Global and Stochastic Analysis
Vol. 12 No. 5 (September, 2025)

Received: 30th May 2025 *Revised: 20th June 2025* *Accepted: 30th June 2025*

ENSEMBLING DISCOUNTED VAW EXPERTS WITH THE VAW META-LEARNER

DMITRY B. ROKHLIN AND GEORGIY A. KARAPETYANTS

ABSTRACT. The Vovk-Azoury-Warmuth (VAW) forecaster is a powerful algorithm for online regression, but its standard form is designed for stationary environments. Recently Jacobsen and Cutkosky (2024) introduced a discounting factor, γ, to the VAW algorithm (DVAW), enabling it to track changing concepts by down-weighting old data. They also proposed an ensemble method for learning γ on-the-fly. In this paper we use a simplified dynamic regret bound and employ the standard VAW forecaster as a meta-learner to dynamically aggregate the predictions of DVAW experts. The main result contains a bound for the dynamic regret of the proposed ensemble. Computer experiments on synthetic data show that our ensembling approach significantly outperforms both the standard VAW and individual DVAW experts in non-stationary settings, while remaining robust and competitive in stationary ones.

1. Introduction

Online learning provides a powerful framework for sequential decision-making in environments where data arrives as a stream. A central paradigm in this field is Online Convex Optimization (OCO), where a learner makes a sequence of predictions to minimize a cumulative loss [4, 7]. The classic performance measure is the static regret, which benchmarks the learner against the best single fixed decision chosen in hindsight. While fundamental, the assumption of a single best decision is often not suitable for real-world applications where the underlying data-generating process may be non-stationary, causing the optimal decision to drift over time.

To address this limitation, the more challenging measure of dynamic regret has been proposed. It evaluates the learner against an arbitrary sequence of comparator decisions, $\mathbf{u} = (u_1, \ldots, u_T)$, making it a suitable metric for non-stationary environments [13]. The difficulty of a dynamic regret problem is typically quantified by the variability of the comparator sequence, most commonly its path-length, $P_T(\mathbf{u}) = \sum_{t=1}^{T-1} \|u_{t+1} - u_t\|_2$. The state-of-the-art for general OCO problems has established minimax optimal regret bounds of $O(\sqrt{T(1+P_T(\mathbf{u}))})$ [11], with recent work focusing on achieving tighter, problem-dependent bounds that replace the dependence on the time horizon T with instance-specific quantities like gradient variation [12].

Date: Date of Submission May 30, 2025 ; Date of Acceptance June 30, 2025 ; Communicated by Yuri E. Gliklikh.

2010 *Mathematics Subject Classification.* Primary 68W27; Secondary 68Q32.

Key words and phrases. Vovk-Azoury-Warmuth algorithm; discounting; dynamic regret.

The research of D.B. Rokhlin was supported the Russian Science Foundation (RSF), project number 25-11-00094.

A canonical and important problem within this domain is online linear regression with quadratic loss. For this specific setting, the Vovk-Azoury-Warmuth (VAW) forecaster, discovered independently by [10] and [1], is known to be a powerful tool. The VAW algorithm achieves optimal static regret bound, which holds for any comparator without prior knowledge of its norm, a property explored extensively in works like [6]. However, the standard VAW forecaster is designed for stationary environments; it weights all past data equally, which hinders its ability to adapt to changes and can lead to poor performance in dynamic settings.

A recent breakthrough by Jacobsen and Cutkosky [5] directly addressed this limitation by introducing the Discounted VAW (DVAW) algorithm. By incorporating a discount factor γ, DVAW exponentially down-weights older observations, allowing it to "forget" old information and track a changing target. They further showed that the optimal discount factor γ could be learned on-the-fly by running an ensemble of DVAW experts and combining their predictions. Their approach achieves optimal dynamic regret bounds in a fully unconstrained setting.

A key challenge is that the ideal discount factor γ^* is defined implicitly by a nonlinear equation depending on the entire data stream and comparator sequence, making it impossible to know in advance. To overcome this, Jacobsen and Cutkosky proposed to learn γ by running an ensemble of DVAW experts over a carefully constructed grid of discount factors. This, in turn, requires a sophisticated solution for aggregating the experts' predictions in a fully unconstrained setting. Their approach is a custom-designed "range-clipped" meta-algorithm, which operates by first defining a dynamic "trust region" based on the history of observations. It then clips each expert's raw prediction to lie within this safe range before combining them. This mechanism, while crucial for proving theoretical guarantees by controlling for potentially unbounded losses, makes the overall algorithm and its analysis quite intricate.

In this paper, we build upon the DVAW framework of [5] but propose a conceptually simpler and more direct ensembling strategy. Our key insight is to employ the standard VAW forecaster itself as the meta-learner to dynamically aggregate predictions from a pool of DVAW experts, creating what we term the VE-DVAW (VAW-Ensembled DVAW) architecture. First we derive a simplified dynamic regret bound for a single DVAW expert, expressing it in terms of the comparator's path-length (Lemma 3.1). The theoretical result of this paper (Theorem 3.2) is a dynamic regret bound for the VE-DVAW algorithm, demonstrating that our simpler method retains strong theoretical guarantees without the need for custom clipping mechanisms. Also, through computer experiments on synthetic datasets, we show that our VE-DVAW ensemble significantly outperforms the standard VAW algorithm in non-stationary scenarios while remaining robust in stationary ones, confirming its practical utility.

2. Preliminaries

We consider the standard online linear regression setting. In each round $t = 1, \dots, T$, the learner receives a feature vector $x_t \in \mathbb{R}^d$, makes a prediction $\hat{y}_t = \langle w_t, x_t \rangle$ using a weight vector $w_t \in \mathbb{R}^d$, and then observes the true label $y_t \in \mathbb{R}$. The learner incurs the squared loss $\ell_t(w) = \frac{1}{2}(\langle w, x_t \rangle - y_t)^2$, $t \geq 1$.

Assume that $\|x_t\|_2 \leq a$, $|y_t| \leq Y$, and put $\ell_0(w) = \lambda \|w\|_2^2/2$. In the VAW algorithm [2, Section 11.8], [7, Section 7.11] the weight w_t is allowed to depend on the feature mapping

x_t, indicating that features x_t are available at time t before predicting the label y_t:

$$w_t = \operatorname*{argmin}_{w\in\mathbb{R}^d} \left\{ \frac{1}{2}\langle x_t, w\rangle^2 + \sum_{s=0}^{t-1} \ell_s(w) \right\}.$$

Explicitly,

$$w_t = S_t^{-1}\sum_{i=1}^{t-1} y_i x_i, \quad S_t = \lambda I_d + \sum_{i=1}^{t} x_i x_i^\top.$$

Moreover, S_t^{-1} can be computed recursively by the Sherman-Morrison formula:

$$S_t^{-1} = S_{t-1}^{-1} - \frac{S_{t-1}^{-1}x_t(S_{t-1}^{-1}x_t)^T}{1 + x_t^T S_{t-1}^{-1}x_t}, \quad S_0^{-1} = \lambda^{-1} I_d.$$

The static regret

$$R_T(u) = \frac{1}{2}\sum_{t=1}^{T}(\langle x_t, w_t\rangle - y_t)^2 - \frac{1}{2}\sum_{t=1}^{T}(\langle x_t, u\rangle - y_t)^2, \quad u \in \mathbb{R}^d$$

of the VAW algorithm satisfies the bound [2, Theorem 11.8], [5, Theorem 1]:

$$R_T(u) \le \frac{\lambda}{2}\|u\|_2^2 + \frac{dY^2}{2}\ln\left(1 + \frac{\sum_{t=1}^{T}\|x_t\|_2^2}{\lambda d}\right). \tag{2.1}$$

Recently [5] modified the VAW algorithm by introducing a discounting factor, which allows to forget old data. Another modification proposed in [5] concerns the introduction of an additional "hint" $\widetilde{y}_t$ for y_t at each step:‘

$$w_t = \arg\min_{w\in\mathbb{R}^d} \left\{ \frac{1}{2}(\tilde{y}_t - \langle x_t, w\rangle)^2 + \gamma\sum_{s=0}^{t-1}\gamma^{t-1-s}\ell_s(w) \right\},$$

where $\gamma \in (0,1]$, $\tilde{y}_1 = 0$, and $\tilde{y}_t \in [-\widetilde{Y}, \widetilde{Y}]$ for $t > 1$. We will call this algorithm DVAW (discounted VAW) and denote it by $\mathscr{A}_\gamma(\lambda)$. For $\gamma = 1$, $\widetilde{y}_t = 0$ the DVAW algorithm coincides with VAW. In this case we will use the notation $\mathscr{A}(\lambda)$. For $\gamma = 0$ DVAW predicts $\widetilde{y}_t$ by convention.

Recursively define $\Sigma_t = x_t x_t^\top + \gamma\Sigma_{t-1}$, $\Sigma_0 = \lambda I$. Then

$$w_t = \Sigma_t^{-1}\left[\tilde{y}_t x_t + \gamma\sum_{s=1}^{t-1}\gamma^{t-1-s} y_s x_s\right].$$

To derive the recursive formula for Σ_t apply the Woodbury matrix identity [3] for a rank-1 update:

$$(A + uv^\top)^{-1} = A^{-1} - \frac{A^{-1}uv^\top A^{-1}}{1 + v^\top A^{-1}u}$$

to $A = \gamma\Sigma_{t-1}$, $u = v = x_t$:

$$\Sigma_t^{-1} = \left(\gamma\Sigma_{t-1} + x_t x_t^\top\right)^{-1} = \left(\frac{1}{\gamma}\Sigma_{t-1}^{-1}\right) - \frac{\left(\frac{1}{\gamma}\Sigma_{t-1}^{-1}\right)x_t x_t^\top\left(\frac{1}{\gamma}\Sigma_{t-1}^{-1}\right)}{1 + x_t^\top\left(\frac{1}{\gamma}\Sigma_{t-1}^{-1}\right)x_t}$$

$$= \frac{1}{\gamma}\left(\Sigma_{t-1}^{-1} - \frac{\Sigma_{t-1}^{-1}x_t x_t^\top\Sigma_{t-1}^{-1}}{\gamma + x_t^\top\Sigma_{t-1}^{-1}x_t}\right).$$

Discounting appears to be an important modification that allows to bound the dynamic regret

$$R_T(\mathbf{u}) = \sum_{t=1}^{T} \ell_t(w_t) - \sum_{t=1}^{T} \ell_t(u_t)$$

against a comparator sequence $\mathbf{u} = (u_1, \dots, u_T)$, where each $u_t \in \mathbb{R}^d$. Namely, for the dynamic regret $R_T^{\mathscr{A}_\gamma}(\lambda)(\mathbf{u})$ of the algorithm $\mathscr{A}_\gamma(\lambda)$, the following bound holds true [5, Theorem 3.1]:

$$R_T^{\mathscr{A}_\gamma(\lambda)}(\mathbf{u}) \leq \frac{\gamma\lambda}{2}\|u_1\|_2^2 + \frac{d}{2}\max_{1\leq t\leq T}\Delta_t^2 \ln\left(1 + \frac{\sum_{t=1}^{T}\gamma^{T-t}\|x_t\|_2^2}{\lambda d}\right)$$
$$+ \gamma\sum_{t=1}^{T-1}[F_t^\gamma(u_{t+1}) - F_t^\gamma(u_t)] + \frac{d}{2}\ln(1/\gamma)\Delta_{1:T}^2, \tag{2.2}$$

where $F_t^\gamma(w) = \gamma^t \frac{\lambda}{2}\|w\|_2^2 + \sum_{s=1}^{t}\gamma^{t-s}\ell_s(w)$,

$$\Delta_t^2 = (y_t - \tilde{y}_t)^2, \quad \Delta_{1:T}^2 = \sum_{t=1}^{T}(y_t - \tilde{y}_t)^2.$$

3. A simplified regret bound and the VAW-ensembled algorithm

Let us introduce the path length $P_T(\mathbf{u}) = \sum_{t=1}^{T-1}\|u_{t+1} - u_t\|_2$ and rewrite the estimate (2.2) in term of this quantity.

Lemma 3.1. *Let $\|u_t\|_2 \leq R$. Then the dynamic regret of the DVAW algorithm $\mathscr{A}_\gamma(\lambda)$ is bounded as follows:*

$$R_T^{\mathscr{A}_\gamma(\lambda)}(\mathbf{u}) \leq \eta a(aR+Y)P_T(\mathbf{u}) + \frac{d}{2\eta}\Delta_{1:T}^2 + \lambda RP_T(\mathbf{u})$$
$$+ \frac{d}{2}\max_{1\leq t\leq T}\Delta_t^2 \ln\left(1 + \frac{a^2 T}{\lambda d}\right) + \frac{\lambda}{2}R^2, \tag{3.1}$$

where $\eta = \gamma/(1-\gamma)$, $\gamma \in [0,1)$. For a static comparator sequence with $P_t(\mathbf{u}) = 0$ the inequality (3.1) holds true also for $\gamma = 1$ ($\eta = +\infty$).

Proof. Note that

$$\gamma\sum_{t=1}^{T-1}(F_t^\gamma(u_{t+1}) - F_t^\gamma(u_t))$$
$$= \gamma\sum_{t=1}^{T-1}\left(\gamma^t\frac{\lambda}{2}\|u_{t+1}\|^2 + \sum_{s=1}^{t}\gamma^{t-s}\ell_s(u_{t+1}) - \left(\gamma^t\frac{\lambda}{2}\|u_t\|^2 + \sum_{s=1}^{t}\gamma^{t-s}\ell_s(u_t)\right)\right)$$
$$= \gamma\sum_{t=1}^{T-1}\left(\gamma^t\frac{\lambda}{2}(\|u_{t+1}\|^2 - \|u_t\|^2) + \sum_{s=1}^{t}\gamma^{t-s}(\ell_s(u_{t+1}) - \ell_s(u_t))\right)$$

We bound the inner terms:

$$\frac{\lambda}{2}(\|u_{t+1}\|^2 - \|u_t\|^2) = \frac{\lambda}{2}\langle u_{t+1} - u_t, u_{t+1} + u_t\rangle \leq \frac{\lambda}{2}\|u_{t+1} - u_t\|_2\|u_{t+1} + u_t\|_2$$
$$\leq \lambda R\|u_{t+1} - u_t\|_2$$

For the second part of the inner sum we use the difference of squares formula and the assumptions $\|x_s\|_2 \le a$, $\|u_t\|_2 \le R$:

$$\begin{aligned}\ell_s(u_{t+1}) - \ell_s(u_t) &= \frac{1}{2}\langle x_s, u_{t+1} - u_t\rangle(\langle x_s, u_{t+1}\rangle + \langle x_s, u_t\rangle - 2y_s) \\ &\le \frac{1}{2}\|x_s\|_2\|u_{t+1} - u_t\|_2(2R\|x_s\|_2 + 2Y) \le a(aR+Y)\|u_{t+1} - u_t\|_2.\end{aligned}$$

Substitute these bounds back:

$$\begin{aligned}\gamma\sum_{t=1}^{T-1}(F_t^\gamma(u_{t+1}) - F_t^\gamma(u_t)) &\le \gamma\sum_{t=1}^{T-1}\left(\gamma^t\lambda R\|u_{t+1} - u_t\|_2 + \sum_{s=1}^{t}\gamma^{t-s}a(aR+Y)\|u_{t+1} - u_t\|_2\right) \\ &= \gamma\sum_{t=1}^{T-1}\|u_{t+1} - u_t\|_2\left(\gamma^t\lambda R + a(aR+Y)\sum_{s=1}^{t}\gamma^{t-s}\right) \\ &\le \left(\lambda R + \frac{\gamma}{1-\gamma}a(aR+Y)\right)P_T(\mathbf{u})\end{aligned}$$

for $\gamma \in [0,1)$. From (2.2) now it follows that

$$\begin{aligned}R_T(\mathbf{u}) &\le \frac{\gamma\lambda}{2}R^2 + \frac{d}{2}\max_{1\le t\le T}\Delta_t^2\ln\left(1 + \frac{a^2T}{\lambda d}\right) \\ &\quad + \left(\lambda R + \frac{\gamma}{1-\gamma}a(aR+Y)\right)P_T(\mathbf{u}) + \frac{d}{2}\left(\frac{1}{\gamma} - 1\right)\Delta_{1:T}^2 \\ &\le \eta a(aR+Y)P_T(\mathbf{u}) + \frac{d}{2\eta}\Delta_{1:T}^2 + \lambda RP_T(\mathbf{u}) + \frac{d}{2}\max_{1\le t\le T}\Delta_t^2\ln\left(1 + \frac{a^2T}{\lambda d}\right) + \frac{\lambda}{2}R^2,\end{aligned}$$

where we used the inequality

$$\ln\frac{1}{\gamma} \le \frac{1}{\gamma} - 1.$$

For $\gamma = 1$ and $P_T(\mathbf{u}) = 0$ the inequality (3.1) follows from (2.2) trivially. □

For $P_T(\mathbf{u}) > 0$ minimum of the right-hand side (3.1) over $\eta > 0$ is attained at

$$\eta^* = \sqrt{\frac{d\Delta_{1:T}^2}{2a(aR+Y)P_T(u)}}. \tag{3.2}$$

Plugging this value back into (3.1) we get

$$R_T(\mathbf{u}) \le \sqrt{2da(aR+Y)\Delta_{1:T}^2P_T(\mathbf{u})} + \lambda RP_T(\mathbf{u}) + \frac{d}{2}\max_{1\le t\le T}\Delta_t^2\ln\left(1 + \frac{a^2T}{\lambda d}\right) + \frac{\lambda}{2}R^2.$$

Note that that this formula also holds true for $P_t(\mathbf{u}) = 0$ (formally, $\eta^* = 0$ in this case).

However, η^*, given by (3.2), cannot be used in any algorithm, since it depends on the future values of labels via $\Delta_{1:T}^2$ and on the comparator sequence via $P_t(\mathbf{u})$. To overcome this difficulty, following [5], we consider an ensemble of DVAW algorithms $\mathscr{A}_{\gamma_k}(\overline{\lambda})$ with parameters γ_k taken from the set S_γ, defined as follows:

$$b > 1, \quad \eta_{\min} = 2d, \quad \eta_{\max} = dT,$$

$$\mathscr{S}_\eta = \{\eta_i = \eta_{\min}b^i \wedge \eta_{\max} : i \in \mathbb{Z}_+\},$$

$$\mathscr{S}_\gamma = \left\{\gamma_i = \frac{\eta_i}{1+\eta_i} : i \in \mathbb{Z}_+\right\} \cup \{0\}.$$

Note that the set S_γ contains $M = O(\log_b(\eta_{\max}/\eta_{\min})) = O(\log_b T)$ elements $\gamma_0, \dots, \gamma_{M-1}$. The vectors of predictions $z_t = (z_{t,0}, \dots, z_{t,M-1})$, $z_{t,k} = \langle w_{t,k}, x_t\rangle$, where $w_{t,k}$ are generated by $\mathscr{A}_{\gamma_k}(\overline{\lambda})$, serve as an expert advice input to the VAW (meta) algorithm $\mathscr{A}(\lambda)$, which mixes them to produce a master prediction. The approach of [5] is more complex: it retains the regret estimate in the form of (2.2) and employs a more sophisticated meta-algorithm.

Theorem 3.2. *Let $\|u_t\|_2 \le R$. For DVAW forecasters $\mathscr{A}_{\gamma_k}(\overline{\lambda})$, $\gamma_k \in \mathscr{S}_\gamma$ take VAW $\mathscr{A}(\lambda)$ as a meta-algorithm. Then*

$$\begin{aligned}
R_T^{\mathscr{A}(\lambda)}(\mathbf{u}) &\le \frac{\lambda}{2} + \frac{MY^2}{2}\ln\left(1+\frac{Z_T^2}{\lambda M}\right) \\
&\quad + (1+b)\sqrt{\frac{d}{2}a(aR+Y)P_T(\mathbf{u})\Delta_{1:T}^2} + \frac{1}{2}(Y+\tilde{Y})^2 \\
&\quad + \overline{\lambda}RP_T(\mathbf{u}) + \frac{d}{2}\max_{1\le t\le T}\Delta_t^2 \ln\left(1+\frac{a^2T}{\overline{\lambda}d}\right) + \frac{\overline{\lambda}}{2}R^2, \qquad (3.3)\\
Z_T^2 &= \left[(M-1)((Y+\widetilde{Y})^2+4Y^2)+\widetilde{Y}^2\right]T + 2(M-1)d(Y+\widetilde{Y})^2\ln\left(1+\frac{a^2T}{\overline{\lambda}d}\right).
\end{aligned}$$

In particular, for $\overline{\lambda} \propto 1/T$ we have

$$R_T^{\mathscr{A}(\lambda)}(\mathbf{u}) = O\left((MY^2 + d(Y+\widetilde{Y})^2)\ln T + (1+b)\sqrt{da(aR+Y)P_T(\mathbf{u})\Delta_{1:T}^2}\right), \quad T\to\infty,$$

since $Z_T^2/M = O(T)$.

Proof. Let $z_t = (z_{t,0}, \dots, z_{t,M-1})$, $z_{t,k} = \langle w_{t,k}, x_t\rangle$, where $w_{t,k}$ are generated by the DVAW algorithms with $\gamma = \gamma_k$. For any $k \in \{0, \dots, M-1\}$, which can depend on $\eta^* = \eta^*(\boldsymbol{y}, \mathbf{u})$, we have

$$\begin{aligned}
R_T^{\mathscr{A}(\lambda)}(\mathbf{u}) &= \frac{1}{2}\sum_{t=1}^T(\langle z_t, \alpha_t\rangle - y_t)^2 - \frac{1}{2}\sum_{t=1}^T(\langle z_t, e_k\rangle - y_t)^2 \\
&\quad + \frac{1}{2}\sum_{t=1}^T(z_{t,k}-y_t)^2 - \frac{1}{2}\sum_{t=1}^T(\langle u_t, x_t\rangle - y_t)^2 \\
&= \underbrace{R_T^{\mathscr{A}(\lambda)}(e_k)}_{\text{Meta-learner's regret w.r.t. expert } k} + \underbrace{R_T^{\mathscr{A}_{\gamma_k}}(\mathbf{u})}_{\text{Regret of expert k}},
\end{aligned}$$

where α_t are produced by the VAW algorithm $\mathscr{A}(\lambda)$, and e_k is the one-hot vector with 1 at index k. Recall that $z_{t,0} = \widetilde{y}_t$ by convention. We need to estimate two terms in the right-hand side.

(1) *Bounding meta-learner's regret $R_T^{\mathscr{A}(\lambda)}(e_k)$.* For the regret

$$R_T^{\mathscr{A}(\lambda)}(e_k) = \frac{1}{2}\sum_{t=1}^T(\langle z_t, \alpha_t\rangle - y_t)^2 - \frac{1}{2}\sum_{t=1}^T(\langle z_t, e_k\rangle - y_t)^2$$

of the VAW algorithm we will use the bound (2.1):

$$R_T^{\mathscr{A}}(\alpha) \le \frac{\lambda}{2}\|e_k\|_2^2 + \frac{MY^2}{2}\ln\left(1+\frac{\sum_{t=1}^T \|z_t\|_2^2}{\lambda M}\right).$$

To estimate $\|z_t\|_2$ put $\mathbf{u}=\mathbf{0}$ in (2.2). For $k\ge 1$, we get

$$\begin{aligned}\sum_{t=1}^T z_{t,k}^2 &= \sum_{t=1}^T (\langle w_{t,k},x_t\rangle - y_t + y_t)^2 \le 2\sum_{t=1}^T (\langle w_{t,k},x_t\rangle - y_t)^2 - 2\sum_{t=1}^T y_t^2 + 4\sum_{t=1}^T y_t^2 \\ &= 4R_T^{\mathscr{A}_{\gamma_k}(\overline{\lambda})}(0) + 4\sum_{t=1}^T y_t^2 \\ &\le 2d \max_{1\le t\le T}\Delta_t^2 \ln\left(1+\frac{a^2T}{\overline{\lambda}d}\right) + 2d\ln\left(\frac{1}{\gamma_k}\right)\Delta_{1:T}^2 + 4Y^2T \\ &\le 2d(Y+\widetilde{Y})^2\ln\left(1+\frac{a^2T}{\overline{\lambda}d}\right) + (Y+\widetilde{Y})^2T + 4Y^2T,\end{aligned}$$

where we used that

$$\ln\frac{1}{\gamma_k} \le \frac{1-\gamma_k}{\gamma_k} = \frac{1}{\eta_k} \le \frac{1}{\eta_{\min}} = \frac{1}{2d}, \quad \Delta_{1:T}^2 \le (Y+\widetilde{Y})^2T$$

in the last inequality. For $k=0$,

$$\sum_{t=1}^T z_{t,0}^2 = \sum_{t=1}^T \widetilde{y}_t^2 \le \widetilde{Y}^2T.$$

It follows that

$$\begin{aligned}\sum_{t=1}^T\sum_{k=0}^{M-1} z_{t,k}^2 \le Z_T^2 := \Big[(M-1)((Y+\widetilde{Y})^2+4Y^2)+\widetilde{Y}^2\Big]\,T \\ + 2(M-1)d(Y+\widetilde{Y})^2\ln\left(1+\frac{a^2T}{\overline{\lambda}d}\right),\end{aligned}$$

and

$$R_T^{\mathscr{A}}(e_k) \le \frac{\lambda}{2} + \frac{MY^2}{2}\ln\left(1+\frac{Z_T^2}{\lambda M}\right). \tag{3.4}$$

(2) *Bounding the regret of expert k.* For $k\ge 1$,

$$\begin{aligned}R_T^{\mathscr{A}_{\gamma_k}(\overline{\lambda})}(\mathbf{u}) \le \eta_k a(aR+Y)P_T(\mathbf{u}) + \frac{d}{2\eta_k}\Delta_{1:T}^2 + \overline{\lambda}RP_T(\mathbf{u}) \\ + \frac{d}{2}\max_{1\le t\le T}\Delta_t^2\ln\left(1+\frac{a^2T}{\overline{\lambda}d}\right) + \frac{\overline{\lambda}}{2}R^2\end{aligned}$$

by Lemma 3.1. For $k=0$,

$$R_T^{\mathscr{A}_0(\overline{\lambda})}(\mathbf{u}) = \frac{1}{2}\sum_{t=1}^T(\tilde{y}_t-y_t)^2 - \frac{1}{2}\sum_{t=1}^T(\langle u_t,x_t\rangle - y_t)^2 \le \frac{1}{2}\sum_{t=1}^T(\tilde{y}_t-y_t)^2 = \frac{1}{2}\Delta_{1:T}^2.$$

Select k by the following rules:

(1) $k=0$, if $\eta^*\le\eta_{\min}=2d$;
(2) take k such that $\eta_k\le\eta^*\le b\eta_k$, if $\eta_{\min}\le\eta^*\le\eta_{\max}$, where $\eta_k\in S_\eta$;
(3) $\eta_k=\eta_{\max}=dT$, if $\eta^*\ge\eta_{\max}=dT$.

Let us estimate $R_T^{\mathscr{A}_{\gamma_k}}(\overline{\lambda})$ in each of these cases.

(1) $\eta^* \le \eta_{\min}$. From the bound

$$\eta^* = \sqrt{\frac{d\Delta_{1:T}^2}{2a(aR+Y)P_T(\mathbf{u})}} \le \eta_{\min} = 2d$$

it follows that

$$\sqrt{\frac{\Delta_{1:T}^2}{2}} \le 2\sqrt{da(aR+Y)P_T(\mathbf{u})}.$$

Hence,

$$R_T^{\mathscr{A}_0(\overline{\lambda})}(\mathbf{u}) \le \sqrt{\frac{1}{2}\Delta_{1:T}^2}\sqrt{\frac{1}{2}\Delta_{1:T}^2} \le 2\sqrt{da(aR+Y)P_T(\mathbf{u})}\sqrt{\frac{1}{2}\Delta_{1:T}^2} \tag{3.5}$$

(2) Let $\eta_{\min} \le \eta^* \le \eta_{\max}$. Take k such that $\eta_k \le \eta^* \le b\eta_k$. Then

$$\begin{aligned}\psi(\eta_k) &:= \eta_k a(aR+Y)P_T(\mathbf{u}) + \frac{d}{2\eta_k}\Delta_{1:T}^2 \le \eta^* a(aR+Y)P_T(\mathbf{u}) + \frac{bd}{2\eta^*}\Delta_{1:T}^2 \\ &= (1+b)\sqrt{\frac{d}{2}a(aR+Y)P_T(\mathbf{u})\Delta_{1:T}^2}.\end{aligned}$$

Hence,

$$\begin{aligned}R_T^{\mathscr{A}_k}(\mathbf{u}) &\le \psi(\eta_k) + \overline{\lambda}RP_T(\mathbf{u}) + \frac{d}{2}\max_{1\le t\le T}\Delta_t^2 \ln\left(1+\frac{a^2T}{\overline{\lambda}d}\right) + \frac{\overline{\lambda}}{2}R^2 \\ &\le (1+b)\sqrt{\frac{d}{2}a(aR+Y)P_T(\mathbf{u})\Delta_{1:T}^2} + \overline{\lambda}RP_T(\mathbf{u}) \\ &+ \frac{d}{2}\max_{1\le t\le T}\Delta_t^2 \ln\left(1+\frac{a^2T}{\overline{\lambda}d}\right) + \frac{\overline{\lambda}}{2}R^2.\end{aligned} \tag{3.6}$$

(3) Let $\eta^* \ge \eta_{\max}$. Take $\eta_k = \eta_{\max} = dT$. If $P_T(\mathbf{u}) > 0$, then

$$\begin{aligned}\psi(\eta_{\max}) &\le \eta^* a(aR+Y)P_T(\mathbf{u}) + \frac{1}{2T}\Delta_{1:T}^2 \\ &\le \sqrt{\frac{d}{2}a(aR+Y)P_T(\mathbf{u})\Delta_{1:T}^2} + \frac{1}{2}(Y+\tilde{Y})^2.\end{aligned}$$

Clearly, the last estimate remains true also for $P_T(\mathbf{u}) = 0$. It follows that

$$\begin{aligned}R_T^{\mathscr{A}_k}(\mathbf{u}) &\le \psi(\eta_k) + \overline{\lambda}RP_T(\mathbf{u}) + \frac{d}{2}\max_{1\le t\le T}\Delta_t^2 \ln\left(1+\frac{a^2T}{\overline{\lambda}d}\right) + \frac{\overline{\lambda}}{2}R^2 \\ &\le \sqrt{\frac{d}{2}a(aR+Y)P_T(\mathbf{u})\Delta_{1:T}^2} + \frac{1}{2}(Y+\tilde{Y})^2 + \overline{\lambda}RP_T(\mathbf{u}) \\ &+ \frac{d}{2}\max_{1\le t\le T}\Delta_t^2 \ln\left(1+\frac{a^2T}{\overline{\lambda}d}\right) + \frac{\overline{\lambda}}{2}R^2.\end{aligned} \tag{3.7}$$

Collecting (3.5), (3.6), (3.7), we conclude that

$$R_T^{\mathscr{A}_{\gamma_k}}(\mathbf{u}) \le (1+b)\sqrt{\frac{d}{2}a(aR+Y)P_T(\mathbf{u})\Delta_{1:T}^2} + \frac{1}{2}(Y+\tilde{Y})^2$$
$$+\overline{\lambda}RP_T(\mathbf{u}) + \frac{d}{2}\max_{1\le t\le T}\Delta_t^2 \ln\left(1+\frac{a^2T}{\overline{\lambda}d}\right) + \frac{\overline{\lambda}}{2}R^2.$$

Together with (3.4) this imples the desired inequlality (3.3). □

We refer to the algorithm proposed in Theorem 3.2 as VE-DVAW (VAW-Ensembled DVAW).

Remark 3.3. The basic result of [5] allows to directly deduce the regret bound $R_T(\mathbf{u}) = O(\ln T + \sqrt{TP_T(\mathbf{u})})$ for their ensemble (Theorem 4.2) under the same assumptions as in Theorem 3.2. The bound $R_T(\mathbf{u}) = O((\ln T)^2 + \sqrt{TP_T(\mathbf{u})})$ of Theorem3.2 for our simple VE-DVAW architecture is slightly worse (recall that $M = \ln T$).

Remark 3.4. The authors of [5] also explore using the learner's own prediction as a hint: $\widetilde{y}_t = \langle x_t, w_t\rangle$, to deduce small-loss bounds. However, this creates a circular definition, as the weight vector w_t depends on $\widetilde{y}_t$ and vice-versa. The paper does not specify a procedure for resolving this implicit definition. So, it seems that this approach is not justified.

4. Computer experiments

The main goal of our computer experiments is to compare the VAW algorithm with the ensemble of DVAW forecasters, considered in Theorem 3.2. We generated six synthetic datasets, each with $T = 10000$ samples and $d = 5$ input features. The data are generated from a time-varying linear model:

$$y_t = \langle w_t^*, x_t\rangle + \varepsilon_t,$$

where the inputs x_t are drawn from a standard normal distribution, $x_t \sim \mathscr{N}(0, I_d)$, and the noise term ε_t is drawn from $\mathscr{N}(0, 0.2^2)$, unless specified otherwise. We note that drawing features from a Gaussian distribution, as is common in experimental setups, means that the boundedness assumptions on $\|x_t\|_2$, $|y_t|$, required for our theoretical analysis are only satisfied with high probability for sufficiently large constants a and Y. The scenarios are designed to test different aspects of adaptivity.

(1) **Stationary environment:** The true weight vector is constant throughout the experiment. This serves as a baseline to evaluate performance in a stable environment:
$$w_t^* = [1.0, -0.5, 0.2, -0.8, 1.2]^\top.$$
(2) **Concept shif:** The underlying weight vector changes suddenly at two points in time. This tests the ability to recover quickly from drastic changes:
$$w_t^* = \begin{cases} [1.0, -0.5, 0.2, -0.8, 1.2]^\top & \text{if } t \le 3333, \\ [-1.0, 1.0, 0.0, 1.0, -0.5]^\top & \text{if } 3333 < t \le 6666, \\ [0.5, 0.5, -0.5, -0.5, 1.5]^\top & \text{if } t > 6666. \end{cases}$$

(3) **Gradual drift:** The true weights evolve according to a random walk. This simulates a continuously evolving environment and tests the tracking capability of the algorithms:

$$w_t^* = w_{t-1}^* + v_t, \quad \text{where } v_t \sim \mathcal{N}(0, (0.05)^2 I_d).$$

The process is initialized with $w_0^* = [1.0, -0.5, 0.2, -0.8, 1.2]^\top$.

(4) **Periodic drift:** Each component of the true weight vector follows a periodic function to model cyclical or seasonal effects:

$$w_{t,j}^* = c_j + A_j \sin\left(\frac{2\pi}{p_j} t\right), \quad j = 1, \dots, 5,$$

where:
- $c = [1.0, -0.5, 0.2, -0.8, 1.2]^\top$ is the stationary center (as in Scenario 1),
- $A = [0.5, 0.8, 0.3, 0.6, 0.4]^\top$ contains the oscillation amplitudes,
- $p = [100, 250, 500, 150, 300]^\top$ specifies the period lengths (in timesteps).

(5) **Changing noise:** The true weight vector is stationary (as in Scenario 1), but the variance of the observation noise changes over time:

$$\varepsilon_t \sim \mathcal{N}(0, \sigma_t^2), \quad \text{where } \sigma_t = \begin{cases} 0.1 & \text{if } t \le 3333 \quad \text{(low noise)}, \\ 1.0 & \text{if } 3333 < t \le 6666 \quad \text{(high noise)}, \\ 0.1 & \text{if } t > 6666 \quad \text{(low noise)}. \end{cases}$$

(6) **Covariate shift:** The true weight vector is stationary (as in Scenario 1), but the distribution of the input features x_t shifts:

$$x_t \sim \begin{cases} \mathcal{N}(0, I_d) & \text{if } t \le 5000 \\ \mathcal{N}(\mu, I_d) & \text{if } t > 5000 \end{cases}$$

where the mean shifts to $\mu = [2, 2, -2, -2, 0]^\top$.

We compare the performance of three algorithms, whose parameters are detailed below. In all cases the hint for the DVAW experts was set to $\tilde{y}_t = 0$.

(a) **VAW**: The standard Vovk-Azoury-Warmuth algorithm, serving as our primary baseline. This is equivalent to a DVAW forecaster with the discounting factor $\gamma = 1.0$ and the regularization parameter $\lambda = 0.1$.

(b) **VE-DVAW (Th)**: the ensemble method described in Theorem 3.2. It uses the VAW meta-learner with regularization $\lambda = 0.1$ to aggregate a pool of DVAW experts. The experts share the regularization parameter $\overline{\lambda} = 10/T = 0.001$. The grid of discount factors $\mathscr{S}_\gamma$ is constructed according to the theory with $b = 2$, $\eta_{\min} = 2d = 10$, and $\eta_{\max} = dT = 50000$. This results in a pool of 16 distinct experts. The γ values, rounded to six decimal places for clarity, are:

$$\begin{aligned} \mathscr{S}_\gamma = \{&0.000000, 0.909091, 0.952381, 0.975610, 0.987654, 0.993789, \\ &0.996895, 0.998439, 0.999220, 0.999609, 0.999805, 0.999902, \\ &0.999951, 0.999975, 0.999980, 1.000000\}. \end{aligned}$$

(c) **VE-DVAW (Pr)**: An ensemble with the same architecture and parameters as VE-DVAW (Th), except that it uses a smaller, manually-selected “practical” set of four discount factors: $\gamma \in \{0.70, 0.85, 0.95, 1.00\}$.

The averaged MSE for each algorithm across the six scenarios, averaged over 5 runs, are summarized in Table 1. The best results are shown in bold. These results clearly illustrate the trade-off between the static VAW algorithm and the adaptive ensembles. In scenarios with a fixed underlying model (Stationary, Changing noise, and Covariate shift), the standard VAW performs best, as expected, since there is no benefit to forgetting past data. However, the performance loss of the ensemble methods is minimal, demonstrating their robustness. Note that for Stationary scenario the best possible MSE is 0.04, which is the variance of noise.

Conversely, in all scenarios featuring parameter drift (Concept shift, Gradual drift, and Periodic drift), both ensemble methods significantly outperform the standard VAW, which is unable to adapt to the changes. Notably, the VE-DVAW (Th) with its theoretically derived grid of experts shows an advantage in tracking gradual drifts, as seen in the Gradual and Periodic drift scenarios.

These results empirically validate that the proposed ensembling method provides substantial gains in non-stationary settings at a small cost in stationary ones.

TABLE 1. MSE of Meta-Learners vs. Individual Experts ($T = 10000$).

Scenario	**Algorithms**					
	VAW ($\gamma = 1.0$)	**DVAW** ($\gamma = 0.70$)	**DVAW** ($\gamma = 0.85$)	**DVAW** ($\gamma = 0.95$)	**VE-DVAW** (Th)	**VE-DVAW** (Pr)
Stationary	**0.0444**	2.1575	1.1367	0.2937	0.0467	0.0468
Concept Shift	2.6261	2.1089	1.1149	0.3084	**0.1133**	0.1162
Gradual Drift	21.6851	40.0723	20.7532	4.9391	**0.5746**	0.8117
Periodic Drift	0.7942	2.6318	1.4205	0.5994	**0.3147**	0.3750
Changing Noise	**0.3459**	2.4841	1.4711	0.6177	0.3495	0.3488
Covariate Shift	**0.0445**	3.4762	1.7214	0.4025	0.0468	0.0468

To perform an additional rough evaluation of the quality of the proposed ensemble, we used the well-known Ader algorithm [11], which has a similar regret bound:

$$R_T(\mathbf{u}) = O(\sqrt{T(1+P_T(\mathbf{u}))}).$$

However, this bound holds true under the assumption that w_t belongs to a bounded feasible set, and the gradients of loss functions are uniformly bounded. As noted previously, our experimental setup with Gaussian features does not satisfy these conditions. The gradients $\nabla \ell_t(w) = (\langle w, x_t \rangle - y_t)x_t$ are unbounded due to the unboundedness of x_t, even if w belongs to a bounded set.

Putting aside these theoretical issues, we considered the parameter grid

$$(D,G) \in \{5,15,30\} \times \{10,100,1000\}$$

and created $N = 9$ Ader "expert" algorithms for all parameter combinations. We used the PyNOL library[1] to run these algorithms in parallel and created the exponentially weighted average (EWA) [2] meta-algorithm, taking convex combinations of their predictions with the weights w_t updated by

$$w_{t,i} = \frac{w_{t-1,i} e^{-\eta(z_{t,i}-y_t)^2}}{\sum_{j=1}^N w_{t-1,j} e^{-\eta(z_{t,j}-y_t)^2}},$$

where

$$\eta = \sqrt{2\frac{\ln N}{T}}, \quad w_1 = (1/N, \dots, 1/N).$$

The results for the best individual experts chosen from the grid in hindsight and the EWA-Ader meta-algorithm are presented in Table 2. The "Best individual expert" column shows the parameters (D,G) and the corresponding MSE for the single expert from the grid that achieved the lowest MSE when averaged over 5 independent runs. The "EWA-Ader" column shows the MSE of the online ensemble, also averaged over the same 5 runs. Interestingly, Ader++ algorithm [11], which has the same theoretical regret bound as Ader, demonstrated worse results, and we do not show them here.

TABLE 2. EWA-Ader vs the best-performing individual Ader expert chosen in hindsight ($T = 10000$).

Scenario	**Best individual expert** (Chosen from grid in hindsight)	**EWA-Ader** (Online tuning)
Stationary	$(D=5, G=10)$ **0.0564**	0.0648
Concept Shift	$(D=15, G=10)$ **0.0928**	0.1003
Gradual Drift	$(D=30, G=10)$ **0.2036**	0.2101
Periodic Drift	$(D=30, G=10)$ **0.2658**	0.2741
Changing Noise	$(D=5, G=10)$ **0.3938**	0.4028
Covariate Shift	$(D=5, G=10)$ **0.0623**	0.0708

The EWA-Ader ensemble's performance is only slightly worse than that of the best individual expert. Furthermore, it adapts to scenarios with parameter drift (Concept shift, Gradual drift, and Periodic drift) even better than VE-DVAW, but underperforms in scenarios with a fixed underlying model (Stationary, Changing noise, and Covariate shift).

[1] `https://github.com/li-lf/PyNOL`

5. Conclusion

In this paper we addressed the challenge of adapting the Vovk-Azoury-Warmuth forecaster, a powerful algorithm for stationary online regression, to dynamic environments. Building on the work of [5], we proposed an ensemble method that leverages a pool of discounted VAW (DVAW) experts, each configured with a different discount factor γ. We employed the standard VAW forecaster as a meta-learner to dynamically aggregate the predictions of these experts. We simplified the dynamic regret bound of [5] for the DVAW algorithm, and directly obtained the simplified dynamic regret bound of the form $O((\ln T)^2 + \sqrt{TP_T(\mathbf{u})})$ for the proposed ensemble. Here $P_T(\mathbf{u})$ is the path length of the comparator sequence.

Our empirical evaluation on several synthetic datasets demonstrated that the proposed ensemble, which we call VE-DVAW, significantly outperforms the standard VAW algorithm in non-stationary settings, including abrupt concept shifts, gradual drift, and periodic changes. Crucially, the ensemble remains robust and highly competitive in stationary environments, incurring only a minimal performance penalty. Furthermore, the VE-DVAW showed performance competitive with a tuned ensemble of Ader algorithms, confirming its place among effective methods for this problem class.

An interesting direction for future work is to extend the proposed adaptive ensembling techniques to non-linear regression in Reproducing Kernel Hilbert Spaces (RKHS). Combining our VE-DVAW architecture with methods like random Fourier features [8] could potentially lead to efficient, scalable, and adaptive algorithms for non-linear regression in dynamic environment. We refer to [9] dealing with a similar architecture in the static environment.

References

[1] Azoury, K.S. and Warmuth, M.K.: Relative loss bounds for on-line density estimation with the exponential family of distributions, *Machine Learning*, **43**, no.3 (2001), 211–246.

[2] Cesa-Bianchi, N. and Lugosi, G.: *Prediction, learning, and games*, Cambridge University Press, 2006.

[3] Hager, W.W.: Updating the inverse of a matrix, *SIAM Review*, **31**, no.2 (1989), 221-239.

[4] Hazan, E.: *Introduction to Online Convex Optimization*. 2nd ed., MIT Press, 2022.

[5] Jacobsen, A. and Cutkosky, A.: Online linear regression in dynamic environments via discounting, *Proceedings of the 41st International Conference on Machine Learning*, **235**, (2024) 21083-21120.

[6] Mayo, J.J., Hadiji, H., van Erven, T.: Scale-free unconstrained online learning for curved losses, *Proceedings of Thirty Fifth Conference on Learning Theory*, **178**, (2022) 4464–4497.

[7] Orabona, F.: *A Modern Introduction to Online Learning*, arXiv:1912.13213v7 [cs.LG] (2025).

[8] Rahimi, A. and Recht, B.: Random features for large-scale kernel machines *Advances in neural information processing systems*, **20**, (2007) 1177-1184.

[9] Rokhlin, D.B. and Gurtovaya, O.V.: *Random feature-based double Vovk-Azoury-Warmuth algorithm for online multi-kernel learning*, arXiv:2503.20087v2 [cs.LG] (2025).

[10] Vovk, V.: Competitive on-line statistics, *International Statistical Review*, **69**, no.2 (2001), 213–248.

[11] Zhang, L., Lu, S. and Zhou, Z.-H.: Adaptive online learning in dynamic environments, *Advances in neural information processing systems*, **31**, (2018), 1327-1337.

[12] Zhao, P., Zhang, Y.-J., Zhang, L., Zhou, Z.-H.: Adaptivity and non-stationarity: Problem-dependent dynamic regret for online convex optimization, *Journal of Machine Learning Research*, **25**, no.98 (2024), 1–52.

[13] Zinkevich, M.: Online convex programming and generalized infinitesimal gradient ascent, *Proceedings of the 20th International Conference on Machine Learning*, (2003), 928–936.

DMITRY B. ROKHLIN, INSTITUTE OF MATHEMATICS, MECHANICS AND COMPUTER SCIENCES AND REGIONAL SCIENTIFIC AND EDUCATIONAL MATHEMATICAL CENTER OF THE SOUTHERN FEDERAL UNIVERSITY, RUSSIAN FEDERATION
Email address: dbrohlin@sfedu.ru

GEORGIY A. KARAPETYANTS, INSTITUTE OF MATHEMATICS, MECHANICS AND COMPUTER SCIENCES OF THE SOUTHERN FEDERAL UNIVERSITY, RUSSIAN FEDERATION
Email address: karape@sfedu.ru