Optimal Parameter-free Online Learning with Switching Cost

Anonymous Author(s)
Affiliation
Address
email

Abstract

Parameter-freeness in online learning refers to the adaptivity of an algorithm with respect to the optimal decision in hindsight. In this paper, we design such algorithms in the presence of switching cost - the latter penalizes the optimistic updates required by parameter-freeness, leading to a delicate design trade-off. Based on a novel *dual space scaling* strategy, we propose a simple yet powerful algorithm for *Online Linear Optimization (OLO) with switching cost*, which improves the existing suboptimal regret bound [ZCP22a] to the optimal rate. The obtained benefit is extended to the expert setting, and the practicality of our algorithm is demonstrated through a sequential investment task.

1 Introduction

Online learning [CBL06, Haz16, Ora19] is a powerful setting for modeling sequential decision making tasks, such as neural network training, financial investment and robotic control. In each round, an agent picks a prediction x_t in a convex domain $\mathcal{X}$, receives a convex and Lipschitz loss function l_t that depends on $x_1, \dots, x_t$, and suffers the loss $l_t(x_t)$. The goal is to ensure that in any environment, the cumulative loss of the agent is never much worse than that of any fixed prediction $u \in \mathcal{X}$. That is, one aims to upper-bound the regret $\sum_{t=1}^{T} [l_t(x_t) - l_t(u)]$, for all time horizon $T \in \mathbb{N}_+$, comparator $u \in \mathcal{X}$ and loss sequence $l_1, \dots, l_T$.

Conventional solutions have a minimax nature. For example, if $\mathcal{X}$ is bounded, then using gradient descent with a conservative learning rate schedule, one can guarantee the optimal $O(\sqrt{T})$ regret bound independent of u [Zin03]. Despite its popularity, such an approach has a few limitations.

1. It requires setting the learning rate based on the diameter of the domain. Many practical problems are naturally unconstrained, making this approach inapplicable.
2. One may have prior information on the optimal fixed prediction (i.e., the comparator u^* that maximizes the regret), possibly from domain knowledge or transfer learning. In that case, the minimax approach cannot utilize it to obtain a better guarantee.

Recent studies of *parameter-free* online learning [LS15, OP16, CO18] aim to address these issues. The domain does not need to be bounded, and the regret bound is an increasing function of $d(u^*, x_1)$, where $d(\cdot, \cdot)$ is some suitable distance measure. Intuitively, these algorithms are both *optimistic* and *robust*: When we have prior information on u^*, we can pick x_1 such that $d(u^*, x_1)$, and consequently the regret bound, are both low. Meanwhile, even when our initialization x_1 is *wrong* (i.e., $d(u^*, x_1)$ is large), the regret bound is still *almost as good* (up to logarithmic factors) as that of gradient descent with the best learning rate in hindsight. Such properties have shown benefits in many applications, e.g., [OT17, JO19, vdH19].

In this paper, we extend the design of parameter-free algorithms to a classical setting with switching costs. Here the agent is penalized not only by its loss, but also by how fast it changes its predictions.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Practically, switching costs are useful whenever the smooth operation of a system is favored, such as in network routing, control of electrical grid, portfolio management with transaction cost, etc. Mathematically, with a given weight $\lambda \geq 0$ and a norm $\|\cdot\|$[1], our goal is to show a parameter-free bound for the *augmented regret*

$$\sum_{t=1}^{T} [l_t(x_t) - l_t(u)] + \lambda \sum_{t=1}^{T-1} \|x_t - x_{t+1}\| .$$

While gradient descent can incorporate the switching cost by scaling its learning rate, extending parameter-free algorithms is a lot harder. Essentially, parameter-freeness is a form of adaptivity, and just like other adaptive algorithms, its key idea is to quickly respond to the incoming information and hedge aggressively. Switching cost, on the other hand, encourages the agent to stay still. Therefore, achieving our goal requires a delicate balance between the two opposite considerations.

Similar trade-offs between adaptivity and switching cost have led to interesting results in the past. For example, [Gof14] showed that the gradient variance adaptivity well-studied in the standard online learning setting is impossible with normed switching costs, thus establishing a clear separation. [DM19] showed that a common analytical technique for switching costs is incompatible to the so-called *"strong adaptivity"* (i.e., a form of adaptivity w.r.t. nonstationary comparators). Regarding our goal, [ZCP22a] proposed the first parameter-free algorithm with switching cost, but the obtained regret bound is sub-optimal in multiple ways. The present work aims at closing this gap.

1.1 Main contribution

We develop parameter-free algorithms for two fundamental settings: (i) *Online Linear Optimization* (OLO) with switching cost; (ii) *Learning with Expert Advice* (LEA) with switching cost.

1. For one-dimensional unconstrained OLO with switching cost, assuming loss gradients $|g_t| \leq 1$ and initial prediction[2] $x_1 = 0$, we propose an algorithm that guarantees

$$\sum_{t=1}^{T} g_t(x_t - u) + \lambda \sum_{t=1}^{T-1} |x_t - x_{t+1}| = C\sqrt{\lambda T} + |u|\, O\left(\sqrt{\lambda T \log(C^{-1}|u|)}\right),$$

   where $C > 0$ is any hyperparameter chosen by the user. Our bound achieves multiple forms of optimality with respect to λ, $|u|$ and T. Notably, a doubling trick can convert it to $C + |u|\, O\left(\sqrt{\lambda T \log(C^{-1}\lambda |u| T)}\right)$, which is a substantial improvement over the existing suboptimal bound $C + |u|\, O\left(\lambda\sqrt{T}\log(C^{-1}\lambda |u| T)\right)$ [ZCP22a]. Our improvement relies on a novel *dual space scaling* strategy for potential methods. Compared to [ZCP22a], our algorithm and analysis are both simpler. Extensions to bounded and high-dimensional domains are presented.
2. Next, we convert this result from OLO to LEA. We demonstrate how classical techniques [LS15, OP16] are *designed* to have large switching costs, and then propose a fix with a clear geometric interpretation. This leads to the first parameter-free algorithm for LEA with switching cost.

Concluding these theoretical results, our algorithm is applied to a portfolio management task with transaction costs. Numerical results support its superiority over the existing approach [ZCP22a].

1.2 Background and notation

Online learning basics Throughout this paper we will only consider linear losses. The generality of our setting is preserved, since convex losses can be reduced to linear losses through the relation $\sum_{t=1}^{T}[l_t(x_t) - l_t(u)] \leq \sum_{t=1}^{T} \langle \nabla l_t(x_t), x_t - u\rangle$ [Haz16, Ora19]. Online learning with linear losses is called *Online Linear Optimization* (OLO). As its important special case, *Learning with Expert Advice* (LEA) considers OLO on a probability simplex, but aims at a different form of regret bound due to its different geometry.

Classical minimax approaches in online learning include *Online Mirror Descent* (OMD) and *Follow the Regularized Leader* (FTRL), with *Online Gradient Descent* (OGD) being their most well-known

[1] We use the L_1 norm for a unified view of OLO and LEA. Our strategy can be extended to other norms.
[2] For general x_1, we can replace $|u|$ in the regret bound by $|u - x_1|$.

special case. We write "gradient descent" as the minimax baseline for the ease of exposition. Moreover, both OMD and FTRL have elegant duality interpretations [Ora19, Section 6.4.1 and 7.3], involving simultaneous updates on the primal space (the domain $\mathcal{X}$) and the dual space (the space of gradients). We will exploit this duality in our analysis.

Parameter-free online learning Also known as *comparator-adaptivity*, parameter-free online learning aims at matching the optimally-tuned gradient descent in hindsight, without knowing the correct tuning parameter (i.e., the optimal comparator u^*). The associated regret bound can appear in different forms, depending on the specific learning setting.

1. For LEA, a parameter-free bound has the form $O(\sqrt{T \cdot \mathrm{KL}(u||\pi)})$, where u and π are distributions on the expert space representing the comparator and a user-chosen prior. Such an idea was initiated in [CFH09], and the analysis was improved and extended by a series of works [CV10, LS15, KVE15, CLW21, NBC+21]. Notably, a parameter-free LEA algorithm naturally induces a bound on the ε*-quantile regret* - the regret with respect to the ε-quantile best expert; this is particularly meaningful when the number of experts is large. Lower bounds were considered in [NBC+21].

   We will present an improvement of the $\sqrt{\mathrm{KL}}$ divergence in this paper. Frameworks that generalize root KL to f*-divergences* have been studied in [Alq21, NBC+21], but to our knowledge, no existing algorithm guarantees a better divergence term than root KL, even without switching costs.

2. For unconstrained OLO, typical parameter-free bounds are $C + \|u\| \, O\left(\sqrt{T \log(C^{-1} \|u\|_* T)}\right)$ or $C\sqrt{T} + \|u\| \, O\left(\sqrt{T \log(C^{-1} \|u\|_*)}\right)$, where a prior x_1 can be added by letting $u \leftarrow u - x_1$. These two bounds are both *Pareto-optimal* [ZCP22b], as they represent different trade-offs on the *loss* (the regret at $u = x_1$) and the *asymptotic* regret (when $\|u - x_1\|$ is large). Existing works [MO14, CO18, FRS18, MK20, JC22] were mostly independent of the LEA setting, but unified views were presented in [FRS15, OP16]. Lower bounds were discussed in [SM12, Ora13, ZCP22b].

Switching cost Motivated by numerous applications, switching costs in online decision making have been studied from many different angles. For example, beside online learning, the online algorithm community has investigated settings like *smoothed online optimization* [CGW18, GLSW19, LQL20] and *convex body chasing* [BLLS19, Sel20], where the loss function l_t is observed *before* the agent picks the prediction x_t. There, the switching cost is the key consideration that prevents the trivial strategy $x_t \in \arg\min_x l_t(x)$. As for online learning, an additional complication is that x_t (e.g., the investment portfolio) should be selected without knowing l_t (e.g., tomorrow's stock price).

Even within online learning, there are several ways to model the switching cost. In cases like network routing, every switch means changing the packet route, which can be costly. Therefore, one needs a *lazy* agent whose amount of switches (or its expectation) [KV05, GVW10, AT18, CYLK20, SK21] is as low as possible - a good modeling candidate is $\mathbf{1}[x_t \neq x_{t+1}]$. Alternatively, one could take a *smooth* view [ABL+13, BCKP21, WWYZ21, ZJLY21] where the agent can perform as many switches as it wishes, as long as the cumulative distance of switching is low - in this view, switching cost can be a norm $\|x_t - x_{t+1}\|$ or its smoothed variant $\|x_t - x_{t+1}\|^2$. The present work considers the L_1 norm switching cost motivated by the transaction cost in some financial applications. Notably, for LEA, the L_1 norm unifies the lazy view and the smooth view [DM19, Section 5.2].

Although switching costs have been extensively studied, existing works on the combination of adaptivity and switching cost are quite sparse. As one should carefully trade-off these two opposite requirements, there have been interesting impossibility results [Gof14, DM19], highlighted in our introduction. In this regard, one should not believe that every classical adaptivity can be naturally achieved with switching cost. Fortunately, we show that the optimal parameter-freeness can indeed be achieved, thus improving the suboptimal result in [ZCP22a].

Relation to downstream problems More generally, incorporating switching costs amounts to considering a *history-dependent* adversary: it can pick loss functions that depend not only on the instantaneous prediction x_t, but also on the previous prediction x_{t-1}. One could further generalize this setting to *online learning with memory* [CBDS13, AHM15], where the loss depends on a fixed-length prediction history, and finally to *dynamical systems* [ABH+19, SSH20, Sim20], where the entire history matters. In fact, a common procedure in *nonstochastic control* [ABH+19] is to bound the risk in the future by a properly scaled switching cost. Achieving parameter-freeness with switching cost may benefit these important downstream problems as well, by making algorithms *easy to combine* [Cut19, Cut20, ZCP22a].

Notation Let f^* be the Fenchel conjugate of a function f. $\Delta(d)$ represents the d-dimensional probability simplex; KL and TV denote the KL divergence and the total variation distance, respectively. For two integers $a \leq b$, $[a : b]$ is the set of all integers c such that $a \leq c \leq b$. $\log$ represents the natural logarithm when the base is omitted. Throughout this paper, "increasing" and "positive" are not strict (i.e., include equality as well).

For a twice differentiable function $V(t, S)$ where t represents time and S represents a spatial variable, let $\nabla_t V$, $\nabla_{tt} V$, $\nabla_S V$ and $\nabla_{SS} V$ be the first and second order partial derivatives. In addition, we define discrete derivatives as

$$\begin{aligned} \bar{\nabla}_t V(t, S) &:= V(t, S) - V(t-1, S), \\ \bar{\nabla}_S V(t, S) &:= (1/2) \cdot [V(t, S+1) - V(t, S-1)], \\ \bar{\nabla}_{SS} V(t, S) &:= V(t, S+1) + V(t, S-1) - 2V(t, S). \end{aligned} \tag{1}$$

2 OLO with switching cost

This section presents our main result, a parameter-free OLO algorithm with switching cost. We will start with the 1D unconstrained setting, followed by extensions to general cases.

2.1 The 1D unconstrained setting

We consider the domain $\mathcal{X} = \mathbb{R}$, a Lipschitz constant $G > 0$ for the loss gradients, and a weight $\lambda \geq 0$ for switching costs. In the t-th round, the agent predicts $x_t \in \mathbb{R}$, receives a loss gradient $g_t \in [-G, G]$ that depends on past predictions $x_{1:t}$, and suffers an augmented loss $g_t x_t + \lambda |x_t - x_{t-1}|$ (w.l.o.g., let $x_0 = x_1 = 0$). We consider the augmented regret for all $u \in \mathbb{R}$ and $T \in \mathbb{N}_+$:

$$\mathrm{Regret}_T^{\lambda}(u) := \sum_{t=1}^{T} g_t(x_t - u) + \lambda \sum_{t=1}^{T-1} |x_t - x_{t+1}| . \tag{2}$$

Ignoring the dependence on G for now, our goal is to show a parameter-free bound $\tilde{O}(|u| \sqrt{\lambda T})$, more specifically the optimal rates $C + |u| O(\sqrt{\lambda T \log(C^{-1} \lambda |u| T)})$ or $C\sqrt{\lambda T} + |u| O(\sqrt{\lambda T \log(C^{-1} |u|)})$ for any hyperparameter $C > 0$. These two cases are equivalent via the standard doubling trick [SS11].

For minimax algorithms like bounded domain gradient descent, one can use scaled learning rates $\eta_t \propto 1/\sqrt{\lambda t}$ to ensure that both sums in (2) are $O(\sqrt{\lambda T})$, thus obtaining a combined $O(\sqrt{\lambda T})$ regret bound. However, such a divide-and-conquer approach does not apply to parameter-free algorithms, as one cannot *separately* show the desirable bound on the two sums in (2). To see this, suppose one could guarantee the second sum alone is at most $1 + |u| O(\sqrt{T \log(|u| T)})$; here we only focus on the dependence on $|u|$ and T. Since this cumulative switching cost is an algorithmic quantity independent of the comparator, we can take infimum with respect to u and obtain a "budget" of 1 for this sum. Following this argument, $|x_T| \leq |x_1| + \sum_{t=1}^{T-1} |x_t - x_{t+1}| = O(1)$. That is, the algorithm should only predict around the origin, which clearly leads to large regret with respect to far-away comparators, under certain loss sequences.

The challenge can be motivated in another way. As shown in [Ora19, Figure 9.1], the one-step switching cost $|x_t - x_{t+1}|$ of parameter-free algorithms can grow exponentially with respect to t, whereas such a quantity is uniformly bounded in gradient descent. In fact, the exponential growth is the key mechanism for standard parameter-free algorithms (i.e., without switching cost) to cover an unconstrained domain fast enough. This is however problematic when switching is penalized, as one can no longer control the switching cost by uniformly scaling $|x_t - x_{t+1}|$.

2.2 Switching-adjusted potential

To address these issues, one should bound the switching cost and the standard OLO regret in a *unified* framework, instead of treating them separately. The prior work [ZCP22a] used the coin-betting approach from [OP16, CO18]. In the t-th round, the algorithm maintains a quantity Wealth_{t-1}; by picking a *betting fraction* $\beta_t \in [0, 1]$, the prediction is set to $x_t = \beta_t \mathrm{Wealth}_{t-1}$. To ensure low switching cost, the betting fraction β_t in [ZCP22a] is capped by a decreasing upper bound $O(1/\sqrt{t})$. Such a hard threshold is very conservative, which could be the reason of their suboptimal result.

Algorithm 1 One-dimensional unconstrained OLO with switching costs.

Require: A hyperparameter $C > 0$, the Lipschitz constant G, and a potential function $V(t, S)$ that implicitly depends on λ and G. Initialize $S_0 = 0$.

1: **for** $t = 1, 2, \ldots$ **do**

2: Predict $x_t = \bar{\nabla}_S V(t, S_{t-1})$, and receive the loss gradient g_t. Let $S_t = S_{t-1} - g_t/G$.

3: **end for**

In contrast, we follow the more general potential framework explored by a parallel line of works [MO14, FRS18, MK20, ZCP22b]. Coin-betting is essentially derived from certain types of potentials [OP16], and many theoretical results using coin-betting can be recovered by the latter. In general, a potential algorithm is defined with a potential function $V(t, S)$, where t represents the time index, and S represents a "sufficient statistic". In each round, the algorithm computes $S_{t-1} = -\sum_{i=1}^{t-1} g_i/G$, and the prediction x_t is the derivative $\nabla_S V$ evaluated at (t, S_{t-1}). We will specifically consider Algorithm 1, which is a variant based on the discrete derivative $\bar{\nabla}_S V$, cf. (1).

One could think of the potential framework as the dual approach of FTRL - the potential function and the regularizer are naturally Fenchel conjugates. While the FTRL analysis relies on a one-step regret bound on the *primal space* (the domain $\mathcal{X}$, cf. [Ora19, Lemma 7.1]), the potential framework constructs a similar one-step relation on the *dual space* (the space of S_t, cf. [ZCP22b, Lemma 3.1]). Along this interpretation, **our key idea is to incorporate switching costs by scaling on the dual space, rather than only on the primal space.** That is, given a potential function that works without switching costs, we scale the sufficient statistic sent to its second argument by a function of λ.

To better demonstrate this idea, let us first consider a quadratic potential $V(t, S) = (1/2) \cdot CGS^2$. The potential method suggests the prediction $x_t = \nabla_S V(t, S_{t-1}) = C \sum_{i=1}^{t-1} g_i = x_{t-1} - Cg_{t-1}$, which is simply gradient descent with learning rate C. Scaling on the primal space means scaling V directly, while scaling on the dual space means scaling the sufficient statistic S. It is clear that both cases are equivalent to scaling the effective learning rate, which is the standard way to incorporate switching costs in bounded domain gradient descent. In other words, for this gradient descent potential, the two types of scaling are essentially the same.

Now, to achieve optimal parameter-freeness, we need a better potential where scaling on the dual space actually makes a difference. With some α that will eventually depend on λ, we consider Algorithm 1 induced by the potential

$$V_\alpha(t, S) = C\sqrt{\alpha t} \left[2 \int_0^{S/\sqrt{4\alpha t}} \left(\int_0^u \exp(x^2) dx \right) du - 1 \right]. \quad (3)$$

When the Lipschitz constant $G = 1$, it has been shown [ZCP22b] that $\alpha = 1/2$ leads to parameter-freeness without switching cost. Here we use $\alpha = 4\lambda G^{-1} + 2$, which amounts to scaling *both* the primal space and the dual space: on the primal space, we scale up the overall prediction by $\Theta(\sqrt{\lambda G^{-1} + 1})$, and on the dual space we scale down the sufficient statistic S by $\Theta(1/\sqrt{\lambda G^{-1} + 1})$. The latter gives us the optimal parameter-free bound (i.e., Pareto-optimal rate in $|u|$ and T), while the former helps us obtain the optimal rate in λ. Due to incorporating λ into the potential function V_α, we call our approach the *switching-adjusted potential method*.

Finally, although the definition of V_α seems mysterious at first glance, it is actually derived from a clean continuous-time analysis presented in Appendix A.1. Such a continuous limit perspective provides an intuitive justification for our scaling strategy.

2.3 Optimal parameter-free bound

Despite its simplicity, our approach improves the existing result [ZCP22a] by a considerable margin. We next present our 1D optimal parameter-free bound, discuss its significance, and sketch its proof.

Theorem 1. *If $\alpha = 4\lambda G^{-1} + 2$, then Algorithm 1 induced by the potential V_α guarantees*

$$\text{Regret}_T^\lambda(u) \le \sqrt{(4\lambda G + 2G^2)T} \left[C + |u| \left(\sqrt{4 \log \left(1 + \frac{|u|}{C} \right)} + 2 \right) \right],$$

for all $u \in \mathbb{R}$ and $T \in \mathbb{N}_+$.

Theorem 1 simultaneously achieves several forms of optimality.

1. Pareto-optimal loss-regret trade-off: considering the dependence on u and T, $\mathrm{Regret}_T^{\lambda}(u) = O\left(|u|\sqrt{T\log|u|}\right)$, while the *cumulative loss* satisfies $\mathrm{Regret}_T^{\lambda}(0) = O(\sqrt{T})$. An existing lower bound [ZCP22b, Theorem 10] shows that even without switching cost, all algorithms satisfying a $O(\sqrt{T})$ loss bound must suffer a $\Omega\left(|u|\sqrt{T\log|u|}\right)$ regret bound. In this sense, our algorithm attains a *Pareto-optimal* loss-regret trade-off, in a strictly generalized setting with switching costs.
2. On T alone: $\mathrm{Regret}_T^{\lambda}(u) = O(\sqrt{T})$. Despite achieving parameter-freeness (i.e., adaptivity to u), the asymptotic rate on T is still the optimal one, matching the well-known minimax lower bound.
3. On λ alone: $\mathrm{Regret}_T^{\lambda}(u) = O(\sqrt{\lambda})$. Our bound has the optimal dependence on the switching cost weight [GVW10, Theorem 5].

To compare Theorem 1 to [ZCP22a], we have to convert them to the same loss-regret trade-off, i.e., both guaranteeing $\mathrm{Regret}_T^{\lambda}(0) = O(1)$ or $\mathrm{Regret}_T^{\lambda}(0) = O(\sqrt{T})$. Here we take the first approach - details are presented in Appendix A.4. By a doubling trick, assuming $G = 1$ for clarity, our bound can be converted to $C + |u|\, O\left(\sqrt{\lambda T\log(C^{-1}\lambda\,|u|\,T)}\right)$, which improves the rate $C + |u|\, O\left(\lambda\sqrt{T}\log(C^{-1}\lambda\,|u|\,T)\right)$ from [ZCP22a, Theorem 1]. Specifically, our converted upper bound also attains Pareto-optimality in this regime (i.e., matching the lower bound $\Omega\left(|u|\sqrt{T\log(|u|\,T)}\right)$ in [Ora13]), whereas the existing approach does not.

The proof of Theorem 1 is sketched as follows, with the formal analysis deferred to Appendix A.3. It mostly follows a standard potential argument, which is another benefit over the existing approach - the idea of this proof is easier to interpret and generalize.

Proof sketch of Theorem 1 The first step is to show a one-step bound on the growth rate of the potential. If there is no switching cost, then the *Discrete Ito formula* [Kle13, HLPR20, ZCP22b] can serve this purpose, which applies to any convex potential V.

Lemma 2.1 (Lemma 3.1 of [ZCP22b])**.** *If the potential function $V(t, S)$ is convex in S, then against any adversary, Algorithm 1 guarantees for all $t \in \mathbb{N}_+$,*

$$V(t, S_t) - V(t-1, S_{t-1}) \leq -G^{-1}g_t x_t + \bar{\nabla}_t V(t, S_{t-1}) + (1/2)\cdot \bar{\nabla}_{SS}V(t, S_{t-1}).$$

Our key observation is the following lemma, which incorporates switching costs into V_α.

Lemma 2.2. *For all $\alpha > 0$, consider Algorithm 1 induced by the potential function V_α. For all $t \in \mathbb{N}_+$,*

$$|x_t - x_{t+1}| \leq \bar{\nabla}_S V_\alpha(t, S_{t-1}+1) - \bar{\nabla}_S V_\alpha(t, S_{t-1}-1).$$

Combining the above, if we define

$$\Delta_t := \bar{\nabla}_t V_\alpha(t, S_{t-1}) + \frac{1}{2}\bar{\nabla}_{SS}V_\alpha(t, S_{t-1}) + G^{-1}\lambda\left[\bar{\nabla}_S V_\alpha(t, S_{t-1}+1) - \bar{\nabla}_S V_\alpha(t, S_{t-1}-1)\right], \tag{4}$$

then a telescopic sum yields a *cumulative loss bound*

$$\mathrm{Regret}_T^{\lambda}(0) \leq \sum_{t=1}^{T}(g_t x_t + \lambda\,|x_t - x_{t+1}|) \leq -G\cdot V_\alpha(T, S_T) + G\sum_{t=1}^{T}\Delta_t.$$

To proceed, we need to control the residual term Δ_t, which may seem problematic due to its complicated form. Fortunately, a careful analysis shows that Δ_t *vanishes* with a proper choice of α!

Lemma 2.3. *If $\alpha \geq 4\lambda G^{-1} + 2$, then for all t and against any adversary, $\Delta_t \leq 0$.*

Finally, with the updated loss bound $\mathrm{Regret}_T^{\lambda}(0) \leq -G\cdot V_\alpha(T, S_T)$, our regret bound follows from the classical loss-regret duality [MO14, Ora19].

2.4 Extension to bounded and higher-dimensional domains

Generalizing the above 1D setting, we discuss the extension of Algorithm 1 to bounded domains and higher-dimensional domains. Due to limited space, details are presented in Appendix A.5.

First, for a constrained domain $\mathcal{X} \subset \mathbb{R}$, we can use a well-known black-box reduction [CO18, Section 4] on top of Algorithm 1 such that the exact bound in Theorem 1 carries over (w.r.t. any $u \in \mathcal{X}$). Similar strategies apply to higher-dimensional problems, but here we emphasize the 1D special case due to an additional feature: if the domain $\mathcal{X}$ has a finite diameter D, then the *switching cost alone* of the combined algorithm has a $\tilde{O}(D\sqrt{\tau})$ bound on *any time interval of length* τ. This could be useful when switching costs have high priority [SK21, WWYZ21] and should be independently bounded. Moreover, it allows the combination of parameter-free algorithms [ZCP22a] in settings with long term prediction effects (e.g., switching cost or memory).

Theorem 2. *Consider the setting of Section 2.1, but on a (smaller) closed and convex domain $\mathcal{X} \subset \mathbb{R}$. Let x^* be an arbitrary point in $\mathcal{X}$. For all $C > 0$, Algorithm 3 in Appendix A.5 guarantees*

$$\mathrm{Regret}_T^{\lambda}(u) \leq \sqrt{(4\lambda G + 2G^2)T}\left[C + |u - x^*|\left(\sqrt{4\log\left(1 + \frac{|u - x^*|}{C}\right)} + 2\right)\right],$$

for all $u \in \mathcal{X}$ and $T \in \mathbb{N}_+$. Moreover, if $\mathcal{X}$ has a finite diameter D, then on any time interval $[T_1 : T_2] \subset \mathbb{N}_+$, the same algorithm guarantees

$$\sum_{t=T_1}^{T_2-1} |x_t - x_{t+1}| \leq 22\sqrt{T_2 - T_1}\left[2D + C + 2D\sqrt{\log(1 + DC^{-1})}\right].$$

From a technical perspective, the second part of Theorem 2 is particularly interesting due to its *non-black-box* use of the reduction approach: we characterize how this reduction (implicitly) controls the unconstrained base algorithm, resulting in the "concentration" of its sufficient statistic S_t (i.e., $S_t = O(\sqrt{t})$) as if losses are stochastic. A similar bound was presented in [ZCP22a], but it critically relies on hard-thresholding a betting fraction, which, as we have shown, is suboptimal. In contrast, we use a different analysis on the improved base algorithm (Algorithm 1) to achieve this switching cost bound and an improved regret bound *simultaneously*.

As for higher dimensions, let us consider the setting where $\mathcal{X} = \mathbb{R}^d$, $\|g_t\|_\infty \leq G$, and the switching costs are measured by the L_1 norm. This serves as a nice bridge towards our LEA approach and financial applications. We run Algorithm 1 on each coordinate separately [SM12], and scale the hyperparameter C by $1/d$.

Theorem 3. *Consider OLO with switching costs on the domain $\mathcal{X} = \mathbb{R}^d$; assume loss gradients satisfy $\|g_t\|_\infty \leq G$. For all $C > 0$, Algorithm 4 in Appendix A.5 guarantees ($\alpha = 4\lambda G^{-1} + 2$)*

$$\sum_{t=1}^{T} \langle g_t, x_t - u\rangle + \lambda \sum_{t=1}^{T-1} \|x_t - x_{t+1}\|_1 \leq G\sqrt{\alpha T}\left[C + \|u\|_1\left(\sqrt{4\log\left(1 + \frac{\|u\|_\infty d}{C}\right)} + 2\right)\right],$$

for all $u \in \mathbb{R}^d$ and $T \in \mathbb{N}_+$.

3 LEA with switching cost

Our Algorithm 1 can also be applied to *LEA with switching cost*, resulting in the first parameter-free algorithm there. Conversion techniques without switching costs were studied in [LS15, OP16], and since then, they have become standard tools for the online learning community. Here we present a different view on this conversion problem, based on its connection to the constrained domain reduction [CO18] adopted in our OLO analysis. In particular, it leads to a mechanism for incorporating switching costs, with a clear geometric interpretation.

The setting of *LEA with switching cost* is a special case of the high-dimensional OLO problem. Let d be the number of experts. Then, compared to the setting of Theorem 3, we simply change $\mathcal{X}$ to the probability simplex $\Delta(d)$. The main difference with OLO is the form of parameter-free bounds - here we aim at $\mathrm{Regret}_T^{\lambda}(u) = O(\sqrt{T \cdot \mathrm{KL}(u||\pi)})$, where $\pi \in \Delta(d)$ is a prior chosen at the beginning. Achieving such a root KL bound relies on special conversion techniques.

Existing approaches [LS15, OP16] have the following procedure. Given a 1D OLO algorithm that predicts on $\mathbb{R}_+$, independent copies are created for each coordinate and updated using certain surrogate losses. A meta-algorithm queries the coordinate-wise predictions $\{w_{t,i}; i \in [1 : d]\}$,

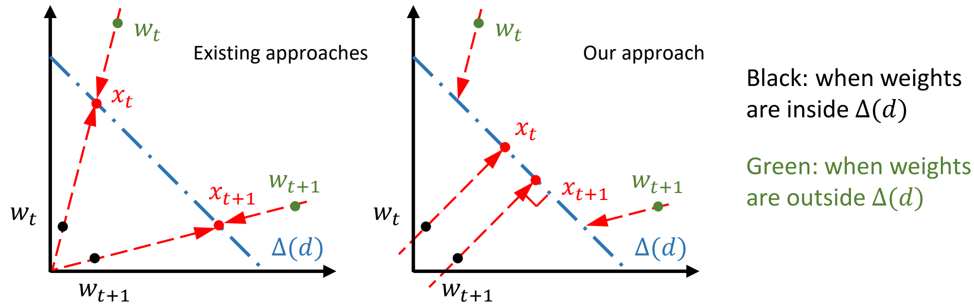

Figure 1: Switching costs in LEA-OLO reductions. Left: existing approaches. Right: ours, where the projection of w_t contains two cases. (i) $\|w_t\|_1 \geq 1$, shown in green; (ii) $\|w_t\|_1 < 1$, shown in black.

collects them into a weight vector $w_t = [w_{t,1}, \ldots, w_{t,d}]$, and finally predicts the scaled weight $x_t = w_t/\|w_t\|_1$ on $\Delta(d)$. Despite its general success, such an approach has a discontinuity problem when switching costs are incorporated: if two consecutive weights w_t and w_{t+1} are both close to the origin, then simply scaling them to $\Delta(d)$ can lead to a large switching cost, even when $\|w_t - w_{t+1}\|_1$ is small. This problem is exacerbated by the typical setting[3] of $w_1 = 0$, due to the associated analysis. A graphical demonstration is provided in Figure 1 (Left).

Our solution is based on *a unified view* of the LEA-OLO reduction and the constrained domain reduction [CO18]. Starting without switching costs, we observe that the general Banach version of the latter can also convert OLO to LEA, therefore specialized techniques are not required for this task. Algorithmically, we set $x_t \in \arg\min_{x\in\Delta(d)} \|x - w_t\|_1$ as opposed to $x_t = w_t/\|w_t\|_1$. The surrogate losses for the base algorithms are also different, which we elaborate in Appendix B.3.

A major benefit of this unified view is the non-uniqueness of the L_1 norm projection: if $\|w_t\| < 1$, then any $x_t \in \Delta(d)$ satisfying $\{x_{t,i} \geq w_{t,i}; \forall i\}$ minimizes $\|x - w_t\|_1$ on $\Delta(d)$. This brings more flexibility to the algorithm design: for the setting with switching costs, we adopt (i) the orthogonal projection $x_t = w_t + d^{-1}(1 - \|w_t\|_1)$ when $\|w_t\|_1 \leq 1$, and (ii) the scaling $x_t = w_t/\|w_t\|_1$ when $\|w_t\|_1 > 1$. The orthogonal projection is better for controlling switching costs, as shown in Figure 1 (Right). Concretely, this leads to the first parameter-free algorithm for LEA with switching cost.

Theorem 4. *For LEA with switching cost, given any prior π in the relative interior of $\Delta(d)$, Algorithm 5 from Appendix B.2 guarantees*

$$\sum_{t=1}^{T} \langle g_t, x_t - u\rangle + \lambda \sum_{t=1}^{T-1} \|x_t - x_{t+1}\|_1 = \left[\sqrt{\mathrm{TV}(u||\pi)\cdot \mathrm{KL}(u||\pi)} + 1\right] \cdot O\left(\sqrt{(\lambda G + G^2)T}\right),$$

for all $u \in \Delta(d)$ and $T \in \mathbb{N}_+$.

We emphasize two strengths of this bound.

1. Since it is parameter-free, such a bound only implicitly depends on d through the divergence term $\sqrt{\mathrm{TV}\cdot\mathrm{KL}}$. In favorable cases we may have a good prior π such that $\mathrm{TV}(u||\pi)\cdot\mathrm{KL}(u||\pi) = O(1)$; this will save us a $\sqrt{\log d}$ factor compared to minimax algorithms (with switching costs), such as *Follow the Lazy Leader* [KV05] and *Shrinking Dartboard* [GVW10].
2. Even without switching costs, we improve the $\sqrt{\mathrm{KL}}$ divergence term in existing parameter-free bounds [CFH09, LS15, OP16] to $\sqrt{\mathrm{TV}\cdot\mathrm{KL}}$. The latter is better since (i) TV is always less than 1, and (ii) there exist $p, q \in \Delta(d)$ such that $\mathrm{TV}(p||q)\cdot\mathrm{KL}(p||q) \leq 1$ but $\mathrm{KL}(p||q) \geq \sqrt{\log d} - o(1)$ (cf. Appendix B.3). In other words, compared to $\sqrt{\mathrm{KL}}$, the $\sqrt{\mathrm{TV}\cdot\mathrm{KL}}$ bound is never worse (up to constants), and can save at least a $(\log d)^{1/4}$ factor in certain cases. Generalizations of root KL to *f-divergences* have been considered in [Alq21, NBC$^+$21], but to our knowledge, no existing algorithm guarantees a better divergence term than root KL.

[3]When $w_t = 0$, x_t can be arbitrary on $\Delta(d)$ by definition. However, as w_t changes continuously w.r.t. the observed information, it could hover around 0 at some point, thus experiencing the sketched problem.

4 Unconstrained investment with transaction cost

Finally, we present applications to a portfolio selection problem with transaction costs. Online portfolio selection has been studied by multiple communities, resulting in a large amount of literature (see [LH14, Doc16] for general expositions). Here we consider an *unconstrained* setting, allowing both short selling (i.e., holding negative amount of assets) and margin trading (i.e., borrowing money to buy assets). Its connections and differences to the classical rebalancing setting [Cov91, CO96, HSSW98, KV02, LWZ18] are detailed in Appendix C.1.

We consider a market with d assets and discrete trading period $t \in \mathbb{N}_+$. In the t-th round, an algorithm chooses a portfolio vector $x_t = [x_{t,1}, \dots, x_{t,d}] \in \mathbb{R}^d$, where $x_{t,i}$ is the *number of shares* of the i-th asset that the algorithm suggests to hold. Compared to the previous round, we need to buy $x_{t,i} - x_{t-1,i}$ shares[4] (or sell, if negative), which requires paying a $\lambda |x_{t,i} - x_{t-1,i}|$ transaction cost[5]. Then, the market reveals a number $g_{t,i} \in [-G, G]$, which represents the price change per share (of the i-th asset) in this round. This effectively increases the value of our portfolio by $\langle g_t, x_t \rangle$.

The considered performance metric is the increased amount of *wealth* on any time horizon $[1 : T] \subset \mathbb{N}_+$, and such wealth includes the total value of our portfolio *plus cash*. Our goal is to show that the performance of our algorithm is never much worse than that of any unconstrained *Buy-and-Hold* (BAH) strategy, which picks a portfolio vector $u \in \mathbb{R}^d$ at the beginning and holds that amount throughout the considered time horizon. That is, we aim to upper bound $\sum_{t=1}^{T} \langle -g_t, x_t - u \rangle + \lambda \sum_{t=1}^{T-1} \|x_t - x_{t+1}\|_1$ for all $u \in \mathbb{R}^d$ and $T \in \mathbb{N}_+$. This is exactly the setting of Theorem 3 with flipped gradients, therefore the same theoretical result carries over.

To complement the theory, we present some numerical results on a synthetic market. Let $G = 1$, $\lambda = 0.1$, and the market contains five assets with different return characteristics. Each $g_{t,i}$ is the summation of a i.i.d. noise, a periodic fluctuation and a constant trend, e.g.,

$$g_{t,i} = 0.6 \cdot \text{Uniform}[-1, 1] + 0.2 \sin\left[(t/500 + 1)\, \pi\right] + 0.2.$$

Two algorithms are tested, our Algorithm 4 (i.e., "ours"), and the baseline [ZCP22a, Algorithm 1, adapted]. Both algorithms require a confidence parameter (our C, and the *initial wealth* for the baseline, also denoted by C). They are set to 1 following the practice of *parameter-free* algorithms [OP16, CLO20, ZCP22a]. Each algorithm is tested in 50 random trials, and the increased wealth $\sum_{\tau=1}^{t} \langle g_\tau, x_\tau \rangle - \lambda \sum_{\tau=1}^{t-1} \|x_\tau - x_{\tau+1}\|_1$ (mean $\pm$ std) is plotted in Figure 2, higher is better. In this setting, our algorithm beats the baseline by a considerable margin, due to being a lot less conservative.

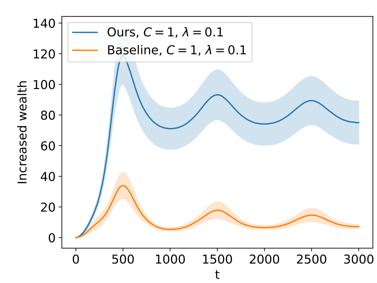

Figure 2: Synthetic market. Both algorithms are in their default parameter-free implementation.

Finally, detailed settings and further experiments, including preliminary results on historical US stock data, are deferred to Appendix C.2 and C.3. Specifically, we also test different λ to show that our algorithm scales to transaction costs better.

5 Conclusion

The present work investigates the design of parameter-free algorithms in the presence of switching cost. By carefully trading off these two opposite considerations, we propose a simple algorithm for OLO with switching cost, which improves the suboptimal regret bound [ZCP22a] to the optimal rate. Extensions of this algorithm lead to new results for bounded domain OLO, parameter-free LEA, and unconstrained portfolio selection.

Limitation and future work Our result requires a known G and a time-invariant λ, which could be generalized in future works. Different from [ZCP22a], we did not discuss applications to control theory, which is interesting on its own. Also, one may combine our portfolio selection approach with adversarial rebalancing and stochastic modeling, in order to further improve its practical performance.

[4] W.l.o.g., assume $x_0 = x_1$.

[5] The coefficient λ can depend on i, the sign of $x_{t,i} - x_{t+1,i}$ and the sign of $x_{t,i}$, but for simplicity we use the same λ for all cases.

References

[ABH+19] Naman Agarwal, Brian Bullins, Elad Hazan, Sham Kakade, and Karan Singh. Online control with adversarial disturbances. In *International Conference on Machine Learning*, pages 111–119. PMLR, 2019.

[ABL+13] Lachlan Andrew, Siddharth Barman, Katrina Ligett, Minghong Lin, Adam Meyerson, Alan Roytman, and Adam Wierman. A tale of two metrics: Simultaneous bounds on competitiveness and regret. In *Conference on Learning Theory*, pages 741–763. PMLR, 2013.

[AHM15] Oren Anava, Elad Hazan, and Shie Mannor. Online learning for adversaries with memory: price of past mistakes. *Advances in Neural Information Processing Systems*, 28, 2015.

[Alq21] Pierre Alquier. Non-exponentially weighted aggregation: Regret bounds for unbounded loss functions. In *International Conference on Machine Learning*, pages 207–218. PMLR, 2021.

[AT18] Jason Altschuler and Kunal Talwar. Online learning over a finite action set with limited switching. In *Conference On Learning Theory*, pages 1569–1573. PMLR, 2018.

[BCKP21] Aditya Bhaskara, Ashok Cutkosky, Ravi Kumar, and Manish Purohit. Power of hints for online learning with movement costs. In *International Conference on Artificial Intelligence and Statistics*, pages 2818–2826. PMLR, 2021.

[BK99] Avrim Blum and Adam Kalai. Universal portfolios with and without transaction costs. *Machine Learning*, 35(3):193–205, 1999.

[BLLS19] Sébastien Bubeck, Yin Tat Lee, Yuanzhi Li, and Mark Sellke. Competitively chasing convex bodies. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 861–868, 2019.

[CBDS13] Nicolo Cesa-Bianchi, Ofer Dekel, and Ohad Shamir. Online learning with switching costs and other adaptive adversaries. *Advances in Neural Information Processing Systems*, 26, 2013.

[CBL06] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[CFH09] Kamalika Chaudhuri, Yoav Freund, and Daniel J Hsu. A parameter-free hedging algorithm. *Advances in neural information processing systems*, 22, 2009.

[CGW18] Niangjun Chen, Gautam Goel, and Adam Wierman. Smoothed online convex optimization in high dimensions via online balanced descent. In *Conference On Learning Theory*, pages 1574–1594. PMLR, 2018.

[CLO20] Keyi Chen, John Langford, and Francesco Orabona. Better parameter-free stochastic optimization with ode updates for coin-betting. *arXiv preprint arXiv:2006.07507*, 2020.

[CLW21] Liyu Chen, Haipeng Luo, and Chen-Yu Wei. Impossible tuning made possible: A new expert algorithm and its applications. In *Conference on Learning Theory*, pages 1216–1259. PMLR, 2021.

[CO96] Thomas M Cover and Erik Ordentlich. Universal portfolios with side information. *IEEE Transactions on Information Theory*, 42(2):348–363, 1996.

[CO18] Ashok Cutkosky and Francesco Orabona. Black-box reductions for parameter-free online learning in banach spaces. In *Conference On Learning Theory*, pages 1493–1529. PMLR, 2018.

[Cov91] Thomas M Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, 1991.

[Cut19] Ashok Cutkosky. Combining online learning guarantees. In *Conference on Learning Theory*, pages 895–913. PMLR, 2019.

[Cut20] Ashok Cutkosky. Parameter-free, dynamic, and strongly-adaptive online learning. In *International Conference on Machine Learning*, pages 2250–2259. PMLR, 2020.

[CV10] Alexey Chernov and Vladimir Vovk. Prediction with advice of unknown number of experts. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pages 117–125, 2010.

[CYLK20] Lin Chen, Qian Yu, Hannah Lawrence, and Amin Karbasi. Minimax regret of switching-constrained online convex optimization: No phase transition. *Advances in Neural Information Processing Systems*, 33:3477–3486, 2020.

[DM19] Amit Daniely and Yishay Mansour. Competitive ratio vs regret minimization: achieving the best of both worlds. In *Algorithmic Learning Theory*, pages 333–368. PMLR, 2019.

[Doc16] Robert Dochow. *Online algorithms for the portfolio selection problem*. Springer, 2016.

[FRS15] Dylan J Foster, Alexander Rakhlin, and Karthik Sridharan. Adaptive online learning. *Advances in Neural Information Processing Systems*, 28:3375–3383, 2015.

[FRS18] Dylan J Foster, Alexander Rakhlin, and Karthik Sridharan. Online learning: Sufficient statistics and the burkholder method. In *Conference On Learning Theory*, pages 3028–3064. PMLR, 2018.

[GLSW19] Gautam Goel, Yiheng Lin, Haoyuan Sun, and Adam Wierman. Beyond online balanced descent: An optimal algorithm for smoothed online optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

[Gof14] Eyal Gofer. Higher-order regret bounds with switching costs. In *Conference on Learning Theory*, pages 210–243. PMLR, 2014.

[GVW10] Sascha Geulen, Berthold Vöcking, and Melanie Winkler. Regret minimization for online buffering problems using the weighted majority algorithm. In *Conference on Learning Theory*, pages 132–143, 2010.

[Haz16] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[HLPR20] Nicholas JA Harvey, Christopher Liaw, Edwin A Perkins, and Sikander Randhawa. Optimal anytime regret for two experts. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1404–1415. IEEE, 2020.

[HSSW98] David P Helmbold, Robert E Schapire, Yoram Singer, and Manfred K Warmuth. On-line portfolio selection using multiplicative updates. *Mathematical Finance*, 8(4):325–347, 1998.

[JC22] Andrew Jacobsen and Ashok Cutkosky. Parameter-free mirror descent. *arXiv preprint arXiv:2203.00444*, 2022.

[JO19] Kwang-Sung Jun and Francesco Orabona. Parameter-free locally differentially private stochastic subgradient descent. *arXiv preprint arXiv:1911.09564*, 2019.

[Kle13] Achim Klenke. *Probability theory: a comprehensive course*. Springer Science & Business Media, 2013.

[KV02] Adam Tauman Kalai and Santosh Vempala. Efficient algorithms for universal portfolios. *Journal of Machine Learning Research*, pages 423–440, 2002.

[KV05] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[KVE15] Wouter M Koolen and Tim Van Erven. Second-order quantile methods for experts and combinatorial games. In *Conference on Learning Theory*, pages 1155–1175. PMLR, 2015.

[LH14] Bin Li and Steven CH Hoi. Online portfolio selection: A survey. *ACM Computing Surveys (CSUR)*, 46(3):1–36, 2014.

[LQL20] Yingying Li, Guannan Qu, and Na Li. Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit. *IEEE Transactions on Automatic Control*, 66(10):4761–4768, 2020.

[LS15] Haipeng Luo and Robert E Schapire. Achieving all with no parameters: Adanormalhedge. In *Conference on Learning Theory*, pages 1286–1304. PMLR, 2015.

[LWZ18] Haipeng Luo, Chen-Yu Wei, and Kai Zheng. Efficient online portfolio with logarithmic regret. *Advances in Neural Information Processing Systems*, 31, 2018.

[MK20] Zakaria Mhammedi and Wouter M Koolen. Lipschitz and comparator-norm adaptivity in online learning. In *Conference on Learning Theory*, pages 2858–2887. PMLR, 2020.

[MO14] H Brendan McMahan and Francesco Orabona. Unconstrained online linear learning in hilbert spaces: Minimax algorithms and normal approximations. In *Conference on Learning Theory*, pages 1020–1039. PMLR, 2014.

[MR22] Zakaria Mhammedi and Alexander Rakhlin. Damped online newton step for portfolio selection. *arXiv preprint arXiv:2202.07574*, 2022.

[NBC+21] Jeffrey Negrea, Blair Bilodeau, Nicolò Campolongo, Francesco Orabona, and Dan Roy. Minimax optimal quantile and semi-adversarial regret via root-logarithmic regularizers. *Advances in Neural Information Processing Systems*, 34, 2021.

[OLL17] Laurent Orseau, Tor Lattimore, and Shane Legg. Soft-bayes: Prod for mixtures of experts with log-loss. In *International Conference on Algorithmic Learning Theory*, pages 372–399. PMLR, 2017.

[OP16] Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. *Advances in Neural Information Processing Systems*, 29, 2016.

[Ora13] Francesco Orabona. Dimension-free exponentiated gradient. *Advances in Neural Information Processing Systems*, 26, 2013.

[Ora19] Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

[OT17] Francesco Orabona and Tatiana Tommasi. Training deep networks without learning rates through coin betting. *Advances in Neural Information Processing Systems*, 30:2160–2170, 2017.

[Sel20] Mark Sellke. Chasing convex bodies optimally. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1509–1518. SIAM, 2020.

[Sim20] Max Simchowitz. Making non-stochastic control (almost) as easy as stochastic. *Advances in Neural Information Processing Systems*, 33:18318–18329, 2020.

[SK21] Uri Sherman and Tomer Koren. Lazy oco: Online convex optimization on a switching budget. In *Conference on Learning Theory*, pages 3972–3988. PMLR, 2021.

[SM12] Matthew Streeter and Brendan Mcmahan. No-regret algorithms for unconstrained online convex optimization. *Advances in Neural Information Processing Systems*, 25, 2012.

[SS11] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2):107–194, 2011.

[SSH20] Max Simchowitz, Karan Singh, and Elad Hazan. Improper learning for non-stochastic control. In *Conference on Learning Theory*, pages 3320–3436. PMLR, 2020.

[vdH19] Dirk van der Hoeven. User-specified local differential privacy in unconstrained adaptive online learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[WWYZ21] Guanghui Wang, Yuanyu Wan, Tianbao Yang, and Lijun Zhang. Online convex optimization with continuous switching constraint. *Advances in Neural Information Processing Systems*, 34, 2021.

[ZAK22] Julian Zimmert, Naman Agarwal, and Satyen Kale. Pushing the efficiency-regret pareto frontier for online learning of portfolios and quantum states. *arXiv preprint arXiv:2202.02765*, 2022.

[ZCP22a] Zhiyu Zhang, Ashok Cutkosky, and Ioannis Paschalidis. Adversarial tracking control via strongly adaptive online learning with memory. In *International Conference on Artificial Intelligence and Statistics*, pages 8458–8492. PMLR, 2022.

[ZCP22b] Zhiyu Zhang, Ashok Cutkosky, and Ioannis Paschalidis. PDE-based optimal strategy for unconstrained online learning. *arXiv preprint arXiv:2201.07877*, 2022.

[Zin03] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, pages 928–936, 2003.

[ZJLY21] Lijun Zhang, Wei Jiang, Shiyin Lu, and Tianbao Yang. Revisiting smoothed online learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] The present work is mainly theoretical.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix C.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] Our experiments are not computationally demanding.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]