OEUVRE: OnlinE Unbiased Variance-Reduced Loss Estimation

Kanad Pardeshi [1]
Machine Learning Department
Carnegie Mellon University

Bryan Wilder
Machine Learning Department
Carnegie Mellon University

Aarti Singh
Machine Learning Department
Carnegie Mellon University

Abstract

Online learning algorithms continually update their models as data arrive, making it essential to accurately estimate the expected loss at the current time step. The prequential method is an effective estimation approach which can be practically deployed in various ways. However, theoretical guarantees have previously been established under strong conditions on the algorithm, and practical algorithms have hyperparameters which require careful tuning. We introduce OEUVRE, an estimator that evaluates each incoming sample on the function learned at the current and previous time steps, recursively updating the loss estimate in constant time and memory. We use algorithmic stability, a property satisfied by many popular online learners, for optimal updates and prove consistency, convergence rates, and concentration bounds for our estimator. We design a method to adaptively tune OEUVRE's hyperparameters and test it across diverse online and stochastic tasks. We observe that OEUVRE matches or outperforms other estimators even when their hyperparameters are tuned with oracle access to ground truth.

1 INTRODUCTION

Online learning is an important part of modern-day machine learning, and algorithms that update their models over time are widely used in applications such as recommendation algorithms and portfolio selection in online time series. In such settings, a measure of the model's *expected loss* at the current time step is a naturally useful quantity for tasks like monitoring model performance over time. This measure can also be used to determine early stopping in training when the labels are costly to obtain, and when we would want to stop training due to required compute cost or time once the model is sufficiently accurate. Moreover, a good estimate of the model's current performance can play a crucial role in online model selection (Foster et al., 2017).

A common approach to online loss estimation is the prequential method (Dawid, 1984; Vovk et al., 2005), which enables performance estimation for online models in a sample-efficient way. Here, the incoming sample is first used to evaluate the current model's performance before using it to update the model. The prequential method can be used with different methods of combining evaluations, such as fading factors, sliding windows, (Gama et al., 2013) and exponential moving averages. Moreover, drift detection methods like ADWIN (Bifet and Gavalda, 2007), EDDM (Baena-Garcıa et al., 2006), and Page-Hinkley (Page, 1954; Sebastião and Fernandes, 2017) can be modified for online loss estimation, resulting in adaptive sliding windows.

A key challenge in online loss estimation is that while the incoming data may be i.i.d., the observed sequence of losses on these samples is not i.i.d. as the learned functions evolve over time. Existing theoretical results for prequential estimation (Gama et al., 2013) require the strong assumption that the sequence of learned functions f_t converges to a fixed limit f^*, and show that the estimate is consistent with this assumption. Drift detection methods (Bifet and Gavalda, 2007; Dos Reis et al., 2016) provide detection guarantees for general data streams but do not exploit properties of the learning algorithm for convergence guarantees. In practice, these methods have hyperparameters which need to be set carefully for accurate loss estimation.

In this work, we present OnlinE Unbiased Variance-Reduced loss Estimation (OEUVRE), an online loss estimator which evaluates the incoming sample on the current model *and* the model from the previous time step. We use these two evaluations to update the

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

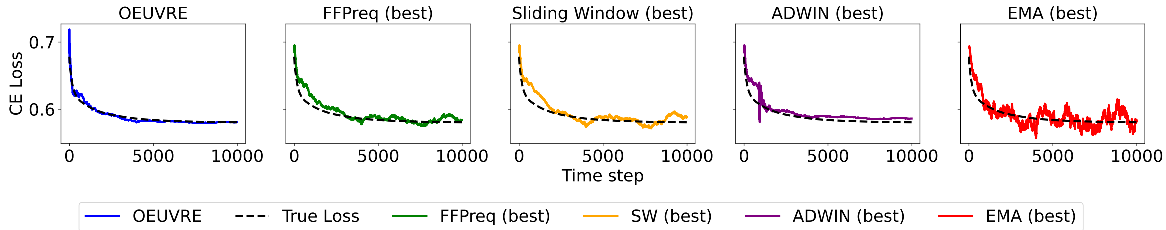


Figure 1: We illustrate the behavior of OEUVRE and several baselines on a representative run of the Diabetes Health Indicators dataset. The hyperparameter for each baseline was chosen using grid search to minimize RMSE. We see that our proposed estimator provides a more accurate continuous estimate of the true loss compared to the baselines without the need for hyperparameter tuning.

estimate, with the past evaluation helping to reduce the variance of the updated estimate. The influence of the past estimate is controlled by a sequence of weights (γ_t), where each γ_t is chosen to minimize a variance bound. This minimization requires knowledge of the extent of change in the learned function between two consecutive time steps, and *algorithmic stability* provides us with exactly that information. Stability bounds are known for many popular online algorithms such as Follow-the-Leader (FTL), Follow-the-Regularized-Leader (FTRL), and Online Mirror Descent (OMD), and OEUVRE can readily be used with these frameworks.

The simple step of evaluating the incoming sample on an additional model leads to powerful theoretical results when coupled with algorithmic stability. We show that in the setting with i.i.d. data and for losses with bounded variance, OEUVRE converges in L^2 to the expected loss of the currently learned model, along with proving convergence rates that scale with the stability rate of the algorithm. We prove that OEUVRE forms a martingale, which enables us to establish novel asymptotic convergence and concentration results for online loss estimation. The rates in our results are closely related to the rate of algorithmic stability, thus linking statistical performance to optimization dynamics. To our knowledge, this is the first work to systematically leverage the stability of learning algorithms for online performance estimation.

We then design a method to improve OEUVRE's practical performance without affecting its rate of convergence by adaptively estimating its hyperparameters from observed data. We demonstrate the empirical effectiveness of our estimator through experiments on linear regression, logistic regression, prediction with expert advice, and neural networks. We observe that OEUVRE consistently provides accurate estimates of the expected loss of the currently learned model across tasks and noise regimes. OEUVRE matches or outperforms existing baselines, even when the hyperparameters of each baseline are tuned for optimal estimation. We provide a comparison of OEUVRE against baseline methods for a representative run in Figure 1.

2 RELATED WORK

To monitor online performance at every step, practitioners commonly use prequential (interleaved test-then-train) evaluation, yielding estimators such as the plain prequential average, sliding windows, fading-factor variants, and exponential moving averages (EMA) (Gama et al., 2013). The prequential average is consistent under i.i.d. data when the learned predictors f_t converge to a limit $f^\star$ (Gama et al., 2013). Moreover, although concentration bound can be obtained for the prequential average (Gama et al., 2009), these bound hold for the *time-averaged* loss, and not the loss for the current time step. Our approach instead leverages algorithmic stability, a much weaker convergence assumption, to design an estimator for the current model's expected loss with stronger L^2 convergence given i.i.d. data. Sliding windows and fixed-decay EMAs impose a fixed bias–variance tradeoff; OEUVRE can adapt to the problem setting by reducing the variance proportional to the difficulty of the problem through the stability of the learning algorithm and provide a consistent estimate.

Drift detection methods like ADWIN (Bifet and Gavalda, 2007), EDDM (Baena-Garcıa et al., 2006), Page-Hinkley (Page, 1954; Sebastião and Fernandes, 2017) and KSWIN (Dos Reis et al., 2016) are designed to identify changes in the data distribution through hypothesis testing or thresholding on observed values. While these methods can be modified to create adaptive sliding window estimators for loss estimation, they are primarily designed for non-stationary environments with concept drift. In contrast, OEUVRE is specifically designed for loss estimation, providing explicit convergence guarantees that exploit algorithmic stability properties of the learner. Moreover, the sensitivity of drift detection methods depends on tunable hyperparameters, whereas OEUVRE's hyperparameters can be set adaptively (Section 6) without affecting convergence rates given that the model is trained on i.i.d. data.

Algorithmic stability (Bousquet and Elisseeff, 2002; Bousquet et al., 2020) bounds the extent to which a new sample can change the function learned by a learning algorithm on a dataset. Uniform stability has previously been connected to generalization bounds (Bousquet and Elisseeff, 2002; Bousquet et al., 2020). Stability guarantees are known for a wide variety of commonly used algorithms, including regularized Empirical Risk Minimization (Bousquet and Elisseeff, 2002), Stochastic Gradient Descent (Hardt et al., 2016), and online learning frameworks (Saha et al.). Our work is the first to connect uniform stability with loss estimation, and our theoretical guarantees establish a link between the estimator's convergence rate and the algorithm's stability rate. We list some useful known stability bounds in Table 1 and elaborate on this connection in Section 4.

Variance reduction techniques (Botev and Ridder, 2017; Gower et al., 2020) and stochastic approximation (Borkar and Borkar, 2008; Lai, 2003) have been highly successful in stochastic optimization. They underpin methods for stochastic gradient descent (Johnson and Zhang, 2013; Defazio et al., 2014; Shalev-Shwartz and Zhang, 2013; Schmidt et al., 2017) and Q-learning (Wainwright, 2019b; Khamaru et al., 2021; Wang et al., 2024), where they yield faster convergence rates. Broad mechanisms include storing snapshots and gradient tables, as well as classical control-variate and importance-sampling strategies. In contrast, our work adapts the spirit of variance reduction to the setting of *loss estimation*: we leverage evaluations at consecutive time steps to more accurately estimate the expected performance of the learning algorithm at the current step.

3 PROBLEM SETUP

We assume a sequence of samples $\{z_t\}_{t\in\mathbb{N}^+}$ arriving in an online fashion, where $z_t = (x_t, y_t)$, $x_t \in \mathcal{X}$ are the features, $y_t \in \mathcal{Y}$ are the labels, and (x_t, y_t) are drawn i.i.d. from the distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. The i.i.d. assumption is standard in analyzing online loss estimation (Gama et al., 2013). Using these samples, we learn a sequence of functions $(f_t)_{t\geq 1}$ using some online learning algorithm, where function $f_t : \mathcal{X} \to \mathcal{Y}$ is learned using the first $(t-1)$ samples, $\{z_s\}_{s\in[t-1]}$. We further assume that all f_t's belong to a hypothesis class $\mathcal{G}$. We define $\{\mathcal{F}_t\}_{t\in\mathbb{N}}$ as the canonical filtration defined on the sample sequence $\{z_t\}$.

We measure the performance of the learning algorithm using a loss function $\ell : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$. We define $\ell_t(z) = \ell_t((x, y)) = \ell(f_t(x), y)$. The expected loss of f_t is given by $\mathbb{E}_{z\sim\mathcal{D}}[\ell_t(z)] = \mathbb{E}[\ell(f_t(x), y) \mid \mathcal{F}_{t-1}]$. Our expected loss is conditioned on the first $(t-1)$ samples, therefore f_t is constant inside the expectation.

Our goal is to estimate $\mathbb{E}_{z\sim\mathcal{D}}[\ell_t(z)]$. Although samples $\{z_t\}$ are drawn i.i.d., the sequence of loss functions $\ell_t(\cdot)$ changes as a result of learning f_t from the data. Thus any sample $\ell_t(z_t)$ that we draw will not be i.i.d. with t. We assume that $\mathrm{Var}(\ell_t \mid \mathcal{F}_{t-1})$ exists and is upper bounded by b^2 for some $b > 0$. This mild assumption is satisfied by a wide variety of loss functions, including bounded loss functions and bounded hypothesis classes. Throughout the paper, we distinguish between *conditional* and *total* expectation. The conditional expectation $\mathbb{E}[\cdot \mid \mathcal{F}_{t-1}]$ assumes that the first $(t-1)$ samples are fixed and the expectation is restricted to the t-th sample. On the other hand, the total expectation $\mathbb{E}[\cdot]$ is over all t samples. Similar definitions hold for conditional and total variance.

3.1 The OEUVRE Estimator

The OEUVRE estimator provides a sequence of estimates $\{L_t\}$, where L_t is an estimate of $\mathbb{E}_{z\sim\mathcal{D}}[\ell_t(z)]$. The estimator is recursively defined as follows:

$$\begin{aligned} L_1 &:= \ell_1(z_1) \\ L_t &:= \ell_t(z_t) + (1-\gamma_t)\left[L_{t-1} - \ell_{t-1}(z_t)\right] \end{aligned} \tag{1}$$

Here, $\ell_t(z_t)$ is the evaluation of the performance of f_t on the future sample (z_t), and $\ell_{t-1}(z_t)$ is evaluating the performance of f_{t-1} on z_t. $(\gamma_t)_{t\geq 2}$ is a sequence of *pre-determined* weights such that each γ_t lies in $[0, 1]$. We observe that each function f_t is evaluated on two samples, z_t and z_{t+1}. Both these samples lie outside of the training data of f_t, since it is trained on $\{z_1, \ldots, z_{t-1}\}$. Conversely, each sample z_t is used for evaluation for two functions, f_{t-1} and f_t. This is in stark contrast with previous evaluation methods, which only evaluate each sample on exactly one function.

Algorithm 1 summarizes the OEUVRE estimator. Each recursive update of the OEUVRE estimator requires $\mathcal{O}(1)$ time and memory, making it highly efficient. We emphasize the order of evaluation and function updates: we first calculate $\ell_{t-1}(z_t)$ using f_{t-1}; then update f_{t-1} to f_t using z_{t-1}; finally, calculate $\ell_t(z_t)$ using f_t. Thus, at time step t, the incoming sample z_t is used only for evaluation. z_t is used for training at the next time step $t+1$. This ordering is a feature of prequential evaluation, where the incoming sample is used for testing and estimating the loss of the learned function at that time step.

We define $M_t = L_t - \mathbb{E}[\ell_t(Z) \mid \mathcal{F}_{t-1}]$, which is our

Algorithm 1 OEUVRE estimator

Input: Data points $z_t = (x_t, y_t)$, loss function ℓ, learning algorithm $A : \mathcal{G} \times (\mathcal{X} \times \mathcal{Y}) \to \mathcal{G}$
Parameters: $c_0 = 1, b_0 = 2$, burn-in period t_0, small constant ϵ
$\hat{c} \leftarrow c_0$, $\hat{b} \leftarrow b_0$, $C = \emptyset$, $B = \emptyset$
for $t \in \mathbb{N}$ **do**
  // Initialize/reset after burn-in
  if $t = 1$ or $t = t_0$ **then**
    $L_t \leftarrow \ell_t(z_t) = \ell(f_t(z_t))$
    continue
  // Evaluation and function update
  $\ell_{t-1}(z_t) \leftarrow \ell(f_{t-1}(x_t, y_t))$
  $f_t \leftarrow A(f_{t-1}, z_{t-1})$
  $\ell_t(z_t) \leftarrow \ell(f_t(x_t, y_t))$
  // OEUVRE estimate update
  Set γ_t^* with $\sigma_t = \hat{c}r(t)$, $b = \hat{b}$ in Equation 3
  $L_t = \ell_t(z_t) + (1 - \gamma_t^*)(L_{t-1} - \ell_{t-1}(z_t))$
  // Estimated constants update (Sec. 6)
  if $t < t_0$ **then**
    $C \leftarrow C \cup (\ell_t(z_t) - \ell_{t-1}(z_t))/r(t)$
    $B \leftarrow B \cup \ell_t(z_t)$
    $\hat{c}^2 = \max(\epsilon, \widehat{\mathrm{Var}}(C))$, $\hat{b}^2 = \max(\epsilon, \widehat{\mathrm{Var}}(B))$

estimator L_t centered around the conditional expectation of the loss. This centered sequence is crucial to our analysis, L_t accurately estimating the conditional mean $\mathbb{E}[\ell_t(Z) \mid \mathcal{F}_{t-1}]$ means that M_t is close to zero.

Proposition 1. *(Unbiasedness in total expectation) For all $t \geq 1$, $\mathbb{E}[M_t] = 0$. Thus, $\mathbb{E}[L_t] = \mathbb{E}[\ell_t(Z)]$.*

This result is a consequence of a martingale constructed using M_t (Proposition 6 in Section 5). Thus, our estimator is unbiased in *total expectation*. This is true irrespective of the choice of (γ_t). While the estimator is not necessarily conditionally unbiased, we later show that with the right choice of (γ_t), L_t is a consistent estimator for $\mathbb{E}[\ell_t(Z) \mid \mathcal{F}_{t-1}]$.

Choice of (γ_t)**.** Intuitively, γ_t controls how much of the information from previous evaluations is carried over to the next time step. A lower γ_t implies more weight being assigned to the past estimate, which is desirable if f_t is not too different from f_{t-1}. On the other hand, γ_t being close to one reduces the impact of the past estimate on the current one. This is desirable if f_t is very different from f_{t-1}, and a higher γ_t translates to less weight being given to the 'memory' consisting of past evaluations.

A natural choice is setting the weight sequence (γ_t) such that the variance of the estimator is minimized. The following proposition establishes a recursion-based upper bound on the total variance of L_t.

Proposition 2. *Let*

$$\sigma_t^2 \geq \sup_{(z_1,\ldots,z_{t-1})} Var(\ell_t(X,Y) - \ell_{t-1}(X,Y) \mid \mathcal{F}_{t-1})$$

be a deterministic sequence. Define the following recursion:

$$\begin{aligned} V_1^{\uparrow} &:= b^2 \\ V_t^{\uparrow} &:= (\gamma_t b + (1-\gamma_t)\sigma_t)^2 + (1-\gamma_t)^2 V_{t-1}^{\uparrow} \end{aligned} \tag{2}$$

Then, for all $t \geq 1$, 1) $Var(M_t) \leq V_t^{\uparrow}$, and 2) $\sum_{i=1}^{t} Var(M_i \mid \mathcal{F}_{i-1}) \leq V_t^{\uparrow}$.

We prove this result in Appendix B.2. $V_t^{\uparrow}$ acts as an upper bound on both the total variance and the sum of conditional variances of M_t. We note that this recursion is in terms of $\sigma_t^2 \geq \sup \mathrm{Var}(\ell_t(Z) - \ell_{t-1}(Z) \mid \mathcal{F}_{t-1})$. This is an upper bound on the conditional variance of the difference between the losses of two consecutive functions evaluated on the same point, $\ell_t(Z)$ and $\ell_{t-1}(Z)$. The key challenge in finding the right γ_t is knowing σ_t^2. As we shall see in Section 4, σ_t^2 can be upper-bounded using results from algorithmic stability. For a large variety of online learning algorithms, we can show that there is a known sequence (σ_t) which satisfies the condition in Proposition 2 for all t. Since we require (γ_t) to be a deterministic sequence, we make use of these upper bounds to obtain our optimal (γ_t).

Equation 2 is quadratic in γ_t, and solving for γ_t minimizing $V_t^{\uparrow}$ at each step gives us the following optimal choice of γ_t:

$$\gamma_t^* = \begin{cases} 0, & V_{t-1}^{\uparrow} \leq \sigma_t(b - \sigma_t) \\ 1, & \sigma_t \geq b \\ \frac{V_{t-1}^{\uparrow} - \sigma_t(b-\sigma_t)}{V_{t-1}^{\uparrow} + (b-\sigma_t)^2}, & \text{otherwise} \end{cases} \tag{3}$$

Intuitively, the first case corresponds to a situation where σ_t is moderately large when compared to V_t. We can then set $\gamma_t = 1$, which means the past estimate is not down-weighted. On the other hand, σ_t is very large in the second case, corresponding to a sudden change in the learned function. We then set $\gamma_t = 1$, effectively resetting the estimator. The third case corresponds to a situation where σ_t is of an appropriate size, and we set γ_t to a value in $(0, 1)$ to minimize the variance. Using the optimal γ_t in Equation 2, the optimal variance bound is:

$$V_t^{\uparrow} = \begin{cases} \sigma_t^2 + V_{t-1}, & V_{t-1}^{\uparrow} \leq \sigma_t(b - \sigma_t) \\ b^2, & \sigma_t \geq b \\ \frac{b^2 V_{t-1}^{\uparrow}}{V_{t-1}^{\uparrow} + (b-\sigma_t)^2}, & \text{otherwise} \end{cases} \tag{4}$$

We now show that the estimator is consistent with the optimal choice of γ_t.

Lemma 3. *(Consistency) If $\sigma_t \to 0$, then with the optimal choice of $\{\gamma_t^*\}$, $\gamma_t^* \to 0$. Moreover, $Var(M_t) \to 0$, i.e., $L_t - \mathbb{E}\left(\ell_t(Z) \mid \mathcal{F}_{t-1}\right) \xrightarrow{L^2} 0$.*

We prove this result in Appendix B.3. Lemma 3 tells us that OEUVRE converges to the *conditional expectation* of the loss, i.e., the expected loss of the currently learned function, in L^2. We thus have a stronger consistency result while using a weaker assumption than previous work on prequential estimation (Gama et al., 2013), where it is assumed that $\ell_t \to \ell^*$ to show that the prequential estimator converges to $\mathbb{E}[\ell^*(Z)]$.

We can also obtain closed-form upper bounds for convergence rates:

Theorem 4. *Let*

$$\gamma_t = \begin{cases} 1/t & \text{if } \sigma_t = o(1/t) \text{ or } \sigma_t = 1/t, \\ \Omega(\sigma_t) & \text{otherwise} \end{cases} \tag{5}$$

Then, with this choice of γ_t in Equation 2, we have $Var(M_t) \leq V_t^{\uparrow} = \mathcal{O}(\gamma_t) = \mathcal{O}(\max\{1/t, \sigma_t\})$.

We prove this theorem in Appendix B.4. The γ_t designed above can potentially be a non-optimal choice of weights. Since γ_t^* is chosen to minimize the variance, $\mathrm{Var}(M_t)$ with the optimal choice of γ_t^* would also be upper bounded by this rate. This result establishes the connection between the stability of the learning algorithm and the closed-form convergence rate of the OEUVRE estimator.

To build intuition, we consider the simple example of a sequence of static functions $f_t = f$, where $\sigma_t = \mathrm{Var}(\ell_t(X, Y) - \ell_{t-1}(X, Y)) = 0$ for all t. Using this sequence of σ_t in Equation 3 gives us $\gamma_t = 1/t$. As $f_t(X_t) = f_{t-1}(X_t) = f(X_t)$, the OEUVRE estimator becomes the empirical mean of the past evaluations. Intuitively, this corresponds to 'best-case' behavior where there is no change between subsequent functions and we can expect the decay rates for γ_t^* to be slower than $1/t$ when f_t's evolve with time.

4 LEARNING ALGORITHMS WITH KNOWN STABILITY RESULTS

In Proposition 2, we need information about $\sigma_t^2 > \sup \mathrm{Var}(\ell_t(Z) - \ell_{t-1}(Z)) \mid \mathcal{F}_{t-1}$ to determine γ_t. These bounds can be readily obtained from results on *uniform stability* (Bousquet and Elisseeff, 2002; Bousquet et al., 2020; Saha et al.).

Definition 5. *(Uniform stability): Let $\mathcal{A}$ be a learning algorithm, and (f_t) be the sequence of functions learned by $\mathcal{A}$, with f_t being learned from samples $\{z_s\}_{s=1}^{t-1}$. Then, $\mathcal{A}$ is β_t-uniformly stable if, for all t and for all sequence of samples $\{z_s\}_{s=1}^{t}$,*

$$\|\ell(f_t(x), y) - \ell(f_{t-1}(x), y)\|_\infty \leq \beta_t$$

Informally, uniform stability bounds the degree of change that any incoming sample can have on the loss of the learned function at any point (x, y). Since this is a uniform bound, we have $\beta_t^2 > \mathrm{Var}(\ell_t(Z) - \ell_{t-1}(Z))$ for all t, which means that we can use $\sigma_t^2 = \beta_t^2$ in our variance recursion (Equation 2), calculate the optimal weight sequence γ_t^* (Equation 3), and use these weights in our estimator (Equation 1).

Our theory (Lemmas 3, Theorem 7) only requires bounds on $\sup \mathrm{Var}(\ell_t(Z) - \ell_{t-1}(Z) | \mathcal{F}_{t-1})$, which is a bound on the L^2 norm. This is a weaker condition than uniform stability, since the L^∞ norm is at least as large as the L^2 norm. However, we utilize uniform stability bounds because they are well-established for common algorithms (Table 1). When available, L^2 stability bounds can be directly plugged into our framework, yielding tighter variance recursions and faster convergence rates. The development of systematic L^2 stability theory for online learning would immediately translate to improved OEUVRE guarantees.

Uniform stability bounds are known for a wide variety of popular online learning algorithms. We state some important examples in Table 1. While we present the order of the stability results, exact stability rates have multiplicative constants based on the Lipschitz constant of the loss function and the hypothesis class, the strong convexity of the loss function, bounds on the function values, etc. Most of these results were established in (Saha et al.). As we elaborate in Section 6, the rate constants need not be known in practice and can be estimated to achieve good empirical performance.

5 CONVERGENCE OF (L_t)

Martingale structure. We recall that $M_t = L_t - \mathbb{E}[\ell_t(z) \mid \mathcal{F}_{t-1}]$ is the centered version of our estimator L_t. We also define $\Gamma_t = \prod_{i=2}^{t}(1 - \gamma_i)$, along with $\Gamma_1 = 1$. This product naturally arises from expanding the OEUVRE recursion (Equation 1), representing the accumulation of the decay factors from all steps of the recursion. Since we assume that (γ_t) is a predetermined sequence, (Γ_t) is also deterministic.

Proposition 6. *$(M_t/\Gamma_t)_{t\geq 1}$ is a martingale w.r.t the canonical filtration $(\mathcal{F}_t)_{t\geq 0}$, with $\mathbb{E}[M_1/\Gamma_1 \mid \mathcal{F}_0] = 0$*

We prove this result in Appendix B.1. This martingale structure allows us to use tools from martingale theory to prove strong convergence results.

Algorithmic Paradigm	**Example Algorithms**	**Order of Stability Bound**
Follow the Leader (FTL)	Online Convex Programming (strongly convex functions)	$\mathcal{O}(1/t)$
Follow the Regularized Leader (FTRL)	Online ridge regression	$\mathcal{O}(1/\sqrt{t})$
Regularized Dual Averaging (RDA)	–	$\mathcal{O}(1/\sqrt{t})$
Implicit Online Learning (IOL)	Implicit Mirror Descent	$\mathcal{O}(\eta_t)$
Online Mirror Descent (OMD)	Online gradient descent, Hedge	$\mathcal{O}(\eta_t)$
OMD (with Polyak Averaging)	Online gradient descent	$\mathcal{O}(1/t)$

Table 1: Stability results for popular online learning paradigms (Saha et al.), where η_t corresponds to the learning rate schedule chosen for the algorithm and is typically set to be $\Theta(1/\sqrt{t})$. We recommend choosing σ_t to have the same decay order as the order of the stability bound, with the constant factors estimated adaptively (Section 6).

5.1 Asymptotic Convergence

Theorem 7. *Let v^* be a random variable taking values in $[\nu^2, b^2]$, where $\nu > 0$. Let the following conditions hold:*

(i) $Var(\ell_t \mid \mathcal{F}_{t-1}) \in [\nu^2, b^2]$.

(ii) $Var(\ell_t \mid \mathcal{F}_{t-1}) \xrightarrow{p} v^*$.

(iii) $(\ell_t(z) - (1-\gamma_t)\ell_{t-1}(z))$ has bounded kurtosis.

(iv) The weight sequence (γ_t) satisfies Equation 5 with the modified condition that $\gamma_t = \omega(\sigma_t)$.

Then, $M_t/\sqrt{V_t} \xrightarrow{d} \mathcal{N}(0,1)$, where $V_t = \sum_{i=1}^t Var(M_i \mid \mathcal{F}_{i-1})$.

We prove this theorem in Appendix B.5. Conditions (i) and (ii) ensure that the conditional variances of the losses converge to some non-degenerate random variable. Condition (iii) is a mild restriction on the kurtosis of consecutive difference, and should be satisfied by loss distributions with light tails, such as sub-Gaussian and sub-exponential distributions. Condition (iv) is a restriction on the weight sequence $\{\gamma_t\}$, which is similar to that in Theorem 4. We discuss the conditions in detail in Appendix B.5. While V_t might not be known in practice, it can be upper-bounded by the recursive variance $V_t^{\uparrow}$ from Proposition 2, which can be used to get asymptotically valid (though more conservative) confidence intervals.

5.2 Fixed-time and Time-uniform Concentration Bounds

Uniform stability bounds can also be used to establish concentration bounds for M_t:

Theorem 8. *Let the loss function ℓ be bounded in $[0, b]$. Let $\{\sigma_t\}$ be a sequence upper-bounding the uniform stability of the function sequence learned by algorithm $\mathcal{A}$. Let γ_t be defined as in Equation 3 or 5. Then,*

(a) (Fixed-time bound) For any $\epsilon > 0$ and $t \in \mathbb{N}$,

$$\mathbb{P}\left(|M_T| \geq \epsilon\right) \leq 2\exp\left(-\frac{\epsilon^2}{2V_T^{\uparrow}}\right)$$

(b) (Time-uniform bound) For any $c > 0$ and $T \in \mathbb{N}$

$$\mathbb{P}\left(\exists t \in [T] : |M_t| \geq \sqrt{\frac{c}{2V_T^{\uparrow}}} \cdot \left(\frac{V_T^{\uparrow}\Gamma_t}{\Gamma_T} + \frac{V_t^{\uparrow}\Gamma_T}{\Gamma_t}\right)\right) \leq 2\exp\left(-c\right)$$

We prove this theorem in Appendix B.6. Part (a) is an application of the Azuma-Hoeffding inequality (Wainwright, 2019a), and part (b) is applies the master theorem from (Howard et al., 2020) to the martingale (M_t/Γ_t) and minimizing the bound for the final time step T. The confidence interval in part (a) has width $\mathcal{O}(\sqrt{V_t^{\uparrow}})$ at time t, and the confidence sequence in part (b) decreases in size with time, having width $\mathcal{O}(\sqrt{V_T^{\uparrow}})$ at the final time step T. Similar results can be obtained for other distributional assumptions on the loss using results in (Howard et al., 2020).

Interpretation of convergence results. We note that Theorems 7 and 8 establish the rate of convergence of L_t to the *conditional expectation* of the loss, which is the expected loss of the currently learned function. If the function class $\mathcal{G}$ for the learning algorithm is such that exact bounds on the uniform stability are known, then Theorems 7 and 8 can be used to construct confidence intervals/sequences for the OEUVRE estimator. On the other hand, if we only know the *order* of uniform stability instead of exact bounds, the above theorems still show that OEUVRE exhibits strong convergence properties, since a stability rate of order β_t implies convergence rates and concentration of order $\sqrt{\beta_t}$ from Theorem 4.

In our running example of a static sequence of functions, this corresponds to a width of The extreme instance of static functions $f_t = f$ results in the well-known $\mathcal{O}(1/\sqrt{T})$ rate for confidence intervals/sequences. The $\mathcal{O}(1/\sqrt{T})$ rate also holds for learning algorithms with $\mathcal{O}(1/T)$ stability rates, which holds from FTL with ERM and SGD for strongly convex loss functions with a learning rate decay of the order $1/t$. On the other hand, algorithms with $\mathcal{O}(1/\sqrt{T})$ stability rates such as FTRL, OMD, and RDA result in intervals/sequences with width $\mathcal{O}(1/T^{1/4})$.

6 ADAPTIVE CONSTANT ESTIMATION

OEUVRE requires two types of constants: 1) the upper bound on the variance of loss function evaluations, b^2, and 2) the constants for the stability rate $\sigma_t = cr(t)$, where $r(t)$ is the rate of uniform stability and $c > 0$ is the associated constant. In practice these constants are unknown, and we need to know additional properties of the function class and loss functions to estimate them. However, we can estimate these practically without affecting the rate of convergence of the estimator. We begin by showing that the rate of convergence of is not impacted by misspecifying σ_t by a constant factor.

Lemma 9. *Let $\sigma_t = cr(t)$, where $c > 0$. . Let $V_t^{\uparrow}$ be the recursion from Equation 2.*

$$V_1^{\uparrow} = b^2, \qquad V_t^{\uparrow} = (\gamma_t b + (1-\gamma_t)cr(t))^2 + V_{t-1}^{\uparrow}$$

Let $V_t^{\dagger}$ be the recursion from Equation 2 with the same weight sequence (γ_t) but with misspecified constants $\hat{b}, \hat{c} > 0$ instead of b, c.

$$V_1^{\dagger} = \hat{b}^2, \qquad V_t^{\dagger} = (\gamma_t \hat{b} + (1-\gamma_t)\hat{c}r(t))^2 + V_{t-1}^{\dagger}$$

Then, $V_t^{\uparrow} \leq V_t^{\dagger} / \left(\min\left\{1, (\hat{c}/c)^2\right\} \cdot \min\left\{1, (\hat{b}/b)^2\right\}\right)$.

We prove this lemma in Appendix B.7. We can improve the empirical performance of OEUVRE by approximating the constants using the first k samples, setting $\hat{b}^2$ and $\hat{c}^2$ as the empirical variances over $B = \{\ell_t(z_t)\}_{t=1}^k$ and $C = \{(\ell_t(z_t) - \ell_{t-1}(z_t))/r_t\}_{t=1}^k$ respectively.

We can then run OEUVRE initialized at sample $k+1$ with $\hat{b}$ and $\hat{c}$. We observe empirically that adaptive estimation in this manner can result in significantly better performance, and all our experiments in Section 7 employ this method. Empirically, we observe that the first 30 time steps are sufficient to approximate $\hat{c}$ and $\hat{b}$ for good performance of OEUVRE in later time steps. By estimating these hyperparameters adaptively and providing stability order for the algorithm, we specify OEUVRE completely, and there are no other hyperparameters which need to be set.

7 EXPERIMENTS

We test the OEUVRE estimator through a series of experiments on diverse tasks consisting of synthetic and real-world datasets. For all experiments, we compare the performance of our estimator against the following baseline methods: 1) Sliding Window (SW), 2) Fading Factor Prequential (FFPreq) (Gama et al., 2013), 3) Exponential Moving Average (EMA), and 4) Adaptive Windowing (ADWIN) (Bifet and Gavalda, 2007). We evaluate the performance of the estimators against the ground truth expected loss using two metrics: the time-averaged Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) over all time steps. For all experiments, we use the first $t_0 = 30$ time steps as a burn-in period to initialize the constants for OEUVRE; these time steps are included in the metric calculation. We repeat each experiment over 10 seeded runs.

For each experimental setting, we sweep over a broad range of hyperparameter values for each baseline estimator, choosing the setting with the lowest mean RMSE in hindsight. Calculating the RMSE to choose the best hyperparameter requires knowledge of the true expected loss–an oracle advantage that is unavailable in practice. This results in a very strong benchmark, and we compare the *best setting* from each baseline estimator against OEUVRE with adaptive tuning, which does not need explicit hyperparameter selection. Please refer to Appendix C for more details.

Linear Regression. We begin with a synthetic online linear regression task, where the data is generated from a linear model with additive Gaussian noise. For the learning algorithm, we use OGD with the learning rate schedule $\eta_0/\sqrt{t}$, where η_0 is the initial learning rate. We see in Table 1 that OGD with this learning rate is $1/\sqrt{t}$-uniformly stable. The goal is to estimate the scaled mean squared error (MSE) of the linear model. We run each experiment for $T = 10,000$ time steps, varying both the number of dimensions and the noise level across experiments. We present our results in Figure 2, where we compare OEUVRE with the best baseline estimator and the median baseline across all methods and hyperparameters.

We observe that OEUVRE achieves competitive mean RMSE when compared against the best baseline across dimensions and noise levels. While the difference in performance is not large, it is significant given that the competitor is chosen from broad hyperparameter sweeps of four different baseline methods. The MAE for OEUVRE is also very close to the best baseline, being significantly lower than the median baseline for smaller dimensions ($d = 25 - 100$).

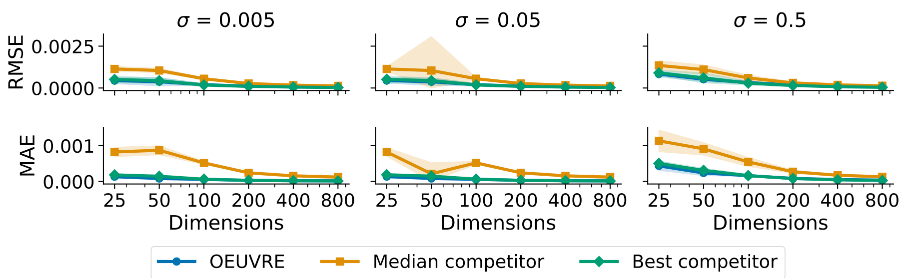


Figure 2: Performance comparison of OEUVRE against the best baseline and the median baseline for the online linear regression task. OEUVRE achieves competitive RMSE and bias when compared to the best baseline.

Prediction with Expert Advice. We then consider the prediction with expert advice task, where we have K experts, each making a prediction at all time steps. We draw the loss incurred by each expert from either a Bernoulli or a Beta distribution in an i.i.d. manner. We use the Hedge algorithm (Mourtada and Gaiffas, 2019) with a learning rate of $\eta = \sqrt{\log(K)/t}$ to learn the distribution over experts. From Table 1, we see that this algorithm is $1/\sqrt{t}$-uniformly stable. The goal is to estimate the expected loss of the currently learned distribution over experts. We run each experiment for $T = 10,000$ time steps, varying both the number of experts and the loss distributions across experiments. We compare OEUVRE with the best baseline estimator and the median baseline across all methods and hyperparameters in Figure 3a.

We observe that for Beta-distributed losses, the OEUVRE estimator achieves significantly lower RMSE when compared to the best baseline, while achieving comparable MAE. For Bernoulli-distributed losses, the RMSEs are comparable for smaller dimensions ($d = 25, 50$), after which OEUVRE outperforms the best baseline. The MAE for OEUVRE is closer to the median baseline for smaller dimensions, becoming comparable to the best baseline for larger dimensions.

Neural Networks. Having validated OEUVRE on tasks with known stability bounds, we now stress-test whether its properties extend to settings beyond our theoretical coverage. We consider two popular image classification tasks, MNIST (Deng, 2012) and EMNIST-digits (Cohen et al., 2017), and train a simple neural network on them with AdamW. The goal is to estimate the expected cross-entropy (CE) loss on the test set, which we approximate by evaluating the model on 1024 randomly drawn test samples at each time step. To update the estimators at each time step, we evaluate the model on 128 randomly drawn test samples and calculate the mean CE loss. We run each experiment for 10 epochs.

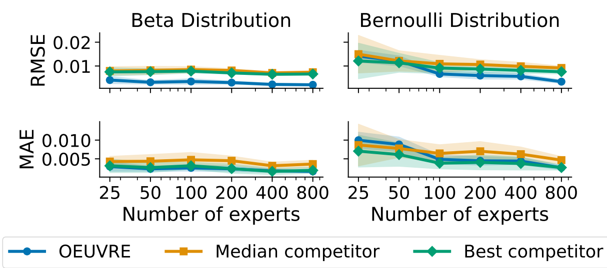
Though stability results are difficult to establish for neural networks with adaptive learning rates, we include these experiments to demonstrate OEUVRE's practical robustness beyond its theoretical guarantees. We approximate σ_t as $C' \|\eta_t \odot g_t\|_\infty$, where η_t is a vector of the learning rate per neuron, and g_t is the gradient vector. This heuristic proxy is guided by the intuition that, assuming the neural network is C-Lipschitz, $|\ell_t - \ell_{t-1}| \leq C \|w_t - w_{t-1}\|_2 \leq C' \|\eta_t \odot g_t\|_\infty$, where w_t is the vector of neuron weights at time t. Thus, the L^∞ norm acts as a bound on the extent of function updates. We present our results in Figure 3b.

We observe that for both MNIST and EMNIST-digits, OEUVRE achieves lower mean RMSE and MAE than the competing methods. In Figure 7, we observe that the time-averaged bias of OEUVRE is significantly lower than those of the competitors, indicating that its estimate follows the true expected loss more closely. However, the variance across runs for the bias for MNIST is high, indicating that the heuristic σ_t approximation may be inaccurate for adaptive optimizers. Further investigation of stability-based tuning for adaptive learning rates remains important future work. We observe similar results when reducing the number of test samples shown to the estimators at each time step from 128 to 8 (Figure 4).

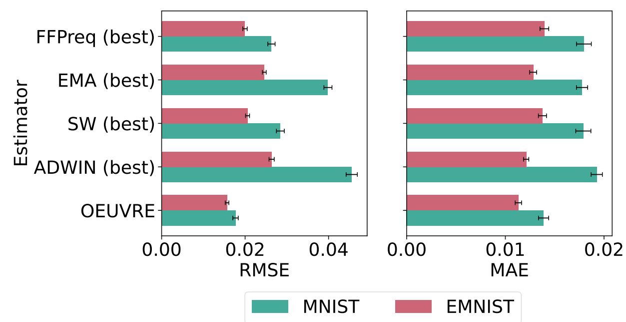
8 DISCUSSION

Our work suggests several directions for future research. The theoretical guarantees for OEUVRE currently assume i.i.d. data; we conduct some experiments on datasets with drift in Appendix E, and OEUVRE performs reasonably well for gradual and seasonal drift, with performance degrading in cases with abrupt drift. Extending OEUVRE to non-stationary settings via mixing assumptions on the data or combining OEUVRE with windowing is a key next step.

Our analysis also requires pre-determined weight sequences, excluding algorithms with adaptive learning rates like Adam, though our experiments suggest heuristic approximations can be effective in practice. Finally, our framework relies on convenient L^∞ uniform stability bounds, but could immediately benefit from the development of tighter, systematic L^2 sta-



(a)



(b)

Figure 3: Comparison of OEUVRE with best-performing baseline methods for (a) prediction with expert advice task and (b) neural networks task. In (a), it achieves smaller RMSE and comparable MAE when compared to the best baseline. In (b), it achieves the lowest RMSE and MAE for both datasets.

bility theory for online learning, which would directly improve our variance bounds and convergence rates.

Acknowledgement

This material is based upon work supported by the AI Research Institutes Program funded by the National Science Foundation under the AI Institute for Societal Decision Making (NSF AI-SDM), Award No. 2229881.

References

Manuel Baena-Garcıa, José del Campo-Ávila, Raul Fidalgo, Albert Bifet, Ricard Gavalda, and Rafael Morales-Bueno. Early drift detection method. In *Fourth international workshop on knowledge discovery from data streams*, volume 6, pages 77–86, 2006.

Albert Bifet and Ricard Gavalda. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining*, pages 443–448. SIAM, 2007.

Vivek S Borkar and Vivek S Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 100. Springer, 2008.

Zdravko Botev and Ad Ridder. Variance reduction. *Wiley statsRef: Statistics reference online*, 136:476, 2017.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.

Olivier Bousquet, Yegor Klochkov, and Nikita Zhivotovskiy. Sharper bounds for uniformly stable algorithms. In *Conference on Learning Theory*, pages 610–626. PMLR, 2020.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

A Philip Dawid. Present position and potential developments: Some personal views statistical theory the prequential approach. *Journal of the Royal Statistical Society: Series A (General)*, 147(2):278–290, 1984.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

Denis Moreira Dos Reis, Peter Flach, Stan Matwin, and Gustavo Batista. Fast unsupervised online drift detection using incremental kolmogorov-smirnov test. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1545–1554, 2016.

Hadi Fanaee-T. Bike Sharing. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5W894.

Dylan J Foster, Satyen Kale, Mehryar Mohri, and Karthik Sridharan. Parameter-free online learning via model selection. *Advances in Neural Information Processing Systems*, 30, 2017.

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

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

Robert M Gower, Mark Schmidt, Francis Bach, and Peter Richtá rik. Variance-reduced methods for machine learning. *Proceedings of the IEEE*, 108(11): 1968–1983, 2020.

Peter Hall and Christopher C Heyde. *Martingale limit theory and its application*. Academic press, 2014.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234. PMLR, 2016.

Steven R Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform chernoff bounds via nonnegative supermartingales. 2020.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

Koulik Khamaru, Eric Xia, Martin J Wainwright, and Michael I Jordan. Instance-optimality in optimal value estimation: Adaptivity via variance-reduced q-learning. *arXiv preprint arXiv:2106.14352*, 2021.

Tze Leung Lai. Stochastic approximation. *The annals of Statistics*, 31(2):391–406, 2003.

Jacob Montiel, Max Halford, Saulo Martiello Mastelini, Geoffrey Bolmier, Raphael Sourty, Robin Vaysse, Adil Zouitine, Heitor Murilo Gomes, Jesse Read, Talel Abdessalem, et al. River: machine learning for streaming data in python. *Journal of Machine Learning Research*, 22(110):1–8, 2021.

Jaouad Mourtada and Stéphane Gaiffas. On the optimality of the hedge algorithm in the stochastic regime. *Journal of Machine Learning Research*, 20 (83):1–28, 2019.

Ewan S Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

Ankan Saha, Prateek Jain, and Ambuj Tewari. The Interplay Between Stability and Regret in Online Learning. URL `http://arxiv.org/abs/1211.6158`.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1):83–112, 2017.

Raquel Sebastião and José Maria Fernandes. Supporting the page-hinkley test with empirical mode decomposition for change detection. In *International symposium on methodologies for intelligent systems*, pages 492–498. Springer, 2017.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14(1): 567–599, 2013.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019a.

Martin J Wainwright. Variance-reduced q-learning is minimax optimal. *arXiv preprint arXiv:1906.04697*, 2019b.

Shengbo Wang, Nian Si, Jose Blanchet, and Zhengyuan Zhou. Sample complexity of variance-reduced distributionally robust q-learning. *Journal of Machine Learning Research*, 25(341):1–77, 2024.

Supplement for OEUVRE: OnlinE Unbiased Variance-Reduced Loss Estimation

A OEUVRE with Batching

The OEUVRE estimator can be modified to be used in batch settings, where the learner updates on a batch of B samples at every time step. This could be the case when training is expensive, which makes it preferable when enough samples have been gathered. It might also be that the distribution of samples over time is non-uniform, with most samples arriving in rapid succession and with large gaps between these spikes. In such situations, training might be done between the spikes of samples.

We assume that at each time step t, a batch of B i.i.d. samples $Z_t = \{z_{t1}, \ldots, z_{tB}\}$ arrives and is used to update the learned algorithm. Function f_t is trained on the past set of samples $Z_1 \cup \ldots \cup Z_{t-1}$, and we define $\ell_t = \ell \circ f_t$ as before. Our modified OEUVRE estimator is

$$L_t = \frac{1}{B}\sum_{i=1}^{B} \ell_t(z_{ti}) + (1-\gamma_t)\left[L_{t-1} - \frac{1}{B}\sum_{i=1}^{B} \ell_{t-1}(z_{ti})\right].$$

We note that since Z_t is evaluated on f_t and f_{t-1}, we need to maintain a cache of size B containing the future samples. Since our learned function is trained on one time step less than those available, the learning algorithms incurs only an additively higher cumulative regret at the benefit of more accurate estimation of their performance.

For most common algorithms (like FTL with ERM, FTL, and OMD), the batched versions of these algorithms have the same uniform stability for time step t as the single-sample version. Thus, the weight sequence $\{\gamma_t\}$ is identical in both cases. By averaging our loss evaluations over B i.i.d. samples, the variance of the estimator reduces by a factor of B. This reduction in variance by batching is also exhibited by other evaluation methods like prequential evaluation, sliding windown estimation and exponential moving average. We also observe that the batches need not be of the same size; the calculation of γ_t only depends on uniform stability guarantees which do not get worse with batching, and OEUVRE can be used with dynamic batch sizes while achieving similar convergence guarantees.

B Deferred proofs

B.1 Proof of Proposition 6

Proof. We define $\Delta_t(x, y) = \ell_t(x, y) - \mathbb{E}[\ell_t(x, y) \mid \mathcal{F}_{t-1}]$. We then have

$$\begin{aligned}\mathbb{E}[\Delta_t(x, y) \mid \mathcal{F}_{t-1}] &= \mathbb{E}[\ell_t(x, y) \mid \mathcal{F}_{t-1}] - \mathbb{E}[\ell_t(x, y) \mid \mathcal{F}_{t-1}] \\ &= 0\end{aligned}$$

Now, M_t can be expressed as

$$\begin{aligned}M_t &= L_t - \mathbb{E}[\ell_t(x, y) \mid \mathcal{F}_{t-1}] \\ &= \ell_t(x_t, y_t) - \mathbb{E}[\ell_t(x, y) \mid \mathcal{F}_{t-1}] - (1-\gamma_t)\left[\ell_{t-1}(x_t, y_t) - \mathbb{E}[\ell_{t-1}(x, y) \mid \mathcal{F}_{t-1}]\right] \\ &\quad + (1-\gamma_t)\left[L_{t-1} - \mathbb{E}[\ell_{t-1}(x, y) \mid \mathcal{F}_{t-1}]\right] \\ &= \Delta_t(x_t, y_t) - (1-\gamma_t)\Delta_{t-1}(x_t, y_t) + (1-\gamma_t)M_{t-1}\end{aligned}$$

Dividing throughout by Γ_t, we have

$$\frac{M_t}{\Gamma_t} = \frac{\Delta_t(x_t, y_t)}{\Gamma_t} - \frac{\Delta_{t-1}(x_t, y_t)}{\Gamma_{t-1}} + \frac{M_{t-1}}{\Gamma_{t-1}}$$

Taking the conditional expectation on both sides, we get

$$\begin{aligned}\mathbb{E}\left[\frac{M_t}{\Gamma_t} \mid \mathcal{F}_{t-1}\right] &= \mathbb{E}\left[\frac{\Delta_t(x_t, y_t)}{\Gamma_t} \mid \mathcal{F}_{t-1}\right] - \mathbb{E}\left[\frac{\Delta_{t-1}(x_t, y_t)}{\Gamma_{t-1}} \mid \mathcal{F}_{t-1}\right] + \mathbb{E}\left[\frac{M_{t-1}}{\Gamma_{t-1}} \mid \mathcal{F}_{t-1}\right] \\ &= \frac{M_{t-1}}{\Gamma_{t-1}}\end{aligned}$$

Thus, (M_t/Γ_t) is a martingale. Moreover, we have

$$\begin{aligned}\mathbb{E}\left[\frac{M_1}{\Gamma_1} \mid \mathcal{F}_0\right] &= \mathbb{E}[L_1 - \mathbb{E}[\ell_1(x, y)]] \\ &= \mathbb{E}[\ell_1(x, y)] - \mathbb{E}[\ell_1(x, y)] = 0\end{aligned}$$

□

B.2 Proof of Proposition 2

Proof. We have

$$\begin{aligned}\mathrm{Var}(M_t) &= \mathrm{Var}\left(\Delta_t(x_t, y_t) + (1-\gamma_t)[M_{t-1} - \Delta_{t-1}(x_t, y_t)]\right) \\ &= \mathrm{Var}\left(\Delta_t(x_t, y_t) - (1-\gamma_t)\Delta_{t-1}(x_t, y_t) + (1-\gamma_t)M_{t-1}\right) \\ &= \mathrm{Var}\left(\Delta_t(x_t, y_t) - (1-\gamma_t)\Delta_{t-1}(x_t, y_t)\right) + (1-\gamma_t)^2\mathrm{Var}(M_{t-1}) + 2\mathbb{E}\left[\left(\Delta_t(x_t) - (1-\gamma_t)\Delta_{t-1}(x_t)\right) M_{t-1}\right]\end{aligned}$$

The third time can be simplified as

$$\begin{aligned}&2\mathbb{E}\left[\mathbb{E}\left[\left(\Delta_t(x_t) - (1-\gamma_t)\Delta_{t-1}(x_t)\right) M_{t-1}\right] \mid \mathcal{F}_{t-1}\right] \\ &\quad = 2\mathbb{E}\left[M_{t-1}\mathbb{E}\left[\left(\Delta_t(x_t) - (1-\gamma_t)\Delta_{t-1}(x_t)\right)\right] \mid \mathcal{F}_{t-1}\right] \\ &\quad \overset{(i)}{=} 0,\end{aligned}$$

where (i) comes from $\mathbb{E}[\Delta_t(x_t) \mid \mathcal{F}_{t-1}] = 0$ and $\mathbb{E}\Delta_{t-1} \mid \mathcal{F}_{t-1} = 0$. Thus, our variance simplifies to

$$\begin{aligned}\mathrm{Var}(M_t) &= \mathrm{Var}\left(\Delta_t(x_t, y_t) - (1-\gamma_t)\Delta_{t-1}(x_t, y_t)\right) + (1-\gamma_t)^2\mathrm{Var}(M_{t-1}) \\ &\leq \left(\sqrt{\mathrm{Var}(\gamma_t\Delta_t(x_t, y_t))} + \sqrt{\mathrm{Var}((1-\gamma_t)(\Delta_t(x_t, y_t) - \Delta_{t-1}(x_t, y_t)))}\right)^2 + (1-\gamma_t)^2\mathrm{Var}(M_{t-1}) \\ &\leq (\gamma_t b + (1-\gamma_t)\sigma_t)^2 + (1-\gamma_t)^2\mathrm{Var}(M_{t-1})\end{aligned}$$

□

B.3 Proof of Lemma 3

Proof. We first show that $V_t \to 0$, followed by $\gamma_t \to 0$.

Showing $V_t \to 0$. Since $\sigma_t \to 0$, there is some $t_0(\epsilon)$ such that for all $t \geq t_0(\epsilon)$, $\sigma_t < \epsilon$. We now show that there should exist some $t_V(\epsilon) \geq t_0(\epsilon)$ such that, for all $t \geq t_V(\epsilon)$, $V_t < 2b\epsilon$. We divide our analysis into three cases:

(a) Case 1, $V_{t-1} \leq \sigma_t(b - \sigma_t)$: In this case,

$$V_t \leq V_{t-1} + \sigma_t^2 \leq b\sigma_t < 2b\epsilon$$

(b) Case 2, $\sigma_t > b$: In this case,

$$V_t \leq b^2 < b\sigma_t < 2b\epsilon$$

(c) Case 3, none of the above: For all $t \geq t_0(\epsilon)$, we also have

$$V_t \leq \frac{b^2 V_{t-1}}{V_{t-1} + (b - \sigma_t)^2} \leq \frac{b^2 V_{t-1}}{V_{t-1} + (b - \epsilon)^2}$$

Thus, the third term is an upper bound on V_t. Now, consider another recursion W_t, $t \geq t_0(\epsilon)$, defined as

$$W_t = \begin{cases} V_{t_0(\epsilon)}, & t = t_0(\epsilon) \\ \frac{b^2 W_{t-1}}{W_{t-1} + (b-\epsilon)^2}, & \text{otherwise} \end{cases}$$

We have $W_{t_0(\epsilon)} = V_{t_0(\epsilon)}$, and $W_{t_0(\epsilon)+1} \geq V_{t_0(\epsilon)+1}$. Moreover, the following function $f(x)$ is strictly increasing in x:

$$f(x) = \frac{b^2 x}{x + (b - \epsilon)^2}$$

Thus, by induction, $W_t \geq V_t \forall t \geq t_0(\epsilon)$. Thus, if we show that there is some $t_V(\epsilon)$ such that for all $t \geq t_V(\epsilon)$, $W_t < b\epsilon$, then $V_t < b\epsilon$ for all $t \geq t_V(\epsilon)$.

Now, the definition of W_t is of the nature of a fixed point recursion in W_t. We calculate the fixed points of this recursion:

$$\begin{aligned} x &= \frac{b^2 x}{x + (b - \epsilon)^2} \\ x(x - (b^2 - (b - \epsilon)^2)) &= 0 \\ x(x - \epsilon(2b - \epsilon)) &= 0 \end{aligned}$$

Thus, the two fixed points of the recursion are 0 and $\epsilon(2b - \epsilon)$.

For all $x \in (0, \epsilon(2b - \epsilon))$, we have $x < f(x)$. Thus, x is mapped to a value higher than itself. For all $x > \epsilon(2b - \epsilon)$, we have $x > f(x)$. Thus, x is mapped to a value lower than itself. This indicates that $\epsilon(2b - \epsilon)$ is the 'attracting' fixed point for $f(x)$. Thus, $W_t \to \epsilon(2b - \epsilon)$.

This means that there should be some $t_V(\epsilon)$ such that for all $t \geq t_V(\epsilon), W_t < 2b\epsilon$. This concludes the case-wise analysis.

Over our three cases, our choice of ϵ was arbitrary. Thus, we should have $V_t \to 0$.

Showing $\gamma_t \to 0$. We observe the recursion for γ_t^* in the non-trivial case, i.e., when $\gamma_t \notin \{0, 1\}$:

$$\gamma_t^* = \frac{V_{t-1} - \sigma_t(b - \sigma_t)}{V_{t-1} + (b - \sigma_t)^2}$$

Now, as $\sigma_t \to 0$ and $V_t \to 0$, we observe that the numerator tends to zero, and the denominator tends to b^2. Thus, $\gamma_t^* \to 0$. □

B.4 Proof of Theorem 4

We prove this result by establishing a result for non-optimal choices for γ_t

Non-optimal choices of γ_t. Although the optimal γ_t^* sequence results in the lowest variance, it might be difficult to establish the convergence rate for $V_t^{\uparrow}$ in closed form. The following lemma generalizes the consistency result for L_t for other choices of γ_t along with providing an upper bound on the convergence rate.

Lemma 10. *Let $\{\gamma_t\}$ be a weight sequence such that there exists some $t_0 \geq 1$ with $\gamma_t = \Omega(\sigma_t)$, and $1/\gamma_t - 1/\gamma_{t-1} \leq 1$ for all $t \geq t_0$. Then $\Gamma_t^2\, Var(M_t) \leq V_t^{\uparrow} = \mathcal{O}(\gamma_t)$.*

Our choice of γ_t in Theorem 4 ensures that the two conditions on γ_t are satisfied.

Proof. We prove the result for $t_0 = 1$ first. We extend the weight sequence to $t = 1$ by letting $\gamma_1 = 1$. $c = \sup_{t\geq 1} c_t/\gamma_t$. We prove that $V_t \leq (b+c)^2\gamma_t$ by induction. For t = 1, we have $V_1 = \text{Var}(\Delta_1) \leq b^2 \leq (b+c)^2 = (b+c)^2\gamma_1^2$. For $t > 1$, we have

$$
\begin{aligned}
1 &\overset{(i)}{\geq} \frac{1}{\gamma_t} - \frac{1}{\gamma_{t-1}} \\
\gamma_t\gamma_{t-1} &\geq \gamma_{t-1} - \gamma_t \\
\gamma_t &\geq \gamma_{t-1}(1-\gamma_t) \\
\gamma_t(1-\gamma_t) &\geq \gamma_{t-1}(1-\gamma_t)^2 \\
\gamma_t &\geq \gamma_t^2 + (1-\gamma_t)^2\gamma_{t-1} \\
(b+c)^2\gamma_t &\geq (b+c^2)\gamma_t^2 + (1-\gamma_t)^2(b+c)^2\gamma_{t-1} \\
&\geq (b\gamma_t + c\gamma_t)^2 + (1-\gamma_t)^2(b+c)^2\gamma_{t-1} \\
&\geq (b\gamma_t + \sigma_t)^2 + (1-\gamma_t)^2(b+c)^2\gamma_{t-1} \\
&\geq (b\gamma_t + (1-\gamma_t)\sigma_t)^2 + (1-\gamma_t)^2(b+c)^2\gamma_{t-1} \\
&\overset{(ii)}{\geq} (b\gamma_t + (1-\gamma_t)\sigma_t)^2 + (1-\gamma_t)^2V_{t-1} \\
&= V_t,
\end{aligned}
$$

where (i) is our assumption on $\{\gamma_t\}$, and (ii) comes from the induction hypothesis. Thus, we have $V_t \leq (b+c)^2\gamma_t$, which proves our result for $t_0 = 1$.

We can extend the result for $t_0 > 1$ by letting

$$
c = \max\left\{\sup_{t\geq t_0} \sigma_t/\gamma_t, \sqrt{\frac{V_{t_0}}{\gamma_{t_0}}} - b\right\}.
$$

This would mean that at time step t_0, we would have $V_{t_0} \leq (b+c)^2\gamma_{t_0}$, along with $\sigma_t \leq c\gamma_t$ for all $t \geq t_0$. We can then do induction on t starting from t_0 as done for $t_0 = 1$. □

B.5 Proof of Theorem 7

Elaborating on conditions. We discuss the conditions in detail here. Condition (i) assumes that the conditional variance of the loss at time t is both upper and lower bounded by positive constants. While the upper bound is something we assume originally, the lower bound typically holds in non-realizable settings. In realizable settings, it can be made to hold by inserting a small amount of white noise in the labels y_t, which would lead to similar convergence rates without impacting the estimator performance.

Condition (ii) states that the conditional variances converge in probability to some random variable v^*. This condition is satisfied by strongly convex loss landscapes, where the sequence of loss functions converge in probability to the optimal loss function. However, the condition we state is more general and can hold for non-convex settings, where the sequence of loss functions can be attracted to multiple local optima. Condition (iii) is a restriction on the kurtosis of the update of the OEUVRE estimator. Importantly, the kurtosis is total, i.e., it is calculated over the entire data sequence $\{Z_i\}_{i\leq t}$. The bound on the kurtosis need not be known, and this condition should be satisfied in most practical settings with light tails for the losses. Condition (iv) is a restriction on $\{\gamma_t\}$ similar to that in Lemma 10, with the chief difference being γ_t having to decay strictly slower than σ_t.

Proof of 7. We define a triangular array $\{U_{n,i}\}$ of normalized martingale differences, with $U_{n,1} = M_1/\overline{V}_n$, and

$$
U_{n,i} = \left[\frac{M_i}{\Gamma_i} - \frac{M_{i-1}}{\Gamma_{i-1}}\right]\frac{1}{\sqrt{\overline{V}_n}} = \left[\frac{\Delta_i(x_i,y_i)}{\Gamma_i} - \frac{\Delta_{i-1}(x_i,y_i)}{\Gamma_{i-1}}\right]\frac{1}{\sqrt{\overline{V}_n}},
$$

where $\Delta_i(\cdot) = \ell_t(\cdot) - \mathbb{E}[\ell_t(x,y) \mid \mathcal{F}_{t-1}$, and $\overline{V}_n = \mathrm{Var}(M_n/\Gamma_n)$. Since $U_{n,i}$ is a martingale difference sequence for each n,

$$\mathrm{Var}(\sum_{i=1}^{n} U_{n,i}) = \mathrm{Var}\left(\frac{M_n}{\Gamma_n} \cdot \frac{1}{\sqrt{\overline{V}_n}}\right) = 1,$$

which means that each $U_{n,i}$ is normalized to have variance 1.

We also define $\Lambda_1 = M_1/\Gamma_1 = M_1$, and

$$\Lambda_i = U_{n,i} \cdot \sqrt{\overline{V}_n} = \left[\frac{\Delta_i(x_i)}{\Gamma_i} - \frac{\Delta_{i-1}(x_i)}{\Gamma_{i-1}}\right],$$

which is the un-normalized martingale difference. Clearly, we have

$$M_n = \sum_{i=1}^{n} \Lambda_i \implies \overline{V}_n = \mathrm{Var}(M_n) = \sum_{i=1}^{n} \mathrm{Var}(\Lambda_i).$$

We now state three lemmas which will help us prove the theorem.

Lemma 11. *Let $Var(\Delta_i(x,y)) \to \overline{v}^*$ and $Var(\ell(x,y)) \in [\nu^2, b^2]$ for all $f \in \mathcal{G}$. Then,*

1. $\liminf_i$ *$Var(\Lambda_i) \geq \gamma_1 \nu^2 > 0$.*
2. *$Var(\Lambda_i)/Var(\Lambda_{i-1}) \to 1$.*

Lemma 12. *Let $\{A_i\}_{i\in\mathbb{N}^+}$ be a sequence of positive real numbers such that 1) $A_i/A_{i-1} \to 1$, and 2) there exists some $m > 0$ such that $\liminf_i A_i \geq m$. Then,*

$$R_n = \frac{\sum_{i=1}^{n} A_i^2}{\left(\sum_{i=1}^{n} A_i\right)^2} \to 0$$

Lemma 13. *Let $Var(\Delta_i(x_i,y_i) \mid \mathcal{F}_{i-1}) \xrightarrow{p} v^*$, where v^* is a random variable with domain $[\nu^2, b^2]$. Then,*

$$\frac{V_n}{\overline{V}_n} = \frac{\sum_{i=1}^{n} v_i}{\sum_{i=1}^{n} \overline{v}_i} \xrightarrow{p} \frac{v^*}{\overline{v}^*},$$

where $V_n = \sum_{i=1}^{n} Var(M_i/\Gamma_i \mid \mathcal{F}_{i-1}) = \sum_{i=1}^{n} Var(\Lambda_i/\Gamma_i \mid \mathcal{F}_{i-1})$, and $V_n = \sum_{i=1}^{n} Var(U_{n,i} \mid \mathcal{F}_{i-1}) = \sum_{i=1}^{n} Var(\Lambda_i \mid \mathcal{F}_{i-1})$.

We can now prove the original theorem.

Proof of Theorem 7. We prove the following two conditions which are sufficient for the martingale CLT to hold (Hall and Heyde, 2014):

1. *Lindeberg condition*: For all $\epsilon > 0$, $\sum_{i=1}^{n} \mathbb{E}[U_{n,i}^2 \mathbb{I}(|U_{n,i}| > \epsilon)] \to 0$.

   For each term in the Lindeberg condition, we have the following inequality:

$$\begin{aligned}
\mathbb{E}[U_{n,i}^2 \mathbb{I}(|U_{n,i}| > \epsilon)] &\overset{(i)}{\leq} \sqrt{\mathbb{E}[U_{n,i}^4]} \cdot \sqrt{\mathbb{E}[\mathbb{I}(|U_{n,i}| > \epsilon)^2]} \\
&\leq \sqrt{\mathbb{E}[U_{n,i}^4]} \cdot \sqrt{\mathbb{E}[\mathbb{I}(U_{n,i}^4 > \epsilon^4)]} \\
&\leq \sqrt{\mathbb{E}[U_{n,i}^4]} \cdot \sqrt{\mathbb{P}(U_{n,i}^4 > \epsilon^4)} \\
&\overset{(ii)}{\leq} \sqrt{\mathbb{E}[U_{n,i}^4]} \cdot \sqrt{\frac{\mathbb{E}[U_{n,i}^4]}{\epsilon^4}} \\
&= \frac{\mathbb{E}\left[U_{n,i}^4\right]}{\epsilon^2}
\end{aligned}$$

(i) is a consequence of Holder's inequality, and (ii) is obtained through Markov's inequality applied to the final term. Now, if $U_{n,i}$ has kurtosis upper bounded by $\kappa_{\max}$, we would have

$$\begin{aligned}\mathbb{E}[U_{n,i}^2\mathbb{I}(|U_{n,i}|>\epsilon)] &\leq \frac{\kappa_{\max}\left(\mathbb{E}[U_n,i^2]\right)^2}{\epsilon^2} \\ &= \frac{\kappa_{\max}}{\epsilon^2}\cdot \mathrm{Var}(U_{n,i})^2 \\ &= \frac{\kappa_{\max}}{\epsilon^2}\cdot\left(\frac{\mathrm{Var}(\Lambda_i)}{\overline{V}_n}\right)^2 \\ &= \frac{\kappa_{\max}}{\epsilon^2}\cdot\left(\frac{\mathrm{Var}(\Lambda_i)}{\sum_{j=1}^n \mathrm{Var}(\Lambda_j)}\right)^2\end{aligned}$$

The sum of all these expectations is thus upper bounded by

$$\sum_{i=1}^n \mathbb{E}\left[U_{ni}^2\mathbb{I}(|U_{ni}>\epsilon|)\right] \leq \frac{\kappa_{\max}}{\epsilon^2}\frac{\sum_{i=1}^n \mathrm{Var}(\Lambda_i)^2}{\left(\sum_{j=1}^n \mathrm{Var}(\Lambda_j)\right)^2}$$

We now apply Lemma 11 followed by Lemma 12 to get

$$\frac{\sum_{i=1}^n \mathrm{Var}(\Lambda_i)^2}{\left(\sum_{j=1}^n \mathrm{Var}(\Lambda_j)\right)^2}\to 0,$$

which proves the Lindeberg condition

2. *Convergence of quadratic variation*: $\sum_{i=1}^n \mathrm{Var}(U_{n,i}\mid\mathcal{F}_{i-1})\to\eta^2$, where η^2 is a bounded random variable with $\mathbb{P}(\eta^2>0)=1$.

   From Lemma 13, we infer that

$$\sum_{i=1}^n \mathrm{Var}(U_{n,i}\mid\mathcal{F}_{i-1}) = \frac{1}{\overline{V}_n}\sum_{i=1}^n \mathrm{Var}(\Lambda_i\mid\mathcal{F}_{i-1}) = \frac{V_n}{\overline{V}_n}\xrightarrow{\mathrm{p}}\frac{v^*}{\overline{v}^*},$$

   where $v^*/\overline{v}^*$ is a strictly positive bounded random variable from our assumption that $\mathrm{Var}(\ell(z))\in[\nu^2,b^2]$ for all $\ell\in\mathcal{G}$.

Since the two conditions hold, the martingale CLT follows. □

B.5.1 Proofs of supporting lemmas

Proof of Lemma 11. We observe that

$$\begin{aligned}\mathrm{Var}(\Lambda_i) &= \mathrm{Var}\left(\frac{\Delta_i(x_i,y_i)}{\Gamma_i}-\frac{\Delta_{i-1}(x_i,y_i)}{\Gamma_{i-1}}\right) \\ &= \mathrm{Var}\left(\frac{\Delta_i(x_i,y_i)}{\Gamma_i}-\frac{(1-\gamma_i)\Delta_{i-1}(x_i,y_i)}{\Gamma_i}\right) \\ &= \mathrm{Var}\left(\frac{\gamma_i\Delta_{i-1}(x_i,y_i)}{\Gamma_i}+\frac{\Delta_i(x_i,y_i)-\Delta_{i-1}(x_i,y_i)}{\Gamma_i}\right) \\ &= \frac{\gamma_i^2}{\Gamma_i^2}\mathrm{Var}\left(\underbrace{\Delta_{i-1}(x_i,y_i)}_{\text{Term 1}}+\underbrace{\frac{\Delta_i(x_i,y_i)-\Delta_{i-1}(x_i,y_i)}{\Gamma_i}}_{\text{Term 2}}\right)\end{aligned}$$

We calculate the variances of the two terms separately.

1. Term 1: We have

$$\mathrm{Var}(\Delta_{i-1}(x_i, y_i)) = \overline{v}_{i-1}.$$

From our assumption of $\min_{f\in\mathcal{G}} \mathrm{Var}(f(x,y)) \geq \nu^2$, we have $\overline{v}_{i-1} \geq \nu^2$.

2. Term 2: We have

$$\begin{aligned}\mathrm{Var}(\Delta_i(x,y) - \Delta_{i-1}(x,y)) &= \mathrm{Var}(\Delta_i(x,y) - \Delta_{i-1}(x,y)) \\ &= \mathbb{E}\left[((\ell_i(x,y) - \ell_{i-1}(x,y)) - (\mathbb{E}[\ell_i(x,y) \mid \mathcal{F}_{i-1}] - \mathbb{E}[\ell_{i-1}(x,y) \mid \mathcal{F}_{i-1}]))^2\right] \\ &= \mathbb{E}\left[(\ell_i(x,y) - \ell_{i-1}(x,y))^2\right] + \mathbb{E}\left[(\mathbb{E}[\ell_i(x,y) \mid \mathcal{F}_{i-1}] - \mathbb{E}[\ell_{i-1}(x,y) \mid \mathcal{F}_{i-1}])^2\right] \\ &\quad - 2\mathbb{E}\left[(\ell_i(x,y) - \ell_{i-1}(x,y))(\mathbb{E}[\ell_i(x,y) \mid \mathcal{F}_{i-1}] - \mathbb{E}[\ell_{i-1}(x,y) \mid \mathcal{F}_{i-1}])\right]\end{aligned}$$

Simplifying the third term, we get

$$\begin{aligned}&\mathbb{E}\left[(\ell_i(x,y) - \ell_{i-1}(x,y))(\mathbb{E}[\ell_i(x,y) \mid \mathcal{F}_{i-1}] - \mathbb{E}[\ell_{i-1}(x,y) \mid \mathcal{F}_{i-1}])\right] \\ &\quad = \mathbb{E}\left[\mathbb{E}\left[(\ell_i(x,y) - \ell_{i-1}(x,y))(\mathbb{E}[\ell_i(x,y) \mid \mathcal{F}_{i-1}] - \mathbb{E}[\ell_{i-1}(x,y) \mid \mathcal{F}_{i-1}])\right] \mid \mathcal{F}_{i-1}\right] \\ &\quad = \mathbb{E}\left[(\mathbb{E}[\ell_i(x,y) \mid \mathcal{F}_{i-1}] - \mathbb{E}[\ell_{i-1}(x,y) \mid \mathcal{F}_{i-1}]) \cdot \mathbb{E}\left[(\ell_i(x,y) - \ell_{i-1}(x,y))\right] \mid \mathcal{F}_{i-1}\right] \\ &\quad = \mathbb{E}\left[(\mathbb{E}[\ell_i(x,y) \mid \mathcal{F}_{i-1}] - \mathbb{E}[\ell_{i-1}(x,y) \mid \mathcal{F}_{i-1}])^2\right]\end{aligned}$$

Substituting this term back in the variance expression, we get

$$\begin{aligned}\mathrm{Var}(\Delta_i(x,y) - \Delta_{i-1}(x,y)) &= \mathbb{E}\left[(\ell_i(x,y) - \ell_{i-1}(x,y))^2\right] - \mathbb{E}\left[(\mathbb{E}[\ell_i(x,y) \mid \mathcal{F}_{i-1}] - \mathbb{E}[\ell_{i-1}(x,y) \mid \mathcal{F}_{i-1}])^2\right] \\ &\leq \mathbb{E}\left[(\ell_i(x,y) - \ell_{i-1}(x,y))^2\right] \overset{(i)}{\leq} \sigma_i^2,\end{aligned}$$

where (i) comes from our uniform stability assumption on the learning algorithm.

The second term's variance is thus upper bounded as

$$\mathrm{Var}\left(\frac{\Delta_i(x,y) - \Delta_{i-1}(x,y)}{\gamma_i}\right) \leq \frac{\sigma_i^2}{\gamma_i^2}.$$

From our choice of γ_i such that $\gamma_i = \omega(\sigma_i)$, we have that $\mathrm{Var}\left(\frac{\Delta_i(x,y) - \Delta_{i-1}(x,y)}{\gamma_i}\right) \to 0$.

We thus infer that the variance is completely dominated by the first term as i increases. Since we assume that $\overline{v}_i \to \overline{v}^* > 0$, we thus have

$$\mathrm{Var}\left(\Delta_{i-1}(x_i, y_i) + \frac{\Delta_i(x_i, y_i) - \Delta_{i-1}(x_i, y_i)}{\Gamma_i}\right) \to \overline{v}^* > 0. \tag{6}$$

$$\begin{aligned}\frac{1}{\gamma_i} - \frac{1}{\gamma_{i-1}} &\overset{(i)}{\leq} 1 \\ \gamma_{i-1} - \gamma_i &\leq \gamma_{i-1}\gamma_i \\ \gamma_{i-1}(1 - \gamma_i) &\leq \gamma_i \\ 1 &\leq \frac{\gamma_i}{\gamma_{i-1}(1 - \gamma_i)} \\ \gamma_{i-1} &\leq \frac{\gamma_i}{(1 - \gamma_i)} \\ \frac{\gamma_{i-1}}{\Gamma_{i-1}} &\leq \frac{\gamma_i}{\Gamma_i},\end{aligned}$$

where (i) comes from our assumption on $\{\gamma_i\}$. Thus, γ_i/Γ_i is a non-increasing sequence. Coupled with 6, we can conclude that $\liminf_i \text{Var}(\Lambda_i) \geq \gamma_1 \overline{v}^* > 0$.

Now, let us consider $\text{Var}(\Lambda_i)/\text{Var}(\Lambda_{i-1})$. We have

$$\frac{\text{Var}(\Lambda_i)}{\text{Var}(\Lambda_{i-1})} = \frac{\gamma_i^2}{\Gamma_i^2} \cdot \frac{\Gamma_{i-1}^2}{\gamma_{i-1}^2} \cdot \frac{\text{Var}\left(\Delta_{i-1}(x_i, y_i) + \frac{\Delta_i(x_i,y_i) - \Delta_{i-1}(x_i,y_i)}{\Gamma_i}\right)}{\text{Var}\left(\Delta_{i-2}(x_{i-1}, y_{i-1}) + \frac{\Delta_{i-1}(x_{i-1},y_{i-1}) - \Delta_{i-2}(x_{i-1},y_{i-1})}{\Gamma_{i-1}}\right)}$$
$$= \left(\frac{\gamma_i}{\gamma_{i-1}(1-\gamma_i)}\right)^2 \cdot \frac{\text{Var}\left(\Delta_{i-1}(x_i, y_i) + \frac{\Delta_i(x_i,y_i) - \Delta_{i-1}(x_i,y_i)}{\Gamma_i}\right)}{\text{Var}\left(\Delta_{i-2}(x_{i-1}, y_{i-1}) + \frac{\Delta_{i-1}(x_{i-1},y_{i-1}) - \Delta_{i-2}(x_{i-1},y_{i-1})}{\Gamma_{i-1}}\right)}$$

From 6, we know that the second term goes to one. For the first term, we know from proposition [PROPOSITION] that $\gamma_i/\gamma_{i-1} \to 1$ and $(1-\gamma_i) \to 1$, and thus, $\gamma_i/(\gamma_{i-1}(1-\gamma_i)) \to 1$. Thus, we conclude that $\text{Var}(\Lambda_i)/\text{Var}(\Lambda_{i-1}) \to 1$. □

Proof of Lemma 12. We have

$$R_n = \frac{\sum_{i=1}^n A_i^2}{\left(\sum_{i=1}^n A_i\right)^2} \leq \frac{\left(\sum_{i=1}^n A_i\right) \cdot \max_i \in [n] A_i}{\left(\sum_{i=1}^n A_i\right)^2}$$
$$= \frac{\max_{i \in [n]} A_i}{\sum_{i=1}^n A_i}$$

We now consider two cases based on how many times $A_{\max,n} = \max_{i \in [n]} A_i$ changes with n.

1. *Case 1: $A_{\max,n}$ changes finitely many times*: In this case, there exists some n_0 such that for all $n \geq n_0$, $A_{\max,n} = A_{\max,n_0}$. Since $\liminf_i A_i \geq m$ for all i, we would have $\sum_{i=1}^n A_i \to \infty$. Thus, $A_{\max,n}/\sum_{i=1}^n A_i \to 0$.

2. *Case 2: $A_{\max,n}$ changes infinitely often*: Let us consider the quantity $A_n/(\sum_{i=1}^n A_i)$, i.e., the ratio of the last term to the sum of all the terms. We have two situations depending on whether A_n is the maximum term:

   (i) If A_n is the maximum term, then $A_{\max,n}/(\sum_{i=1}^n A_i) = A_n/(\sum_{i=1}^n A_i)$.

   (ii) If A_n is not the maximum term, then $A_{\max,n} = A_{\max,n-1}$, and $A_{\max,n}/(\sum_{i=1}^n A_i) \leq A_{\max,n}/(\sum_{i=1}^{n-1} A_i)$.

   Thus, the sequence $A_{\max,n}/(\sum_{i=1}^n A_i)$ can increase only when $A_{\max,n} = A_n$, decreasing otherwise. If we show that $A_n/(\sum_{i=1}^n A_i) \to 0$, then we can conclude that $A_{\max,n}/(\sum_{i=1}^n A_i) \to 0$ as well.

   To do this, we define $S_n = (\sum_{i=1}^n A_i)/A_n$, which is the inverse of the ratio defined above. We can express S_n through the following recursion:

$$S_n = \begin{cases} 1, & n = 1 \\ 1 + \frac{A_{n-1}}{A_n} S_{n-1}, & n > 1 \end{cases}$$

   Consider some arbitrary $\epsilon > 0$. We know that $A_{n-1}/A_n \leq 1$, with $A_{n-1}/A_n \to 1$. Thus, there exists some $n_0(\epsilon)$ such that for all $n \geq n_0(\epsilon)$, $A_{n-1}/A_n \geq 1 - \epsilon$. Let S'_n be another sequence defined for $n \geq n_0(\epsilon)$ as

$$S'_n = \begin{cases} 1, & n = n_0(\epsilon) \\ 1 + (1-\epsilon) S'_{n-1}, & n > n_0(\epsilon) \end{cases}$$

   Clearly, $S'_n \leq S_n$ for all $n \geq n_0(\epsilon)$. We also note that $\lim_{n\to\infty} S'_n = \frac{1}{\epsilon}$. Thus, we have

$$\frac{1}{\epsilon} = \lim_{n\to\infty} S'_n \leq \lim_{n\to\infty} S_n.$$

   Since ϵ is arbitrary, we have $\lim_{n\to\infty} S_n = \infty$. Thus, $A_n/\sum_{i=1}^n A_i \to 0$.

Since we have shown that $A_{\max,n}/(\sum_{i=1}^{n} A_i \to 0)$ for both cases, we can conclude that $R_n \to 0$. □

Proof of Lemma 13. Let $v_i = \mathrm{Var}(\Lambda_i \mid \mathcal{F}_{i-1})$ and $\overline{v}_i = \mathrm{Var}(\Lambda_i)$. We then have $V_n = \sum_{i=1}^{n} v_i$ and $\overline{V}_n = \sum_{i=1}^{n} \overline{v}_i$. We also have

$$\begin{aligned} v_i &= \mathrm{Var}(\Lambda_i \mid \mathcal{F}_{i-1}) \\ &= \mathrm{Var}\left(\frac{\Delta_i(x_i, y_i)}{\Gamma_i} - \frac{\Delta_{i-1}(x_i, y_i)}{\Gamma_{i-1}} \mid \mathcal{F}_{i-1}\right) \\ &\overset{(i)}{=} \frac{\gamma_i^2}{\Gamma_i^2}\mathrm{Var}\left(\Delta_i(x_i, y_i) + \frac{(\Delta_i(x_i, y_i) - \Delta_{i-1}(x_i, y_i))}{\gamma_i} \mid \mathcal{F}_{i-1}\right), \end{aligned}$$

where (i) comes from a decomposition similar to that in Lemma 11. For notational convenience we use

$$u_i = \mathrm{Var}\left(\Delta_i(x_i, y_i) + \frac{(\Delta_i(x_i, y_i) - \Delta_{i-1}(x_i, y_i))}{\gamma_i} \mid \mathcal{F}_{i-1}\right)$$

From our assumption that $\gamma_i = \omega(\sigma_i)$, we know that $\sigma_i/\gamma_i \to 0$. Moreover, as $\mathrm{Var}(\ell(x, y) \in [\nu^2, b^2]$ for all $\ell \in \mathcal{G}$, we know that $\mathrm{Var}(\Delta_i(x_i, y_i)) \geq \nu^2$ for all i. Thus, there should be some i_0 such that for all $i \geq i_0$, $\sigma_i/\gamma_i < \nu^2$, where we would have We then have

$$\left(\sqrt{\mathrm{Var}(\Delta_i(x_i, y_i) \mid \mathcal{F}_{t-1})} - \frac{\sigma_i}{\gamma_i}\right)^2 \leq u_i \leq \left(\sqrt{\mathrm{Var}(\Delta_i(x_i, y_i) \mid \mathcal{F}_{t-1})} + \frac{\sigma_i}{\gamma_i}\right)^2$$

As $\sigma_i/\gamma_i \to 0$, we would thus have $u_i \xrightarrow{\mathrm{p}} v^*$.

We also note that u_i is bounded from above by $(b + \sup_i \sigma_i/\gamma_i)$, where the supremum is finite since $\sigma_i, \gamma_i > 0$ and $\sigma_i/\gamma_i \to 0$. Thus, we would have $u_i \xrightarrow{L_1} v^*$, which implies that $\mathbb{E}[|u_i - v^*|] \to 0$.

Let us define two sequences:

$$U_n = \sum_{i=1}^{n} \frac{\gamma_i^2}{\Gamma_i^2}\mathbb{E}[|u_i - v^*|], \quad \text{and} \quad W_n = \sum_{i=1}^{n} \frac{\gamma_i^2}{\Gamma_i^2}$$

We note that W_n is a monotonically increasing sequence, and from our observation that γ_i^2/Γ_i^2 is an increasing sequence in Lemma 11, we have $W_n \to \infty$. We then have, for $n > 1$,

$$\frac{U_n - U_{n-1}}{W_n - W_{n-1}} = \mathbb{E}[|u_i - v^*|] \to 0$$

Thus, by the Stolz-Cesaro theorem, we have $U_n/W_n \to 0$. By repeated application of the triangle inequality, we have

$$\begin{aligned} \mathbb{E}\left[\left|\frac{V_n}{W_n} - v^*\right|\right] &= \mathbb{E}\left[\left|\frac{\sum_{i=1}^{n} \gamma_i^2 u_i/\Gamma_i^2}{W_n} - v^*\right|\right] \\ &\leq \frac{1}{W_n} \cdot \sum_{i=1}^{n} \frac{\gamma_i^2}{\Gamma_i^2}\mathbb{E}[|u_i - v^*|] = \frac{U_n}{W_n} \to 0 \end{aligned}$$

Thus, we have $V_n/W_n \xrightarrow{L_1} v^*$, which in turn implies convergence in probability.

By a similar argument but for total variance, we obtain $\overline{V}_n/W_n \to \overline{v}^* = \mathbb{E}[v^*]$ (since $\overline{v}_n$ is a deterministic quantity, $\overline{V}_n/W_n$ is also deterministic). By applying Slutsky's theorem, we thus get

$$\frac{V_n}{\overline{V}_n} = \frac{V_n/W_n}{\overline{V}_n/W_n} \to \frac{v^*}{\overline{v}^*}$$

□

B.6 Proof of Theorem 8

Fixed-time concentration bound

Proof. We consider the martingale M_t/Γ_t defined on the filtration $\{\mathcal{F}_t\}$. The corresponding martingale difference sequence is given by

$$\begin{aligned} U_t &= \frac{M_t}{\Gamma_t} - \frac{M_{t-1}}{\Gamma_{t-1}} \\ &= \frac{\Delta_t(z_t) - (1-\gamma_t)\Delta_{t-1}(z_t)}{\Gamma_t} \\ &= \frac{\gamma_t \Delta_t(z_t)}{\Gamma_t} + \frac{(\Delta_t(z_t) - \Delta_{t-1}(z_t))}{\Gamma_{t-1}} \end{aligned}$$

We note that the first term is bounded between an interval of size $[-b\gamma_t/\Gamma_t, b\gamma_t/\Gamma_t]$. From our assumption of uniform stability, the second term is bounded within an interval of size $[-\sigma_t/\Gamma_{t-1}, \sigma_t/\Gamma_{t-1}]$. Thus, the sum of the two terms is bounded between $[-(b\gamma_t + (1-\gamma_t)\sigma_t)/\Gamma_t, (b\gamma_t + (1-\gamma_t)\sigma_t)/\Gamma_t]$. From the Azuma-Hoeffding inequality (Wainwright, 2019a), we thus have

$$\begin{aligned} \mathbb{P}\left(\left|\frac{M_T}{\Gamma_T}\right| \geq \frac{\epsilon}{\Gamma_T}\right) &\leq 2\exp\left(-\frac{2\epsilon^2/\Gamma_T^2}{4\sum_{t=1}^T (b\gamma_t + (1-\gamma_t)\sigma_t)^2/\Gamma_t^2}\right) \\ &= 2\exp\left(-\frac{\epsilon^2/\Gamma_T^2}{2V_T^{\uparrow}/\Gamma_T^2}\right) \\ \implies \mathbb{P}\left(|M_T| \geq \epsilon\right) &\leq 2\exp\left(-\frac{\epsilon^2}{2V_T^{\uparrow}}\right) \end{aligned}$$

□

Time-uniform concentration bound

Proof. From the master theorem from (Howard et al., 2020), we know that for a martingale with sub-Gaussian increments U_t, we have

$$\mathbb{P}\left(\exists t \in [T] : \left|\sum_{i=1}^T U_T\right| \geq \frac{x}{2} + \frac{x}{m}V_t\right) \leq 2\exp\left(-\frac{x^2}{2m}\right),$$

where $U_t = M_t/\Gamma_t - M_{t-1}/\Gamma_{t-1}$, and $V_T = \sum_{t=1}^T \mathrm{Var}(U_t \mid \mathcal{F}_{t-1})$. For our martingale M_t/Γ_t, we know that $V_t^{\uparrow}\Gamma_t^2 \geq V_t$ for all t, which implies that

$$\begin{aligned} \mathbb{P}\left(\exists t \in [T] : \left|\sum_{i=1}^T U_T\right| \geq \frac{x}{2} + \frac{x}{m}\frac{V_t^{\uparrow}}{\Gamma_t^2}\right) &\leq \mathbb{P}\left(\exists t \in [T] : \left|\sum_{i=1}^T U_T\right| \geq \frac{x}{2} + \frac{x}{m}V_t\right) \\ &\leq 2\exp\left(-\frac{x^2}{2m}\right) \\ \implies \mathbb{P}\left(\exists t \in [T] : \left|\frac{M_t}{\Gamma_t}\right| \geq \frac{x}{2} + \frac{x}{m}\frac{V_t^{\uparrow}}{\Gamma_t^2}\right) &\leq 2\exp\left(-\frac{x^2}{2m}\right) \\ \mathbb{P}\left(\exists t \in [T] : |M_t| \geq \frac{x}{2}\Gamma_t + \frac{x}{m}V_t^{\uparrow}\Gamma_t\right) &\leq 2\exp\left(-\frac{x^2}{2m}\right) \end{aligned}$$

Let $x^2/2m = c$, which means that $m = x^2/2c$. The bound inside the probability at time T is

$$\frac{x}{2}\Gamma_T + \frac{x}{m}V_T^{\uparrow}\Gamma_T = \frac{x}{2}\Gamma_T + \frac{2c}{x}\cdot\frac{V_T^{\uparrow}}{\Gamma_T}.$$

Minimizing this quantity w.r.t. x, we get

$$x = \frac{\sqrt{2cV_T^{\uparrow}}}{\Gamma_T}, \quad \text{and} \quad m = \frac{cV_T^{\uparrow}}{\Gamma_T^2}$$

Thus, our bound becomes

$$\mathbb{P}\left(\exists t \in [T] : |M_t| \geq \sqrt{\frac{c}{2V_T}} \cdot \frac{(V_T\Gamma_t^2 + V_t\Gamma_t^2)}{\Gamma_T\Gamma_t}\right) \leq 2\exp\left(-c\right)$$

□

B.7 Proof of Lemma 9

Proof. Let

$$\begin{aligned} V_1 &= V_1' = b^2 \\ V_t &= (\gamma_t b + (1-\gamma_t)\gamma_t)^2 + V_{t-1} \\ V_t' &= (\gamma_t b + (1-\gamma_t)c\gamma_t)^2 + V_{t-1}' \end{aligned}$$

We consider two cases:

1. $c \geq 1$: Here, we have

   $$V_1 = b^2 = V_1' V_t - V_{t-1} \qquad = (\gamma_t b + (1-\gamma_t)\sigma_t)^2 \leq (\gamma_t b + (1-\gamma_t)c\sigma_t)^2 = V_t' - V_{t-1}'$$

   Adding the inequalities up for $t \in \{1, \ldots, T\}$, we get $V_T \leq V_T'$.

2. $c < 1$: Here, we have

   $$\begin{aligned} V_1 = b^2 < b^2/c^2 &= V_1'/c^2 \\ V_t - V_{t-1} = (\gamma_t b + (1-\gamma_t)\sigma_t)^2 &= \frac{(\gamma_t bc + (1-\gamma_t)c\sigma_t)^2}{c^2} \\ \leq \frac{(\gamma_t b + (1-\gamma_t)c\sigma_t)^2}{c^2} &= V_t' - V_{t-1}' \end{aligned}$$

   Adding the inequalities up for $t \in \{1, \ldots, T\}$, we get $V_T \leq V_T'/c^2$.

□

C Implementation Details

Metrics. Let $\overline{\ell}_t = \mathbb{E}[\ell_t(Z) \mid \mathcal{F}_{t-1}]$ be the true expected loss of the model at time t, and $\hat{\ell}_t$ be the estiamted loss. We consider the following metrics to compare the performance of different estimators:

- *Root mean squared error (MSE)*: $\sqrt{\frac{1}{T}\sum_{t=1}^{T}(\hat{\ell}_t - \overline{\ell_t})^2}$.
- *Mean absolute error (MAE)*: $\frac{1}{T}\sum_{t=1}^{T}\left|\hat{\ell}_t - \overline{\ell_t}\right|$.
- *Bias*: $\frac{1}{T}\sum_{t=1}^{T}(\hat{\ell}_t - \overline{\ell_t})$.

Baseline estimators. We consider the following hyperparameter sweeps for the baseline estimators:

- *Sliding window (SW)*: Window size $\in \{10, 50, 50, 100, 200, 400, 600, 800, 1000\}$.
- *Exponential moving average (EMA)*: Decay factor $\in \{0.1, 0.05, 0.01, 0.005, 0.001\}$.
- *Fading factor prequential (FFPreq)*: Decay factor $\in \{0.8, 0.9, 0.95, 0.99, 0.999, 0.9999, 0.99999\}$.
- *ADWIN*: Sensitivity $\delta \in \left\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\right\} \cup \left\{5 \cdot 10^{-2}, 5 \cdot 10^{-3}, 5 \cdot 10^{-4}, 5 \cdot 10^{-5}\right\} \cup \{0.1, 0.2, \ldots, 0.9\}$.

At time step t, these estimators are given the prequential evaluation $\ell_t(z_t)$ for update.

C.1 Linear Regression

Data. We generate synthetic data for d-dimensional linear regression by first sampling the true weight $\mathbf{w}^*$ uniformly from the d-dimensional unit hypersphere. We then sample $\mathbf{x}_t \sim \mathcal{N}(0, \Sigma)$, where $\Sigma \in \mathbb{R}^{d \times d}$ is a covariance matrix such that $\Sigma_{ii} = 1$ for all $i \in [d]$. We then set $y_t = (\mathbf{w}^*)^T \mathbf{x}_t + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ We vary the standard deviation of the noise, $\sigma \in \{0.005, 0.05, 0.5\}$.

Loss. We consider the scaled mean squared error $\ell_t(\hat{y}, y) = (y - \hat{y})^2/d$ as the loss function. The division by d ensures that we can use the same learning rate across all dimensions and noise levels. We set the true loss at time t as $\bar{\ell}_t = (\mathbf{w}_t - \mathbf{w}^*)^T \Sigma (\mathbf{w}_t - \mathbf{w}^*) + \sigma^2$, where $\mathbf{w}_t$ is the weight vector of the model at time t. All estimators try to estimate this true loss with access to $\ell_t(z_t) = \ell(\mathbf{w}_t^T \mathbf{x}_t, y_t)$. OEUVRE additionally has access to $\ell_{t-1}(z_t) = \ell(\mathbf{w}_{t-1}^T \mathbf{x}_t, y_t)$.

Learning algorithm. We use online gradient descent with a learning rate of $\eta_t = \eta_0/\sqrt{t}$, with $\eta_0 = 0.01$. We initialize the weight vector $\mathbf{w}_0$ to be the zero vector.

C.2 Prediction with Expert Advice

Data. We generate synthetic data for prediction with expert advice with d experts by sampling each expert i's loss at time t from distribution $\mathcal{D}_i$ in an iid manner. The distributions $\mathcal{D}_i$ are independent of each other. We use the following choices of distributions:

- *Beta distribution*: $\mathcal{D}_i = \text{Beta}(\alpha_i, \beta_i)$, where $\alpha_i, \beta_i \in \{1, \cdots, 9\}$.
- *Bernoulli distribution*: $\mathcal{D}_i = \text{Bernoulli}(p_i)$, where $p_i \sim \mathcal{U}(0.01, 0.99)$ are sampled uniformly.

Loss. Let $\bar{\boldsymbol{\ell}}$ be the vector of expected losses of the experts, and $\mathbf{p}_t$ be the distribution over the experts learned by the algorithm at time t. The expected loss is given by $\bar{\boldsymbol{\ell}}^T \mathbf{p}_t$. The loss observed at time t is given by $\ell_t(z_t) = \boldsymbol{\ell}_t^T \mathbf{p}_t$, where $\boldsymbol{\ell}_t$ is the vector of losses of the experts at time t. OEUVRE additionally has access to $\ell_{t-1}(z_t) = \boldsymbol{\ell}_t^T \mathbf{p}_{t-1}$.

Learning algorithm. We use the Hedge algorithm (Mourtada and Gaiffas, 2019) with a learning rate of $\eta_t = \sqrt{\log(d)/t}$. We initialize the distribution over experts $\mathbf{p}_0$ to be the uniform distribution.

C.3 Neural Networks

Data. We use the MNIST (Deng, 2012) and EMNIST-digits (Cohen et al., 2017) datasets for our experiments.

Loss. We use the cross-entropy loss for our experiments. The goal is to estimate the expected loss of the currently learned model on the test split of the data. We obtain a proxy for the expected test loss by evaluating on 1024 test samples at each time step. Each estimator is given 128 independently sampled test samples at each time to update its estimate. OEUVRE additionally has access to the loss of the previous model on the same 128 samples.

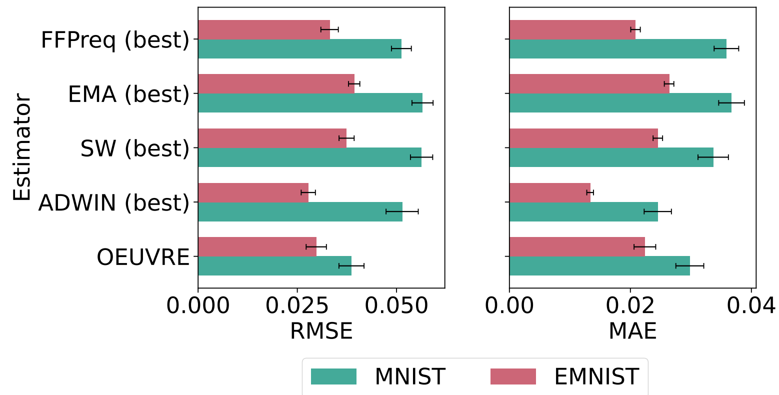


Figure 4: Performance of different loss estimation methods on the MNIST and EMNIST dataset with batch size 8.

Learning algorithm. We use a simple neural network with one hidden layer consisting of 128 ReLU units. We train the model using mini-batch SGD using the AdamW optimizer with a learning rate of 5×10^{-4}. Each experiment is repeated for 10 epochs.

C.4 Logistic Regression

Data. We use the Diabetes Health Indicators dataset to perform binary classification. We shuffle the data randomly and use the first 10,000 samples to train our model.

Loss. We use the cross-entropy loss for our experiments. The goal is to estimate the expected loss of the currently learned model on the entire training set.

Learning algorithm. We use a logistic regression model trained using online gradient with a learning rate of $\eta_t = \eta_0/\sqrt{t}$, with $\eta_0 = 0.05$ and use Polyak-Ruppert (PR) averaging. PR-averaging results in an algorithm which is $1/t$-uniformly stable (Table 1). We initialize the weight vector $\mathbf{w}_0$ to be the zero vector.

D Additional experimental results

In this section, we provide two additional sets of experimental results. We first show the comparative performance of OEUVRE against the best settings of all baselines for the linear regression and prediction with expert advice tasks in Figures 5 and 6, respectively. We also compare the performance of these estimators on another metric: the time-averaged bias against the ground truth. We present these results for all three tasks. For the linear regression and prediction with expert advice tasks, we also add the simple Prequential estimator, which computes the time-averaged prequential loss at each time step, as a baseline.

In general, we observe that the best-performing abaseline varies across tasks and metrics. OEUVRE, however, consistently matches or outperforms the best-performing baseline, proving to be a good general performer across tasks and metrics.

Linear regression. We present the results for the linear regression task in Figure 5. We observe that OEUVRE's performance is comparable to the best settings of SW, EMA, and FFPreq, with ADWIN and the Prequential estimator performing much worse. This is true across all three metrics of RMSE, MAE, and bias.

Prediction with expert advice. We present the results for the prediction with expert advice task in Figure 6. EMA performs the worst across both distributions and all metrics. We observe that for the Beta distribution, OEUVRE clearly gives the lowest RMSE and bias across dimensions. For the Bernoulli distribution, OEUVRE's RMSE is comparable to the best, and its bias is the lowest across all dimensions. OEUVRE's MAE is high for lower dimensions ($d = 25, 50$), but it then decreases and becomes comparable to the other estimators for higher dimensions.

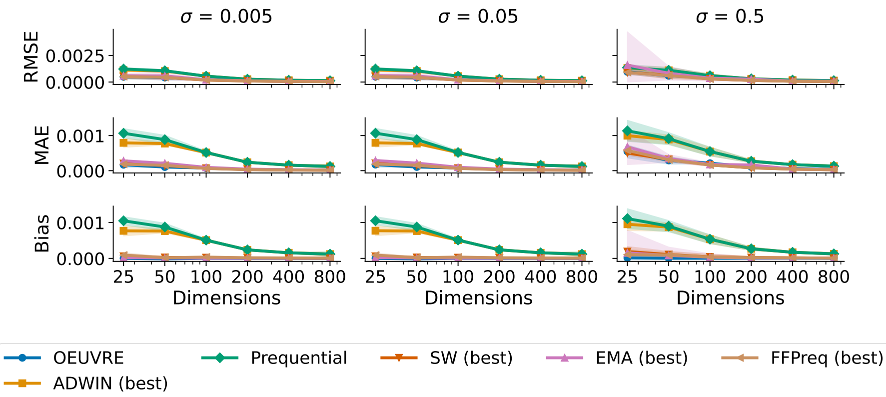


Figure 5: Performance comparison of OEUVRE against the best hyperparameter setting for each baseline for the linear regression task. OEUVRE achieves competitive RMSE, MAE, and bias when compared to the best baseline.

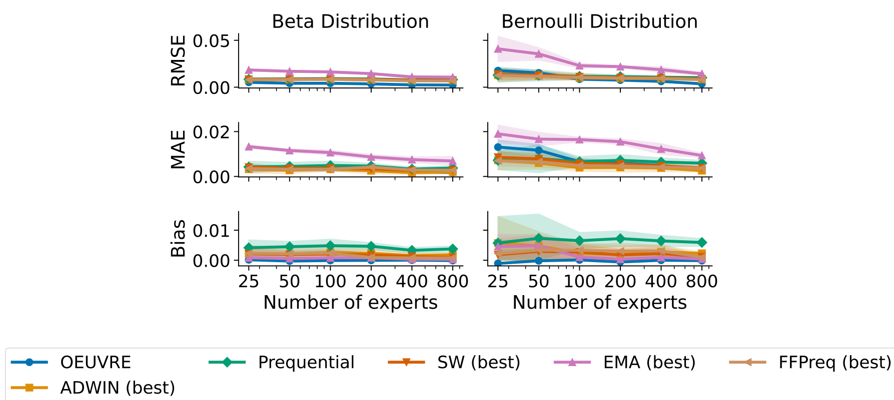


Figure 6: Performance comparison of OEUVRE against the best hyperparameter setting for each baseline for the prediction with expert advice task. OEUVRE achieves competitive RMSE, MAE, and bias when compared to the best baseline.

Neural networks. We present the results for the neural network task in Figures 7 and 8 for batch sizes of 128 and 8, respectively. We observe that OEUVRE has the lowest RMSE, MAE and bias for both MNIST and EMNIST datasets for batch size 128. However, the variance of the bias across runs is high. For batch size 8, the performance is more mixed. For the MNIST dataset, ADWIN has lower RMSE and MAE than OEUVRE. For EMNIST, OEUVRE has the lowest RMSE and the second-lowest MAE. OEUVRE's bias is the lowest acros bothd datasets; however, we again observe high variance across runs.

E Experiments on non-stationary datasets

We conduct further experiments comparing OEUVRE to the baselines on real-world non-stationary datasets. We consider the following datasets: 1) *UCI Bike sharing* (Fanaee-T, 2013), 2) *Bikes* from the River package (Montiel et al., 2021), 3) *Chick Weights* from the River package (Montiel et al., 2021), and 4) *Wine Quality* from the River package (Montiel et al., 2021). We use a linear regression model trained using online gradient descent to perform regression on these datasets with learning rate $\eta_t = \eta_0/\sqrt{t}$, with $\eta_0 = 0.01$. The goal is to estimate the expected MSE of the currently learned model at the current time step. The ground truth is estimated by calculating the loss of the model on a lookahead of 50 samples.

Bike Sharing dataset. We present our results for this dataset in Figure 9 and Table 2b. We observe that OEUVRE follows the true loss fairly well, achieving performance comparable to the best EMA and SW baselines.

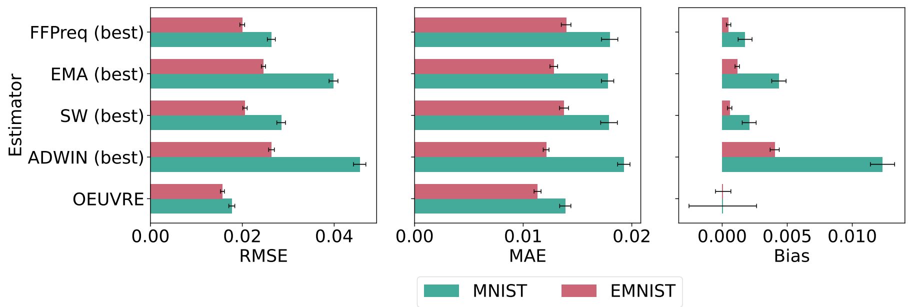


Figure 7: Performance comparison of OEUVRE against the best hyperparameter setting for each baseline for the neural network task with 128 samples for estimation at each time step. OEUVRE achieves competitive RMSE, MAE, and bias when compared to the best baseline.

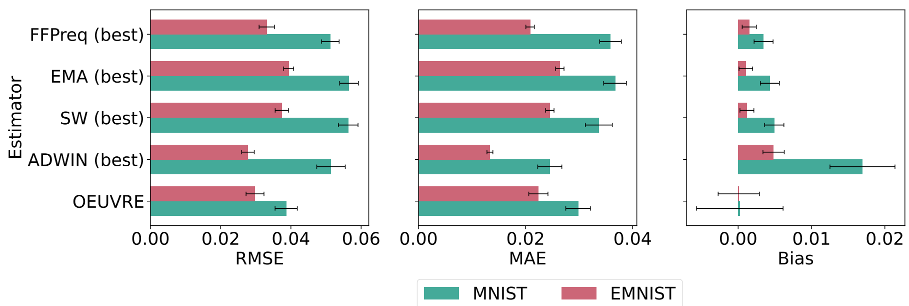


Figure 8: Performance comparison of OEUVRE against the best baseline and the median baseline for the neural network task with 8 samples for estimation at each time step. OEUVRE achieves competitive RMSE and bias when compared to the best baseline.

This dataset has gradual and seasonal concept drift, which also allows OEUVRE's estimates to change gradually.

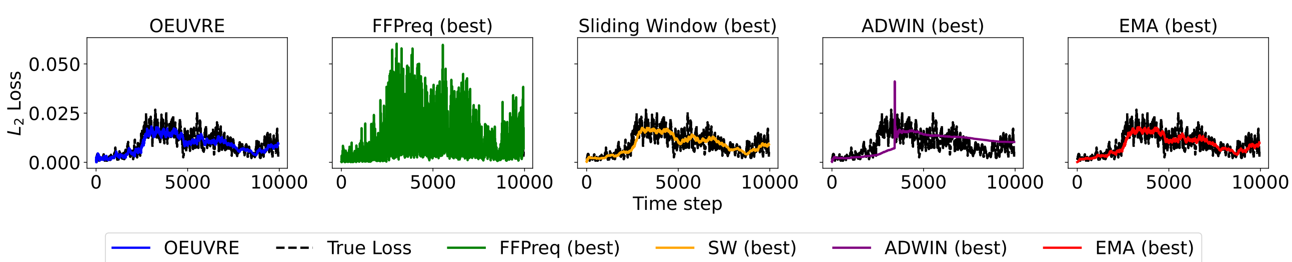


Figure 9: Performance of different algorithms on the Bike Sharing dataset.

Bikes dataset. We present our results for this dataset in Figure 10 and Table 2a. We observe that EMA and SW with optimal hyperparameters perform the best, followed by OEUVRE and FFPreq. The Bikes dataset has more extreme changes in the observed loss as compared to the Bike Sharing dataset, which makes it slightly more difficult for OEUVRE to adapt quickly enough.

Chick Weights dataset. We present our results for this dataset in Figure 11 and Table 2c. We observe that the true loss increases rapidly with time, which makes it difficult for all estimators to track the true loss. Although the best settings of EMA, SW, and FFPreq perform better than OEUVRE, OEUVRE still achieves a competitive result when compared with the simple prequential and ADWIN estimators.

Wine Quality dataset. We present our results for this dataset in Figure 12 and Table 2d. The ground truth has a lot of variability with time, which makes id difficult for OEUVRE to track it accurately – we observe in

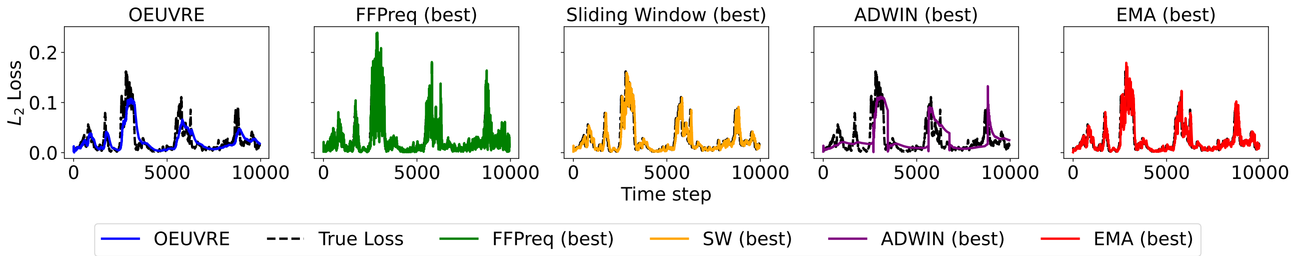


Figure 10: Performance of different algorithms on the Bikes dataset.

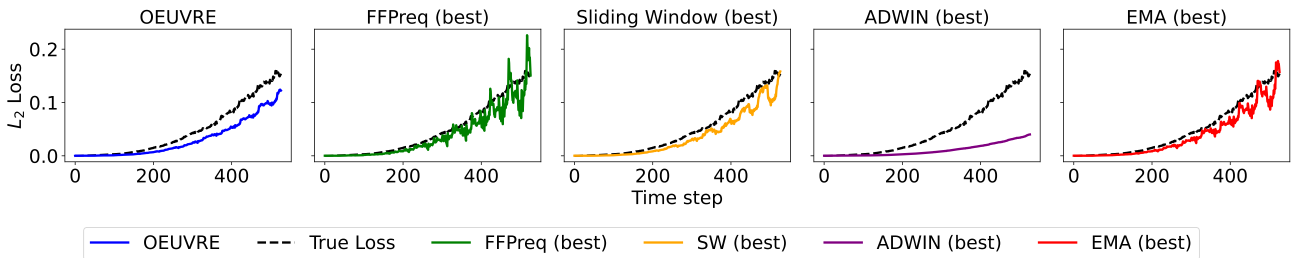


Figure 11: Performance of different algorithms on the Chick Weights dataset.

Figure 12 that OEUVRE's estimate does not change as rapidly. SW and EMA with optimal hyperparameters perform the best, and OEUVRE's performance is third-best overall.

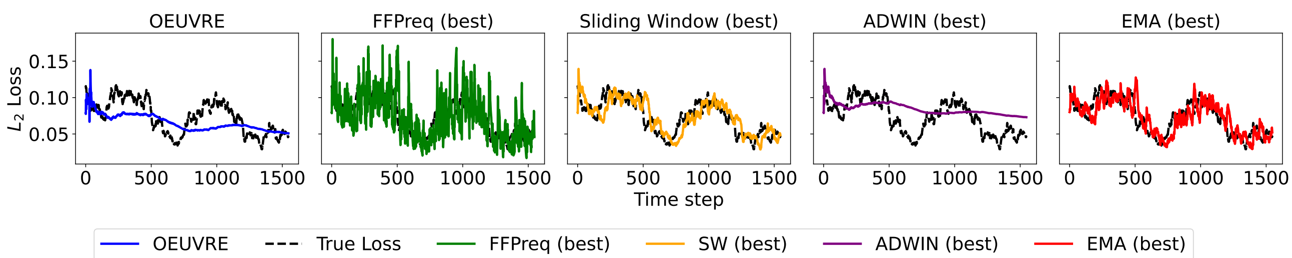


Figure 12: Performance of different algorithms on the Wine Quality dataset.

Estimator	RMSE	MAE	Bias
Prequential	0.0293	0.0209	-0.0012
ADWIN	0.0245	0.0161	0.0012
SW (best)	0.0142	0.0084	-0.0003
EMA (best)	0.0138	0.0080	-0.0004
OEUVRE	0.0158	0.0096	-0.0010
FFPreq (best)	0.0176	0.0098	-0.0003

(a) Bikes dataset

Estimator	RMSE	MAE	Bias
Prequential	0.0054	0.0040	-0.0019
ADWIN	0.0053	0.0040	0.0004
SW (best)	0.0033	0.0026	-0.0002
EMA (best)	0.0033	0.0026	-0.0002
OEUVRE	0.0035	0.0026	-0.0004
FFPreq (best)	0.0080	0.0055	-0.0000

(b) Bike Sharing dataset

Estimator	RMSE	MAE	Bias
Prequential	0.0520	0.0369	-0.0369
ADWIN	0.0520	0.0369	-0.0369
SW (best)	0.0155	0.0112	-0.0111
EMA (best)	0.0151	0.0108	-0.0100
OEUVRE	0.0171	0.0127	-0.0127
FFPreq (best)	0.0169	0.0110	-0.0084

(c) Chick Weights dataset

Estimator	RMSE	MAE	Bias
Prequential	0.0230	0.0192	0.0110
ADWIN	0.0230	0.0192	0.0110
SW (best)	0.0151	0.0119	0.0004
EMA (best)	0.0163	0.0126	-0.0001
OEUVRE	0.0193	0.0164	-0.0044
FFPreq (best)	0.0259	0.0200	-0.0003

(d) Wine Quality dataset

Table 2: Performance comparison of different estimators on non-stationary datasets. OEUVRE achieves competitive RMSE, MAE, and bias when compared to the best baseline.