

---

# A CLOSER LOOK AT TEMPORAL VARIABILITY IN DYNAMIC ONLINE LEARNING

---

A PREPRINT

**Nicolò Campolongo**  
Università di Milano  
nicolo.campolongo@unimi.it

**Francesco Orabona**  
Boston University  
francesco@orabona.com

February 15, 2021

## ABSTRACT

This work focuses on the setting of dynamic regret in the context of online learning with full information. In particular, we analyze regret bounds with respect to the temporal variability of the loss functions. By assuming that the sequence of loss functions does not vary much with time, we show that it is possible to incur improved regret bounds compared to existing results. The key to our approach is to use the loss function (and not its gradient) during the optimization process. Building on recent advances in the analysis of Implicit algorithms, we propose an adaptation of the Implicit version of Online Mirror Descent to the dynamic setting. Our proposed algorithm is adaptive not only to the temporal variability of the loss functions, but also to the path length of the sequence of comparators when an upper bound is known. Furthermore, our analysis reveals that our results are tight and cannot be improved without further assumptions. Next, we show how our algorithm can be applied to the setting of learning with expert advice or to settings with composite loss functions. Finally, when an upper bound to the path-length is not fixed beforehand we show how to combine a greedy strategy with existing strongly-adaptive algorithms to compete optimally against different sequences of comparators simultaneously.

## 1 Introduction

Online learning is a powerful tool in modeling many practical scenarios. Furthermore, in recent years it has led to advancements in various areas of machine learning in general, both practically and theoretically. Formally, given a convex set  $\mathcal{V} \subseteq \mathbb{R}^d$ , a time horizon  $T$  and a sequence of cost functions  $\ell_1, \dots, \ell_t$ , in the online learning setting the goal is to design algorithms such that for any comparator model  $\mathbf{u} \in \mathcal{V}$  the regret is minimized,

$$R_T(\mathbf{u}) \triangleq \sum_{t=1}^T \ell_t(\mathbf{x}_t) - \sum_{t=1}^T \ell_t(\mathbf{u}),$$

where  $\mathbf{x}_t$  is the output of the algorithm at time  $t$ . In particular, the objective is to have algorithms whose regret can be provably upper bounded by a quantity which grows sublinearly in  $T$ .

While the static regret is a well-studied objective and many algorithms have a sublinear regret upper bound, sometimes competing with the best comparator is not meaningful. Indeed, there are situations where the environment is not stationary. In this case, rather than comparing the performance of an algorithm against a single fixed model, it is preferable to compete against a “moving” target, i.e., a sequence of different comparators. In this work, we focus on online learning in the dynamic setting, considering the *full-information* feedback, where in every round the loss function is revealed.

To model dynamic environments, stronger notions of regret are used. In particular, we consider the *general dynamic regret* [Zinkevich, 2003, Hall and Willett, 2013] against the sequence  $\mathbf{u}_{1:T} \triangleq (\mathbf{u}_1, \dots, \mathbf{u}_T)$  as

$$R_T(\mathbf{u}_{1:T}) \triangleq \sum_{t=1}^T \ell_t(\mathbf{x}_t) - \sum_{t=1}^T \ell_t(\mathbf{u}_t). \quad (1)$$

It can be shown that it is impossible to achieve sublinear dynamic regret in the worst-case. However, if one

puts some restrictions on the sequence  $\mathbf{u}_{1:T}$  and makes some regularities assumptions, then Eq. (1) can be sublinear in  $T$ . There are various measures which can be used to model the regularity of the environment. A natural measure of non-stationarity introduced in Zinkevich [2003] is the *path-length*<sup>1</sup> of the sequence  $\mathbf{u}_{1:T}$ , which we denote by

$$C_T(\mathbf{u}_{1:T}) \triangleq \sum_{t=2}^T \|\mathbf{u}_t - \mathbf{u}_{t-1}\|. \quad (2)$$

Another measure of non-stationarity is given by the *temporal variability* of the loss functions [Besbes et al., 2015]. Formally, let  $\ell_{1:T}$  be the shorthand for  $(\ell_1, \dots, \ell_T)$ , the temporal variability of a sequence  $\ell_{1:T}$  is defined as

$$V_T(\ell_{1:T}) \triangleq \sum_{t=2}^T \max_{\mathbf{x} \in \mathcal{V}} |\ell_t(\mathbf{x}) - \ell_{t-1}(\mathbf{x})|. \quad (3)$$

In the remaining we will use the shorthands  $C_T$  for  $C_T(\mathbf{u}_{1:T})$  and  $V_T$  for  $V_T(\ell_{1:T})$  when the context is clear. A particular case of dynamic regret is the so-called *restricted* setting [Besbes et al., 2015, Jadbabaie et al., 2015, Yang et al., 2016]. In this setting, the sequence of comparators is given by the local minimizers of the loss functions, i.e.,  $\mathbf{u}_{1:T}^* := (\mathbf{u}_1^*, \dots, \mathbf{u}_T^*)$ , where  $\mathbf{u}_t^* = \operatorname{argmin}_{\mathbf{x} \in \mathcal{V}} \ell_t(\mathbf{x})$ .

Most recent developments in online learning have been driven by the use of two paradigms: *Online Mirror Descent* (OMD) and *Follow The Regularized Leader* (FTRL) (see the surveys Shalev-Shwartz [2012], Orabona [2019]). Both of them usually achieve the same regret bounds thanks to the linearization trick: given the convexity of the loss functions one can exploit the fact that  $\ell_t(\mathbf{x}_t) - \ell_t(\mathbf{u}) \leq \langle \mathbf{g}_t, \mathbf{x}_t - \mathbf{u} \rangle$ , where  $\mathbf{g}_t \in \partial \ell_t(\mathbf{x}_t)$  is a subgradient of the loss function. One can therefore shift her goal to minimize this new objective over time. On the other hand, we choose to not use subgradients in the optimization process but the loss function directly. We will show that this is the key factor in order to obtain dynamic regret bounds depending on the temporal variability  $V_T$ .

**Contributions.** The main results of this paper are summarized below:

- In Section 4, we show that there exists a simple strategy which achieves an upper bound of  $\mathcal{O}(V_T)$  on the dynamic regret. We also provide a lower bound which shows that this regret bound is tight. Next, we show when this strategy fails and why we need different algorithms.
- In Section 5, using recent advances in the analysis of implicit updates in online learning we design an algorithm which is adaptive to both  $C_T$  and  $V_T$ . Using an adaptation of OMD

to the implicit case, we will provide an algorithm which incurs a dynamic regret bound of  $\mathcal{O}(V_T, \sqrt{T(1+\tau)})$ , for all sequence of comparators whose path-length  $C_T$  is upper bounded by  $\tau$ .

- Finally, when the complexity of the class of comparators is not fixed in advance in terms of path-length (i.e., an upper bound  $C_T \leq \tau$  is not fixed beforehand), in Section 6 we show how to combine the strategy from Section 4 with an existing algorithm and get the optimal bound of  $\tilde{\mathcal{O}}(\min\{V_T, \sqrt{T(1+C_T(\mathbf{u}_{1:T}))}\})^2$  for any possible sequence  $\mathbf{u}_{1:T}$ .

## 2 Related work

In this section, we are going to review the two lines of work most related to ours: algorithms designed for non-stationary environments and implicit updates in online learning. We recap existing results and highlight both similarities and differences compared to our results.

**Path-length.** The notion of dynamic regret was first introduced in the seminal work of Zinkevich [2003], which proved that *Online Gradient Descent* incurs a regret bound of  $\mathcal{O}(\sqrt{T}(1+C_T))$ . This result was later extended by Hall and Willett [2013] who considered a modified (and possibly richer) definition of path-length. A lower bound of  $\Omega(\sqrt{T(1+C_T)})^3$  in terms of path-length is shown in Zhang et al. [2018a, Theorem 2], who also provide an algorithm which matches it.

**Temporal Variability.** Besbes et al. [2015] provided an analysis of restarted gradient descent in the setting of stochastic optimization with noisy gradients which incurs  $\mathcal{O}(T^{2/3}(V_T' + 1)^{1/3})$ , where  $V_T'$  is an upper bound on  $V_T$  known in advance. Jadbabaie et al. [2015] gave an algorithm achieving a restricted dynamic regret of  $\tilde{\mathcal{O}}(\sqrt{G_T} + \min(\sqrt{(G_T+1)C_T}, ((G_T+1)T)^{1/3}(V_T+1)^{2/3}))$ , where  $G_T = \sum_{t=1}^T \|\nabla f_t(\mathbf{x}_t) - \mathbf{p}_t\|_*^2$  and  $\mathbf{p}_1, \dots, \mathbf{p}_T$  is a predictable sequence computable at the start of round  $t$ . Importantly, this bound is obtained without prior knowledge of  $G_T$ ,  $C_T$  and  $V_T$  but under the assumption that all of them can be *observable*. If one limits the algorithm to not use predictable sequences, then the bound reduces to  $\tilde{\mathcal{O}}(\sqrt{T} + \min\{\sqrt{T(1+C_T)}, T^{1/3}(V_T+1)^{2/3}\})$ . In Section 5, we design an algorithm similar in spirit to the one from Jadbabaie et al. [2015], which incurs an improved regret bound of  $\min\{\sqrt{T(1+C_T)}, V_T\}$  when  $C_T$  is fixed in advance or can be observed.

**Adaptive Regret.** A parallel line of work on non-stationary environments involves the study of the weakly

<sup>2</sup>The  $\tilde{\mathcal{O}}$  notation hides poly-logarithmic terms.

<sup>3</sup>To avoid clutter, w.l.o.g. we suppress parameters other than  $T$  and  $C_T$  in the asymptotic notation.

<sup>1</sup>One could also consider other versions of path-length, such as its squared version [Yang et al., 2016].

and strongly-adaptive regret [Hazan and Seshadhri, 2007, Daniely et al., 2015], which aims to minimize the static regret over any possible (sub)interval over the time horizon  $T$ . Importantly, it has been shown that strongly-adaptive regret bounds imply dynamic regret bounds. Recently, Cutkosky [2020] provided a strongly-adaptive algorithm that achieves the optimal dynamic regret bound in terms of path-length, for any sequence of comparators. On the other hand, in Zhang et al. [2018b] it is shown that dynamic regret of the strongly adaptive algorithm given in Jun et al. [2017] is  $\tilde{\mathcal{O}}(T^{2/3}(V_T + 1)^{1/3})$ . To summarize, using existing strongly-adaptive algorithms, a dynamic regret bound of  $\tilde{\mathcal{O}}(\min\{\sqrt{T(1 + C_T)}, T^{2/3}(V_T + 1)^{1/3}\})$  for any sequence  $\mathbf{u}_{1:T}$  can be achieved, without knowing its path-length in advance. In Section 6, we show instead how to improve this bound to  $\mathcal{O}(\min\{\sqrt{T(1 + C_T)}, V_T\})$ .

**Implicit Algorithms.** Implicit algorithms are known in the optimization literature as proximal methods [Parikh and Boyd, 2014] and can be traced back to the work of Moreau [1965]. In the online learning community, they have been introduced in Kivinen and Warmuth [1997]. In a recent work Campolongo and Orabona [2020] showed that implicit updates can outperform their linearized counterparts when the temporal variability is low, in the static setting. The dynamic regret for proximal algorithms in the online setting has been also studied in the case of strongly convex losses in Dixit et al. [2019] and for composite losses in Ajallooeian et al. [2020], but both these works assume different notions of feedback from ours.

### 3 Definitions

For a function  $f : \mathbb{R}^d \rightarrow (-\infty, +\infty]$ , we define a *sub-gradient* of  $f$  in  $\mathbf{x} \in \mathbb{R}^d$  as a vector  $\mathbf{g} \in \mathbb{R}^d$  that satisfies  $f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{g}, \mathbf{y} - \mathbf{x} \rangle$ ,  $\forall \mathbf{y} \in \mathbb{R}^d$ . We denote the set of subgradients of  $f$  in  $\mathbf{x}$  by  $\partial f(\mathbf{x})$ . We denote by  $\mathbf{e}_i$  the standard basis vectors, for  $i = 1, \dots, d$ . We denote the expected value of a random variable  $\mathbf{x}$  by  $\mathbb{E}[\mathbf{x}]$  and the indicator function of the event  $A$  by  $\mathbb{1}\{A\}$ . We denote the *dual norm* of  $\|\cdot\|$  by  $\|\cdot\|_*$ . A proper function  $f : \mathbb{R}^d \rightarrow (-\infty, +\infty]$  is  $\mu$ -*strongly convex* over a convex set  $V \subseteq \text{int dom } f$  w.r.t.  $\|\cdot\|$  if  $\forall \mathbf{x}, \mathbf{y} \in V$  and  $\mathbf{g} \in \partial f(\mathbf{x})$ , we have  $f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{g}, \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2} \|\mathbf{x} - \mathbf{y}\|^2$ . Let  $\psi : X \rightarrow \mathbb{R}$  be strictly convex and continuously differentiable on  $\text{int } X$ . The *Bregman Divergence* w.r.t.  $\psi$  is  $B_\psi : X \times \text{int } X \rightarrow \mathbb{R}_+$  defined as  $B_\psi(\mathbf{x}, \mathbf{y}) = \psi(\mathbf{x}) - \psi(\mathbf{y}) - \langle \nabla \psi(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle$ . We assume that  $\psi$  is strongly convex w.r.t. a norm  $\|\cdot\|$  in  $\text{int } X$ . We also assume w.l.o.g. the strong convexity constant to be 1, which implies

$$B_\psi(\mathbf{x}, \mathbf{y}) \geq \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2, \quad \forall \mathbf{x} \in X, \mathbf{y} \in \text{int } X. \quad (4)$$

---

#### Algorithm 1 Greedy optimizer

---

**Require:** Non-empty closed convex set  $\mathcal{V} \subset X \subset \mathbb{R}^d$ ,  $\mathbf{x}_1 \in \mathcal{V}$

- 1: **for**  $t = 1, \dots, T$  **do**
- 2:   Output  $\mathbf{x}_t \in \mathcal{V}$
- 3:   Receive  $\ell_t : \mathbb{R}^d \rightarrow \mathbb{R}$  and pay  $\ell_t(\mathbf{x}_t)$
- 4:   Update  $\mathbf{x}_{t+1} = \arg \min_{\mathbf{x} \in \mathcal{V}} \ell_t(\mathbf{x})$
- 5: **end for**

---

### 4 A greedy strategy and its limitations

In this section, we presents a greedy strategy that achieves a dynamic regret bound in terms of temporal variability of the loss functions of  $\mathcal{O}(V_T)$ . Furthermore, we show that this bound is optimal proving a matching lower bound. Then, we discuss why this strategy might be harmful if the environment is stationary.

The strategy adopted to achieve a bound in terms of temporal variability  $V_T$  is depicted in Algorithm 1. In each round the algorithm plays the minimizer of the observed loss function in the previous round. Despite being reported in Jadbabaie et al. [2015], we could not find a formal proof of the regret bound of this algorithm. Hence, for completeness we next state a theorem which provides a regret bound to Algorithm 1.

**Theorem 4.1.** *Let  $\mathcal{V} \subset X \subset \mathbb{R}^d$  be non-empty closed convex sets. The regret of Algorithm 1 against any sequence  $\mathbf{u}_{1:T}$  with  $\mathbf{u}_t \in \mathcal{V}$  for all  $t$  is bounded as  $R_T(\mathbf{u}_{1:T}) \leq \max(V_T, \mathcal{O}(1))$ .*

*Proof.* From the update of Algorithm 1, for any  $\mathbf{u}_t \in \mathcal{V}$  we have that

$$\begin{aligned} & \sum_{t=1}^T (\ell_t(\mathbf{x}_t) - \ell_t(\mathbf{u}_t)) \\ &= \sum_{t=1}^T (\ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}_{t+1}) + \ell_t(\mathbf{x}_{t+1}) - \ell_t(\mathbf{u}_t)) \\ &\leq \sum_{t=1}^T (\ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}_{t+1})) \\ &= \ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + \sum_{t=2}^T (\ell_t(\mathbf{x}_t) - \ell_{t-1}(\mathbf{x}_t)) \\ &\leq \ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + V_T \\ &= \max(V_T, \mathcal{O}(1)). \quad \square \end{aligned}$$

**Remark.** The theorem above holds for any sequence of comparators, and in particular for  $\mathbf{u}_{1:T}^* = (\mathbf{u}_1^*, \dots, \mathbf{u}_T^*)$  used in the *restricted* setting. At first sight, this result might seem to be in contrast with the result given in Besbes et al. [2015], which reports a lower bound of  $\Omega((V_T^{1/3} + 1)T^{2/3})$ . However, it should be noted that in Besbes et al. [2015] the feedback is different and not directly comparable to our setting. Indeed, they assume

only access to noisy functions and gradients and therefore their Theorem 2 is not applicable.

In Yang et al. [2016] it is shown that the same strategy of Algorithm 1 achieves an upper bound of  $\mathcal{O}(\max(C_T(u_1^*, \dots, u_T^*), 1))$  when the path-length is taken into account. While the result regarding the path-length is tight, one might wonder if the same could be said about the temporal variability. In the next theorem, we provide a lower bound which shows that the bound in Theorem 4.1 is tight.

**Theorem 4.2.** *Let  $\mathcal{V} = [-1, 1]$ , and  $C$  be a positive constant independent of  $T$ . Then, for any algorithm  $\mathcal{A}$  on  $\mathcal{V}$ , and any  $\sigma \in (1/\sqrt{T}, 1)$ , there exists a sequence of loss functions  $\ell_1, \dots, \ell_T$  with temporal variability less than or equal to  $2\sigma T$  such that*

$$R(u_{1:T}) \geq CV_T^\gamma, \quad (5)$$

for any  $\gamma \in (0, 1)$ .

*Proof.* Similarly to Yang et al. [2016], we consider a simple 1-d problem and employ the following sequence of loss functions. Define  $\ell_t(x_t) = \frac{1}{2}(x_t - \varepsilon_t)^2$ , where  $\varepsilon_1, \dots, \varepsilon_T$  is a sequence of random variables sampled uniformly at random between the two values  $\{-\sigma, \sigma\}$ . Note that we have  $\mathbb{E}[\varepsilon_t] = 0$  and  $\text{Var}(\varepsilon_t) = \mathbb{E}[\varepsilon_t^2] = \sigma^2$ . Obviously, the optimal choice in every round is  $u_t = \varepsilon_t$ . Assume  $T \geq 1$ . Then, the restricted dynamic regret is given by

$$\begin{aligned} \mathbb{E}[R_T(u_{1:T})] &= \mathbb{E}\left[\sum_{t=1}^T \ell_t(x_t) - \ell_t(\varepsilon_t)\right] \\ &= \sum_{t=1}^T \frac{1}{2}\mathbb{E}[x_t^2] + \frac{1}{2}\mathbb{E}[\varepsilon_t^2] \geq \frac{\sigma^2}{2}T, \end{aligned} \quad (6)$$

where the expectation is taken with respect to the randomness in the sequence of loss functions and any algorithm  $\mathcal{A}$ , while the inequality is due to the fact that  $x_t$  is independent from  $\varepsilon_t$  and  $\mathbb{E}[\varepsilon_t] = 0$ . Now, note that we can upper bound the temporal variability as follows

$$\begin{aligned} V_T &= \sum_{t=1}^{T-1} \max_{x \in \mathcal{V}} |\ell_t(x) - \ell_{t+1}(x)| \\ &= \sum_{t=1}^{T-1} \max_{x \in \mathcal{V}} \left| \frac{1}{2}(x - \varepsilon_t)^2 - \frac{1}{2}(x - \varepsilon_{t+1})^2 \right| \\ &= \sum_{t=1}^{T-1} \max_{x \in \mathcal{V}} |x(\varepsilon_{t+1} - \varepsilon_t)| \\ &\leq \sum_{t=1}^{T-1} |\varepsilon_{t+1} - \varepsilon_t| \\ &\leq 2\sigma T. \end{aligned} \quad (7)$$

Observe that if we set  $\sigma = C'/2$  for a positive constant  $C'$ , then we recover the result in Proposition 1 of Besbes

et al. [2015] which says that it is impossible to achieve sublinear dynamic regret unless  $V_T = o(T)$ .

Now, we let  $\sigma = T^{-\mu}$ , with  $\mu = (1 - \gamma)/(2 - \gamma)$  and  $\mu \in (0, 1/2)$ . Then, from Eq. (6) we have that  $R(u_{1:T}) \geq T^{1-2\mu}/2$ , while from Eq. (7) we have that  $T \geq (V_T/2)^{\frac{1}{1-\mu}}$ . Therefore, putting things together we have that  $R(u_{1:T}) \geq \frac{1}{2}(V_T/2)^{\frac{1-2\mu}{1-\mu}} = CV_T^\gamma$ . Note that if  $\gamma = 1$  then  $\mu = 0$  and the regret must be linear in  $T$ . Therefore, we let  $\gamma < 1$ .  $\square$

**Remark.** A lower bound in terms of temporal variability for the static regret on constrained domains has been proved in Campolongo and Orabona [2020], which states that for every  $\tau \geq 0$ , there exists a sequence of loss functions such that  $V_T$  is equal to  $\tau$  and the regret satisfies  $R_T(u) \geq \tau$ . However, their proof technique is different and the result is limited to deterministic algorithms, while the lower bound given in Theorem 4.2 holds for randomized algorithms as well.

Eq. (5) implies that it is impossible to achieve a dynamic regret bound better than  $\mathcal{O}(V_T^\gamma)$ , with  $\gamma < 1$ . Note that the proof of lower bound is given for the *restricted* setting, but it automatically implies a lower bound for the general dynamic regret, which includes the *restricted* setting as a particular case.

**Greedy fails in the static case.** Algorithm 1 incurs a dynamic regret bounded as  $R_T(u_{1:T}) \leq \mathcal{O}(\min\{V_T, C_T(u_{1:T}^*)\})$  for any sequence  $u_{1:T}$ . Based on Theorem 4.2, this bound is tight when considering  $V_T$ . However, there are situations where the greedy strategy is doomed to fail. For example, consider the setting of Learning with Expert Advice with two experts, with  $\ell_t(x) = \langle g_t, x \rangle$  and the following choice of  $g_t$ :

$$g_t = \begin{cases} [1, 0], & t \text{ even} \\ [0, 1], & t \text{ odd} \end{cases}$$

It's immediate to see that  $C_T(u_{1:T}^*) = V_T = T$ , and indeed in this case there is no hope of getting an upper bound sublinear in  $T$  when considering  $u_{1:T}^*$ . However, if we consider the static case, then the regret of Algorithm 1 against either  $e_1$  or  $e_2$  is  $R_T(e_i) = T - T/2 = \mathcal{O}(T)$ , while any algorithm designed for the static setting incurs  $R_T(e_i) \leq \mathcal{O}(\sqrt{T})$ . Hence, in these situations Algorithm 1 is not a good choice.

In general, from Zhang et al. [2018a] we know that a lower bound of  $\Omega(\sqrt{T(1+\tau)})$  holds when considering sequences of comparators whose path-length is upper bounded by  $\tau$ . A natural question arises: is it possible to keep the rate of  $\mathcal{O}(V_T)$  but at the same time to guarantee  $\mathcal{O}(\sqrt{T(1+\tau)})$ ?

To the best of our knowledge there are no algorithms which achieve this goal. Indeed, algorithms designed for dynamic regret such as Jadbabaie et al. [2015] and Besbes et al. [2015] have a regret bound of  $\mathcal{O}((V_T^{1/3} + 1)T^{2/3})$ .

**Algorithm 2** Dynamic IOMD

---

**Require:** Non-empty closed convex set  $\mathcal{V} \subset X \subset \mathbb{R}^d$ ,  $\psi : X \rightarrow \mathbb{R}$ ,  $\mathbf{x}_1 \in \mathcal{V}$ ,  $\gamma$  such that  $B_\psi(\mathbf{x}, \mathbf{z}) - B_\psi(\mathbf{y}, \mathbf{z}) \leq \gamma \|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{V}$ , non increasing sequence  $(\eta_t)_{t=1}^T$

- 1: **for**  $t = 1, \dots, T$  **do**
- 2:   Output  $\mathbf{x}_t \in \mathcal{V}$
- 3:   Receive  $\ell_t : \mathbb{R}^d \rightarrow \mathbb{R}$  and pay  $\ell_t(\mathbf{x}_t)$
- 4:   Update  $\mathbf{x}_{t+1} = \arg \min_{\mathbf{x} \in \mathcal{V}} \ell_t(\mathbf{x}) + B_\psi(\mathbf{x}, \mathbf{x}_t)/\eta_t$
- 5: **end for**

---

The same holds true for strongly adaptive algorithms, as shown in Zhang et al. [2018b]. This is not really surprising: a bound of  $\mathcal{O}(V_T)$  would imply constant regret in the case the loss functions are fixed, i.e.,  $\ell_t = \ell$  for all  $t$ . In this case, using an online-to-batch conversion [Cesa-Bianchi et al., 2004] would result in a convergence rate of  $\mathcal{O}(1/T)$ . However, this would be in contrast with the lower bound of  $\Omega(1/\sqrt{T})$  by Nesterov [2013] on non-smooth batch black-box optimization. Unfortunately, all the algorithms mentioned above make use of gradients and therefore are subject to this lower bound.

However, not all is lost: in the next section, we illustrate how to achieve a bound of  $\mathcal{O}(\min\{V_T, \sqrt{T(1+C_T)}\})$  using an algorithm which makes full use of the loss function (and not just its gradient) for the class of sequences  $\mathbf{u}_{1:T}$  with path-length  $C_T$ . Moreover, in Section 6 we show how the greedy strategy can be combined with another algorithm in order to achieve the same goal for all sequences  $\mathbf{u}_{1:T}$  simultaneously.

## 5 Implicit updates in dynamic environments

In a recent work, Campolongo and Orabona [2020] showed how a modified version of OMD with implicit updates achieves a regret bound in the static setting which is order of  $\mathcal{O}(\min(V_T, \sqrt{T}))$ . In this section, we show how to adapt this algorithm to satisfy a bound of  $\mathcal{O}(\min\{V_T, \sqrt{T(1+C_T)}\})$  on the dynamic regret when the path-length of the sequence of comparators is fixed to  $C_T$ .

OMD with implicit updates is depicted in Algorithm 2. The only difference with its linearized counterpart is in the update rule, which uses directly the loss rather than its (sub)gradient in  $\mathbf{x}_t$ :

$$\mathbf{x}_{t+1} = \arg \min_{\mathbf{x} \in \mathcal{V}} \ell_t(\mathbf{x}) + B_\psi(\mathbf{x}, \mathbf{x}_t)/\eta_t. \quad (8)$$

In order to provide a dynamic regret bound to Algorithm 2, we require a Lipschitz continuity condition on the Bregman divergence. Using this assumption, we can get a bound for the dynamic regret shown in the next lemma. We will use this result in Theorem 5.1 to prove the desired bound.

**Lemma 5.1.** *Let  $\mathcal{V} \subset X \subset \mathbb{R}^d$  be non-empty closed convex sets,  $\psi : X \rightarrow \mathbb{R}$ , and  $\mathbf{x}_1 \in \mathcal{V}$ . Assume there exists  $\gamma \in \mathbb{R}$  such that  $B_\psi(\mathbf{x}, \mathbf{z}) - B_\psi(\mathbf{y}, \mathbf{z}) \leq \gamma \|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{V}$ . Define  $D^2 \triangleq \max_{\mathbf{x}, \mathbf{y} \in \mathcal{V}} B_\psi(\mathbf{x}, \mathbf{y})$ . Let  $(\eta_t)_{t=1}^T$  be a non-increasing sequence. Then, the regret of Algorithm 2 against any sequence  $\mathbf{u}_{1:T}$  with  $\mathbf{u}_t \in \mathcal{V}$  for all  $t$  is bounded as follows*

$$R_T(\mathbf{u}_{1:T}) \leq \frac{D^2}{\eta_T} + \gamma \sum_{t=2}^T \frac{\|\mathbf{u}_t - \mathbf{u}_{t-1}\|}{\eta_t} + \sum_{t=1}^T \delta_t, \quad (9)$$

where  $\delta_t = \ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}_{t+1}) - B_\psi(\mathbf{x}_{t+1}, \mathbf{x}_t)/\eta_t$ .

*Proof.* Let  $\mathbf{g}'_t \in \partial \ell_t(\mathbf{x}_{t+1})$ . From the update rule of Algorithm 2 we have that

$$\begin{aligned} \eta_t(\ell_t(\mathbf{x}_{t+1}) - \ell_t(\mathbf{u}_t)) &\leq \langle \eta_t \mathbf{g}'_t, \mathbf{x}_{t+1} - \mathbf{u}_t \rangle \\ &\leq \langle \nabla \psi(\mathbf{x}_t) - \nabla \psi(\mathbf{x}_{t+1}), \mathbf{x}_{t+1} - \mathbf{u}_t \rangle \\ &= B_\psi(\mathbf{u}_t, \mathbf{x}_t) - B_\psi(\mathbf{u}_t, \mathbf{x}_{t+1}) - B_\psi(\mathbf{x}_{t+1}, \mathbf{x}_t), \end{aligned} \quad (10)$$

where the first inequality follows from the convexity of the loss functions, while the second from the first-order optimality condition.

Now, we consider the first two terms of the r.h.s. of Eq. (10). Using the Lipschitz continuity condition on the Bregman divergence and the fact that  $\eta_t$  is non-increasing over time, we get

$$\begin{aligned} &\sum_{t=1}^T \frac{1}{\eta_t} (B_\psi(\mathbf{u}_t, \mathbf{x}_t) - B_\psi(\mathbf{u}_t, \mathbf{x}_{t+1})) \\ &\leq \frac{D^2}{\eta_1} + \sum_{t=2}^T \left( \frac{B_\psi(\mathbf{u}_t, \mathbf{x}_t)}{\eta_t} - \frac{B_\psi(\mathbf{u}_{t-1}, \mathbf{x}_t)}{\eta_{t-1}} \right) \\ &= \frac{D^2}{\eta_1} + \sum_{t=2}^T \left( \frac{B_\psi(\mathbf{u}_t, \mathbf{x}_t)}{\eta_t} - \frac{B_\psi(\mathbf{u}_{t-1}, \mathbf{x}_t)}{\eta_t} \right. \\ &\quad \left. + \frac{B_\psi(\mathbf{u}_{t-1}, \mathbf{x}_t)}{\eta_t} - \frac{B_\psi(\mathbf{u}_{t-1}, \mathbf{x}_t)}{\eta_{t-1}} \right) \\ &\leq \frac{D^2}{\eta_1} + \gamma \sum_{t=2}^T \frac{\|\mathbf{u}_t - \mathbf{u}_{t-1}\|}{\eta_t} \\ &\quad + \sum_{t=2}^T B_\psi(\mathbf{u}_{t-1}, \mathbf{x}_t) \left( \frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right) \\ &\leq \frac{D^2}{\eta_1} + D^2 \left( \frac{1}{\eta_T} - \frac{1}{\eta_1} \right) + \gamma \sum_{t=2}^T \frac{\|\mathbf{u}_t - \mathbf{u}_{t-1}\|}{\eta_t} \\ &= \frac{D^2}{\eta_T} + \gamma \sum_{t=2}^T \frac{\|\mathbf{u}_t - \mathbf{u}_{t-1}\|}{\eta_t}. \end{aligned}$$

Adding  $\ell_t(\mathbf{x}_t)$  on both sides of Eq. (10) and summing over time yields the regret bound in Eq. (9).  $\square$

Notice that the Lipschitz continuity assumption is not a strong requirement. Indeed, when the function  $\psi$  is Lipschitz on  $\mathcal{V}$ , the Lipschitz condition on the Bregman divergence is automatically satisfied. When this is not true, we can still satisfy this condition changing the domain of interest. For example, in the case of learning with expert advice we have that  $\gamma = \mathcal{O}(\ln T)$  if we use a “clipped” simplex (details in Section 5.1).

We still have to set the learning rate in Algorithm 2 and next theorem shows how to do it in order to have a regret bound which is the minimum between temporal variability and path-length. The proof follows the one of Campolongo and Orabona [2020, Theorem 6.12] and is reported in Appendix A for completeness.

**Theorem 5.1.** *Let  $\tau \geq 0$  be a positive constant. Under the assumptions of lemma 5.1, for any sequence  $\mathbf{u}_{1:T}$  whose path length  $C_T(\mathbf{u}_{1:T})$  is less or equal than  $\tau$ , Algorithm 2 with  $1/\eta_t = \lambda_t = \frac{1}{\beta^2} \sum_{i=1}^{t-1} \delta_t$ , and  $\beta^2 = (D^2 + \gamma\tau)$  incurs dynamic regret against upper bounded as follows*

$$R_T(\mathbf{u}_{1:T}) \leq \min \left\{ 2(\ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + V_T), \right. \\ \left. 2\sqrt{(3D^2 + \gamma\tau) \sum_{t=1}^T \|\mathbf{g}_t\|_*^2} \right\}, \quad (11)$$

where  $V_T = \sum_{t=2}^T \max_{\mathbf{x} \in \mathcal{V}} \ell_t(\mathbf{x}) - \ell_{t-1}(\mathbf{x})$ .

In addition, with  $\beta^2 = D^2$  we get

$$R_T(\mathbf{u}_{1:T}) \leq \left( 2 + \frac{\gamma C_T}{D^2} \right) \min \left\{ \ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + V_T, \sqrt{3D^2 \sum_{t=1}^T \|\mathbf{g}_t\|_*^2} \right\}. \quad (12)$$

If we assume an upper bound  $\|\mathbf{g}_t\|_*^2 \leq \max_{t \in [T]} \|\mathbf{g}_t\|_*^2 \leq 1$ , and that  $\gamma = D = 1$ , then the result in Eq. (11) gives us a dynamic regret bound of  $\mathcal{O}(\min\{V_T, \sqrt{T(1+\tau)}\})$ . This bound is tight for sequences whose path-length  $C_T = \tau$ , matching the lower bounds for both the path-length and temporal variability. Moreover, we show in Appendix A how the algorithm can be adapted using a doubling trick in the same spirit of Jadbabaie et al. [2015], when  $\tau$  is not fixed in advance but the path-length can be observed on the fly.

**Running Time** Compared to standard *Mirror Descent* the update depicted in Eq. (8) does not have a general solution, but it has to be calculated case-by-case depending on the loss function. There are some cases relevant in practical applications where this update is available in closed form, such as regression with square or absolute loss or classification with hinge loss (see for example Crammer et al. [2006]) and the time complexity of the update is similar to OMD. When a closed-form solution is not available, it can be approximated efficiently using numerical methods [Song et al., 2018, Shtof, 2020].

## 5.1 Applications

Next, we are going to show some applications of this algorithm. We point out that the same set of applications and the related regret bounds continue to hold for the algorithm presented later in Theorem 6.2, with some differences that will be highlighted in Section 6.

**Learning with Expert Advice** In this setting the loss is linear, i.e.,  $\ell_t(\mathbf{x}) = \langle \mathbf{g}_t, \mathbf{x} \rangle$ , therefore the implicit and the standard version of *Mirror Descent* coincide. It is known that *Mirror Descent* with negative entropy regularization, i.e.,  $\psi(\mathbf{x}) = \sum_{i=1}^d x_i \ln x_i$ , yields the exponential weights algorithm which is optimal. We recall that the Bregman divergence induced by the negative entropy is the KL divergence, for any 2 points on the simplex  $\Delta_d$ , which can be potentially unbounded. Indeed, using a dynamic learning rate with *Mirror Descent* in this setting gives rise to a vacuous bound [Orabona and Pál, 2018, Theorem 4]. For this reason, we modify the domain of interest: instead of the regular simplex, we use a “clipped” version of it.

$$\Delta_d^\alpha \triangleq \{ \mathbf{x} \in \mathbb{R}_+^d : \|\mathbf{x}\|_1 = 1, x_i \geq \frac{\alpha}{d} \forall i = 1, \dots, d \}. \quad (13)$$

This set makes the diameter w.r.t to KL divergence bounded. Indeed, we have that

$$\max_{\mathbf{x}, \mathbf{y} \in \Delta_d^\alpha} \text{KL}(\mathbf{x}, \mathbf{y}) \leq \sum_{i=1}^d x_i \ln \frac{d}{\alpha} \leq \ln \frac{d}{\alpha}. \quad (14)$$

Furthermore, the update in Eq. (8) using this “clipped” simplex can be computed efficiently, see Herbster and Warmuth [2001, Theorem 7] for details.

We prove a regret bound for Algorithm 2 in the next theorem. We stress that the proof the next result does not follow directly from an application of Theorem 5.1 (it is reported in Appendix B).

**Theorem 5.2.** *Consider the setting of Learning with Expert Advice on the  $d$ -dimensional simplex  $\Delta_d$ . Assume  $0 \leq g_{t,i} \leq L_\infty$  for all  $t = 1, \dots, T$ , and  $i = 1, \dots, d$ . Assume that  $T \geq d$  and set  $\alpha = d/T$ . Furthermore, set  $\eta_{t+1}^{-1} = \lambda_{t+1} = \frac{1}{\beta^2} \sum_{i=1}^t \delta_t$ , with  $\delta_t$  defined as in lemma 5.1. Then, for any sequence of comparators  $\mathbf{u}_{1:T}$  with  $\mathbf{u}_t \in \Delta_d$  such that  $C_T(\mathbf{u}_{1:T}) \leq \tau$ , using  $\beta^2 = (1 + \tau) \ln T$  the regret of Algorithm 2 run on  $\Delta_d^\alpha$  is bounded as*

$$R_T(\mathbf{u}_{1:T}) \leq 2 \min \left\{ \ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + V_T, \right. \\ \left. \sqrt{(1 + (1 + \tau) \ln T) \sum_{t=1}^T \mathbb{E}_t[\mathbf{g}_t^2]} \right\} + 2L_\infty d, \quad (15)$$

where  $\mathbb{E}_t[\mathbf{g}_t^2] = \sum_{i=1}^T x_{t,i} g_{t,i}^2$  and  $V_T = \sum_{t=2}^T \max_{\mathbf{x} \in \Delta_d} \ell_t(\mathbf{x}) - \ell_{t-1}(\mathbf{x})$ .

**Discussion.** Note that  $\mathbb{E}_t[\mathbf{g}_t^2] = \sum_{i=1}^d x_{t,i} g_{t,i}^2 \leq \|\mathbf{g}_t\|_\infty^2$ . Furthermore, if the sequence  $\mathbf{u}_{1:T}$  is only composed by “corners” of the simplex, then  $C_T(\mathbf{u}_{1:T})$  is given

roughly by the number of “shifts”, i.e.,  $C_T(\mathbf{u}_{1:T}) = 2 \sum_{t=2}^T \mathbb{1}\{\mathbf{u}_t \neq \mathbf{u}_{t-1}\}$ . In this setting, we know that the *Fixed-Share* algorithm [Herbster and Warmuth, 1998] achieves a regret bound of  $\mathcal{O}(\sqrt{TS \ln \frac{dT}{S}})$ , where  $C_T(\mathbf{u}_{1:T}) = \mathcal{O}(S)$  is the number of shifts and is fixed in advance. Our algorithm achieves a similar bound, but it can be sometimes better. For example, assume  $\ell_t(\mathbf{x}) \in [0, 1]$  and that  $\ell_t$  stays fixed for all except  $S$  rounds, then we have that  $V_T = S$ . In this case, *Fixed-Share* regret is  $\mathcal{O}(\sqrt{TS})$ , while the bound in Eq. (15) reduces to  $\mathcal{O}(S)$ . It should be noted that both algorithms assume oracle knowledge of  $S$ . In order to remove this assumption, we refer to a different algorithm described in Section 6. Moreover, it can be shown (see Appendix B) that the regret bound in Eq. (15) can lead to a first-order regret bound which depends on the loss of the sequence of competitors. To the best of our knowledge a similar result is not *easily* obtainable for the *Fixed-Share* algorithm (see Cesa-Bianchi et al. [2012, Section 7.3] for a comparison).

**Composite Losses** In this paragraph, we assume that the losses received are composed by two parts: one convex part changing over time and the other one fixed and known to the algorithm. These losses are called *composite* [Duchi et al., 2010]. This setting was also studied in the implicit case in Song et al. [2018] for the static regret. For example, we might have  $\ell_t(\mathbf{x}) = \tilde{\ell}_t(\mathbf{x}) + \beta \|\mathbf{x}\|_1$ . In this case, considering a bounded domain  $\mathcal{V}$  the update rule for Algorithm 2 will be

$$\mathbf{x}_{t+1} = \underset{\mathbf{x} \in \mathcal{V}}{\operatorname{argmin}} \tilde{\ell}_t(\mathbf{x}) + \beta \|\mathbf{x}\|_1 + B_\psi(\mathbf{x}, \mathbf{x}_t), \quad (16)$$

which will promote sparsity in our model. We show in Appendix C that the algorithm *AdaImplicit* [Campolongo and Orabona, 2020] designed for the static regret already satisfies a regret bound order of  $\mathcal{O}(\min(V_T, \sqrt{T}))$  (improving over the existing result of Song et al. [2018]). Next, we extend the analysis to the dynamic scenario. In particular, using Algorithm 2 we can give the following theorem, whose proof is reported in Appendix C.

**Theorem 5.3.** *Let  $\mathcal{V} \subset X \subseteq \mathbb{R}^d$  be a non-empty closed convex set. Let  $\ell_t(\mathbf{x}) = \tilde{\ell}_t(\mathbf{x}) + r(\mathbf{x})$ , where  $r : X \rightarrow \mathbb{R}$  is a convex function. Then, under the assumptions of lemma 5.1 the regret of Algorithm 2 run with  $1/\eta_t = \lambda_t = \frac{1}{\beta^2} \sum_{i=1}^{t-1} \delta_i$  and  $\beta^2 = D^2 + \gamma\tau$  against any sequence of comparators  $\mathbf{u}_{1:T}$  whose path-length  $C_T$  is less or equal than  $\tau$  is bounded as*

$$R_T(\mathbf{u}_{1:T}) \leq \min \left\{ 2(\ell_1(\mathbf{x}_1) - \ell_{T+1}(\mathbf{x}_{T+1}) + V_T), \right. \\ \left. 2\sqrt{(3D^2 + \gamma\tau) \sum_{t=1}^T \|\mathbf{g}_t\|_*^2} \right\},$$

where  $V_T = \sum_{t=2}^T \max_{\mathbf{x} \in \mathcal{V}} \tilde{\ell}_t(\mathbf{x}) - \tilde{\ell}_{t-1}(\mathbf{x})$  and  $\mathbf{g}_t \in \partial \ell_t(\mathbf{x}_t)$ .

**Remark.** Comparing to the regret bound given in Theorem 5.1, we can see that the result above contains a subtle

---

### Algorithm 3 Anytime $(\mathcal{A}, \mathcal{B})$ -PROD

---

**Require:** Algorithms  $\mathcal{A}, \mathcal{B}$ ,  $\eta_1 = w_{1,\mathcal{A}} = w_{1,\mathcal{B}} = 1/2$

```

1: for  $t = 1, \dots, T$  do
2:   Let  $p_{t,\mathcal{A}} = \frac{\eta_t w_{t,\mathcal{A}}}{\eta_t w_{t,\mathcal{A}} + w_{t,\mathcal{B}}/2}$ ,  $p_{t,\mathcal{B}} = 1 - p_{t,\mathcal{A}}$ 
3:   Get  $\mathbf{a}_t$  from  $\mathcal{A}$  and  $\mathbf{b}_t$  from  $\mathcal{B}$ 
4:   Set  $\mathbf{x}_t = p_{t,\mathcal{A}} \mathbf{a}_t + p_{t,\mathcal{B}} \mathbf{b}_t$ 
5:   Receive  $\ell_t : \mathbb{R}^d \rightarrow [0, 1]$  and pay  $\ell_t(\mathbf{x}_t)$ 
6:   Feed  $\ell_t$  to  $\mathcal{A}$  and  $\mathcal{B}$ 
7:   Set  $r_t = \ell_t(\mathbf{b}_t) - \ell_t(\mathbf{a}_t)$ 
8:   Set  $\eta_{t+1} = \sqrt{(1 + \sum_{i=1}^t (\ell_i(\mathbf{b}_i) - \ell_i(\mathbf{a}_i))^2)^{-1}}$ 
9:   Set  $w_{t+1,\mathcal{A}} = w_{t,\mathcal{A}}(1 + \eta_t r_t)^{\eta_{t+1}/\eta_t}$ 
10: end for
```

---

difference: the temporal variability is given only in terms of the variable part of the losses,  $\ell_t$ .

All the results up to this point are given under the assumption that the class of strategies we want to compete against is fixed before the start of the game, i.e., an upper bound to  $C_T$  is fixed beforehand. This can be limiting in practice: in a truly realistic online setting, knowing the right upper bound beforehand might be hard. Therefore, in the next section we are going to provide an algorithm which adapts to the values of  $C_T$  for any possible sequence of comparators, but at the same time guarantees a bound in  $V_T$ .

## 6 Adapting to different path-lengths

In this section, we present an approach to obtain the optimal bound of  $\mathcal{O}(\min\{V_T, \sqrt{T(1+C_T)}\})$  on the dynamic regret for all the sequences of comparators simultaneously. Our approach is based on smartly combining different algorithms.

Using existing algorithms [see, e.g., Cutkosky, 2020, Zhang et al., 2018a] we can achieve the optimal dynamic regret bound for all possible sequences of comparators. In particular, a recent result from Cutkosky [2020] shows the condition that strongly-adaptive algorithms need to satisfy in order to incur the optimal bound of  $\mathcal{O}(\sqrt{T(1+C_T)})$ , which we report in the next theorem (proof in Appendix D).

**Theorem 6.1** (Adapted from Cutkosky [2020]). *Let  $\mathcal{V} \subset X \subset \mathbb{R}^d$  be non-empty closed convex sets,  $\psi : X \rightarrow \mathbb{R}$ . Define  $D^2 = \max_{\mathbf{x}, \mathbf{y} \in \mathcal{V}} B_\psi(\mathbf{x}, \mathbf{y})$ . Given a sequence of loss functions  $\ell_1, \dots, \ell_T$ , assume there exists an algorithm that for any interval  $I = [s, e] \subseteq [1, T]$  and sequence  $\mathbf{u}_{s:e}$  guarantees a dynamic regret bounded as*

$$R_I(\mathbf{u}_{s:e}) = \mathcal{O} \left( (D + C_I) \sqrt{|I|} \right), \quad (17)$$

where  $|I| = (e - s)$ . Then, for any interval  $J = [s', e'] \subseteq [1, T]$  it also guarantees

$$R_J(\mathbf{u}_{s':e'}) \leq \mathcal{O} \left( \sqrt{|J|D(C_J + D)} \right), \quad (18)$$

where  $C_J = \sum_{t=s'+1}^{e'} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|$ .

On the other hand, in order to achieve a bound on the temporal variability, we can simply adopt the greedy strategy given in the Algorithm 1, which in every step plays the minimizer of the last seen loss function. Hence, we need to combine these two strategies.

A plain expert algorithm would fail to give a constant bound w.r.t. the temporal variability. Instead, we use a modification of the *ML-Prod* algorithm [Gaillard et al., 2014] proposed in Sani et al. [2014] and depicted in Algorithm 3. This algorithm takes as input two base learners (aka experts)  $\mathcal{A}$  and  $\mathcal{B}$  and guarantees a regret which is (almost) constant against  $\mathcal{B}$  and  $\mathcal{O}(\sqrt{T} \ln \ln T)$  against  $\mathcal{A}$  in the worst case. The idea is to use a strongly adaptive algorithm which satisfies the assumptions of Theorem 6.1 such as the one from Cutkosky [2020] as algorithm  $\mathcal{A}$ , and the greedy strategy in Algorithm 1 as  $\mathcal{B}$ . In the next theorem, we provide an upper bound to the dynamic regret of the resulting algorithm (proof in Appendix D).

**Theorem 6.2.** *Let  $\mathcal{V} \subset X \subset \mathbb{R}^d$  be non-empty closed convex sets,  $\psi : X \rightarrow \mathbb{R}$ . Define  $D^2 = \max_{\mathbf{x}, \mathbf{y} \in \mathcal{V}} B_\psi(\mathbf{x}, \mathbf{y})$ . Let  $\ell_1, \dots, \ell_t$  be a sequence of convex loss functions such that  $\ell_t : \mathbb{R}^d \rightarrow [0, 1]$  for all  $t$ . Then, for all sequence of  $\mathbf{u}_{1:T}$  with path-length  $C_T$ , running Algorithm 3 with  $\mathcal{A}$  as a strongly-adaptive algorithm satisfying the condition from Theorem 6.1, and  $\mathcal{B}$  as Algorithm 1, guarantees*

$$R_T(\mathbf{u}_{1:T}) \leq \mathcal{O} \left( \min \left\{ V_T, \sqrt{C} + \sqrt{TD(C_T + D)} \right\} \right),$$

where  $C$  is an upper bound to the loss of algorithm  $\mathcal{A}$ .

**Discussion.** The algorithm from last theorem is able to guarantee the optimal dependence on  $C_T$  for any sequence of comparators  $\mathbf{u}_{1:T}$ , without requiring any prior knowledge. However, compared to the algorithms from Section 5, Theorem 6.2 requires the losses to be bounded in a range known to the algorithm. Furthermore, note that all known strongly-adaptive algorithms require a running time of  $\mathcal{O}(d \ln T)$  per-update, which is higher than  $\mathcal{O}(d)$  required by OMD (and its Implicit version when the update is available in closed form). Finally, we point out that the actual regret bound depends on the specific choice of the algorithm  $\mathcal{A}$  used. Next, we sketch some applications in the same spirit of Section 5.1.

**Applications.** To make a comparison with the setting of Learning with Expert Advice previously covered, consider again the “shifting” scenario with  $S$  shifts of Section 5.1. In this case we can adopt a strategy similar to the strongly-adaptive algorithm *CBCE* from Jun et al. [2017]. It is known that a strongly-adaptive algorithm incurs a dynamic regret bounded by  $\tilde{\mathcal{O}}(\sqrt{TS})$ , without requiring the knowledge of  $S$  in advance [see, e.g., Appendix A in Jun et al., 2017], contrarily to Algorithm 2. Moreover, from our bound in Theorem 6.2 if the loss functions stay fixed for all but  $S$  rounds, thanks to the guarantee of Algorithm 1 the regret bound is  $\mathcal{O}(V_T) = \mathcal{O}(S)$ , which again

is like the guarantee of Algorithm 2 with the important difference that knowing in advance the number of shifts is not required.

On the other hand, when considering for example the setting of composite losses, or more in general Euclidean domains and the  $L_2$  norm, we can adopt the algorithm from Cutkosky [2020], which is adaptive to the sum of the gradients  $\sum_{t=1}^T \|\mathbf{g}_t\|_2^2$  and therefore can have a potentially better bound compared to *CBCE*, which has a worst-case regret bound of  $\mathcal{O}(\sqrt{T})$ .

## 7 Conclusion

In this work, we have shown that existing bounds in the dynamic setting with full information feedback can be improved, by establishing a lower bounds on the dynamic regret in terms of temporal variability of the loss functions and showing algorithms with matching upper bounds. In particular, we designed an algorithm using implicit updates that can adapt to both the temporal variability and the path-length of the sequence of comparators. Furthermore, when the desired path-length is not fixed in advance, we showed how to combine existing algorithms in order to achieve the optimal bound.

An interesting question remains open: is it possible to obtain a dynamic regret bound of  $\mathcal{O}(\min\{V_T, \sqrt{T(1+C_T)}\})$  for all sequence of comparators with a single algorithm? If so, what is its running time? As observed in previous work, all strongly-adaptive algorithms [Cutkosky, 2020, Jun et al., 2017] have a running time of  $\mathcal{O}(T \ln T)$  and it is currently not known whether it can be improved. Future research directions therefore could aim at designing faster and more practical algorithms which can adapt to unknown path-lengths, or in alternative prove that this goal cannot be achieved.

## References

- Amirhossein Ajalloeian, Andrea Simonetto, and Emiliano Dall’Anese. Inexact online proximal-gradient method for time-varying convex optimization. In *2020 American Control Conference (ACC)*, pages 2850–2857. IEEE, 2020.
- Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations research*, 63(5):1227–1244, 2015.
- Nicolò Campolongo and Francesco Orabona. Temporal variability in implicit online learning. *arXiv preprint arXiv:2006.07503*, 2020.
- N Cesa-Bianchi, P Gaillard, G Lugosi, and G Stoltz. Mirror descent meets fixed share (and feels no regret). In *Conference on Neural Information Processing Systems*, volume 2, pages 989–997. Neural information processing systems foundation, 2012.
- Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning



- algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.
- Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7 (Mar):551–585, 2006.
- Ashok Cutkosky. Parameter-free, dynamic, and strongly-adaptive online learning. In *International Conference on Machine Learning*, volume 2, 2020.
- Amit Daniely, Alon Gonen, and Shai Shalev-Shwartz. Strongly adaptive online learning. In *International Conference on Machine Learning*, pages 1405–1411, 2015.
- Rishabh Dixit, Amrit Singh Bedi, Ruchi Tripathi, and Ketan Rajawat. Online learning with inexact proximal online gradient descent algorithms. *IEEE Transactions on Signal Processing*, 67(5):1338–1352, 2019.
- John C Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. In *COLT*, pages 14–26. Citeseer, 2010.
- Pierre Gaillard, Gilles Stoltz, and Tim Van Erven. A second-order bound with excess losses. In *Conference on Learning Theory*, pages 176–196, 2014.
- Eric Hall and Rebecca Willett. Dynamical models and tracking regret in online convex programming. In *International Conference on Machine Learning*, pages 579–587, 2013.
- Elad Hazan and Comandur Seshadhri. Adaptive algorithms for online decision problems. In *Electronic colloquium on computational complexity (ECCC)*, volume 14, 2007.
- Mark Herbster and Manfred K Warmuth. Tracking the best expert. *Machine learning*, 32(2):151–178, 1998.
- Mark Herbster and Manfred K Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1(281-309):10–1162, 2001.
- Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization: Competing with dynamic comparators. In *Artificial Intelligence and Statistics*, pages 398–406, 2015.
- Kwang-Sung Jun, Francesco Orabona, Stephen Wright, and Rebecca Willett. Improved strongly adaptive online learning using coin betting. In *Artificial Intelligence and Statistics*, pages 943–951. PMLR, 2017.
- Jyrki Kivinen and Manfred K Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *information and computation*, 132(1):1–63, 1997.
- Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.
- Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.
- F. Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.
- Francesco Orabona and Dávid Pál. Scale-free online learning. *Theor. Comput. Sci.*, 716:50–69, 2018. doi: 10.1016/j.tcs.2017.11.021. URL <https://doi.org/10.1016/j.tcs.2017.11.021>.
- Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.
- Amir Sani, Gergely Neu, and Alessandro Lazaric. Exploiting easy data in online optimization. In *Advances in Neural Information Processing Systems*, pages 810–818, 2014.
- Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.
- Alex Shtof. Proximal point - convex on linear losses, 2020. URL <https://alexshstf.github.io/2020/02/15/ProximalConvexOnLinear.html>.
- Chaobing Song, Ji Liu, Han Liu, Yong Jiang, and Tong Zhang. Fully implicit online learning. *arXiv preprint arXiv:1809.09350*, 2018.
- Tianbao Yang, Lijun Zhang, Rong Jin, and Jinfeng Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *International Conference on Machine Learning*, pages 449–457, 2016.
- Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive online learning in dynamic environments. In *Advances in neural information processing systems*, pages 1323–1333, 2018a.
- Lijun Zhang, Tianbao Yang, Rong Jin, and Zhi-Hua Zhou. Dynamic regret of strongly adaptive methods. In *International Conference on Machine Learning*, pages 5882–5891, 2018b.
- Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.

## A Dynamic IOMD

In this section we prove a regret bound for Algorithm 2 and its variant using a doubling trick mentioned in Section 5.

The following lemma is a generalization of Orabona [2019, Lemma 7.12] which we will use in subsequent proofs.

**Lemma A.1.** *Let  $\{a_t\}_{t=1}^\infty, \{b_t\}_{t=1}^\infty$  be two sequence of non-negative real numbers. Suppose that  $\{\Delta_t\}_{t=1}^\infty$  is a sequence of non-negative real numbers satisfying  $\Delta_1 = 0$  and<sup>4</sup>  $\Delta_{t+1} \leq \Delta_t + \min\{db_t, ca_t^2/(2\Delta_t)\}$ , for any  $t \geq 1$ .*

*Then, for any  $T \geq 0$ ,  $\Delta_{T+1} \leq \sqrt{d^2 \sum_{t=1}^T b_t^2 + c \sum_{t=1}^T a_t^2}$ .*

*Proof.* Observe that

$$\Delta_{T+1}^2 = \sum_{t=1}^T \Delta_{t+1}^2 - \Delta_t^2 = \sum_{t=1}^T \underbrace{(\Delta_{t+1} - \Delta_t)^2}_{(a)} + \sum_{t=1}^T \underbrace{2(\Delta_{t+1} - \Delta_t)\Delta_t}_{(b)}.$$

We bound the sequences (a) and (b) separately. For (a), from the assumption on the recurrence and using the first term in the minimum we have that  $(\Delta_{t+1} - \Delta_t)^2 \leq d^2 b_t^2$ . On the other hand, for (b) using the second term in the minimum in the recurrence we get  $2(\Delta_{t+1} - \Delta_t)\Delta_t \leq ca_t^2$ . Putting together the results we have that  $\Delta_{T+1}^2 \leq d^2 \sum_{t=1}^T b_t^2 + c \sum_{t=1}^T a_t^2$  and the lemma follows.  $\square$

We are now ready to show a regret bound for Algorithm 2, as stated in Theorem 5.1. The statement of the theorem is reported next for completeness.

**Theorem 5.1.** *Let  $\tau \geq 0$  be a positive constant. Under the assumptions of lemma 5.1, for any sequence  $\mathbf{u}_{1:T}$  whose path length  $C_T(\mathbf{u}_{1:T})$  is less or equal than  $\tau$ , Algorithm 2 with  $1/\eta_t = \lambda_t = \frac{1}{\beta^2} \sum_{i=1}^{t-1} \delta_i$ , and  $\beta^2 = (D^2 + \gamma\tau)$  incurs dynamic regret against upper bounded as follows*

$$R_T(\mathbf{u}_{1:T}) \leq \min \left\{ 2(\ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + V_T), \right. \\ \left. 2\sqrt{(3D^2 + \gamma\tau) \sum_{t=1}^T \|\mathbf{g}_t\|_*^2} \right\}, \quad (11)$$

where  $V_T = \sum_{t=2}^T \max_{\mathbf{x} \in \mathcal{V}} \ell_t(\mathbf{x}) - \ell_{t-1}(\mathbf{x})$ .

In addition, with  $\beta^2 = D^2$  we get

$$R_T(\mathbf{u}_{1:T}) \leq \left( 2 + \frac{\gamma C_T}{D^2} \right) \min \left\{ \ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + \right. \\ \left. V_T, \sqrt{3D^2 \sum_{t=1}^T \|\mathbf{g}_t\|_*^2} \right\}. \quad (12)$$

*Proof.* First, note that  $(\lambda_t)_{t=1}^T$  is an increasing sequence, since  $\delta_t \geq 0$ . Indeed, from the optimality of the update rule of Algorithm 2 we have

$$\ell_t(\mathbf{x}_{t+1}) + \lambda_t B_\psi(\mathbf{x}_{t+1}, \mathbf{x}_t) \leq \ell_t(\mathbf{x}_t) + \lambda_t B_\psi(\mathbf{x}_t, \mathbf{x}_t) = \ell_t(\mathbf{x}_t),$$

which implies  $\delta_t := \ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}_{t+1}) - \lambda_t B_\psi(\mathbf{x}_{t+1}, \mathbf{x}_t) \geq 0$ . Hence, by using the prescribed learning rate  $\lambda_t$ , we can rewrite the bound in Eq. (9) as follows

$$R_T(\mathbf{u}_{1:T}) \leq \lambda_T D^2 + \gamma \sum_{t=2}^T \lambda_t \|\mathbf{u}_t - \mathbf{u}_{t-1}\| + \beta^2 \lambda_{T+1} \\ \leq (D^2 + \beta^2) \lambda_{T+1} + \gamma \lambda_{T+1} \sum_{t=2}^T \|\mathbf{u}_t - \mathbf{u}_{t-1}\| \\ \leq (D^2 + \beta^2 + \gamma\tau) \lambda_{T+1},$$

where in the second inequality we have used the fact that  $(\lambda_t)_{t=1}^T$  is an increasing sequence.

---

<sup>4</sup>With a small abuse of notation, let  $\min(x, y/0) = x$ .

The rest of the proof is similar to the one in Campolongo and Orabona [2020, Theorem 6.2]. From the choice of  $\lambda_t$ , we have that

$$\begin{aligned}
\beta^2 \lambda_{T+1} &= \sum_{t=1}^T \delta_t = \sum_{t=1}^T \ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}_{t+1}) - \lambda_t B_\psi(\mathbf{x}_{t+1}, \mathbf{x}_t) \\
&\leq \sum_{t=1}^T \ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}_{t+1}) \\
&\leq \ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + \sum_{t=2}^T \max_{\mathbf{x} \in \mathcal{V}} \ell_t(\mathbf{x}) - \ell_{t-1}(\mathbf{x}) \\
&= \ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + V_T,
\end{aligned} \tag{19}$$

where the first inequality derives from the fact that Bregman divergences are always positive.

On the other hand, from the definition of  $\delta_t$  we have that

$$\delta_t \leq \ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}_{t+1}) \leq \langle \mathbf{g}_t, \mathbf{x}_t - \mathbf{x}_{t+1} \rangle \leq \|\mathbf{g}_t\|_* \|\mathbf{x}_t - \mathbf{x}_{t+1}\|. \tag{20}$$

Now, note that from the assumptions in lemma 5.1 we have that for any  $\mathbf{x}, \mathbf{y} \in \mathcal{V}$

$$D^2 \geq B_\psi(\mathbf{x}, \mathbf{y}) \geq \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2.$$

Therefore,  $\|\mathbf{x} - \mathbf{y}\| \leq \sqrt{2}D$  and substituting back in Eq. (20)

$$\delta_t \leq \sqrt{2}D \|\mathbf{g}_t\|_*.$$

On the other hand, by not discarding the negative Bregman divergence term in Eq. (20) we get

$$\delta_t \leq \|\mathbf{g}_t\|_* \sqrt{2B_\psi(\mathbf{x}_{t+1}, \mathbf{x}_t) - \lambda_t B_\psi(\mathbf{x}_{t+1}, \mathbf{x}_t)} \leq \frac{\|\mathbf{g}_t\|_*^2}{2\lambda_t},$$

where the last step derives from the fact that  $bx - \frac{a}{2}x^2 \leq \frac{b^2}{2a}, \forall x \in \mathbb{R}$ , with  $x = \sqrt{B_\psi(\mathbf{x}_{t+1}, \mathbf{x}_t)}$ .

To summarize, we have that

$$\lambda_{t+1} = \lambda_t + \delta_t \leq \lambda_t + \frac{1}{\beta^2} \min \left\{ \sqrt{2}D \|\mathbf{g}_t\|_*, \frac{\|\mathbf{g}_t\|_*^2}{2\lambda_t} \right\}.$$

Now, applying lemma A.1 with  $\Delta_t = \lambda_t, a_t = b_t = \|\mathbf{g}_t\|_*, d = \frac{\sqrt{2}D}{\beta^2}, c = \frac{1}{\beta^2}$  yields

$$\lambda_{T+1} \leq \sqrt{\left( \frac{2D^2}{\beta^4} + \frac{1}{\beta^2} \right) \sum_{t=1}^T \|\mathbf{g}_t\|_*^2}. \tag{21}$$

Therefore, putting together Eqs. (19) and (21) and using the suggested values for  $\beta$ , we get the stated results.  $\square$

### A.1 Adapting on the fly

The result given in the previous paragraph was limited to all sequences of comparators whose path-length is fixed beforehand. Following Jadbabaie et al. [2015], the approach given above can be generalized to any sequence of  $\mathbf{u}_{1:T}$  whose path length  $C_T$  can be calculated on the fly.

**Doubling trick.** The idea is to run Algorithm 2 in phases and tune the learning rate  $\lambda_t$  appropriately. At the beginning of each phase  $i$ , we start monitoring the path length  $C_i$ . Once it reaches a certain threshold, we restart the algorithm doubling the threshold. Formally, we introduce a quantity  $Q_i$  for phase  $i$  and set the learning rate  $\lambda_t$  of the algorithm as  $\lambda_t^i = \frac{1}{\beta_i^2} \sum_{s=1}^{t-1} \delta_s$ , with  $\beta_i^2 = D^2 + \gamma Q_i$ . The resulting algorithm is shown in Algorithm 4.

We are now going to analyze the regret bound incurred by Algorithm 4. First, we need the following lemma which bounds the number of times the algorithm is restarted.

**Algorithm 4** Dynamic IOMD with Doubling Trick

---

**Require:** Non-empty closed convex set  $\mathcal{V} \subset X \subset \mathbb{R}^d$ ,  $\psi : X \rightarrow \mathbb{R}$ ,  $\mathbf{x}_1 \in \mathcal{V}$ ,  $\gamma$  such that  $B_\psi(\mathbf{x}, \mathbf{z}) - B_\psi(\mathbf{y}, \mathbf{z}) \leq \gamma \|\mathbf{x} - \mathbf{y}\|$ ,  $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{V}$ ,  $\beta_0^2 > 0$ , observable sequence  $\mathbf{u}_{1:T}$

- 1:  $i \leftarrow 0, \lambda_1^0 \leftarrow 0, Q_0 \leftarrow \sqrt{2}D, C_0 \leftarrow 0$
- 2: **for**  $t = 1, \dots, T$  **do**
- 3:   Output  $\mathbf{x}_t \in \mathcal{V}$
- 4:   Receive  $\ell_t : \mathbb{R}^d \rightarrow \mathbb{R}$  and pay  $\ell_t(\mathbf{x}_t)$
- 5:   Update  $C_i \leftarrow C_i + \|\mathbf{u}_t - \mathbf{u}_{t-1}\|$
- 6:   **if**  $C_i > Q_i$  **then**
- 7:      $i \leftarrow i + 1$
- 8:      $Q_i \leftarrow \sqrt{2}D2^i, \lambda_{t+1}^i \leftarrow 0, C_i \leftarrow 0, \beta_i^2 \leftarrow D^2 + \gamma Q_i$
- 9:     Update  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t$
- 10:   **else**
- 11:     Update  $\mathbf{x}_{t+1} \leftarrow \arg \min_{\mathbf{x} \in \mathcal{V}} \ell_t(\mathbf{x}) + \lambda_t^i B_\psi(\mathbf{x}, \mathbf{x}_t)$
- 12:     Set  $\delta_t \leftarrow \ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}_{t+1}) - \lambda_t^i B_\psi(\mathbf{x}_{t+1}, \mathbf{x}_t)$
- 13:     Update  $\lambda_{t+1}^i \leftarrow \lambda_t^i + \frac{1}{\beta_i^2} \delta_t$
- 14:   **end if**
- 15: **end for**

---

**Lemma A.2.** Let  $t_i$  be the first time-step of epoch  $i$ , with  $t_0 = 1$ . Suppose Algorithm 4 is run for a total of  $N + 1$  epochs. Let  $C_i = \sum_{t=t_i}^{t_{i+1}-1} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|$ , with  $\|\mathbf{u}_1 - \mathbf{u}_0\| \triangleq 0$ . Let  $C_T = \sum_{i=0}^N C_i$ . Then, we have that  $N$  satisfies

$$N \leq \log_2 \left( \frac{C_T}{\sqrt{2}D} + 1 \right). \quad (22)$$

*Proof.* First, recall that  $\sum_{i=0}^{N-1} a^i = \frac{a^N - 1}{a - 1}$ . Now, note that the sum in the first  $N$  epochs of the quantity we are monitoring is at most equal to the final sum over all  $N + 1$  epochs.

Therefore, we have the following

$$\sum_{i=0}^{N-1} \sqrt{2}D2^i \leq \sqrt{2}D(2^N - 1) \leq \sum_{i=0}^N \sum_{t=t_i}^{t_{i+1}-1} \|\mathbf{u}_t - \mathbf{u}_{t-1}\| = C_T,$$

where  $\|\mathbf{u}_{t_0} - \mathbf{u}_{t_0-1}\| = \|\mathbf{u}_1 - \mathbf{u}_0\| \triangleq 0$  by definition. Solving for  $N$  yields the desired result.  $\square$

Next, we provide a theorem which gives a regret bound to Algorithm 4.

**Theorem A.1.** Let  $\mathcal{V} \subset X \subset \mathbb{R}^d$  be a non-empty closed convex set. Assume Algorithm 4 is run for  $N$  epochs. Then, under the assumptions of Lemma 5.1 the regret against any sequence of comparators  $\mathbf{u}_{1:T}$  with  $\mathbf{u}_t \in \mathcal{V}$  is bounded as

$$R_T(\mathbf{u}_{1:T}) \leq (2 + c) \min \left( (\ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + V_T), \sqrt{\left( 3D^2 \left( \log_2 \frac{C_T}{\sqrt{2}D} + 1 \right) + \gamma C_T \right) \sum_{t=1}^T \|\mathbf{g}_t\|_*^2} \right), \quad (23)$$

where  $c \triangleq \frac{\sqrt{2}}{D + \gamma\sqrt{2}}$  and  $C_T = \sum_{t=2}^T \|\mathbf{u}_t - \mathbf{u}_{t-1}\|$ .

*Proof.* Let  $V_i = \sum_{t=t_i+1}^{t_{i+1}-1} \max_{\mathbf{x} \in \mathcal{V}} |\ell_t(\mathbf{x}) - \ell_{t-1}(\mathbf{x})|$ . Using the result from Theorem 5.1, assuming the knowledge of  $C_i$  during each phase  $i$  we have that

$$R(\mathbf{u}_{1:T}) = \sum_{i=0}^N \sum_{t=t_i}^{t_{i+1}-1} R(\mathbf{u}_{t_i:t_{i+1}-1}) \leq \sum_{i=0}^N \sum_{t=t_i}^{t_{i+1}-1} \frac{D^2 + \gamma C_i + \beta_i^2}{\beta_i^2} \min(B_1^i, B_2^i),$$

where in the last inequality we used Theorem 5.1 with  $B_1^i = \ell_{t_i}(\mathbf{x}_1) - \ell_{t_{i+1}-1}(\mathbf{x}_{t_{i+1}}) + V_i$  and  $B_2 = \sqrt{(2D^2 + \beta^2) \sum_{t=t_i}^{t_{i+1}-1} \|\mathbf{g}_t\|_*^2}$ .

Note that with the adopted  $\beta_i$  from Algorithm 4 we have that

$$\frac{D^2 + \gamma C_i + \beta_i^2}{\beta_i^2} = \frac{D^2 + \gamma C_i + D^2 + \gamma Q_i}{D^2 + \gamma Q_i} = 2 + \gamma \frac{(C_i - Q_i)}{D^2 + \gamma Q_i} \leq 2 + \gamma \frac{\sqrt{2}D}{D^2 + \gamma\sqrt{2}D2^i},$$

where the last inequality derives from the fact that the last term in  $C_i$  which causes the algorithm to restart is such that  $\|\mathbf{x} - \mathbf{y}\| \leq \sqrt{2}D$ ,  $\forall \mathbf{x}, \mathbf{y} \in \mathcal{V}$ .

Therefore, we have

$$\begin{aligned}
R(\mathbf{u}_{1:T}) &\leq \sum_{i=0}^N \sum_{t=t_i}^{t_{i+1}-1} \left( 2 + \gamma \frac{\sqrt{2}}{D + \gamma 2^{i+\frac{1}{2}}} \right) \min(B_1^i, B_2^i) \\
&\leq \sum_{i=0}^N (2+c) \min \left\{ \ell_{t_i}(\mathbf{x}_{t_i}) - \ell_{t_{i+1}-1}(\mathbf{x}_{t_{i+1}}) + V_i, \sqrt{(3D^2 + \gamma\sqrt{2}D2^i) \sum_{t=t_i}^{t_{i+1}-1} \|\mathbf{g}_t\|_\star^2} \right\} \\
&\leq (2+c) \sum_{i=0}^N \min \left\{ \ell_{t_i}(\mathbf{x}_{t_i}) - \ell_{t_{i+1}-1}(\mathbf{x}_{t_{i+1}}) + V_i, \sqrt{(3D^2 + \gamma C_i) \sum_{t=t_i}^{t_{i+1}-1} \|\mathbf{g}_t\|_\star^2} \right\} \\
&\leq (2+c) \min \left\{ \underbrace{\sum_{i=0}^N (\ell_{t_i}(\mathbf{x}_{t_i}) - \ell_{t_{i+1}-1}(\mathbf{x}_{t_{i+1}}) + V_i)}_{(a)}, \underbrace{\sum_{i=0}^N \sqrt{(3D^2 + \gamma C_i) \sum_{t=t_i}^{t_{i+1}-1} \|\mathbf{g}_t\|_\star^2}}_{(b)} \right\},
\end{aligned}$$

where in the second inequality we used the definition of  $c$ . We now analyze (a) and (b) separately.

For (b), using the Cauchy-Schwartz inequality we have that

$$\begin{aligned}
\sum_{i=0}^N \sqrt{(3D^2 + \gamma C_i) \sum_{t=t_i}^{t_{i+1}-1} \|\mathbf{g}_t\|_\star^2} &\leq \sqrt{\sum_{i=0}^N (3D^2 + \gamma C_i)} \cdot \sqrt{\sum_{i=0}^N \sum_{t=t_i}^{t_{i+1}-1} \|\mathbf{g}_t\|_\star^2} \\
&= \sqrt{3ND^2 + \gamma C_T} \cdot \sqrt{\sum_{t=1}^T \|\mathbf{g}_t\|_\star^2} \\
&\leq \sqrt{\left( 3D^2 \left( \log_2 \frac{C_T}{\sqrt{2}D} + 1 \right) + \gamma C_T \right) \sum_{t=1}^T \|\mathbf{g}_t\|_\star^2}.
\end{aligned}$$

On the other hand, for (a) we have

$$\sum_{i=0}^N (\ell_{t_i}(\mathbf{x}_{t_i}) - \ell_{t_{i+1}-1}(\mathbf{x}_{t_{i+1}}) + V_i) \leq \ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + V_T.$$

Therefore, putting together the results for (a) and (b), we get the stated bound.  $\square$

To summarize, for sequence of comparators whose path-length is observable, from Equation (23) we have a worst-case regret bound of

$$R(\mathbf{u}_{1:T}) = \tilde{O} \left( \min \left\{ V_T, \sqrt{T(1 + C_T)} \right\} \right). \quad (24)$$

In light of this last result, compared to Jadbabaie et al. [2015], our upper bound from Theorem A.1 strictly improves their result when optimistic predictions are not helpful.

We stress that a doubling trick is necessary for Algorithm 4. Indeed, in order to have a fully adaptive learning rate, we should be able to tune it as a function of two quantities varying over time, namely the path-length observed and the temporal variability of the losses paid by the algorithm. While both quantities are increasing quantities over time, they also appear both at the numerator and denominator of the learning rate  $\lambda_t$ . However, this would result in a non-monotone sequence of learning rates, thus contradicting the assumptions in Lemma 5.1. Also, we would like to point out that to the best of our knowledge there are no existing methods in the literature which tune the learning rates with non-monotone sequences.

## B Learning with Expert Advice

In this section we cover the application of Algorithm 2 to the setting of Learning with Expert Advice, as explained in the main paper in Section 5.1.

**Theorem 5.2.** Consider the setting of Learning with Expert Advice on the  $d$ -dimensional simplex  $\Delta_d$ . Assume  $0 \leq g_{t,i} \leq L_\infty$  for all  $t = 1, \dots, T$ , and  $i = 1, \dots, d$ . Assume that  $T \geq d$  and set  $\alpha = d/T$ . Furthermore, set  $\eta_{t+1}^{-1} = \lambda_{t+1} = \frac{1}{\beta^2} \sum_{i=1}^t \delta_t$ , with  $\delta_t$  defined as in lemma 5.1. Then, for any sequence of comparators  $\mathbf{u}_{1:T}$  with  $\mathbf{u}_t \in \Delta_d$  such that  $C_T(\mathbf{u}_{1:T}) \leq \tau$ , using  $\beta^2 = (1 + \tau) \ln T$  the regret of Algorithm 2 run on  $\Delta_d^\alpha$  is bounded as

$$R_T(\mathbf{u}_{1:T}) \leq 2 \min \left\{ \ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + V_T, \right. \\ \left. \sqrt{(1 + (1 + \tau) \ln T) \sum_{t=1}^T \mathbb{E}_t[\mathbf{g}_t^2]} \right\} + 2L_\infty d, \quad (15)$$

where  $\mathbb{E}_t[\mathbf{g}_t^2] = \sum_{i=1}^T x_{t,i} g_{t,i}^2$  and  $V_T = \sum_{t=2}^T \max_{\mathbf{x} \in \Delta_d} \ell_t(\mathbf{x}) - \ell_{t-1}(\mathbf{x})$ .

*Proof.* Given any sequence  $\mathbf{u}_{1:T}$ , with  $\mathbf{u}_t \in \Delta_d$ , we introduce  $\mathbf{u}'_t = \frac{\alpha}{d} \vec{\mathbf{1}} + (1 - \alpha) \mathbf{u}_t$ , where  $\vec{\mathbf{1}}$  is the  $d$ -dimensional all-ones vector. Note that  $\mathbf{u}'_t \in \Delta_d^\alpha$  by definition. The regret can be decomposed as follows

$$R_T(\mathbf{u}_{1:T}) = \sum_{t=1}^T \ell_t(\mathbf{x}_t) - \sum_{t=1}^T \ell_t(\mathbf{u}_t) = \underbrace{\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{x}_t - \mathbf{u}'_t \rangle}_{(a)} + \underbrace{\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{u}'_t - \mathbf{u}_t \rangle}_{(b)}.$$

Now, note that for (b)

$$\begin{aligned} \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{u}'_t - \mathbf{u}_t \rangle &\leq \sum_{t=1}^T \|\mathbf{g}_t\|_\infty \|\mathbf{u}'_t - \mathbf{u}_t\|_1 \\ &\leq L_\infty \sum_{t=1}^T \left\| \frac{\alpha}{d} \vec{\mathbf{1}} + (1 - \alpha) \mathbf{u}_t - \mathbf{u}_t \right\|_1 \\ &\leq L_\infty \sum_{t=1}^T \left( \frac{\alpha}{d} \|\vec{\mathbf{1}}\|_1 + \alpha \|\mathbf{u}_t\|_1 \right) \\ &= 2L_\infty T \alpha, \end{aligned} \quad (25)$$

where the second-to-last inequality derives from applying the triangle inequality.

We can now analyze (a). Note that for any  $\mathbf{u}'_t, \mathbf{u}'_{t-1}, \mathbf{x}_t \in \mathcal{V}$  the following holds

$$\begin{aligned} B_\psi(\mathbf{u}'_t, \mathbf{x}_t) - B_\psi(\mathbf{u}'_{t-1}, \mathbf{x}_t) &= \psi(\mathbf{u}'_t) - \psi(\mathbf{u}'_{t-1}) - \langle \nabla \psi(\mathbf{x}_t), \mathbf{u}'_t - \mathbf{u}'_{t-1} \rangle \\ &= -B_\psi(\mathbf{u}'_{t-1}, \mathbf{u}'_t) + \langle \nabla \psi(\mathbf{x}_t) - \nabla \psi(\mathbf{u}'_t), \mathbf{u}'_{t-1} - \mathbf{u}'_t \rangle \\ &\leq \|\nabla \psi(\mathbf{x}_t) - \nabla \psi(\mathbf{u}'_t)\|_\infty \|\mathbf{u}'_{t-1} - \mathbf{u}'_t\|_1 \\ &\leq \ln \frac{d}{\alpha} \cdot \|\mathbf{u}'_{t-1} - \mathbf{u}'_t\|_1, \end{aligned}$$

where the last inequality derives from the fact that  $\|\nabla \psi(\mathbf{x}_t) - \nabla \psi(\mathbf{u}'_t)\|_\infty = \max_{i \in [d]} \ln \frac{x_{t,i}}{u'_{t,i}} \leq \ln \frac{d}{\alpha}$ .

Therefore, using the prescribed learning rate and applying lemma 5.1 we get

$$\begin{aligned} \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{x}_t - \mathbf{u}'_t \rangle &\leq \lambda_T D^2 + \gamma \sum_{t=2}^T \lambda_t \|\mathbf{u}'_t - \mathbf{u}'_{t-1}\|_1 + \sum_{t=1}^T \delta_t \\ &\leq \lambda_T \left( D^2 + \ln \frac{d}{\alpha} \sum_{t=2}^T (1 - \alpha) \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_1 \right) + \beta^2 \lambda_{T+1} \\ &\leq \lambda_{T+1} \left( \ln \frac{d}{\alpha} + \tau \ln \frac{d}{\alpha} + \beta^2 \right), \end{aligned} \quad (26)$$

where the last inequality derives from bounding the diameter of  $\Delta_d^\alpha$  with respect to the KL as done in Eq. (14),  $1 - \alpha \leq 1$  and the assumption on  $C_T(\mathbf{u}_{1:T}) \leq \tau$ , while the second-to-last inequality from the definition of  $\mathbf{u}'_t$ .

Similarly to the proof of Theorem 5.1, we have that

$$\beta^2 \lambda_{T+1} = \sum_{t=1}^T \ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}_{t+1}) - \lambda_t B_\psi(\mathbf{x}_{t+1}, \mathbf{x}_t)$$

$$\begin{aligned}
&\leq \sum_{t=1}^T \ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}_{t+1}) \\
&\leq \ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + \sum_{t=2}^T \max_{\mathbf{x} \in \Delta_d^\alpha} \ell_t(\mathbf{x}) - \ell_{t-1}(\mathbf{x}) \\
&\leq \ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + \sum_{t=2}^T \max_{\mathbf{x} \in \Delta_d} \ell_t(\mathbf{x}) - \ell_{t-1}(\mathbf{x}) \\
&= \ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + V_T.
\end{aligned} \tag{27}$$

On the other hand, we can improve the second part of the bound compared to a standard application of Theorem 5.1. Indeed, we have that

$$\delta_t \leq \langle \mathbf{g}_t, \mathbf{x}_t - \mathbf{x}_{t+1} \rangle \leq \langle \mathbf{g}_t, \mathbf{x}_t \rangle := \mathbb{E}_t[\mathbf{g}_t],$$

where  $\mathbb{E}_t[\mathbf{g}_t]$  is the expected value of  $\mathbf{g}_t$  under the distribution  $\mathbf{x}_t$ . Furthermore, using the local norms bound for *Mirror Descent* [Orabona, 2019, Section 6.5] we have that

$$\delta_t = \langle \mathbf{g}_t, \mathbf{x}_t - \mathbf{x}_{t+1} \rangle - \lambda_t B_\psi(\mathbf{x}_{t+1}, \mathbf{x}_t) \leq \frac{1}{2\lambda_t} \|\mathbf{g}_t\|_{(\nabla^2 \psi(\mathbf{z}_t))^{-1}}^2 = \frac{1}{2\lambda_t} \sum_{i=1}^d z_{t,i} g_{t,i}^2,$$

for a certain  $\mathbf{z}_t = \theta_t \mathbf{x}_t + (1 - \theta_t) \mathbf{x}_{t+1}$  and  $\theta_t \in [0, 1]$ . Now, observe that from the fact that  $\delta_t \geq 0$  we get  $\langle \mathbf{g}_t, \mathbf{x}_t \rangle \geq \langle \mathbf{g}_t, \mathbf{x}_{t+1} \rangle$ . Therefore, it follows that  $\langle \mathbf{g}_t, \mathbf{z}_t \rangle = \theta \langle \mathbf{g}_t, \mathbf{x}_t \rangle + (1 - \theta) \langle \mathbf{g}_t, \mathbf{x}_{t+1} \rangle \leq \langle \mathbf{g}_t, \mathbf{x}_t \rangle$ , since the  $g_{t,i} \geq 0$  for all  $i$ . Hence, we have that  $\delta_t \leq \frac{1}{2\lambda_t} \sum_{i=1}^d x_{t,i} g_{t,i}^2 := \frac{\mathbb{E}_t[\mathbf{g}_t^2]}{2\lambda_t}$ .

To summarize, we have that

$$\lambda_{t+1} = \lambda_t + \frac{1}{\beta^2} \delta_t \leq \lambda_t + \frac{1}{\beta^2} \min \left\{ \mathbb{E}_t[\mathbf{g}_t], \frac{\mathbb{E}_t[\mathbf{g}_t^2]}{2\lambda_t} \right\}.$$

Therefore, applying lemma A.1 with  $\Delta_t = \lambda_t$ ,  $a_t^2 = \mathbb{E}_t[\mathbf{g}_t^2]$ ,  $b_t = \mathbb{E}_t[\mathbf{g}_t]$ ,  $d = c = \frac{1}{\beta^2}$  we have that

$$\begin{aligned}
\lambda_{T+1} &\leq \sqrt{\frac{1}{\beta^4} \sum_{t=1}^T (\mathbb{E}_t[\mathbf{g}_t])^2 + \frac{1}{\beta^2} \sum_{t=1}^T \mathbb{E}_t[\mathbf{g}_t^2]} \\
&\leq \frac{1}{\beta^2} \sqrt{(1 + \beta^2) \sum_{t=1}^T \mathbb{E}_t[\mathbf{g}_t]^2},
\end{aligned} \tag{28}$$

where the last step derives from Jensen's inequality, i.e.,  $(\mathbb{E}_t[\mathbf{g}_t])^2 \leq \mathbb{E}_t[\mathbf{g}_t^2]$ . Therefore, putting together Eq. (27) and Eq. (28) we get

$$\lambda_{T+1} \leq \frac{1}{\beta^2} \min \left\{ \ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + V_T, \sqrt{(1 + \beta^2) \sum_{t=1}^T \mathbb{E}_t[\mathbf{g}_t^2]} \right\},$$

Finally, note that from our choice of  $\alpha$  and  $\beta^2$ , we have that

$$\frac{\ln \frac{d}{\alpha} + \tau \ln \frac{d}{\alpha} + \beta^2}{\beta^2} = \frac{(1 + \tau) \ln T + \beta^2}{\beta^2} = 2.$$

Hence, adding together Eq. (26) and the upper bounds to Eq. (25) in Eq. (27) and Eq. (28) yields the stated result.  $\square$

We next provide a corollary which shows that the bound in Theorem 5.2 implies a first-order bound which depends on the loss of the sequence of competitors.

**Corollary B.0.1.** *Assume that  $\max_{i,t} g_{t,i} = L_\infty$ . Then, under the same assumptions of Theorem 5.2, Algorithm 2 guarantees*

$$R_T(\mathbf{u}_{1:T}) \leq 2\sqrt{L_\infty(1 + (1 + \ln \tau) \ln T) L_T(\mathbf{u}_{1:T})} + \mathcal{O}(\ln T).$$

*Proof.* Let  $L_T = \sum_{t=1}^T \ell_t(\mathbf{x}_t)$  and  $L_T(\mathbf{u}_{1:T}) = \sum_{t=1}^T \ell_t(\mathbf{u}_t)$ . Consider the second term in the minimum of the bound in Eq. (15). We have that

$$L_T - L_T(\mathbf{u}_{1:T}) \leq B' \sqrt{\sum_{t=1}^T \sum_{i=1}^d x_{t,i} g_{t,i}^2} + B'' \leq B' \sqrt{L_\infty L_T} + B'' ,$$

where  $B' = 2\sqrt{1 + (1 + \ln \tau) \ln T}$  and  $B'' = 2L_\infty d$ . Rearranging terms, we have that

$$L_T - \underbrace{B' \sqrt{L_\infty}}_{=b} \sqrt{L_T} - \underbrace{(L_T(\mathbf{u}_{1:T}) + B'')}_{=c} \leq 0$$

We use a result which says that given  $x, b, c \in \mathbb{R}_+$ , if  $x - b\sqrt{x} - c \leq 0$  holds, then  $x \leq c + b^2 + b\sqrt{c}$ . Applying this result with  $x = \sqrt{L_T}$  we get

$$L_T \leq L_T(\mathbf{u}_{1:T}) + 2L_\infty d + 4L_\infty(1 + (1 + \ln \tau) \ln T) + 2\sqrt{L_\infty(1 + (1 + \ln \tau) \ln T)L_T(\mathbf{u}_{1:T})} .$$

Rearranging terms yields the stated result.  $\square$

## C Composite losses

We are now going to derive a regret bound on the case of composite losses for the static regret scenario using the algorithm *AdalImplicit* from Campolongo and Orabona [2020].

**Theorem C.1.** Let  $\mathcal{V} \subset X \subseteq \mathbb{R}^d$  be a non-empty closed convex set. Let  $\ell_t(\mathbf{x}) = \tilde{\ell}_t(\mathbf{x}) + r(\mathbf{x})$ , where  $r : X \rightarrow \mathbb{R}$  is a convex function. Let  $B_\psi$  be the Bregman divergence with respect to  $\psi : X \rightarrow \mathbb{R}$ . Assume  $\psi$  to be 1-strongly convex w.r.t.  $\|\cdot\|$  and let  $\lambda_t = 1/\eta_t$ . Then, Algorithm 2 with  $\lambda_1 = 0$  and  $\lambda_t = \frac{1}{D^2} \sum_{i=1}^{t-1} \ell_i(\mathbf{x}_i) - \ell_i(\mathbf{x}_{i+1}) - \lambda_i B_\psi(\mathbf{x}_{i+1}, \mathbf{x}_i)$  for  $t = 2, \dots, T$  incurs the following regret bound against any  $\mathbf{u} \in \mathcal{V}$

$$R_T(\mathbf{u}) \leq \min \left\{ 2(\ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + V_T), 2D\sqrt{3 \sum_{t=1}^T \|\mathbf{g}_t\|_*^2} \right\}, \quad (29)$$

where  $V_T = \sum_{t=2}^T \max_{\mathbf{x} \in \mathcal{V}} \tilde{\ell}_t(\mathbf{x}) - \tilde{\ell}_{t-1}(\mathbf{x})$ , and  $\mathbf{g}_t \in \partial \ell_t(\mathbf{x}_t)$ .

*Proof.* First, let  $\mathbf{g}'_t \in \partial \tilde{\ell}_t(\mathbf{x}_{t+1})$ . Note that for any  $\mathbf{u} \in \mathcal{V}$  we have the following

$$\begin{aligned} \eta_t(\ell_t(\mathbf{x}_{t+1}) - \ell_t(\mathbf{u})) &\leq \eta_t \langle \mathbf{g}'_t + \nabla r(\mathbf{x}_{t+1}), \mathbf{x}_{t+1} - \mathbf{u} \rangle \\ &= \langle \eta_t \mathbf{g}'_t + \nabla \psi(\mathbf{x}_{t+1}) - \nabla \psi(\mathbf{x}_t) + \eta_t \nabla r(\mathbf{x}_{t+1}), \mathbf{x}_{t+1} - \mathbf{u} \rangle - \langle \nabla \psi(\mathbf{x}_{t+1}) - \nabla \psi(\mathbf{x}_t), \mathbf{x}_{t+1} - \mathbf{u} \rangle \\ &\leq \langle \nabla \psi(\mathbf{x}_{t+1}) - \nabla \psi(\mathbf{x}_t), \mathbf{u} - \mathbf{x}_{t+1} \rangle \\ &= B_\psi(\mathbf{u}, \mathbf{x}_t) - B_\psi(\mathbf{u}, \mathbf{x}_{t+1}) - B_\psi(\mathbf{x}_{t+1}, \mathbf{x}_t), \end{aligned}$$

where the second inequality derives from the optimality condition of the update rule.

Remember that  $\delta_t = \ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}_{t+1}) - \frac{B_\psi(\mathbf{x}_{t+1}, \mathbf{x}_t)}{\eta_t}$ . After adding  $\ell_t(\mathbf{x}_t)$  on both sides, taking  $\ell_t(\mathbf{x}_{t+1})$ , dividing both sides by  $\eta_t$  and summing over time we get

$$\begin{aligned} \sum_{t=1}^T (\ell_t(\mathbf{x}_t) - \ell_t(\mathbf{u})) &\leq \sum_{t=1}^T \frac{B_\psi(\mathbf{u}, \mathbf{x}_t) - B_\psi(\mathbf{u}, \mathbf{x}_{t+1})}{\eta_t} + \sum_{t=1}^T \delta_t \\ &\leq \frac{D^2}{\eta_1} + D^2 \sum_{t=2}^T \left( \frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right) + \sum_{t=1}^T \delta_t \\ &\leq 2D^2 \lambda_{T+1} . \end{aligned}$$

Now, define  $\tilde{\delta}_t = \tilde{\ell}_t(\mathbf{x}_t) - \tilde{\ell}_t(\mathbf{x}_{t+1}) - \lambda_t B_\psi(\mathbf{x}_{t+1}, \mathbf{x}_t)$ . Note that

$$\begin{aligned} \sum_{t=1}^T \delta_t &= \sum_{t=1}^T \left[ \tilde{\delta}_t + r(\mathbf{x}_t) - r(\mathbf{x}_{t+1}) \right] \\ &\leq \sum_{t=1}^T \left( \tilde{\ell}_t(\mathbf{x}_t) - \tilde{\ell}_t(\mathbf{x}_{t+1}) \right) + r(\mathbf{x}_1) - r(\mathbf{x}_{T+1}) \end{aligned}$$



**Algorithm 5** Adapt-ML-Prod

**Require:** A rule to sequentially pick the learning rates, vector  $\mathbf{w}_0 = (w_{0,1}, \dots, w_{0,d})$  of nonnegative weights that sum to 1.

- 1: **for**  $t = 1, \dots, T$  **do**
- 2:   Pick the learning rates  $\eta_{t-1,i}$  according to the rule.
- 3:   Define  $\mathbf{p}_t$  such that  $p_{t,i} = \eta_{t-1,i} w_{t-1,i} / \boldsymbol{\eta}_{t-1}^\top \mathbf{w}_{t-1}$
- 4:   Observe  $\mathbf{g}_t$  and incur loss  $\ell_t(\mathbf{p}_t) = \langle \mathbf{g}_t, \mathbf{p}_t \rangle$
- 5:   For each expert  $i$  perform the update

$$w_{t,i} = (w_{t-1,i} (1 + \eta_{t-1,i} (\ell_t(\mathbf{p}_t) - \ell_{t,i})))^{\frac{\eta_{t,i}}{\eta_{t-1,i}}}$$

6: **end for**

$$\begin{aligned} &= \ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + \sum_{t=2}^T \left( \tilde{\ell}_t(\mathbf{x}_t) - \tilde{\ell}_{t-1}(\mathbf{x}_t) \right) \\ &\leq \ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + \sum_{t=1}^T \max_{\mathbf{x} \in \mathcal{V}} \tilde{\ell}_t(\mathbf{x}) - \tilde{\ell}_{t-1}(\mathbf{x}). \end{aligned}$$

The rest of the proof follows from the proof of Theorem 5.1 in Appendix A.  $\square$

Compared to Song et al. [2018], the above regret bound is adaptive to the gradients of the loss function. Furthermore, the regret bound from Song et al. [2018] does not contain the temporal variability  $V_T$ , which could potentially lead to constant regret if the loss function stays fixed over time. Next, we are going to show how one can adapt the previous theorem to the dynamic case.

**Theorem 5.3.** *Let  $\mathcal{V} \subset X \subseteq \mathbb{R}^d$  be a non-empty closed convex set. Let  $\ell_t(\mathbf{x}) = \tilde{\ell}_t(\mathbf{x}) + r(\mathbf{x})$ , where  $r : X \rightarrow \mathbb{R}$  is a convex function. Then, under the assumptions of lemma 5.1 the regret of Algorithm 2 run with  $1/\eta_t = \lambda_t = \frac{1}{\beta^2} \sum_{i=1}^{t-1} \delta_i$  and  $\beta^2 = D^2 + \gamma\tau$  against any sequence of comparators  $\mathbf{u}_{1:T}$  whose path-length  $C_T$  is less or equal than  $\tau$  is bounded as*

$$\begin{aligned} R_T(\mathbf{u}_{1:T}) &\leq \min \left\{ 2(\ell_1(\mathbf{x}_1) - \ell_{T+1}(\mathbf{x}_{T+1}) + V_T), \right. \\ &\quad \left. 2\sqrt{(3D^2 + \gamma\tau) \sum_{t=1}^T \|\mathbf{g}_t\|_*^2} \right\}, \end{aligned}$$

where  $V_T = \sum_{t=2}^T \max_{\mathbf{x} \in \mathcal{V}} \tilde{\ell}_t(\mathbf{x}) - \tilde{\ell}_{t-1}(\mathbf{x})$  and  $\mathbf{g}_t \in \partial \ell_t(\mathbf{x}_t)$ .

*Proof.* To prove the stated bound, we can adapt the proof from Theorem C.1. In particular, from the update rule using Eq. (10) we have that

$$\eta_t(\ell_t(\mathbf{x}_{t+1}) - \ell_t(\mathbf{u}_t)) \leq B_\psi(\mathbf{u}_t, \mathbf{x}_t) - B_\psi(\mathbf{u}_t, \mathbf{x}_{t+1}) - B_\psi(\mathbf{x}_{t+1}, \mathbf{x}_t),$$

where  $\mathbf{g}'_t \in \partial \ell_t(\mathbf{x}_{t+1})$ . Following the proof of Theorem 5.1, summing  $\ell_t(\mathbf{x}_t)$  on both sides and rearranging terms we get

$$\begin{aligned} R_T(\mathbf{u}_{1:T}) &\leq \sum_{t=1}^T \lambda_t (B_\psi(\mathbf{u}_t, \mathbf{x}_t) - B_\psi(\mathbf{u}_t, \mathbf{x}_{t+1})) + \sum_{t=1}^T \left[ \tilde{\delta}_t + \beta(r(\mathbf{x}_t) - r(\mathbf{x}_{t+1})) \right] \\ &\leq 2(D^2 + \gamma C_T + \beta^2) \lambda_{T+1}. \end{aligned}$$

From the last inequality, substituting the value of  $\beta^2$  and following the proof of Theorem C.1 yields the desired result.  $\square$

## D Combining algorithms

In this section, we give in detail the results related to Section 6.

First, we point out that Algorithm 3 is an application of the more general *Adapt-ML-Prod* algorithm, which is given in Algorithm 5. We recall the following theorem which provides a regret bound to Algorithm 5 (proof omitted).

**Theorem D.1.** [Gaillard et al., 2014, Theorem 3] For all sequences of loss vectors  $\mathbf{g}_t \in [0, 1]^d$ , for all rules prescribing sequences of learning rates  $\eta_{t,i}$  that, for each  $i$ , are non-increasing in  $t$ , Algorithm 5 ensures

$$R_T(e_i) \leq \frac{1}{\eta_{0,i}} \ln \frac{1}{w_{0,i}} + \sum_{t=1}^T \eta_{t-1,i} r_{t,i}^2 + \frac{1}{\eta_{T,i}} \ln K_T ,$$

where  $r_{t,i} = \langle \mathbf{g}_t, \mathbf{p}_t \rangle - g_{t,i}$  and  $K_T = 1 + \frac{1}{e} \sum_{t=1}^T \sum_{j=1}^d \left( \frac{\eta_{t-1,j}}{\eta_{t,j}} - 1 \right)$ .

### D.1 Strongly-adaptive algorithms

Strongly adaptive algorithms enjoy a  $\tilde{\mathcal{O}}(\sqrt{T})$  regret bound for any interval  $I = [s, e] \subseteq [1, T]$ . The argument used to prove the optimal dynamic regret bound for strongly adaptive algorithms is reported next for completeness. We first provide a lemma that we will use in the proof of the main result.

**Lemma D.1.** [Cutkosky, 2020] Consider a set  $\mathcal{V} \subset \mathbb{R}^d$  such that  $\max_{\mathbf{x}, \mathbf{y} \in \mathcal{V}} \|\mathbf{x} - \mathbf{y}\| \leq D$ . Define a time interval  $I = [s, e]$  and let  $\mathbf{u}_t \in \mathcal{V}$  for any  $t \in I$ . Let  $C_I = \sum_{t=s+1}^e \|\mathbf{u}_t - \mathbf{u}_{t-1}\|$ . Then it is possible to break the interval  $I$  into  $K$  disjoint intervals  $I = J_1 \cup \dots \cup J_K$  such that for each  $i$  we have that  $C_{J_i} \leq 2D$ . Furthermore, we have that  $K \leq \frac{C_I + D}{D}$ .

*Proof.* We can build the set of subintervals  $J_1, \dots, J_K$  iteratively. Define  $J_1 = [s, t_1]$  as the interval such that  $t_1$  is the first time-step when  $\sum_{t=s+1}^{t_1} \|\mathbf{u}_t - \mathbf{u}_{t-1}\| \geq D$ . Then we have that  $C_{J_1} \leq 2D$  (since  $\|\mathbf{u}_t - \mathbf{u}_{t-1}\| \leq D$  in any step  $t$ ). We can repeat this process: given  $t_{i-1}$ , let  $t_i$  be the time-step such that  $C_{[t_{i-1}, t_i]} \geq D$  and define  $J_i = [t_{i-1}, t_i]$ . If such a  $t_i$  does not exist then set  $i = K$  and  $t_i = e$ . Then for any subinterval we have that  $C_{J_i} \leq 2D$ .

On the other hand, we have  $C_{J_i} \geq D$  for all  $i$  but the last one. Assume that there are  $K$  subintervals. We have that  $\sum_{i=1}^K C_{J_i} \leq C_I$ . Therefore,

$$C_I \geq \sum_{i=1}^K C_{J_i} \geq (K-1)D ,$$

from which the desired result follows.  $\square$

We can now prove that result regarding the dynamic regret of strongly adaptive algorithms satisfying a certain condition on the path-length, as stated in Theorem 6.1 (restated here for completeness).

**Theorem 6.1** (Adapted from Cutkosky [2020]). Let  $\mathcal{V} \subset X \subset \mathbb{R}^d$  be non-empty closed convex sets,  $\psi : X \rightarrow \mathbb{R}$ . Define  $D^2 = \max_{\mathbf{x}, \mathbf{y} \in \mathcal{V}} B_\psi(\mathbf{x}, \mathbf{y})$ . Given a sequence of loss functions  $\ell_1, \dots, \ell_T$ , assume there exists an algorithm that for any interval  $I = [s, e] \subseteq [1, T]$  and sequence  $\mathbf{u}_{s:e}$  guarantees a dynamic regret bounded as

$$R_I(\mathbf{u}_{s:e}) = \mathcal{O} \left( (D + C_I) \sqrt{|I|} \right) , \quad (17)$$

where  $|I| = (e - s)$ . Then, for any interval  $J = [s', e'] \subseteq [1, T]$  it also guarantees

$$R_J(\mathbf{u}_{s':e'}) \leq \mathcal{O} \left( \sqrt{|J| D (C_J + D)} \right) , \quad (18)$$

where  $C_J = \sum_{t=s'+1}^{e'} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|$ .

*Proof.* Note that for any  $\mathbf{x}, \mathbf{y} \in \mathcal{V}$  we have  $\frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2 \leq B_\psi(\mathbf{x}, \mathbf{y}) \leq D^2$ . Hence  $\|\mathbf{x} - \mathbf{y}\| \leq \sqrt{2}D$ . Let  $J_1, \dots, J_K$  be the set of disjoint intervals resulting from the construction in Lemma D.1. We have that  $J = J_1 \cup \dots \cup J_K$ , such that for each  $i$  we have  $C_{J_i} \leq 2\sqrt{2}D$ , and  $K \leq \frac{C_J + \sqrt{2}D}{\sqrt{2}D}$ . Now, on each of intervals  $J_i = [s_i, e_i]$ , using Equation (17) the regret is bounded as

$$R(\mathbf{u}_{s_i:e_i}) \leq \mathcal{O} \left( (D + C_{J_i}) \sqrt{|J_i|} \right) = \mathcal{O} \left( D \sqrt{|J_i|} \right) ,$$

where in the last inequality we used the fact that  $C_{J_i} \leq 2\sqrt{2}D$  from lemma D.1.

Then, for interval  $J$  using the Cauchy-Schwartz inequality we have that

$$R(\mathbf{u}_{s:e}) = \sum_{i=1}^K R(\mathbf{u}_{s_i:e_i}) \leq \sum_{i=1}^K \mathcal{O} \left( D \sqrt{|J_i|} \right) \leq \mathcal{O} \left( D \sqrt{K \sum_{i=1}^K |J_i|} \right) \leq \mathcal{O} \left( D \sqrt{\frac{C_I + D}{D} |J|} \right)$$

$$= \mathcal{O}\left(\sqrt{|J|D(C_I + D)}\right)$$

where in the third inequality we used the fact that  $K \leq \frac{C_I + \sqrt{2}D}{\sqrt{2}D}$  from lemma D.1.  $\square$

In particular, note that the above theorem is valid also with  $J = [1, T]$ .

We next prove Theorem 6.2, providing a regret bound to the algorithm given in Section 6.

**Theorem 6.2.** *Let  $\mathcal{V} \subset X \subset \mathbb{R}^d$  be non-empty closed convex sets,  $\psi : X \rightarrow \mathbb{R}$ . Define  $D^2 = \max_{\mathbf{x}, \mathbf{y} \in \mathcal{V}} B_\psi(\mathbf{x}, \mathbf{y})$ . Let  $\ell_1, \dots, \ell_t$  be a sequence of convex loss functions such that  $\ell_t : \mathbb{R}^d \rightarrow [0, 1]$  for all  $t$ . Then, for all sequence of  $\mathbf{u}_{1:T}$  with path-length  $C_T$ , running Algorithm 3 with  $\mathcal{A}$  as a strongly-adaptive algorithm satisfying the condition from Theorem 6.1, and  $\mathcal{B}$  as Algorithm 1, guarantees*

$$R_T(\mathbf{u}_{1:T}) \leq \mathcal{O}\left(\min\left\{V_T, \sqrt{C} + \sqrt{TD(C_T + D)}\right\}\right),$$

where  $C$  is an upper bound to the loss of algorithm  $\mathcal{A}$ .

*Proof.* Let  $\mathbf{y}_t^i$  be the output of algorithm  $i \in \{\mathcal{A}, \mathcal{B}\}$  at time  $t$ . For any  $i \in \{\mathcal{A}, \mathcal{B}\}$ , we can decompose the regret as follows

$$\begin{aligned} \sum_{t=1}^T (\ell_t(\mathbf{x}_t) - \ell_t(\mathbf{u}_t)) &= \sum_{t=1}^T (\ell_t(\mathbf{x}_t) - \ell_t(\mathbf{y}_t^i)) + \sum_{t=1}^T (\ell_t(\mathbf{y}_t^i) - \ell_t(\mathbf{u}_{1:T})) \\ &\leq \sum_{t=1}^T \sum_{i \in \{\mathcal{A}, \mathcal{B}\}} p_{t,i} \ell_t(\mathbf{y}_t^i) - \sum_{t=1}^T \ell_t(\mathbf{y}_t^i) + R_{T,i}(\mathbf{u}_{1:T}), \end{aligned} \quad (30)$$

where we used Jensen's inequality and the regret guarantee of algorithm  $i$  in the last step.

We now analyze the first part in the above bound. Recall that  $w_{1,\mathcal{A}} = w_{1,\mathcal{B}} = \frac{1}{2}$ . For algorithm  $\mathcal{A}$  we denote  $\mathbf{y}_t^{\mathcal{A}} = \mathbf{a}_t$ . Also, we assume  $\eta_t = \eta_{t,\mathcal{A}}$  and  $\eta_{t,\mathcal{B}} = 1/2$  (cf. Algorithm 3).

From Theorem D.1 we have that

$$\begin{aligned} \sum_{t=1}^T \sum_{i \in \{\mathcal{A}, \mathcal{B}\}} p_{t,i} \ell_t(\mathbf{y}_t^i) - \sum_{t=1}^T \ell_t(\mathbf{a}_t) &\leq \frac{1}{\eta_{0,\mathcal{A}}} \ln \frac{1}{w_{0,\mathcal{A}}} + \sum_{t=1}^T \eta_{t-1,\mathcal{A}} r_{t,\mathcal{A}}^2 + \frac{1}{\eta_{T,\mathcal{A}}} \ln K_T \\ &= 2 \ln 2 + \sum_{t=1}^T \eta_{t-1,\mathcal{A}} (\ell_t(\mathbf{b}_t) - \ell_t(\mathbf{a}_t))^2 + \frac{1}{\eta_{T,\mathcal{A}}} \ln K_T \\ &\leq 2 \ln 2 + \sum_{t=1}^T \frac{(\ell_t(\mathbf{b}_t) - \ell_t(\mathbf{a}_t))^2}{\sqrt{1 + \sum_{i=1}^{t-1} (\ell_t(\mathbf{b}_i) - \ell_t(\mathbf{a}_i))^2}} + \ln K_T \sqrt{1 + \sum_{t=1}^T (\ell_t(\mathbf{b}_t) - \ell_t(\mathbf{a}_t))^2} \\ &\leq 2 \ln 2 + (2 + \ln K_T) \sqrt{1 + \sum_{t=1}^T (\ell_t(\mathbf{b}_t) - \ell_t(\mathbf{a}_t))^2} \\ &\leq 2 \ln 2 + (2 + \ln K_T) \sqrt{T+1}, \end{aligned}$$

where the second-to-last inequality derives from applying Orabona [2019, Lemma 4.13], while the last one from the fact that the losses are bounded in  $[0, 1]$ . Note that  $\ln K_T = \mathcal{O}(\ln \ln T)$  from Gaillard et al. [2014, Corollary 4]. Therefore, using this last result in Eq. (30) and the fact that  $R_{T,\mathcal{A}} \leq$  by assumption we have that

$$R_T(\mathbf{u}_{1:T}) \leq \mathcal{O}\left(\ln K_T \sqrt{T+1} + \sqrt{TD(C_T + D)}\right) = \tilde{\mathcal{O}}\left(\sqrt{TD(C_T + D)}\right). \quad (31)$$

On the other hand, for algorithm  $\mathcal{B}$  by applying again Theorem D.1 we have that

$$\begin{aligned} \sum_{t=1}^T \sum_{i \in \{\mathcal{A}, \mathcal{B}\}} p_{t,i} \ell_t(\mathbf{y}_t^i) - \sum_{t=1}^T \ell_t(\mathbf{b}_t) &\leq \frac{1}{\eta_{0,\mathcal{B}}} \ln \frac{1}{w_{0,\mathcal{B}}} + \sum_{t=1}^T \eta_{t-1,\mathcal{B}} r_{t,\mathcal{B}}^2 + \frac{1}{\eta_{T,\mathcal{B}}} \ln K_T \\ &= 2 \ln 2 + 2 \ln K_T, \end{aligned}$$

since  $\eta_t = \frac{1}{2}$  and  $r_{t,\mathcal{A}} = 0$  for all  $t$  by assumption (see Algorithm 3). Using this last result in Eq. (30) we get

$$R_T(\mathbf{u}_{1:T}) \leq \ell_1(\mathbf{x}_1) - \ell_T(\mathbf{x}_{T+1}) + V_T + 2 \ln 2 + 2 \ln K_T = \tilde{\mathcal{O}}(V_T). \quad (32)$$

Taking the minimum between Eq. (31) and Eq. (32) concludes the proof.  $\square$

**Algorithm 6** Generic strongly-adaptive algorithm**Require:** Non-empty closed convex set  $\mathcal{V} \subset \mathbb{R}^d$ , OLO algorithm  $\mathcal{A}$ , expert algorithm  $\mathcal{B}$ 

- 
- 1: **for**  $t = 1, \dots, T$  **do**
  - 2:   Receive predictions from  $\mathcal{A}_1, \dots, \mathcal{A}_t$ , denoted  $\mathbf{y}_1^t, \dots, \mathbf{y}_t^t$
  - 3:   Receive distribution  $\mathbf{p}_t$  from  $\mathcal{B}$
  - 4:   Output  $\mathbf{x}_t = \sum_{i=1}^t p_{t,i} \mathbf{y}_t^i$
  - 5:   Observe loss  $\ell_t$  and pay  $\ell_t(\mathbf{x}_t)$
  - 6:   Pass  $\ell_t$  to  $\mathcal{A}_1, \dots, \mathcal{A}_t$
  - 7:   Set  $g_{t,i} = \ell_t(\mathbf{y}_t^i)$  and pass  $\mathbf{g}_t$  to  $\mathcal{B}$
  - 8: **end for**
- 

**D.2 Learning with Expert Advice**

In this section we sketch the strategy to get the optimal bound for the setting of Learning with Expert Advice (LEA) by combining the prediction of a strongly-adaptive algorithm and Algorithm 1, as suggested by Theorem 6.2. In particular, we need to design a strongly-adaptive algorithm which satisfies the conditions of Theorem 6.1.

**Strongly-adaptive algorithm.** The dominant approach in the design of strongly adaptive algorithms has been the following. Consider an anytime algorithm  $\mathcal{A}$  with static regret bound of  $\tilde{\mathcal{O}}(\sqrt{t})$  for the interval  $[1, t]$ . At each time-step  $t$  initialize a new copy of  $\mathcal{A}$ . Then, to come up with a prediction at round  $t$ , use an expert algorithm  $\mathcal{B}$  to combine the predictions of the  $t$  existing base learners. The resulting strategy is depicted in Algorithm 6. Let  $\mathbf{y}_t^s$  the output at time  $t$  of the algorithm initialized at time  $s$ , i.e.,  $\mathcal{A}_s$ . The regret over an interval  $I = [s, e]$  against any sequence  $\mathbf{u}_{s:e}$  can then be decomposed as follows

$$\begin{aligned}
 \sum_{t=s}^e (\ell_t(\mathbf{x}_t) - \ell_t(\mathbf{u}_t)) &\leq \sum_{t=s}^e \sum_{i=1}^t p_{t,i} \ell_t(\mathbf{y}_t^i) - \sum_{t=1}^T \ell_t(\mathbf{u}_t) \\
 &= \underbrace{\sum_{t=s}^e (\langle \mathbf{p}_t, \mathbf{g}_t \rangle - \ell_t(\mathbf{y}_t^s))}_{\text{Regret of } \mathcal{B}} + \underbrace{\sum_{t=s}^e (\ell_t(\mathbf{y}_t^s) - \ell_t(\mathbf{u}_t))}_{\text{Regret of } \mathcal{A}_s} .
 \end{aligned} \tag{33}$$

where the first inequality derives from Jensen's inequality. Next, we analyze the two contributions separately.

**Algorithm  $\mathcal{B}$ .** This is the regret of an expert algorithm against a fictitious adversary which always commits to the same choice  $\mathbf{y}_t^s$ . In order to have the desired regret bound of  $\tilde{\mathcal{O}}(\sqrt{|I|})$  a regular expert algorithm with regret bound of  $\tilde{\mathcal{O}}(\sqrt{T})$  does not work. Indeed, we need an expert algorithm which only pays for the timesteps when the base algorithm  $\mathcal{A}_s$  has been active, i.e., for the interval  $[s, e]$ . It can be shown that a *sleeping* experts algorithm suffices (details omitted). Hence, we adopt the algorithm *Sleeping CB* from Jun et al. [2017] as the expert algorithm  $\mathcal{B}$  in Algorithm 6, which gives

$$\sum_{t=s}^e \langle \mathbf{p}_t, \mathbf{g}_t \rangle - \sum_{t=s}^e \ell_t(\mathbf{y}_t^s) \leq \tilde{\mathcal{O}}\left(\sqrt{|I|}\right) , \tag{34}$$

for any interval  $I = [s, e] \subseteq [1, T]$ .

**Algorithm  $\mathcal{A}$ .** Differently from Jun et al. [2017] for the case of Learning with Expert Advice, we cannot use the *Coin Betting* algorithm as base algorithm (i.e.,  $\mathcal{A}$  in Algorithm 6), since it does not satisfy a regret bound as the one required by Eq. (17) involving the path-length of the comparator sequence. Instead, we can adopt the algorithm from Theorem 5.2. Indeed, by using  $\beta^2 = \ln T$  in Theorem 5.2 we get the following regret bound against any sequence  $\mathbf{u}_{1:T}$  in the simplex

$$R_T(\mathbf{u}_{1:T}) \leq (2 + C_T) \sqrt{(1 + \ln T) \sum_{t=1}^T \mathbb{E}[\mathbf{g}_t^2]} + 2L_\infty d = \mathcal{O}\left(C_T \sqrt{T \ln T}\right) . \tag{35}$$

**Regret bound.** For any interval  $I \subseteq [1, T]$ , a strongly-adaptive algorithm run with *Coin Betting* as  $\mathcal{B}$  and the algorithm from Theorem 5.2 as base algorithm  $\mathcal{A}$  would get the regret bound required in Eq. (17). Indeed, from the

decomposition in Eq. (33) we have that for any interval  $I = [s, e]$  and sequence  $\mathbf{u}_{s:e}$

$$\begin{aligned} \sum_{t=s}^e (\ell_t(\mathbf{x}_t) - \ell_t(\mathbf{u}_t)) &\leq \sum_{t=s}^e (\langle \mathbf{p}_t, \mathbf{g}_t \rangle - \ell_t(\mathbf{y}_t^s)) + \sum_{t=s}^e (\ell_t(\mathbf{y}_t^s) - \ell_t(\mathbf{u}_t)) \\ &\leq \tilde{\mathcal{O}}(\sqrt{|I|}) + \mathcal{O}(C_I \sqrt{|I| \ln T}) \\ &= \tilde{\mathcal{O}}(C_I \sqrt{|I|}) , \end{aligned}$$

where the first inequality derives from the regret guarantee of the sleeping expert algorithm in Eq. (34) and the one of the base algorithm in Eq. (35). We can therefore apply Theorem 6.1 with  $J = [1, T]$ . This immediately gives a dynamic regret bound of  $\mathcal{O}(\sqrt{TC_T \ln T})$  for the resulting strongly-adaptive algorithm against any sequence  $\mathbf{u}_{1:T}$ . To conclude, we can combine the predictions of the strongly-adaptive algorithm and those from the greedy strategy with Algorithm 3 and apply Theorem 6.2 to get the desired result.

**Running Time.** Note that the technique sketched in Algorithm 6 requires initializing a new algorithm in any step, which would lead to a total runtime of  $\mathcal{O}(T^2)$ . However, there are techniques to reduce this running time to  $\mathcal{O}(T \ln T)$  such as *Geometric Covering* intervals [Jun et al., 2017].