competitiveness. Polylogarithmic envelopes give only the corresponding polylogarithmic comparison.

Explicit example. If $B = C = 1$, $a = 1/2$, $\bar{L}(t) = t^{-1/2}$ for $t \geq 1$, and $S = \lceil \log_2 \log T \rceil - 1$, then $W \rightarrow 2$ and $\lambda \rightarrow 2$. Here $r_1 \rightarrow \infty$; the rounding error in (8) is at most $B + CS = O(\log \log T) = o(\sqrt{T})$. Since $R_{\text{full}} = 2\sqrt{T} + O(1)$ and $T^{\beta-1/2} \leq \sqrt{e} + o(1)$,

$$\frac{R(\tau)}{R_{\text{full}}} \leq 2\sqrt{e} + o(1) \simeq 3.30.$$

This is an asymptotic upper bound, not a prediction of the observed ratio.

Lower bound without a budget-dependent constant. Let $t_1 < \dots < t_S < T$ be any deterministic full-prefix schedule, $t_{S+1} = T$, and R its expected risk under the lower envelope in part (iii). The case $S = 0$ is immediate. Otherwise $ct_1 \leq R$ and, for $i \geq 1$, $c(t_{i+1} - t_i)t_i^{-a} \leq R$. At every prediction the training prefix has length at most T , hence $R \geq cT^{1-a}$. In particular, $t_i^{1-a} \leq R/c$, and therefore

$$t_{i+1} \leq t_i + (R/c)t_i^a \leq (2R/c)t_i^a.$$

Starting at $t_0 = 1$ and unrolling this inequality gives $T \leq (2R/c)^{1+a+\dots+a^S} = (2R/c)^W$. Thus $R \geq (c/2)T^\beta$, as claimed. The argument applies conditionally to adaptive choices only under a lower envelope that remains valid for the selected predictors, conditional on the information used to choose them. An unconditional bound on $\bar{L}(t)$ cannot substitute for this assumption.

Logarithmic regime. For $a = 1$, use the distinct rounded checkpoints $\lceil T^{i/(S+1)} \rceil < T$. Each epoch contributes at most a constant multiple of $\max(B, C)T^{1/(S+1)}$, giving part (ii). The preceding lower-bound recurrence holds with $a = 1$ and $W = S + 1$, giving $(c/2)T^{1/(S+1)}$. The additional factor $S + 1$ in this regime is not removed by the power-law argument.

Budget monotonicity and additive loss. The feasible set permits at most S updates, so optimal risk is non-increasing in S . An upper bound obtained from a particular family need not itself be monotone. Constant competitiveness also does not imply a vanishing additive gap. For the exact curve $\bar{L}(t) = t^{-a}$, choose $S = \lfloor \log \log T / \log(1/a) \rfloor - K$ with a sufficiently large fixed K . Then $a^{S+1} \log T \geq a^{1-K}$, so the lower bound above can exceed $R_{\text{full}} \sim T^{1-a}/(1-a)$ by a fixed positive multiple of T^{1-a} . Thus an $\Omega(T^{1-a})$ additive gap is compatible with $S = \Theta(\log \log T)$.

D An Example with a Power-Law Learning Curve

In the next example, we construct a hypothesis class $\mathcal{H}$ based on Haar-type functions whose learning curve satisfies $\bar{L}(t) = \Theta(t^{-\alpha})$, i.e., it has matching power-law upper and *lower* bounds. We do this

for every rational positive $\alpha = \frac{p}{q} \leq 1$, $p, q \in \mathbb{Z}_+$. For any irrational α , one can find a rational α' such that $|\alpha - \alpha'| \leq \frac{1}{\log_2 T}$ and use the hypothesis class constructed for α' . Since $\frac{1}{2} \leq \frac{t^{\alpha'}}{t^\alpha} \leq 2$ for $1 \leq t \leq T$, the resulting bounds hold for α up to a factor of two.

Example 1. Let $I_{l,j} = (\frac{j-1}{2^{ql}}, \frac{j}{2^{ql}}]$, $l \in \mathbb{Z}_+$, $j \in \{1, \dots, 2^{ql}\}$ be 2^{ql} intervals that uniformly split $(0, 1]$. In addition, let $g_{l,j}(x) = \sum_{i=2^{q-1}}^{2^q-1} \mathbf{1}(x \in I_{l+1, 2^q j - i}) - \sum_{i=0}^{2^{q-1}-1} \mathbf{1}(x \in I_{l+1, 2^q j - i})$ for $l \in \mathbb{Z}_+$ and $j \in \{1, \dots, 2^{ql}\}$. For $\sigma = (\sigma_{l,j} : l \in \mathbb{Z}_+, j \in \{1, \dots, 2^{ql}\}) \in \{-1, 1\}^{\mathbb{N}}$, let h_σ be the sequence-valued function whose l -th component is

$$h_{\sigma,l}(x) = c_l \sum_{j=1}^{2^{ql}} \sigma_{l,j} g_{l,j}(x), \quad l \in \mathbb{Z}_+, \quad (9)$$

where $c_l = \frac{1}{2^{\frac{pl}{2}}}$, and define the hypothesis class $\mathcal{H} = \{h_\sigma : \sigma \in \{-1, 1\}^{\mathbb{N}}\}$. Labels are thus sequences, and ℓ is the squared ℓ_2 loss $\ell(y, y') = \sum_{l \geq 1} (y_l - y'_l)^2$, which is at most $4 \sum_l c_l^2 \leq 4$. Let $X_1, \dots, X_T \sim U(0, 1)$ be *i.i.d.* random variables uniformly distributed over $(0, 1)$. For the ground truth function $h \in \mathcal{H}$, let

$$\mathcal{O}(\{X_i, h(X_i)\}_{i=1}^t) = \arg \min_{\hat{h} \in \mathcal{H} : \hat{h}(X_i) = h(X_i), i \in \{1, \dots, t\}} \sup_{h' \in \mathcal{H}} \mathbb{E}_{X \sim U(0,1)} [\ell(\hat{h}(X), h'(X))]$$

return the ERM that minimizes the worst-case risk. If there are multiple such ERMs, $\mathcal{O}(\{X_i, h(X_i)\}_{i=1}^t)$ returns the minimum $(\sigma_{l,j} : l \in \mathbb{Z}_+, j \in \{1, \dots, 2^{ql}\})$ in lexicographic order.

Lemma D.1. *For the hypothesis class $\mathcal{H}$, data distribution $X_i \sim U(0, 1)$, and oracle $\mathcal{O}$ given in Example 1, and the ground truth h_σ with $\sigma \equiv 1$, the learning curve satisfies $2^{-pt-\alpha} \leq \bar{L}(t) \leq C_{p,q} t^{-\alpha}$ for all $t \geq 2$, where $\alpha = p/q$ and $C_{p,q}$ depends only on p and q .*

We start with the following result.

Proposition D.2. *Let $h = h_\sigma \in \mathcal{H}$ and let $W_t = \{(l, j) : I_{l,j} \cap \{X_1, \dots, X_t\} \neq \emptyset\}$ be the set of cells that contain a sample. Then $\mathcal{O}(\{X_i, h(X_i)\}_{i=1}^t) = h_{\hat{\sigma}}$ with*

$$\hat{\sigma}_{l,j} = \sigma_{l,j} \text{ for } (l, j) \in W_t, \quad \hat{\sigma}_{l,j} = -1 \text{ for } (l, j) \notin W_t, \quad (10)$$

and

$$\mathbb{E}_{X \sim U(0,1)} [\ell(\mathcal{O}(\{X_i, h(X_i)\}_{i=1}^t)(X), h(X))] = \sum_{l=1}^{\infty} \sum_{j=1}^{2^{ql}} \mathbf{1}(I_{l,j} \cap \{X_i\}_{i=1}^t = \emptyset) \frac{c_l^2}{2^{ql}} (\sigma_{l,j} + 1)^2. \quad (11)$$

Proof. Let $\hat{h} = h_{\hat{\sigma}} \in \mathcal{H}$ satisfy $\hat{h}(X_i) = h(X_i)$ for $i \leq t$. For $x \in I_{l,j}$, the only nonzero term of the l -th component $h_{\sigma,l}(x)$ is $c_l \sigma_{l,j} g_{l,j}(x)$, and $g_{l,j}(x) \in \{-1, 1\}$ is determined by x . Hence the l -th component of the label $h(X_i)$ reveals $\sigma_{l,j}$ for the level- l cell $I_{l,j}$ containing X_i . Consequently $\hat{\sigma}_{l,j} = \sigma_{l,j}$ for all $(l, j) \in W_t$, while the coordinates $(l, j) \notin W_t$ are unconstrained. Since the cells of each level are disjoint and $|I_{l,j}| = 2^{-ql}$,

$$\mathbb{E}_{X \sim U(0,1)} [\ell(\hat{h}(X), h(X))] = \sum_{l=1}^{\infty} \sum_{j=1}^{2^{ql}} \frac{c_l^2}{2^{ql}} (\hat{\sigma}_{l,j} - \sigma_{l,j})^2 = \sum_{(l,j) \notin W_t} \frac{c_l^2}{2^{ql}} (\hat{\sigma}_{l,j} - \sigma_{l,j})^2. \quad (12)$$

Every consistent $\hat{h}$ has the same worst-case risk, since the worst-case h' flips the sign of every coordinate of $\hat{\sigma}$. The tie-breaking rule therefore chooses $\hat{\sigma}_{l,j} = -1$ for $(l, j) \notin W_t$, which gives (10). Substituting it into (12) gives (11). $\square$

Proof of Lemma D.1. Take $\sigma \equiv 1$ in (11). Each of the 2^{ql} cells of level l contains no sample with probability $(1 - 2^{-ql})^t$, so

$$\bar{L}(t) = \sum_{l=1}^{\infty} \frac{4}{2^{pl}} \left(1 - \frac{1}{2^{ql}}\right)^t. \quad (13)$$

Let $l^* \geq 1$ be the smallest integer with $2^{ql^*} \geq t$. Then $2^{-pl^*} \leq t^{-\alpha}$, and $2^{q(l^*-1)} < t$ because $t \geq 2$. The levels $l \geq l^*$ contribute at most $\sum_{l \geq l^*} 4 \cdot 2^{-pl} \leq 8 \cdot 2^{-pl^*} \leq 8t^{-\alpha}$. For $l = l^* - k$ with $k \geq 1$, we have $2^{ql} < t \cdot 2^{-q(k-1)}$ and $2^{-pl} \leq t^{-\alpha} 2^{pk}$, so

$$\frac{4}{2^{pl}} \left(1 - \frac{1}{2^{ql}}\right)^t \leq \frac{4}{2^{pl}} \exp\left(-\frac{t}{2^{ql}}\right) \leq 4t^{-\alpha} 2^{pk} \exp(-2^{q(k-1)}).$$

Summing over k gives $\bar{L}(t) \leq C_{p,q} t^{-\alpha}$ with $C_{p,q} = 8 + 4 \sum_{k \geq 1} 2^{pk} \exp(-2^{q(k-1)}) < \infty$. For the lower bound, $(1 - 2^{-ql})^t \geq (1 - 1/t)^t \geq 1/4$ for $l \geq l^*$. Hence

$$\bar{L}(t) \geq \sum_{l \geq l^*} 2^{-pl} \geq 2^{-pl^*} > 2^{-p} t^{-\alpha},$$

where the last step uses $2^{ql^*} < 2^q t$. $\square$

E Proof of Theorem 3.5

Proof. Fix T and $1 \leq S \leq c \log T / \log \log T$; we construct the curve $\bar{L} = \bar{L}_{T,S}$. Let $\tilde{\alpha}_i = \alpha_1 + \frac{i(\alpha_2 - \alpha_1)}{S+1}$ and $u_i = \frac{1-\alpha_2}{1-\tilde{\alpha}_i}$ for $i \in \{0, \dots, S+1\}$. Let $T_{1,i} = T^{u_i}$ for $i \in \{1, \dots, S+1\}$ and $T_{1,0} = 1$; thus $T_{1,S+1} = T$. Since $(1 - \tilde{\alpha}_i)(1 - \tilde{\alpha}_{i+1}) \leq (1 - \alpha_1)^2$,

$$u_{i+1} - u_i = \frac{(1 - \alpha_2)(\tilde{\alpha}_{i+1} - \tilde{\alpha}_i)}{(1 - \tilde{\alpha}_i)(1 - \tilde{\alpha}_{i+1})} \geq \frac{c_*}{S+1}, \quad c_* := \frac{(1 - \alpha_2)(\alpha_2 - \alpha_1)}{(1 - \alpha_1)^2}. \quad (14)$$

Let $(T_{2,i}, \frac{1}{T_{2,i}})$, $i \in \{0, \dots, S\}$, be the intersection of the tangent line of the curve $f_i(x) = \frac{1}{x^{\tilde{\alpha}_i}}$ at point $(T_{1,i}, \frac{1}{T_{1,i}})$ and the curve $f_{i+1}(x) = \frac{1}{x^{\tilde{\alpha}_{i+1}}}$ such that $T_{2,i} > T_{1,i}$, i.e., we have

$$\frac{-\tilde{\alpha}_i}{T_{1,i}^{\tilde{\alpha}_i+1}}(T_{2,i} - T_{1,i}) + \frac{1}{T_{1,i}^{\tilde{\alpha}_i}} = \frac{1}{T_{2,i}^{\tilde{\alpha}_{i+1}}}. \quad (15)$$

The tangent line vanishes at $\frac{1+\tilde{\alpha}_i}{\tilde{\alpha}_i} T_{1,i}$, so (15) gives $T_{2,i} < AT_{1,i}$ with $A := \frac{1+\alpha_1}{\alpha_1}$. By (14), $T_{1,i+1}/T_{1,i} \geq T^{c_*/(S+1)}$ for $i \geq 1$, and $T_{1,1} \geq T^{1-\alpha_2}$. Since $S+1 \leq 2c \log T / \log \log T$, we have $T^{c_*/(S+1)} \geq (\log T)^{c_*/(2c)}$. Hence, for T large enough, $T_{1,i+1} \geq AT_{1,i} > T_{2,i}$ for all $i \in \{0, \dots, S\}$. Let $\bar{L}(0) = 1$ and

$$\bar{L}(t) = \begin{cases} \frac{-\tilde{\alpha}_i}{T_{1,i}^{\tilde{\alpha}_i+1}}(t - T_{1,i}) + \frac{1}{T_{1,i}^{\tilde{\alpha}_i}} & T_{1,i} \leq t \leq T_{2,i} \text{ for some } i \in \{0, \dots, S\}, \\ \frac{1}{t^{\tilde{\alpha}_{i+1}}} & T_{2,i} \leq t \leq T_{1,i+1} \text{ for some } i \in \{0, \dots, S\}. \end{cases} \quad (16)$$

This function is continuous and non-increasing. Since f_i is convex, its tangent lies below f_i , and it lies above f_{i+1} on $[T_{1,i}, T_{2,i}]$. Hence $t^{-\alpha_2} \leq \bar{L}(t) \leq t^{-\alpha_1}$ for $t \geq 1$, and $\bar{L}(t) \leq t^{-\tilde{\alpha}_i}$ for $t \in [T_{2,i-1}, T_{2,i}]$, where $T_{2,-1} = 1$ and $T_{2,S+1} = T$.

Full-update risk. Using $\sum_{1 \leq t \leq x} t^{-a} \leq x^{1-a}/(1-a)$, $T_{2,i} \leq AT_{1,i}$, and $T_{1,i}^{1-\tilde{\alpha}_i} = T^{1-\alpha_2}$ for $i \geq 1$,

$$\inf_{\Phi} \text{Risk}_{T,T}(\Phi, \mu) = \sum_{t=0}^{T-1} \bar{L}(t) \leq 1 + \sum_{i=0}^{S+1} \frac{T_{2,i}^{1-\tilde{\alpha}_i}}{1 - \tilde{\alpha}_i} \leq 1 + \frac{A(S+2)T^{1-\alpha_2}}{1 - \alpha_2}.$$

Risk with S updates. Consider any full-prefix schedule fixed in advance with at most S updates. The $S+1$ open intervals $(T_{1,i}, T_{1,i+1})$, $i \in \{0, \dots, S\}$, are disjoint, so one of them contains no update. For this i , every round $t \in (T_{1,i}, T_{1,i+1}]$ uses a predictor trained on at most $T_{1,i}$ samples, whose expected loss is at least $\bar{L}(T_{1,i})$. If $i \geq 1$, then by (14) the risk is at least

$$(T_{1,i+1} - T_{1,i} - 1) T_{1,i}^{-\tilde{\alpha}_i} \geq T^{1-\alpha_2} (T^{c_*/(S+1)} - 1) - 1.$$

If $i = 0$, it is at least $(T_{1,1} - 1)\bar{L}(1) = T^{u_1} - 1$, where $u_1 \geq u_0 + \frac{c_*}{S+1} \geq 1 - \alpha_2 + \frac{c_*}{S+1}$. In both cases the risk is at least $\frac{1}{4} T^{1-\alpha_2+c_*/(S+1)}$ for T large enough. Therefore its ratio to the full-update risk is at least

$$\frac{T^{c_*/(S+1)}}{4(1 + A(S+2)/(1 - \alpha_2))} \geq T^{\frac{c_*}{2(S+1)}},$$

where the last step uses $T^{c_*/(2(S+1))} \geq (\log T)^{c_*/(4c)}$ and holds for large T whenever $c < c_*/4$. This proves the bound $T^{\Omega(1/S)}$. When $S \leq o(\frac{\log T}{\log \log T})$, the ratio $T^{c_*/(2(S+1))}$ grows faster than any power of $\log T$. For adaptive full-prefix rules, the same argument applies to the realized schedule when the conditional expected loss of each selected predictor is at least $\bar{L}$ evaluated at its training-set size. $\square$

F Proof of Theorem 3.6

Throughout this section, $\log$ is the natural logarithm, T is sufficiently large, and losses lie in $[0, 1]$. Epoch 0 consists of rounds $1, \dots, \tau_1$, which use $\hat{h}_0$. For $i \geq 1$, epoch i consists of rounds $\tau_i + 1, \dots, \tau_{i+1}$, which use $\hat{h}_{\tau_i}$. Let $n_i = \tau_{i+1} - \tau_i$, let $\mu_i = L(\tau_i)$ be the realized risk of $\hat{h}_{\tau_i}$, and let $\hat{L}_i$ be the average loss in epoch i , i.e., the value $\hat{L}(\tau_{i+1})$ used in lines 7–8 of Algorithm 1. The time τ_{i+1} is fixed at time τ_i . We start with the following well known large deviation result.

Lemma F.1. *Fix $i \geq 1$, and condition on the first τ_i samples and on τ_{i+1} . Then the n_i losses of epoch i are i.i.d. with mean μ_i and variance at most μ_i , and for every $\epsilon > 0$,*

$$\Pr(\pm (n_i \hat{L}_i - n_i \mu_i) \geq \epsilon) \leq \exp\left(-\frac{\epsilon^2}{2n_i \mu_i + \frac{2\epsilon}{3}}\right). \quad (17)$$

Proof. (of Theorem 3.6) Write $\hat{\alpha}_0 = 1 - \frac{\log \log T}{\log T} \geq \alpha$ for the initial estimate and $\hat{\alpha}_i$ for the estimate used to compute τ_{i+1} . By line 9, every estimate lies in $[1/\log T, \hat{\alpha}_0] \subset (0, 1)$, so $\hat{S}$ and $\hat{\beta}$ are well defined and $\hat{\alpha}^{\hat{S}} = 1/\log T$. The first update is at $\tau_1 = \lfloor T^{\hat{\beta}} \rfloor$, where $T^{\hat{\beta}} = (\log T)^{\log T / (\log T - 1)} \in [\log T, e \log T]$; hence $2 \leq \tau_1 \leq e \log T$ and $\tau_1 \geq \frac{1}{2} \log T$. For $i \geq 1$,

$$\tau_{i+1} = \max\{\lfloor 2^{-i} \tau_i^{\hat{\alpha}_i} T^{\hat{\beta}_i} \rfloor, 2\tau_i\} \geq 2\tau_i. \quad (18)$$

Hence $\tau_i \geq 2^{i-1} \tau_1$, and the algorithm makes at most $\log_2 T + 1$ updates.

Step 1: a good event. Let G_0 be the event in the assumption of the theorem. For $i \geq 1$, let E_i be the event that (a) if the test in line 7 fires at the end of epoch i , then $\mu_i/2 \leq \hat{L}_i \leq 3\mu_i/2$, and (b) if $n_i \mu_i \geq 56 \log T$, then the test fires. Write $m = n_i \mu_i$ and $Z = n_i \hat{L}_i$; the test fires if and only if $Z \geq 28 \log T$. If $m \geq 28 \log T$, Lemma F.1 with $\epsilon = m/2$ gives $\Pr(|Z - m| \geq m/2) \leq 2 \exp(-3m/28) \leq 2T^{-3}$, and outside this event both (a) and (b) hold. If $m < 28 \log T$, then (b) holds trivially, and (a) can fail only if $Z - m \geq \epsilon_0 := \max\{28 \log T - m, m/2\}$. If $m \leq 14 \log T$, then $\epsilon_0 \geq 14 \log T \geq m$ and $\epsilon_0^2/(2m + 2\epsilon_0/3) \geq 3\epsilon_0/8 \geq 5 \log T$. Otherwise $\epsilon_0 \geq m/2 \geq 7 \log T$ and $\epsilon_0^2/(2m + 2\epsilon_0/3) \geq 3\epsilon_0/14 \geq \frac{3}{2} \log T$. Hence $\Pr(E_i^c) \leq 2T^{-3/2}$ in all cases. Let $G = G_0 \cap \bigcap_{i \leq \log_2 T + 1} E_i$. By a union bound, $\Pr(G^c) \leq \frac{1}{T} + 2(\log_2 T + 1)T^{-3/2} \leq \frac{2}{T}$.

Step 2: bracketing the estimate. On G , every estimate satisfies $\hat{\alpha} \geq \alpha$. This holds initially. If the test fires at the end of epoch i , then (a) and G_0 give $\frac{2}{3} \hat{L}_i \leq \mu_i \leq C_1 \log \tau_i \cdot \tau_i^{-\alpha}$, so α satisfies the constraint in line 8 and $\alpha_{\max} \geq \alpha$. Moreover, $\hat{L}_i \geq \mu_i/2 \geq C_2 \tau_i^{-\alpha}/2$, so

$$\alpha \leq \alpha_{\max} = \frac{\log(3C_1 \log \tau_i / (2\hat{L}_i))}{\log \tau_i} \leq \alpha + \delta(\tau_i), \quad \delta(\tau) := \frac{\log \log \tau + \log(\frac{3C_1}{C_2})}{\log \tau}. \quad (19)$$

Since $\alpha \leq \hat{\alpha}_0$ and, for large T , $\alpha \geq 1/\log T$ by the range of α , the clipping in line 9 preserves this bracket: the new estimate satisfies $\alpha \leq \hat{\alpha} \leq \alpha + \delta(\tau_i)$. This is the bracket stated in the proof sketch.

Step 3: the risk. Conditionally on the past, the expected loss of a round in epoch i is μ_i , where $\mu_0 \leq 1$. Hence

$$\mathbb{E}[\text{Risk}] = \mathbb{E}\left[\sum_{i \geq 0} n_i \mu_i\right] \leq \mathbb{E}\left[\mathbf{1}_G \sum_{i \geq 0} n_i \mu_i\right] + T \Pr(G^c) \leq \mathbb{E}\left[\mathbf{1}_G \sum_{i \geq 0} n_i \mu_i\right] + 2.$$

On G , $n_0 \mu_0 \leq \tau_1 \leq e \log T$. For $i \geq 1$, $n_i \mu_i \leq C_1 \log T \cdot \tau_{i+1} \tau_i^{-\alpha}$, where for the last epoch $\tau_{i+1} \geq T$ denotes the value given by (18). Since $\hat{\alpha}_i \geq \alpha$, $\tau_i \leq T$, and $\hat{\alpha}_i^{\hat{S}_i} = 1/\log T$,

$$2^{-i} \tau_i^{\hat{\alpha}_i} T^{\hat{\beta}_i} \tau_i^{-\alpha} \leq 2^{-i} T^{\hat{\alpha}_i - \alpha + \hat{\beta}_i} = 2^{-i} T^{1-\alpha} T^{\frac{1-\hat{\alpha}_i}{\log T - 1}} \leq e^2 2^{-i} T^{1-\alpha}.$$

Since rounding down only decreases τ_{i+1} , we obtain $\tau_{i+1}\tau_i^{-\alpha} \leq e^2 2^{-i} T^{1-\alpha} + 2\tau_i^{1-\alpha}$. By (18), $\tau_i \leq 2^{-(S-i)}T$ for $1 \leq i \leq S$, so $\sum_{i=1}^S \tau_i^{1-\alpha} \leq T^{1-\alpha}/(1 - 2^{-(1-\alpha)}) \leq 3T^{1-\alpha}/(1 - \alpha)$. Using $1 - \alpha \geq \log \log T / \log T$, we obtain on G

$$\sum_{i \geq 0} n_i \mu_i \leq e \log T + C_1 \log T \left(e^2 + \frac{6 \log T}{\log \log T} \right) T^{1-\alpha} = \tilde{O}(T^{1-\alpha}).$$

Step 4: the number of updates. We work on G and let $y_i = \log_2(T/\tau_i) \geq 0$ for $i \leq S$. Since $0 < \hat{\alpha}_i < 1$, we have $\hat{\beta}_i \geq 1 - \hat{\alpha}_i$. Since $\lfloor x \rfloor \geq x/2$ for $x \geq 1$, (18) gives $\log_2 \tau_{i+1} \geq -i - 1 + \hat{\alpha}_i \log_2 \tau_i + (1 - \hat{\alpha}_i) \log_2 T$, and hence

$$y_{i+1} \leq \min\{\hat{\alpha}_i y_i + i + 1, y_i - 1\}. \quad (20)$$

Let $a = \sqrt{\alpha}$, so that $1 - a \geq (1 - \alpha)/2$ and $\log(1/a) = \frac{1}{2} \log(1/\alpha)$. Let $\tau_A = (56 \log T / C_2)^{1/(1-\alpha)}$. Since $\log(3C_1/C_2) \geq \log 3 > 1$, δ is decreasing on $[e, \infty)$. Let $\tau_B \geq e$ be the smallest value with $\delta(\tau_B) \leq a - \alpha$. Let i_1 be the first index with $\tau_{i_1} \geq \max\{\tau_A, \tau_B\}$. If epoch i_1 ends before T , then $n_{i_1} \mu_{i_1} \geq \tau_{i_1} C_2 \tau_{i_1}^{-\alpha} \geq 56 \log T$, so by (b) the test fires at its end and, by (19), sets $\hat{\alpha} \leq \alpha + \delta(\tau_{i_1}) \leq a$. Every later firing also occurs at some $\tau_i \geq \tau_{i_1}$ and sets $\hat{\alpha} \leq a$. Hence $\alpha \leq \hat{\alpha}_i \leq a$ for all $i \geq i_0 := i_1 + 1$. Iterating (20) from i_0 , with $y_{i_0} \leq \log_2 T$,

$$y_{i_0+k} \leq a^k \log_2 T + \sum_{m=0}^{k-1} a^{k-1-m} (i_0 + m + 1) \leq a^k \log_2 T + \frac{i_0 + k + 1}{1 - a}.$$

For $k_1 = \lceil 2 \log \log_2 T / \log(1/\alpha) \rceil$, we have $a^{k_1} \log_2 T \leq 1$. Since y_i then decreases by at least one per update, at most $2 + (i_0 + k_1 + 1)/(1 - a)$ further updates occur. If the algorithm stops before index i_0 , the same bound holds trivially. Therefore

$$S \leq i_0 + k_1 + 2 + \frac{i_0 + k_1 + 1}{1 - a} \leq \left(1 + \frac{2}{1 - \alpha}\right)(i_0 + k_1 + 1) + 2.$$

Since $\tau_i \geq 2^{i-1} \tau_1$ and $\tau_1 \geq \frac{1}{2} \log T$, we have $i_0 \leq 4 + \log_2^+(\tau_A / \log T) + \log_2^+ \tau_B$, and

$$\log_2 \frac{\tau_A}{\log T} \leq \frac{\alpha}{1 - \alpha} \log_2 \log T + \frac{\log_2^+(56/C_2)}{1 - \alpha}.$$

Let $\eta = a - \alpha = \sqrt{\alpha}(1 - \sqrt{\alpha})$ and $c = \log(3C_1/C_2)$. Every τ with $\log \tau \geq \frac{2}{\eta} (\log \frac{2}{\eta} + c)$ satisfies $\delta(\tau) \leq \eta$, so $\log \tau_B \leq \max\{1, \frac{2}{\eta} (\log \frac{2}{\eta} + c)\}$.

If $\alpha \leq \frac{1}{2}$, then $\frac{1}{1-\alpha} \leq 2$ and $\eta \geq 0.29\sqrt{\alpha}$. Because $\alpha \log(1/\alpha) \leq 1/e$, the first term above is $O(\log \log T / \log(1/\alpha))$. Moreover, $\log \tau_B = O(\alpha^{-1/2}(1 + \log \frac{1}{\alpha})) = o(\log \log T / \log(1/\alpha))$ because $\alpha = \Omega(\log \log \log T / \log \log T)$. Since also $k_1 = O(\log \log T / \log(1/\alpha))$, we obtain $S = O(\log \log T / \log(1/\alpha))$.

If $\alpha > \frac{1}{2}$, then $\log(1/\alpha) \geq 1 - \alpha$ and $\eta \geq (1 - \alpha)/4$. Using $1 - \alpha \geq \log \log T / \log T$, we get $k_1, \log_2(\tau_A / \log T)$, and $\log \tau_B \leq \frac{8}{1-\alpha} (\log \frac{8}{1-\alpha} + c)$ are all $O(\log \log T / (1 - \alpha))$. Hence $S = O(\log \log T / (1 - \alpha)^2)$.

In both cases $S = O(\frac{\log \log T}{(1-\alpha) \log(1/\alpha)})$, and $S \leq \log_2 T + 1$ always. All statements hold on G , whose probability is at least $1 - \frac{2}{T}$. $\square$

G Proof of Theorem 3.8

Proof. Write $m = \lfloor S/(K+1) \rfloor$ and $N = T/(K+1)$. If $S \leq K$, then $m = 0$ and the claimed bound is of order $(S + K + 1)N \geq T$, so it holds trivially because each loss is at most one. In the following, we assume $S \geq K + 1$.

We show that Algorithm 2 achieves risk $\tilde{O}((S + K + 1)N^{\frac{1}{m+1}})$. Let $\hat{\kappa}_1 < \dots < \hat{\kappa}_k$ be the times when the if condition in line 6 in Algorithm 2 is satisfied and let $\hat{\kappa}_0 = 0$. Let E_ℓ be the event that there exists a distribution change at some time $\kappa \in \{\hat{\kappa}_{\ell-1} + 1, \dots, \hat{\kappa}_\ell\}$, and let G_0 be the event in the assumption of the theorem.

False detections. Index the epochs of the test statistic by the updates: epoch $i \in \{0, \dots, S\}$ starts at the i -th update time τ_i (with $\tau_0 = 0$) and uses the fixed predictor $\hat{h}_{\tau_i}$, and the statistic in line 5 sums its losses from round $\tau_i + 1$. A detection resets j to 0 and sets $d = 1$, which blocks the test until the next update. Hence, whenever the test can fire during epoch i , the rate used in line 5 is the one determined at time τ_i . Fix i and an endpoint $r \leq T$, and condition on the history up to the stopping time τ_i . The losses of $\hat{h}_{\tau_i}$ at rounds $\tau_i + 1, \dots, \tau_i + r$ are then independent with fixed means. Let $A_{i,r}$ be the event that the test fires at time $\tau_i + r$ while epoch i is current and all of these means are at most the rate, and let V be the rate times r . Since $\epsilon \mapsto \epsilon^2/(2V + 2\epsilon/3)$ is increasing, the test fires only if the loss sum exceeds V by the smallest ϵ with $\epsilon^2/(2V + 2\epsilon/3) \geq 2 \ln T$. Bernstein's inequality, as in Lemma F.1, therefore gives $\Pr(A_{i,r}) \leq T^{-2}$. A union bound over at most $S + 1$ epochs and T endpoints gives $\Pr(\bigcup_{i,r} A_{i,r}) \leq (S + 1)/T$.

Now suppose that the test fires at a time t in epoch i , that no change occurs between the last detection and t , and that j updates have occurred since that detection (or since time 0). If $j = 0$, the rate used in line 5 is $C_0 \log T \geq 1$, so the test cannot fire. If $j \geq 1$, the predictor $\hat{h}_{\tau_i}$ was trained on at least $N^{\frac{j}{m+1}}$ samples from the current segment, because the epoch lengths are $\lceil N^{\frac{u}{m+1}} \rceil$. On G_0 , its realized loss on that segment is therefore at most $C_0 \log T / N^{\frac{j}{m+1}}$, the rate used in line 5, so $A_{i,t-\tau_i}$ occurs. Hence the event $G = G_0 \cap \bigcap_{\ell} E_{\ell}$ satisfies $\Pr(G^c) \leq \Pr(G_0^c) + (S + 1)/T \leq (S + 2)/T$. Let Φ_1 be the update rule given in Algorithm 2. Since each loss is at most one,

$$\begin{aligned} \text{Risk}_{S,T}(\Phi_1, \mu) &\leq \mathbb{E} [\text{Risk}_{S,T}(\Phi_1, \{(X_t, Y_t)\}_{t=1}^T \mathbf{1}_G)] + T \Pr(G^c) \\ &\leq \mathbb{E} [\text{Risk}_{S,T}(\Phi_1, \{(X_t, Y_t)\}_{t=1}^T \mathbf{1}_G)] + S + 2. \end{aligned} \quad (21)$$

Risk on G . On G , every detection is preceded by a change after the previous detection, so $k \leq K$. By the condition in line 8, there are at most m model updates between $\hat{\kappa}_{\ell-1}$ and $\hat{\kappa}_{\ell}$, $\ell \in \{1, \dots, k+1\}$, where $\hat{\kappa}_{k+1} = T$. Therefore, there are at most $m(K + 1) \leq S$ model updates in total. Denote

$$\tau_{\ell,1}, \dots, \tau_{\ell,j_{\ell}} = \{\tau_i : i \in \{1, \dots, S\}, \hat{\kappa}_{\ell-1} < \tau_i \leq \hat{\kappa}_{\ell}\}$$

as the collection of model update times between $\hat{\kappa}_{\ell-1}$ and $\hat{\kappa}_{\ell}$, and let $\tau_{\ell,0} = \hat{\kappa}_{\ell-1}$ and $\tau_{\ell,j_{\ell}+1} = \hat{\kappa}_{\ell}$. During the epoch $(\tau_{\ell,j-1}, \tau_{\ell,j}]$, line 5 uses the rate $r_j = C_0 \log T / N^{\frac{j-1}{m+1}}$. The test in line 6 does not fire before the last round of the epoch, so $\epsilon(t)^2 < 2 \ln T (2V + 2\epsilon(t)/3)$ with $V = r_j(t - \tau_{\ell,j-1})$ at $t = \tau_{\ell,j} - 1$ (for $j = 1$, the rate $r_1 = C_0 \log T$ is at least one, so the bound below is immediate). Solving this quadratic inequality and adding one for the last round, the loss of the epoch is at most

$$V_{\ell,j} + 2\sqrt{V_{\ell,j} \ln T} + \frac{4}{3} \ln T + 1 \leq 2V_{\ell,j} + \frac{10}{3} \ln T + 1, \quad V_{\ell,j} = r_j(\tau_{\ell,j} - \tau_{\ell,j-1}).$$

If the epoch ends at or before its scheduled update, its length is at most $\lceil N^{\frac{j}{m+1}} \rceil \leq 2N^{\frac{j}{m+1}}$, so $V_{\ell,j} \leq 2C_0 \log T N^{\frac{1}{m+1}}$. Otherwise $j = m + 1$, no further update is scheduled, and these epochs have total length at most $T = (K + 1)N$; hence their values $V_{\ell,j}$ sum to at most $C_0 \log T (K + 1)N^{\frac{1}{m+1}}$. There are at most $(m + 1)(K + 1) \leq S + K + 1$ epochs. Summing the bounds, the realized risk on G is

$$O((S + K + 1) \log T N^{\frac{1}{m+1}}),$$

which, together with (21) and $S + 2 = O(S + K + 1)$, completes the proof of the upper bound for $\text{Risk}_{S,T}(\mathbf{P})$.

Finally, we show that this bound cannot be improved by any schedule fixed in advance, up to the rounding of $S/(K + 1)$ and logarithmic factors. Let $\kappa_i = iN + 1$ for $i \in \{1, \dots, K\}$, so that every segment has length N . Suppose that, for some constant $c > 0$, every predictor whose training data contain n samples from the current segment has expected loss at least $c/(n + 1)$ on that segment. Fix segment k and its update times $\tau_{k,1} < \dots < \tau_{k,j_k}$, and let $x_j = \tau_{k,j} - \kappa_{k-1} + 2$, with $x_0 = 1$ and $x_{j_k+1} = N + 1$. The predictor used between the $(j - 1)$ -th and j -th of these times has at most $x_{j-1} - 1$ samples from the segment. Hence, by the AM-GM inequality applied to the ratios x_j/x_{j-1} , whose product is $N + 1$, the risk incurred in segment k is at least

$$c \sum_{j=1}^{j_k+1} \frac{x_j - x_{j-1}}{x_{j-1}} \geq c(j_k + 1)(N^{\frac{1}{j_k+1}} - 1).$$

The function $g(y) = yN^{1/y}$ is convex on $(0, \infty)$ and decreasing on $(0, \ln N]$. Since $\sum_k (j_k + 1) \leq S + K + 1$, Jensen's inequality shows that, whenever $S + K + 1 \leq (K + 1) \ln N$, the total risk of the schedule is at least

$$c \sum_{k=1}^{K+1} (g(j_k + 1) - (j_k + 1)) \geq c(S + K + 1)(N^{\frac{K+1}{S+K+1}} - 1).$$

When $K + 1$ divides S , the exponent equals $\frac{1}{m+1}$, so this lower bound matches the upper bound up to logarithmic factors. The same argument applies to adaptive rules under a conditional lower envelope, as in Theorem 3.3(iii). $\square$

H Proof of Theorem 3.10

Throughout this section, $\mathcal{H}$ has VC dimension d , every μ_t is realizable by $\mathcal{H}$ (some $h_t^* \in \mathcal{H}$ satisfies $h_t^*(X) = Y$ almost surely under μ_t), the loss is 0–1, and $\text{err}_\mu(h) = \Pr_{(X,Y) \sim \mu}(h(X) \neq Y)$. Let $C_d = 20d + 70$ and $T \geq 5$. We first show that an empirical risk minimizer trained on a window of consecutive samples behaves almost as in the i.i.d. case.

Lemma H.1 (Drifting windows). *With probability at least $1 - 2T^{-2}$, the following holds simultaneously for all integers $1 \leq a < b \leq c \leq T$: every empirical risk minimizer $\hat{h}$ for the samples (X_s, Y_s) , $a \leq s < b$, satisfies*

$$\text{err}_{\mu_c}(\hat{h}) \leq 8\gamma(c - a) + \frac{C_d \log T}{b - a}.$$

Proof. Fix a, b, c , and let $n = b - a$, $D = c - a$, and $\delta = T^{-5}$. For $a \leq s < b$, $\|\mu_s - \mu_c\|_{\text{TV}} \leq \gamma(c - s) \leq \gamma D$. Couple each sample $Z_s = (X_s, Y_s)$ with $\tilde{Z}_s \sim \mu_c$ by a maximal coupling, so that $\Pr(Z_s \neq \tilde{Z}_s) \leq \gamma D$, independently over s . The number m of indices with $Z_s \neq \tilde{Z}_s$ is a sum of independent Bernoulli variables with total mean at most $n\gamma D$, so Bernstein's inequality gives $m < 2n\gamma D + 2\log(1/\delta)$ with probability at least $1 - \delta$. The target h_c^* makes no error on the samples $\tilde{Z}_s$, hence at most m errors on the samples Z_s . Therefore every empirical risk minimizer $\hat{h}$ makes at most m errors on the Z_s and at most $2m$ errors on the $\tilde{Z}_s$. The samples $\tilde{Z}_s$ are i.i.d. from μ_c . By the relative-deviation bound of Boucheron et al. (2005, Section 5.1), together with the bound $(2n + 1)^d$ on the growth function of $\mathcal{H}$, with probability at least $1 - \delta$ every $h \in \mathcal{H}$ satisfies

$$\text{err}_{\mu_c}(h) \leq 2\hat{e}(h) + \frac{5(2d \log(n + 1) + \log(4/\delta))}{n},$$

where $\hat{e}(h)$ is the error rate of h on the $\tilde{Z}_s$. Combining the two events, with probability at least $1 - 2\delta$,

$$\text{err}_{\mu_c}(\hat{h}) \leq \frac{4m}{n} + \frac{5(2d \log(n + 1) + \log(4/\delta))}{n} \leq 8\gamma D + \frac{C_d \log T}{n}.$$

A union bound over at most T^3 triples (a, b, c) completes the proof. $\square$

Proof of Theorem 3.10. Let $N_0 = \min\{T, (C_d \log T / (8\bar{\gamma}))^{1/2}\}$, and run Algorithm 2 with $C_0 = 2C_d$, $K = \lceil T/N_0 \rceil - 1$, and $S = (K + 1)\lceil \log T \rceil$. Then $N = T/(K + 1) \leq N_0$ and $m = \lfloor S/(K + 1) \rfloor = \lceil \log T \rceil$, so $N^{1/(m+1)} \leq e$. Let G_1 be the event of Lemma H.1.

Detections are more than N rounds apart. Let $\hat{\kappa}$ be the last detection before a time $t \leq \hat{\kappa} + N$, or $\hat{\kappa} = 0$ before the first detection, and let j be the number of updates since $\hat{\kappa}$. If $j = 0$, the rate used in line 5 is $C_0 \log T \geq 1$, so the test in line 6 cannot fire. If $j \geq 1$, the deployed predictor is an empirical risk minimizer for the samples with indices in $(\hat{\kappa}, \tau]$, where $\tau - \hat{\kappa} \geq N^{j/(m+1)}$. On G_1 , its realized risk at every round $u \leq t$ is at most

$$8\gamma N + \frac{C_d \log T}{N^{j/(m+1)}} \leq \frac{2C_d \log T}{N^{j/(m+1)}} = \frac{C_0 \log T}{N^{j/(m+1)}},$$

because $8\gamma N \leq 8\bar{\gamma} N_0 \leq C_d \log T / N_0 \leq C_d \log T / N^{j/(m+1)}$. This is the rate used in line 5. Hence, on G_1 , a detection within N rounds after the previous detection, or after time 0, implies one of the events $A_{i,r}$ defined in Appendix G, whose union has probability at most $(S + 1)/T$; the proof of

that bound uses only the independence of the samples after each update. The event G that G_1 holds and no such detection occurs therefore satisfies $\Pr(G^c) \leq 2T^{-2} + (S+1)/T \leq (S+2)/T$. On G , the ℓ -th detection occurs after time ℓN , so there are at most K detections. At most m updates occur between consecutive detections, so the cap $S = m(K+1)$ in line 8 is never reached.

Risk. The argument of Appendix G bounds the realized risk pathwise by summing per-epoch bounds over at most $(m+1)(K+1)$ epochs. It uses only the test in line 6, the epoch lengths, and the bound on the number of detections, so it applies verbatim on G . Hence, on G , the realized risk is $O((m+1)(K+1)C_0 \log T N^{1/(m+1)}) = O(C_d(K+1) \log^2 T)$. Since each loss is at most one and $\Pr(G^c) \leq (S+2)/T$, the expected risk exceeds this bound by at most $S+2 = O((K+1) \log T)$. Finally, $K+1 \leq T/N_0 + 1 \leq 2 + T(8\bar{\gamma}/(C_d \log T))^{1/2}$ and $S = (K+1)\lceil \log T \rceil$, which give the bounds $\tilde{O}(1 + T\bar{\gamma}^{1/2})$ claimed in Theorem 3.10. $\square$

I Proof of Theorem 3.12

Proof. Throughout, the rule $\mathcal{U}$ retrains on the full prefix at each update and starts from $\hat{h}_0 = \mathcal{O}(\emptyset)$, as in the theorem. We construct two hypothesis classes such that under the same reactive data distribution and the oracle rule, the historical loss sequences under these two hypothesis classes and their respective choices of the ground truth function are hard to distinguish. On the other hand, the loss vectors under these two hypothesis classes follow the same behavior of loss vectors under realizable *i.i.d.* environment and piecewise stationary environment, respectively, and have expected risks far from each other. Then we use Le Cam's two point lemma to show that the existence of a universal risk-driven update algorithm $\mathcal{U}$ results in a contradiction.

To construct the first hypothesis class $\mathcal{H}_1$ based on Example 1 when $\alpha = \frac{1}{3}$, consider a partition of the indices (l, j) of 8^l intervals $I_{l,j}, j \in \{1, \dots, 8^l\}$, for each $l \in \mathbb{Z}_+$ into a set $\mathcal{P}_l = \{S_{l,1}, \dots, S_{l,2^l+2l}\}$ of $2^l + 2l$ disjoint sets of intervals such that

$$|S_{l,v}| = \begin{cases} 1 & v = 1 \\ 2^{v-2} & 2 \leq v \leq 2l+1 \\ 4^l & 2l+2 \leq v \leq 2^l+2l \end{cases}, \text{ and} \\ \bigcup_{v=1}^{2^l+2l} S_{l,v} = \{(l, 1), \dots, (l, 8^l)\} \quad (22)$$

Denote by $\mathcal{S} = (\mathcal{P}_l : l \in \mathbb{Z}_+)$ the collection of partitions $\mathcal{P}_l$ for all l and denote $\sigma = (\sigma_{l,j} : l \in \mathbb{Z}_+, j \in \{1, \dots, 8^l\})$. Consider the hypothesis subclass

$$\mathcal{H}_{\sigma, \mathcal{S}} = \left\{ \sum_{l=1}^{\infty} \sum_{v=1}^{2^l+2l} \theta_{l,v,\sigma,\mathcal{S}} G_{l,v,\sigma,\mathcal{S}}(x) : \theta_{l,v,\sigma,\mathcal{S}} \in \{-1, 1\} \right\}$$

where

$$G_{l,v,\sigma,\mathcal{S}}(x) = \sum_{j=1}^{8^l} \mathbf{1}((l, j) \in S_{l,v}) c_l \sigma_{l,j} g_{l,j}(x)$$

and $c_l = \frac{1}{2^{l/2}}$ for $l \in \mathbb{Z}_+$ and $v \in \{1, \dots, 2^l+2l\}$. Define the hypothesis class

$$\mathcal{H}_1 = \{h(x, \sigma, \mathcal{S}) : h(x, \sigma, \mathcal{S}) \in \mathcal{H}_{\sigma, \mathcal{S}} \text{ for fixed } \sigma, \mathcal{S}\},$$

where $x \in (0, 1)$, and σ and $\mathcal{S}$ take values over all possible choices of $\sigma_{l,j}$ and partition $\mathcal{S}_l$, respectively. As in Example 1, labels are sequences indexed by the level l : we write hypotheses as sums over l , and the l -th summand is the l -th component of the label. The loss is the squared ℓ_2 loss of Example 1, and the oracle $\mathcal{O}^{\mathcal{H}_1}$ is the same¹ as in Example 1. More specifically, the oracle $\mathcal{O}^{\mathcal{H}_1}$ selects $\hat{h}^{\mathcal{H}_1}(x, \sigma, \mathcal{S}) \in \mathcal{H}_1$ with the minimum $(\theta_{l,v,\sigma,\mathcal{S}} : l \in \mathbb{Z}_+, v \in \{1, \dots, 2^l+2l\})$ in lexicographic order such that the worst-case risk is minimized for every σ and $\mathcal{S}$.

¹Note that the assumption on $\mathcal{O}^{\mathcal{H}_1}$ and $\mathcal{O}^{\mathcal{H}_2}$ can be removed by adding an extra vector σ_2 in the function input in $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively.

The second hypothesis class is simply the hypothesis class $\mathcal{H}$ in Example 1 with dummy parameters σ and $\mathcal{S}$

$$\mathcal{H}_2 = \left\{ h(x, \sigma, \mathcal{S}) : h(x, \sigma, \mathcal{S}) = \sum_{l=1}^{\infty} \sum_{j=1}^{8^l} c_l \theta_{l,j} g_{l,j}(x), \theta_{l,j} \in \{-1, 1\} \text{ for any } \sigma, \mathcal{S} \right\}.$$

Similarly, the loss is the squared ℓ_2 loss and the oracle $\mathcal{O}^{\mathcal{H}_2}$ returns $\hat{h}^{\mathcal{H}_2}(x, \sigma, \mathcal{S}) \in \mathcal{H}_2$ with the minimum $(\theta_{l,j} : l \in \mathbb{Z}_+, j \in \{1, \dots, 8^l\})$ in lexicographic order such that the worst-case risk is minimized for every σ and $\mathcal{S}$.

Consider the following reactive environment E with $K = 1$ distribution change, which occurs at the first update time τ_1 of the update rule running in it. The initial distribution is given by $X_t = (U_t, \sigma_0, \mathcal{S}_0) \sim \mu_0 = (U(0, 1), \sigma_0, \mathcal{S}_0)$, where σ_0 is all 1's vector and $\mathcal{S}_0 = \{\mathcal{P}_{l,0} : l \in \mathbb{Z}_+\}$ is a partition such that $\mathcal{P}_{l,0} = \{S_{l,1,0}, \dots, S_{l,2^l+2l,0}\}$ satisfies (22) and each set $S_{l,v,0}$ lies in a single sign class $\{j : \theta_{l,j} = 1\}$ or $\{j : \theta_{l,j} = -1\}$ of the ground truth chosen below. Such a partition exists: for $l > 1$, each sign class contains $2^{l-1}4^l$ indices, so one class is the union of 2^{l-1} sets of size 4^l and the other is the union of the remaining sets.

Consider an update rule with budget $S = 2$ running in E, with initial data distribution μ_0 . At its first update time τ_1 , the data distribution changes to μ_1 , which E computes from the first τ_1 samples in the following way, regardless of whether the rule uses $\mathcal{H}_1$ or $\mathcal{H}_2$. Let $\hat{h}_{\tau_1}^{\mathcal{H}_2}(x) = \sum_{l=1}^{\infty} \sum_{j=1}^{8^l} c_l \hat{\theta}_{l,j} g_{l,j}(x) \in \mathcal{H}_2$ be the output of the oracle $\mathcal{O}^{\mathcal{H}_2}(\{X_t, h^{\mathcal{H}_2}(X_t)\}_{t=1}^{\tau_1})$. In addition, let $h^{\mathcal{H}_2}(x, \sigma, \mathcal{S}) = \sum_{l=1}^{\infty} \sum_{j=1}^{8^l} c_l \theta_{l,j} g_{l,j}(x)$. Denote the vector $\hat{\theta} = (\hat{\theta}_{l,j} : l \in \mathbb{Z}_+, j \in \{1, \dots, 8^l\})$. For each $l \in \mathbb{Z}_+$, let $V_l = \{(l, j) : \theta_{l,j} \neq \hat{\theta}_{l,j}, j \in \{1, \dots, 8^l\}\}$ be the set of indices where there is a mismatch between $\theta_{l,j}$, $l \in \mathbb{Z}_+, j \in \{1, \dots, 8^l\}$, for h and $\hat{\theta}_{l,j}$ for $\hat{h}_{\tau_1}^{\mathcal{H}_2}$.

If $|V_l| < 8^l$, let $|V_l| \bmod 4^l = \sum_{u=1}^{d_l} 2^{b_{l,u}}$ be the binary representation of the remainder of $|V_l|$ divided by 4^l , with $b_{l,1} < \dots < b_{l,d_l}$ and $d_l = 0$ if the remainder is zero, and let

$$A_l = \{b_{l,u} + 2 : 1 \leq u \leq d_l\} \cup \{2l + 2, \dots, 2l + \lfloor |V_l|/4^l \rfloor + 1\}.$$

If $|V_l| = 8^l$, let $A_l = \{1, \dots, 2^l + 2l\}$. In both cases, construct the partition $\mathcal{S}_1 = (\mathcal{P}_{l,1} : l \in \mathbb{Z}_+)$ such that $\mathcal{P}_{l,1} = \{S_{l,1,1}, \dots, S_{l,2^l+2l,1}\}$ satisfies (22) and $\{S_{l,v,1} : v \in A_l\}$ is a partition of V_l ; this is possible because the sets indexed by A_l have total size $|V_l|$. Then $\{S_{l,v,1} : v \notin A_l\}$ is a partition of $\{(l, j) : j \in \{1, \dots, 8^l\} \setminus V_l, \text{ for } l \in \mathbb{Z}_+\}$.

The distribution μ_1 is given by $X_t = (U_t, \sigma_1, \mathcal{S}_1) \sim \mu_1 = (U(0, 1), \sigma_1, \mathcal{S}_1)$, where $\sigma_1 = -\hat{\theta}$ flips every entry in $\hat{\theta}$. Therefore, the data X_t follows μ_0 for $1 \leq t \leq \tau_1$ and μ_1 for $\tau_1 < t \leq T$.

Choose a ground truth function $h^{\mathcal{H}_2} \in \mathcal{H}_2$ such that the corresponding $\theta_{l,j} = 1$ for $j \in \{1, \dots, 8\}$ and $\sum_{j=1}^{8^l} \theta_{l,j} = 0$ for $l > 1$, i.e., half of the $\theta_{l,j}$'s take the value 1 for each $l > 1$. Then, choose a ground truth function $h^{\mathcal{H}_1} \in \mathcal{H}_1$ such that

$$h^{\mathcal{H}_2}(X_t) = h^{\mathcal{H}_1}(X_t) \quad (23)$$

for $t \in \{1, \dots, T\}$. For $t \leq \tau_1$, this holds by letting $\theta_{l,v,\sigma_0,\mathcal{S}_0}$ be the common sign of $\theta_{l,j}$ on $S_{l,v,0}$. For $t > \tau_1$, it holds by selecting

$$\theta_{l,v,-\hat{\theta},\mathcal{S}_1} = \begin{cases} 1, & v \in A_l, \\ -1, & v \in \{1, \dots, 2^l + 2l\} \setminus A_l, \end{cases}$$

for $l \in \mathbb{Z}_+$. This follows from the definition of $G_{l,v,\sigma,\mathcal{S}}$, the facts that $\theta_{l,j} = -\hat{\theta}_{l,j}$ for $(l, j) \in V_l$ and $\theta_{l,j} = \hat{\theta}_{l,j}$ otherwise, and the fact that $\{S_{l,v,1} : v \in A_l\}$ is a partition of V_l . In fact, both identities hold for every $x \in (0, 1)$ and every level l .

Next, we show that

$$\hat{h}_{\tau_i}^{\mathcal{H}_1}(X_t) = \hat{h}_{\tau_i}^{\mathcal{H}_2}(X_t) \quad (24)$$

for $\tau_i < t \leq \tau_{i+1}$, $i \in \{0, 1\}$, where $\tau_0 = 0$ and τ_2 is the time of the second model update decided by $\mathcal{U}$. For $\tau_0 < t \leq \tau_1$, $\hat{h}^{\mathcal{H}_1}$ and $\hat{h}^{\mathcal{H}_2}$ are initial functions returned by $\mathcal{O}^{\mathcal{H}_1}$ and $\mathcal{O}^{\mathcal{H}_2}$, respectively,

where the parameters $\theta_{l,v,\sigma,S} = -1$ in $\mathcal{H}_1$ and $\theta_{l,j} = -1$ in $\mathcal{H}_2$. Since the parameter σ_0 is all one vector for $\tau_0 < t \leq \tau_1$, we have

$$\hat{h}_{\tau_i}^{\mathcal{H}_1}(X_t) = \hat{h}_{\tau_i}^{\mathcal{H}_2}(X_t) = - \sum_{l=1}^{\infty} \sum_{j=1}^{8^l} c_l g_{l,j}(x).$$

We then show that (24) holds for $\tau_1 < t \leq \tau_2$. Note that $h^{\mathcal{H}_2}$ has $\theta_{1,j} = 1$ for all $j \in \{1, \dots, 8\}$, whereas the initial estimates are -1 . As shown in the proof of Proposition D.2, $\hat{\theta}_{1,j}$ is corrected whenever there exists $U_t \in I_{1,j}$ for some $t \leq \tau_1$. Hence, the updated function $\hat{h}_{\tau_1}^{\mathcal{H}_2}$ at time $t = \tau_1$ has a different set of θ compared to the initial all -1 vector, so $\sigma_1 \neq \sigma_0$. This implies that the updated function $\hat{h}_{\tau_1}^{\mathcal{H}_1}$ does not update the initial parameters $\theta_{l,v,-\hat{\theta},S_1} = -1$ for $l \in \mathbb{Z}_+$ and $v \in \{1, \dots, 2l + 2^l\}$. Hence, we have

$$\hat{h}_{\tau_1}^{\mathcal{H}_1}(x, -\hat{\theta}, S_1) = - \sum_{l=1}^{\infty} \sum_{v=1}^{2^l+2l} G_{l,v,-\hat{\theta},S_1}(x) = \sum_{l=1}^{\infty} \sum_{j=1}^{8^l} c_l \hat{\theta}_{l,j} g_{l,j}(x) = \hat{h}_{\tau_1}^{\mathcal{H}_2}(x, -\hat{\theta}, S_1),$$

and thus (24) for $\tau_1 < t \leq \tau_2$.

Note that (23) and (24) imply that the loss vector $\ell^{\tau_2} = (\ell(\hat{h}_{s(t)}(X_t), h(X_t)) : t \leq \tau_2)$ is the same when the algorithm $\mathcal{U}$ is run on hypothesis classes $\mathcal{H}_1$ and $\mathcal{H}_2$ with oracles $\mathcal{O}^{\mathcal{H}_1}$ and $\mathcal{O}^{\mathcal{H}_2}$, respectively. Thus, by the Le Cam's two point lemma, any hypothesis class estimator ψ that takes the loss vector ℓ^{τ_2} (τ_2 can be decided by running $\mathcal{U}$) as input and returns the class $\mathcal{H}_1$ or $\mathcal{H}_2$ that $\mathcal{U}$ runs on has worst-case correct probability at most

$$\begin{aligned} & \min \{ \Pr(\psi(\ell^{\tau_2}) = \mathcal{H}_1 \mid \mathcal{U} \text{ runs on } \mathcal{H}_1), \Pr(\psi(\ell^{\tau_2}) = \mathcal{H}_2 \mid \mathcal{U} \text{ runs on } \mathcal{H}_2) \} \\ & \leq 1 - \frac{1 - TV(\Pr(\ell^{\tau_2} \mid \mathcal{U} \text{ runs on } \mathcal{H}_1), \Pr(\ell^{\tau_2} \mid \mathcal{U} \text{ runs on } \mathcal{H}_2))}{2} = \frac{1}{2}. \end{aligned}$$

To derive a contradiction, in the following, we show that $\mathcal{U}$ results in an estimator ψ with worst case correct probability at least $\frac{2}{3}$, i.e.,

$$\min \{ \Pr(\psi(\ell^{\tau_2}) = \mathcal{H}_1 \mid \mathcal{U} \text{ runs on } \mathcal{H}_1), \Pr(\psi(\ell^{\tau_2}) = \mathcal{H}_2 \mid \mathcal{U} \text{ runs on } \mathcal{H}_2) \} \geq \frac{2}{3}. \quad (25)$$

The following lemma, proved in Appendix J, bounds the expected loss of the full-prefix oracle outputs under $\mathcal{H}_1$ and $\mathcal{H}_2$ when the first update time is fixed.

Lemma I.1. Fix $\tau_1 \geq 1$, let $X_t \sim \mu_0$ for $\tau_0 < t \leq \tau_1$ and $X_t \sim \mu_1$ for $\tau_1 < t \leq T$, and let $\hat{h}_{t_1}^{\mathcal{H}_k} = \mathcal{O}^{\mathcal{H}_k}(\{X_i, h^{\mathcal{H}_k}(X_i)\}_{i=1}^{t_1})$ for $k \in \{1, 2\}$. For the squared ℓ_2 loss, we have

$$\mathbb{E}_{\{X_i, h^{\mathcal{H}_2}(X_i)\}_{i=1}^{t_1}, X_{t_2}} \left[\ell(\hat{h}_{t_1}^{\mathcal{H}_2}(X_{t_2}), h^{\mathcal{H}_2}(X_{t_2})) \right] = \begin{cases} \Omega(\frac{1}{t_1^{\frac{1}{3}}}), O(\frac{1+\log t_1}{t_1^{\frac{1}{3}}}) & 1 \leq t_1 < t_2 \leq T \\ \Theta(1) & 0 = t_1 < t_2 \leq T \end{cases} \quad (26)$$

and

$$\begin{aligned} & \mathbb{E}_{\{X_i, h^{\mathcal{H}_1}(X_i)\}_{i=1}^{t_1}, X_{t_2}} \left[\ell(\hat{h}_{t_1}^{\mathcal{H}_1}(X_{t_2}), h^{\mathcal{H}_1}(X_{t_2})) \right] \\ & = \begin{cases} \min \left\{ O\left(\frac{1+\log \tau_1}{\tau_1^{\frac{1}{3}}}\right), O\left(\frac{1+\log(t_1-\tau_1)}{t_1-\tau_1}\right) \right\} & \tau_1 < t_1 < t_2 \leq T \\ \Omega(\frac{1}{\tau_1^{\frac{1}{3}}}), O\left(\frac{1+\log \tau_1}{\tau_1^{\frac{1}{3}}}\right) & \tau_1 = t_1 < t_2 \leq T \\ O(\frac{1+\log t_1}{t_1^{\frac{1}{3}}}) & 0 < t_1 < t_2 \leq \tau_1 \\ \Theta(1) & 0 = t_1 < t_2 \leq \tau_1 \end{cases} \quad (27) \end{aligned}$$

Define the following hypothesis class estimator

$$\psi(\ell^{\tau_2}) = \begin{cases} \mathcal{H}_1 & \text{if } \tau_2 \leq T^{\frac{5}{6}} \\ \mathcal{H}_2 & \text{if } \tau_2 > T^{\frac{5}{6}}. \end{cases}$$

For $k \in \{1, 2\}$ and an update rule Φ with $S = 2$, let $R_k(\Phi)$ be the expected risk of Φ when it runs in $\mathbf{E}$ with $(\mathcal{H}_k, \mathcal{O}^{\mathcal{H}_k})$, and let $\text{OPT}_k = \inf_{\Phi} R_k(\Phi)$. Both quantities refer to the same reactive environment, whose change occurs at the first update of the rule being evaluated. Suppose, for contradiction, that $R_k(\mathcal{U}) \leq T^c \text{OPT}_k$ for $k = 1, 2$, for a constant $c \in (0, \frac{7}{234})$, where $\frac{7}{234} = \frac{13}{18} - \frac{9}{13}$. We show that the correct probability of $\psi(\ell^{\tau_2})$ is then at least $\frac{2}{3}$ when $\mathcal{U}$ runs on $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively.

We first record a deterministic lower bound. Let $\hat{h}$ be the output of $\mathcal{O}^{\mathcal{H}_2}$ on $n \geq 1$ samples. At each level $l > 1$, at most n of the $8^l/2$ cells with $\theta_{l,j} = 1$ contain a sample, and each remaining such cell contributes $4c_l^2/8^l$ to the risk of $\hat{h}$ by (11). Summing over the levels with $8^l \geq 4n$ shows that the risk of $\hat{h}$, for the fixed sample, is at least

$$\sum_{l>1: 8^l \geq 4n} 2^{-l} \geq (8n)^{-1/3}. \quad (28)$$

When $\mathcal{U}$ runs on $\mathcal{H}_1$, consider the rule Φ_1 that updates at times 1 and $\lceil \sqrt{T} \rceil$. In $\mathbf{E}$, its change occurs at time 1, and its second predictor is trained on $\lceil \sqrt{T} \rceil - 1$ post-change samples. By Lemma I.1 and the boundedness of the loss, $R_1(\Phi_1) = O(\sqrt{T} \log T)$, so $R_1(\mathcal{U}) = O(T^{\frac{1}{2}+c} \log T)$. The losses before τ_1 are i.i.d., since $\hat{h}_0^{\mathcal{H}_1}$ is fixed, and τ_1 is a stopping time with respect to them. Wald's identity gives

$$\mathbb{E} \left[\sum_{t=1}^{\tau_1} \ell(\hat{h}_0^{\mathcal{H}_1}(X_t), h^{\mathcal{H}_1}(X_t)) \right] = \mathbb{E}[\tau_1] \mathbb{E}[\ell(\hat{h}_0^{\mathcal{H}_1}(X_1), h^{\mathcal{H}_1}(X_1))],$$

and the last expectation is $\Theta(1)$. Then, we have $\mathbb{E}[\tau_1] \leq O(T^{\frac{1}{2}+c} \log T)$, and thus by Markov's inequality, $\Pr(\tau_1 > T^{\frac{7}{12}}) \leq \frac{1}{6}$ for large T . After τ_1 , the samples are i.i.d. from μ_1 and the predictor $\hat{h}_{\tau_1}^{\mathcal{H}_1}$ is fixed until τ_2 . Conditionally on the history $\mathcal{F}_{\tau_1}$ up to τ_1 , Wald's identity therefore gives

$$\mathbb{E} \left[\sum_{t=\tau_1+1}^{\tau_2} \ell(\hat{h}_{\tau_1}^{\mathcal{H}_1}(X_t), h^{\mathcal{H}_1}(X_t)) \mid \mathcal{F}_{\tau_1} \right] = \mathbb{E}[\tau_2 - \tau_1 \mid \mathcal{F}_{\tau_1}] R_1,$$

where R_1 is the risk of $\hat{h}_{\tau_1}^{\mathcal{H}_1}$ under μ_1 . By (23) and (24), which hold for every x , R_1 is the risk of $\hat{h}_{\tau_1}^{\mathcal{H}_2}$, so $R_1 \geq (8\tau_1)^{-1/3}$ by (28). On $\{\tau_1 \leq T^{\frac{7}{12}}\}$, this gives $R_1 \geq \frac{1}{2}T^{-\frac{7}{36}}$. Taking expectations, $\mathbb{E}[(\tau_2 - \tau_1)\mathbf{1}\{\tau_1 \leq T^{\frac{7}{12}}\}] \leq O(T^{\frac{25}{36}+c} \log T)$. Since $\frac{25}{36} + c < \frac{5}{6}$, Markov's inequality gives $\Pr(\tau_2 > T^{\frac{5}{6}}, \tau_1 \leq T^{\frac{7}{12}}) \leq \frac{1}{6}$ for large T . Therefore,

$$\Pr(\tau_2 > T^{\frac{5}{6}}) \leq \Pr(\tau_1 > T^{\frac{7}{12}}) + \Pr(\tau_2 > T^{\frac{5}{6}}, \tau_1 \leq T^{\frac{7}{12}}) \leq \frac{1}{3}.$$

We next consider the case when $\mathcal{U}$ runs on $\mathcal{H}_2$. Since $h^{\mathcal{H}_2}$ does not depend on $(\sigma, \mathcal{S})$, the losses behave as in Example 1 with $\alpha = \frac{1}{3}$. The fixed schedule of Theorem 3.3(i) with $S = 2$ also faces a change at its first update, which does not affect the losses under $\mathcal{H}_2$. By Lemma I.1, its risk is $O(T^{\frac{9}{13}} \log T)$, so $\text{OPT}_2 = O(T^{\frac{9}{13}} \log T)$ and $R_2(\mathcal{U}) = O(T^{\frac{9}{13}+c} \log T)$. No update occurs after τ_2 , so conditionally on $\mathcal{F}_{\tau_2}$ the expected remaining loss is $(T - \tau_2)R_2$, where $R_2 \geq (8\tau_2)^{-1/3}$ is the risk of $\hat{h}_{\tau_2}^{\mathcal{H}_2}$ by (28). Hence the expected risk of $\mathcal{U}$ is at least

$$\begin{aligned} \mathbb{E}[(T - \tau_2)(8\tau_2)^{-1/3}\mathbf{1}\{\tau_2 \leq T^{\frac{5}{6}}\}] &\geq (T - T^{\frac{5}{6}})(8T^{\frac{5}{6}})^{-1/3} \Pr(\tau_2 \leq T^{\frac{5}{6}}) \\ &= \Omega(T^{\frac{13}{18}}) \Pr(\tau_2 \leq T^{\frac{5}{6}}). \end{aligned}$$

Since $\frac{13}{18} > \frac{9}{13} + c$, this forces $\Pr(\tau_2 \leq T^{\frac{5}{6}}) < \frac{1}{3}$ for large T . Therefore, the probability that $\psi(\ell^{\tau_2})$ makes a wrong estimation is at most $\frac{1}{3}$ in both environments, which implies (25) and contradicts the Le Cam bound above. This completes the proof. $\square$

J Proof of Lemma I.1

Proof. Throughout, $(h - \hat{h})^2$ denotes the squared ℓ_2 loss of Example 1. By similar arguments as in the proof of Proposition D.2, we have

$$\mathbb{E}_{X_{t_2}} [(h^{\mathcal{H}_2}(X_{t_2}) - \mathcal{O}^{\mathcal{H}_2}(\{X_i, h^{\mathcal{H}_2}(X_i)\}_{i=1}^{t_1}))^2]$$

$$= \begin{cases} \sum_{l=1}^{\infty} \sum_{j=1}^{8^l} \mathbf{1}(I_{l,j} \cap \{X_i\}_{i=1}^{t_1} = \emptyset) \frac{c_l^2}{8^l} (\theta_{l,j} + 1)^2 & t_1 > 0 \\ \Theta(1) & t_1 = 0 \end{cases}, \quad (29)$$

for $t_2 > t_1$, since the value $h(X_t)$ for any $h \in \mathcal{H}_2$ does not depend on $(\sigma, \mathcal{S})$ and the distribution of the loss vector ℓ^T is the same as that of the loss vector ℓ^T in Example 1. Note that

$$\mathbb{E}_{\{X_i, h^{\mathcal{H}_2}(X_i)\}_{i=1}^{t_1}} [\mathbf{1}(I_{l,j} \cap \{X_i\}_{i=1}^{t_1} = \emptyset)] = \left(1 - \frac{1}{8^l}\right)^{t_1}.$$

Combining with (29), we have

$$\begin{aligned} & \mathbb{E}_{\{X_i, h^{\mathcal{H}_2}(X_i)\}_{i=1}^{t_1}, X_{t_2}} [(h^{\mathcal{H}_2}(X_{t_2}) - \mathcal{O}^{\mathcal{H}_2}(\{X_i, h^{\mathcal{H}_2}(X_i)\}_{i=1}^{t_1}))^2] \\ & \leq \sum_{l=1}^{\infty} \frac{4}{2^l} \left(1 - \frac{1}{8^l}\right)^{t_1} \\ & \leq \sum_{l=1}^{\frac{\ln t_1 - \ln \ln t_1}{\ln 8}} \frac{4}{2^l} \left(1 - \frac{1}{8^l}\right)^{t_1} + \sum_{l=\frac{\ln t_1 - \ln \ln t_1}{\ln 8} + 1}^{\infty} \frac{4}{2^l} \left(1 - \frac{1}{8^l}\right)^{t_1} \\ & \leq \frac{2(\frac{\ln t_1 - \ln \ln t_1}{\ln 8})}{t_1} + \frac{4 \ln^{\frac{1}{3}} t_1}{t_1^{\frac{1}{3}}} \leq O\left(\frac{\log t_1}{t_1^{\frac{1}{3}}}\right), \end{aligned}$$

and

$$\begin{aligned} & \mathbb{E}_{\{X_i, h^{\mathcal{H}_2}(X_i)\}_{i=1}^{t_1}, X_{t_2}} [(h^{\mathcal{H}_2}(X_{t_2}) - \mathcal{O}^{\mathcal{H}_2}(\{X_i, h^{\mathcal{H}_2}(X_i)\}_{i=1}^{t_1}))^2] \\ & \geq \sum_{l \geq 2} \frac{2}{2^l} \left(1 - \frac{1}{8^l}\right)^{t_1} \geq \frac{1}{4t_1^{\frac{1}{3}}} = \Omega\left(\frac{1}{t_1^{\frac{1}{3}}}\right), \end{aligned}$$

Therefore, (26) holds. Similarly, consider $\mathcal{H}_1$. Its parameters are indexed by the context $(\sigma, \mathcal{S})$, which is part of the input, so the parameters of $\hat{h}_{t_1}^{\mathcal{H}_1}$ for a context depend only on the samples in that context: $\hat{\theta}_{l,v,\sigma,\mathcal{S}} = \theta_{l,v,\sigma,\mathcal{S}}$ if one of these samples lies in $\cup_{(l,j) \in S_{l,v}} I_{l,j}$, and $\hat{\theta}_{l,v,\sigma,\mathcal{S}} = -1$ otherwise. Since $\sigma_1 \neq \sigma_0$ (Appendix I), only the samples after τ_1 determine the parameters used after the change. Hence

$$\begin{aligned} & \mathbb{E}_{X_{t_2}} [(h^{\mathcal{H}_1}(X_{t_2}) - \hat{h}_{t_1}^{\mathcal{H}_1}(X_{t_2}))^2] \\ & = \begin{cases} \sum_{l=1}^{\infty} \sum_{v=1}^{2^l+2l} \mathbf{1}(\cup_{(l,j) \in S_{l,v,0}} I_{l,j} \cap \{X_i\}_{i=1}^{t_1} = \emptyset) & 0 < t_1 < t_2 \leq \tau_1 \\ \quad \times \frac{|S_{l,v,0}|c_l^2}{8^l} (\theta_{l,v,\sigma_0,\mathcal{S}_0} + 1)^2 & \\ \sum_{l=1}^{\infty} \sum_{v=1}^{2^l+2l} \mathbf{1}(\cup_{(l,j) \in S_{l,v,1}} I_{l,j} \cap \{X_i\}_{i=\tau_1+1}^{t_1} = \emptyset) & \tau_1 \leq t_1 < t_2 \leq T \\ \quad \times \frac{|S_{l,v,1}|c_l^2}{8^l} (\theta_{l,v,\sigma_1,\mathcal{S}_1} + 1)^2 & \\ \Theta(1) & t_1 = 0 \end{cases}. \quad (30) \end{aligned}$$

Since $(\theta + 1)^2 \leq 4$, the expectations of both sums are controlled by the following occupancy estimate. Let $\{S_{l,1}, \dots, S_{l,2^l+2l}\}$ satisfy (22) for each l , and let $U_1, \dots, U_m$, $m \geq 1$, be i.i.d. uniform on $(0, 1)$ and independent of these partitions. The sets with $v \geq 2l + 2$ have size 4^l , and there are fewer than 2^l of them. Since the function $(1 - x)^m x$ on $(0, 1)$ is maximized at $x = \frac{1}{m+1}$, where it is at most $\frac{1}{m+1}$, we have

$$\begin{aligned} & \mathbb{E} \left[\sum_{l=1}^{\infty} \sum_{v=1}^{2^l+2l} \mathbf{1}(\cup_{(l,j) \in S_{l,v}} I_{l,j} \cap \{U_i\}_{i=1}^m = \emptyset) \frac{4|S_{l,v}|c_l^2}{8^l} \right] \\ & = \sum_{l=1}^{\infty} \sum_{v=1}^{2^l+2l} \left(1 - \frac{|S_{l,v}|}{8^l}\right)^m \frac{4|S_{l,v}|c_l^2}{8^l} \end{aligned}$$

$$\begin{aligned}
&\leq \sum_{l=1}^{\infty} \sum_{v=1}^{2l+1} \left(1 - \frac{|S_{l,v}|}{8^l}\right)^m \frac{4|S_{l,v}|c_l^2}{8^l} + \sum_{l=1}^{\infty} \left(1 - \frac{1}{2^l}\right)^m \frac{4}{2^l} \\
&\leq \sum_{l=1}^{\infty} \frac{4(2l+1)c_l^2}{m+1} + \sum_{l \leq \log_2 m} \frac{4}{m+1} + \sum_{l > \log_2 m} \frac{4}{2^l} = O\left(\frac{1 + \log m}{m}\right).
\end{aligned}$$

For $0 < t_1 < t_2 \leq \tau_1$, the partition $\mathcal{S}_0$ is fixed and $X_1, \dots, X_{t_1}$ are i.i.d. from μ_0 , so the estimate with $m = t_1$ bounds the expectation of the first sum by $O(\frac{1+\log t_1}{t_1})$. For $t_1 > \tau_1$, let $n = t_1 - \tau_1$. The partition $\mathcal{S}_1$ depends on the first τ_1 samples. Conditionally on these samples, $\mathcal{S}_1$ is fixed and the inputs $U_{\tau_1+1}, \dots, U_{t_1}$ are i.i.d. uniform and independent of it, so the estimate with $m = n$ bounds the conditional expectation of the second sum by $O(\frac{1+\log n}{n})$, and hence also its expectation. Moreover, a post-change sample can only change a parameter from -1 to its correct value, so the post-change risk of $\hat{h}_{t_1}^{\mathcal{H}_1}$ is nonincreasing in $t_1 \geq \tau_1$ for every realization of the samples. By (23) and (24), which hold for every x , this risk equals the risk of $\hat{h}_{\tau_1}^{\mathcal{H}_2}$ when $t_1 = \tau_1$. With (26), this gives the bounds $\Omega(\tau_1^{-\frac{1}{3}})$ and $O(\frac{1+\log \tau_1}{\tau_1^{1/3}})$ for $t_1 = \tau_1$, and the upper bound $O(\frac{1+\log \tau_1}{\tau_1^{1/3}})$ for $t_1 > \tau_1$. Therefore, we have (27). $\square$

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction state each result with its assumptions: constant-factor guarantees for the fixed schedules, polylogarithmic guarantees for the adaptive algorithms, the scope of the no-free-lunch theorem, and the experimental findings.

Guidelines:

- The answer [\[N/A\]](#) means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [\[No\]](#) or [\[N/A\]](#) answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The conclusion and Appendix B discuss curve assumptions, event-count versus compute costs, and the scope of the experiments.

Guidelines:

- The answer [\[N/A\]](#) means that the paper has no limitation while the answer [\[No\]](#) means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: Each theorem states its assumptions, and complete proofs are given in Appendices A and C–J.

Guidelines:

- The answer [\[N/A\]](#) means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: Appendix B specifies the data, oracles, schedules, seeds, fitting procedure, and evaluation protocol. Code and per-seed outputs accompany the paper.

Guidelines:

- The answer [\[N/A\]](#) means that the paper does not include experiments.
- If the paper includes experiments, a [\[No\]](#) answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The supplementary code includes download commands, dependencies, training and evaluation scripts, raw outputs, and figure/table generation instructions.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://neurips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://neurips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Appendix B.1 specifies preprocessing, model sizes, optimizer settings, training buffers, and the distinction between the MLP reference and the exact ridge benchmark.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Tables and figures report means and sample standard deviations across independent seeds, with the number of runs and sources of variation specified in Appendix B.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Appendix B.6 gives the CPU, memory, thread settings, timing logs, and the limits of the compute accounting. Timings cover the reported experiments, not the entire research project.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The work studies retraining schedules using standard benchmark data and synthetic streams; it does not recruit participants or deploy models.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Appendix B.6 discusses possible reductions in deployment overhead and the risk of delayed correction under distribution shift.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We do not release pretrained generative models, scraped datasets, or other assets with a high risk of misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Dataset creators and source URLs are credited in Appendix B.6. Fashion-MNIST is used under its MIT license; MNIST, which its creators distribute without an explicit license, is used for research only. Dataset files are not redistributed.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: The supplementary experiment code and numerical outputs are documented with reproduction commands, parameters, and limitations. No new dataset is introduced.

Guidelines:

- The answer [\[N/A\]](#) means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[N/A\]](#)

Justification: The experiments do not involve crowdsourcing or recruitment of human participants.

Guidelines:

- The answer [\[N/A\]](#) means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[N/A\]](#)

Justification: The experiments do not involve human-subject recruitment or interventions.

Guidelines:

- The answer [\[N/A\]](#) means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [\[Yes\]](#)

Justification: An LLM-based coding assistant helped prepare this revision, check the sharpened schedule, revise experiment code, execute experiments, and draft text. LLM outputs are not a component of the retraining algorithms or experimental oracles.

Guidelines:

- The answer [\[N/A\]](#) means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.