Understanding Adam optimizer via Online Learning of Updates: Adam is FTRL in Disguise

Kwangjun Ahn [1 2] **Zhiyu Zhang** [3] **Yunbum Kook** [4] **Yan Dai** [5]

Abstract

Despite the success of the Adam optimizer in practice, the theoretical understanding of its algorithmic components still remains limited. In particular, most existing analyses of Adam show the convergence rate that can be simply achieved by non-adative algorithms like SGD. In this work, we provide a different perspective based on online learning that underscores the importance of Adam's algorithmic components. Inspired by Cutkosky et al. (2023), we consider the framework called *online learning of updates/increments*, where we choose the updates/increments of an optimizer based on an online learner. With this framework, the design of a good optimizer is reduced to the design of a good online learner. Our main observation is that Adam corresponds to a principled online learning framework called Follow-the-Regularized-Leader (FTRL). Building on this observation, we study the benefits of its algorithmic components from the online learning perspective.

1. Introduction

Let $F : \mathbb{R}^d \to \mathbb{R}$ be the (training) loss function we want to minimize. In machine learning applications, F is often minimized via an iterative optimization algorithm which starts at some initialization $\mathbf{w}_0$ and recursively updates

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mathbf{\Delta}_t \quad \text{for } t = 0, 1 \dots , \tag{1.1}$$

where $\mathbf{\Delta}_t$ denotes the update/increment[1] chosen by the algorithm at the t-th iteration. Practical optimizers often choose the update $\mathbf{\Delta}_t$ based on the past (stochastic) gradients $\mathbf{g}_{1:t} = (\mathbf{g}_1, \dots, \mathbf{g}_t)$ where $\mathbf{g}_t$ is the stochastic gradient of F collected during the t-th iteration. For instance, stochastic gradient descent (SGD) corresponds to choosing $\mathbf{\Delta}_t = -\alpha_t \mathbf{g}_t$ in (1.1) for some learning rate $\alpha_t > 0$.

For training deep neural networks, one of the most popular choices is the Adam optimizer (Kingma and Ba, 2014). In particular, several recent works have observed that Adam and its variants are particularly effective for training Transformer-based neural network models (Zhang et al., 2020b; Kunstner et al., 2023; Jiang et al., 2023; Pan and Li, 2023; Ahn et al., 2024). Given some learning rate $\gamma_t > 0$ and discounting factors $\beta_1, \beta_2 \in (0, 1)$, Adam chooses $\mathbf{\Delta}_t$ on each coordinate $i = 1, 2, \dots, d$ by combining $\mathbf{g}_{1:t}$ as[2]

$$\mathbf{\Delta}_t[i] = -\gamma_t \frac{(1-\beta_1)\sum_{s=1}^t \beta_1^{t-s}\mathbf{g}_s[i]}{\sqrt{(1-\beta_2^2)\sum_{s=1}^t (\beta_2^{t-s}\mathbf{g}_s[i])^2}} ,$$

where $\mathbf{v}[i]$ denotes the i-th coordinate of a vector $\mathbf{v}$. For a streamlined notation, we define the *scaled learning rate* $\alpha_t \leftarrow \gamma_t \cdot {}^{(1-\beta_1)}/\sqrt{1-\beta_2^2}$ and consider

$$\mathbf{\Delta}_t[i] = -\alpha_t \frac{\sum_{s=1}^t \beta_1^{t-s}\mathbf{g}_s[i]}{\sqrt{\sum_{s=1}^t (\beta_2^{t-s}\mathbf{g}_s[i])^2}} . \tag{Adam}$$

Compared to SGD, the notable components of Adam is the fact that it aggregates the past gradients $\mathbf{g}_{1:t}$ (*i.e.*, **momentum**) with the **discounting factors** β_1, β_2.

Despite the prevalent application of Adam in deep learning, our theoretical grasp of its mechanics remains incomplete, particularly regarding the roles and significance of its core elements: the **momentum** and the **discounting factors**. Most existing theoretical works on Adam and its variants primarily focus on characterizing the convergence rate for convex functions or smooth nonconvex functions (Reddi et al., 2018; Zhou et al., 2019; Chen et al., 2019; Zou et al., 2019; Alacaoglu et al., 2020; Guo et al., 2021; Défossez et al., 2022; Zhang et al., 2022; Li et al., 2023; Wang et al., 2023) for which methods like SGD already achieve the minimax optimal convergence rate. In fact, the latest works in this line (Li et al., 2023; Wang et al., 2023) both mention

[1]MIT [2]Microsoft Research [3]Havard University [4]Georgia Tech [5]Tsinghua University. Correspondence to: Kwangjun Ahn <kjahn@mit.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1]Sometimes, "update" refers to the iterate $\mathbf{w}_t$, but throughout this work, we mean the increment $\mathbf{w}_{t+1} - \mathbf{w}_t$.

[2]For simplicity, we remove the debiasing step and the appearance of ϵ in the denominator used in the original paper.

that their convergence rate of Adam gets worse with momentum (Wang et al., 2023, §6) or the rate of Adam is no better than that of SGD (Li et al., 2023, §7). A notable exception is Crawshaw et al. (2022) where they show the benefits of momentum in a variant of Adam, under the generalized smoothness conditions of Zhang et al. (2020a).

In this work, we take a different approach to understand Adam from a online learning perspective, as outlined below.

1.1. Our Approach and Main Results

Our starting point is the main insight of Cutkosky et al. (2023) that the design of nonconvex optimizers falls under the scope of *online linear optimization*, an iconic setting in online learning. Specifically, one can regard the selection of the update $\mathbf{\Delta}_t$ based on $\mathbf{g}_{1:t}$ as an online prediction procedure. Such a framework will be called **online learning of updates/increments** (OLU).

Building on this framework, we then notice that it is important to choose an online learner that performs well in dynamic environments (Cutkosky et al., 2023). Better dynamic regret leads to better optimization performance (Theorem 2.1), and therefore, the design of good optimizers is reduced to designing good *dynamic online learners*. Along this line, our results can be summarized as follows:

- (Section 2:) Our main observation is that the popular Adam optimizer corresponds to choosing a classical online learner called Follow-the-Regularized-Leader (FTRL) (Gordon, 1999; Kalai and Vempala, 2005; Shalev-Shwartz and Singer, 2006; Abernethy et al., 2008; Nesterov, 2009; Hazan and Kale, 2010). Specifically, when using the framework OLU, Adam is recovered by plugging in a discounted instance of FTRL well-suited for dynamic environment, which we call β-FTRL.
- (Section 3:) We provide the dynamic regret guarantees of β-FTRL (Theorem 3.1 and Theorem 3.2) through a novel **discounted-to-dynamic conversion**. It gives us a new perspective on the role of Adam's algorithmic components, namely the momentum and the discounting factors. Our results suggest that **both components are crucial for designing a good dynamic online learner** (see Subsection 3.2).
- (Section 4:) We justify the importance of a good dynamic regret, via its implications for optimization. Along the way, we discuss optimization settings for which Adam could be potentially beneficial.

2. Adam is FTRL in Disguise

Iterative optimization algorithms are closely connected to adversarial online learning. For example, SGD is often analyzed through online gradient descent (OGD), its online learning counterpart. To exploit this connection in (1.1), the traditional approach is using an online learner to directly choose the *iterates* $\mathbf{w}_t$, as demonstrated by Bottou (1998); Cesa-Bianchi et al. (2004); Duchi et al. (2011); Li and Orabona (2019); Ward et al. (2019) and many more. Diverging from this common approach, we consider a new approach due to Cutkosky et al. (2023) that applies the online learner to choose the *updates* $\mathbf{\Delta}_t$.

2.1. Choosing Updates/Increments via Online Learning

Consider an iconic setting of online learning called *online linear optimization* (OLO). For the consistency with our optimization algorithm (1.1), we will introduce OLO using slightly nonstandard notations. In each round t, the algorithm (or online learner) chooses a point $\mathbf{\Delta}_t \in \mathbb{R}^d$, and then receives a linear loss function $\ell_t(\cdot) = \langle \mathbf{v}_{t+1}, \cdot \rangle$ and suffers the loss of $\ell_t(\mathbf{\Delta}_t)$. In other words, it chooses $\mathbf{\Delta}_t$ based on the previous loss sequence $\mathbf{v}_{1:t} \coloneqq (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_t)$ and then receives the next loss $\mathbf{v}_{t+1}$. The performance of the online learner is measured by the *regret* against a comparator sequence $\mathbf{u}_{0:T-1}$, defined as

$$R_T(\mathbf{u}_{0:T-1}) \coloneqq \sum_{t=1}^{T} \langle \mathbf{v}_t, \mathbf{\Delta}_{t-1} - \mathbf{u}_{t-1} \rangle \,. \qquad (2.1)$$

To be precise, (2.1) is called the *dynamic regret* in the literature (Zinkevich, 2003). Another common metric, *static regret*, is a special case of (2.1) where all $\mathbf{u}_t = \mathbf{u}$; this is denoted as $R_T(\mathbf{u})$.

Now given an online learner LEARNER, we consider an optimization algorithm that outputs the $\mathbf{\Delta}_t$ in (1.1) using LEARNER. More formally,

$$\boxed{\mathbf{\Delta}_t \text{ is chosen by LEARNER based on } \mathbf{g}_{1:t}.} \qquad \text{(OLU)}$$

We call this framework *online learning of updates* (or *online learning of increments*). This framework was first proposed by Cutkosky et al. (2023) (under the name *online-to-nonconvex conversion*) to design algorithms that find critical points for nonsmooth and nonconvex stochastic optimization problems. Under OLU, we want LEARNER of choice to be a good online learner for dynamic environments, as summarized in the informal statement below.

Theorem 2.1 (**Importance of dynamic regret in OLU**; see Theorem 4.1)**.** *In OLU, a better dynamic regret of* LEARNER *leads to a better optimization guarantee. Therefore, we want* LEARNER *to have a low dynamic regret.*

To understand this elegant reduction, let us give examples of how Theorem 4.1 is applied. Recent works (Cutkosky et al., 2023; Zhang and Cutkosky, 2024) choose an online gradient descent (OGD) (Zinkevich, 2003) as LEARNER to design algorithms for finding stationary points for nonconvex and nonsmooth functions. When LEARNER is chosen as OGD, the resulting optimization algorithm under OLU turns out to be SGD with momentum (Zhang and Cutkosky, 2024). However, OGD is known to require a careful tuning of learning rate (Zinkevich, 2003). What if we use an adaptive online learner as LEARNER?

Our main observation is that Adam can be recovered by choosing LEARNER as an adaptive version of Follow-the-Regularized-Leader that is well-suited for dynamic environments, which we gradually elaborate.

2.2. Basics of Follow-the-Regularized-Leader (FTRL)

Follow-the-Regularized-Leader (FTRL) is a classical algorithmic framework in online learning. Unlike the more intuitive descent-type algorithms, the key idea of FTRL is selecting the decisions by solving a convex optimization problem in each round. Throughout, we focus on the 1D case of OLO ($d = 1$) since the update of Adam is coordinate-wise. In particular, the overall regret for the d-dimension would be the sum of the regret of each coordinate.

The 1D linear loss function is given by $\ell_t(\Delta) = v_{t+1}\Delta$ for $v_{t+1} \in \mathbb{R}$. We use the subscript $t + 1$ to highlight that it is only revealed after deciding Δ_t.

FTRL relies on a nonnegative convex *regularizer* Φ, which is set to $\Phi = \frac{1}{2}|\cdot|^2$ in this work. The algorithm initializes at $\Delta_0 = 0$ and in each round outputs

$$\Delta_t = \arg\min_x \Big[\frac{1}{\eta_t}\Phi(x) + \sum_{s=1}^t v_s x\Big] = -\eta_t \sum_{s=1}^t v_s \,, \tag{FTRL}$$

where the effective step size $\eta_t > 0$ is non-increasing in t. The remaining task is to choose good step sizes η_t.

One prominent choice is the *adaptive* step size of the **scale-free FTRL** algorithm (Orabona and Pál, 2018, §3) in the style of McMahan and Streeter (2010); Duchi et al. (2011). Scale-free FTRL chooses $\eta_t = \alpha/\sqrt{\sum_{s=1}^t v_s^2}$ based on a scaling factor $\alpha > 0$, resulting in the update

$$\Delta_t = -\alpha \frac{\sum_{s=1}^t v_s}{\sqrt{\sum_{s=1}^t v_s^2}} \,. \tag{2.2}$$

Here if the denominator is zero, then we set the output $\Delta_t = 0$. The update (2.2) is independent of any constant scaling of loss sequence $v_{1:t}$, making it a *scale-free* update. This is beneficial when the magnitude of loss sequence varies across different coordinates.

We remark that the analysis of scale-free FTRL is in fact quite subtle, as echoed by McMahan (2017); Orabona and Pál (2018). Using a different proof strategy, we prove a static regret bound (Theorem A.1) of scale-free FTRL that slightly strengthens that of Orabona and Pál (2018).

2.3. Adam Corresponds to Discounted-FTRL

Now back to OLU, let us use FTRL to choose the update Δ_t. Denoting the coordinate-wise gradients in optimization by $g_{1:t}$, a naïve approach is to use FTRL directly by setting $v_t \leftarrow g_t$. Unfortunately, this approach is not a good one because FTRL is designed to achieve low static regret, while OLU requires low *dynamic* regret. In fact, it is shown by (Jacobsen and Cutkosky, 2022, Theorem 2) that Algorithms of the form (FTRL) are **not** good dynamic online learners. See also the lower bounds in Theorem 3.3.

One could already see intuitively why this is the case: any algorithm in this form does not "forget the past", as the output Δ_t is the (regularized) minimizer of the cumulative loss $L_t(x) \coloneqq \sum_{s=1}^t v_s x$. Therefore, it is only competitive *w.r.t.* a fixed comparator that minimizes $L_t(x)$, instead of a time-varying comparator sequence.

To address this issue, our approach is to "**discount**" the losses from the distant past. In particular, we implement this by gradually up-scaling the losses over time. The intuition is that when deciding the output Δ_t, the recent losses would have much higher "weights" compared to older ones, which essentially makes the latter negligible.

Theorem 2.2 (Informal; see Theorems 3.1 and 3.2)**.** *For some $\beta \in (0, 1)$, the discounted version of scale-free FTRL that internally replaces v_t by $\beta^{-t}v_t$ is a good dynamic online learner.*

Remarkably, plugging this discounted scale-free FTRL into OLU would almost recover Adam. There are just two small issues: in the Adam update, the (scaled) learning rate α_t is time-varying, and we need two discounting factors β_1 and β_2 for the numerator and the denominator separately. It is not hard to fix this last bit, and we end up with an FTRL instance which given the input g_t picks

$$v_t \leftarrow \beta_1^{-t} g_t \,, \quad \text{and} \quad \eta_t = \frac{\alpha_t(\beta_1/\beta_2)^t}{\sqrt{\sum_{s=1}^t (\beta_2^{-s} g_s)^2}} \,. \tag{2.3}$$

Collecting all the pieces above yields our first main result.

Proposition 2.3 (**Adam is discounted-FTRL in disguise**)**.** *For some learning rate $\alpha_t > 0$ and discounting factors $\beta_1, \beta_2 \in (0, 1]$, FTRL with (2.3) is equivalent to picking*

$$\Delta_t = -\alpha_t \frac{\sum_{s=1}^t \beta_1^{t-s} g_s}{\sqrt{\sum_{s=1}^t (\beta_2^{t-s} g_s)^2}} \,.$$

Applying it as a coordinate-wise LEARNER *in OLU recovers the Adam optimizer in Adam.*

Recall that OLU connects the problem of optimization to the well-established problem of dynamic regret minimization.

Given Proposition 2.3, we make use of this connection to understand the components of Adam from the dynamic regret perspective. That is the main focus of the next section. Before getting into that, we briefly compare our approach with existing derivations of Adam based on FTRL.

2.4. Comparison with the Previous Approach

In fact, Zheng and Kwok (2017) propose a derivation of Adam based on FTRL. However, their approach is quite different than ours, as we detail below.

We first briefly summarize the approach of Zheng and Kwok (2017). Their main idea is to consider the "weighted" version of proximal-FTRL defined as

$$w_t = \arg\min_w \sum_{s=1}^{t} \lambda_s \left(\langle g_s, w \rangle + \frac{1}{2} \|w - w_{s-1}\|_{Q_s}^2 \right),$$

for some weights $\{\lambda_s\}$ and positive semi-definite matrices $\{Q_s\}$. Given this, their main observation is that Adam roughly corresponds to this proximal-FTRL with carefully chosen $\{\lambda_s\}$ and $\{Q_s\}$.

Although their motivation to explain Adam with a version of FTRL is similar to ours, we highlight that their approach is different than ours. In fact, our approach overcomes some of the limitations of Zheng and Kwok (2017).

- Firstly, their derivation actually needs a heuristic adjustment of changing the anchor points of the regularizer from w_{s-1} to w_{t-1}. A priori, it is not clear why such adjustment is needed, and to the best of our knowledge, there is no formal justification given. But with our approach, such an adjustment is naturally derived because under OLU, the online learner chooses the update/increment instead of the iterate.
- Secondly, in Zheng and Kwok (2017), in order to recover Adam, they have to choose $\{w_s\}$ and $\{Q_s\}$ carefully, which also lacks justification. One of the main advantages of our approach is the fact that the discounting factors are theoretically justified via the dynamic regret perspective. More specifically, we show that without the discounting factor, FTRL is not a good dynamic learner.

3. Discounted-FTRL as a Dynamic Learner

This section provides details on Theorem 2.2, focusing on the special case of Adam where $\beta_1, \beta_2 = \beta$ for some $\beta \in (0, 1]$, and $\alpha_t = \alpha$ for some $\alpha > 0$. From Proposition 2.3 and using the same notation as (2.2), this corresponds to the following coordinate-wise update rule:

$$\Delta_t = -\alpha \frac{\sum_{s=1}^{t} \beta^{t-s} v_s}{\sqrt{\sum_{s=1}^{t} (\beta^{t-s} v_s)^2}}, \qquad (\beta\text{-FTRL})$$

and if the denominator is zero, we define $\Delta_t = 0$. We call this algorithm β-FTRL. With $\beta = 1$, it exactly recovers scale-free FTRL (2.2) which is shown to be a poor dynamic learner (Jacobsen and Cutkosky, 2022). Therefore, we will focus on $\beta < 1$ in the dynamic regret analysis.

The earlier informal result (Theorem 2.2) is formalized in Theorem 3.1 (for unbounded domain) and Theorem 3.2 (for bounded domain). We provide the simplified versions here, deferring the detailed adaptive version to Theorem B.4.

Theorem 3.1 (Dynamic regret of β-FTRL; unbounded domain). *For a loss sequence $v_{1:T}$, consider β-FTRL with $\beta < 1$ and some constant $\alpha > 0$. Let*

$$M_\beta := \max_{t \in [1,T]} \frac{\left| \sum_{s=1}^{t} \beta^{t-s} v_s \right|}{\sqrt{\sum_{s=1}^{t} (\beta^{t-s} v_s)^2}}.$$

Then, for any comparator sequence $u_{0:T-1}$ such that $|u_t| \leq \alpha M_\beta$ for all t, the dynamic regret $R_T(u_{0:T-1})$ is upper bounded by

$$\mathcal{O}\left(\frac{(\alpha M_\beta^2 + M_\beta P) G}{\sqrt{1-\beta}} + \sqrt{1-\beta} \cdot \alpha M_\beta^2 G T \right).$$

Here, $G := \max_{t \in [1:T]} |v_t|$, and $P := \sum_{t=1}^{T-1} |u_t - u_{t-1}|$ is the path length.

We sketch the proof of Theorem 3.1 in Subsection 3.3.

Theorem 3.1 may not seem straightforward, so let us start with a high level interpretation. First of all, the path length P is a standard complexity measure of the comparator $u_{0:T-1}$ in the literature (Herbster and Warmuth, 2001), which we examine closely in Subsection 3.1. In the context of dynamic online learning, the above bound could be reminiscent of a classical result from (Zinkevich, 2003, Theorem 2): on a domain of diameter D, the dynamic regret of *online gradient descent* (OGD) with learning rate η can be bounded as

$$R_T(u_{0:T-1}) \leq \mathcal{O}\left(\frac{D^2 + DP}{\eta} + \eta G^2 T \right). \quad (3.1)$$

Intuitively, the choice of η balances the two conflicting terms on the RHS, and a similar tradeoff remains as a recurring theme in the dynamic online learning literature (Hall and Willett, 2015; Zhang et al., 2018a; Jacobsen and Cutkosky, 2022). In an analogous manner, the discounting factor β in Theorem 3.1 largely serves the similar purpose of balancing conflicting factors. A rigorous discussion is deferred to Subsection 3.2.

As a complementary result to Theorem 3.1, we also present a dynamic regret bound for the case of *a priori bounded domain*, where the outputs of online learner should lie in a bounded domain $[-D, D]$. In this case, we project the output of β-FTRL to the given domain:

$$\Delta_t = -\text{clip}_D\left(\alpha \frac{\sum_{s=1}^{t} \beta^{t-s} v_s}{\sqrt{\sum_{s=1}^{t} (\beta^{t-s} v_s)^2}}\right), \quad (\beta\text{-FTRL}_D)$$

where $\text{clip}_D(x) := x \min(\frac{D}{|x|}, 1)$. Then, with the same notations of G and P as in Theorem 3.1, we get the following result (see Subsection B.5 for details).

Theorem 3.2 (Dynamic regret of β-FTRL$_D$; bounded domain). *For $D > 0$, consider any comparator sequence $u_{0:T-1}$ such that $|u_t| \le D$ for all t. Then for any loss sequence $v_{1:T}$, β-FTRL$_D$ with $\beta < 1$ and $\alpha = D$ has the dynamic regret $R_T(u_{0:T-1})$ upper bounded by*

$$\mathcal{O}\left(\frac{DG}{\sqrt{1-\beta}} + \frac{GP}{1-\beta} + \sqrt{1-\beta}DGT\right).$$

Compared to Theorem 3.1, the main difference is that the regret bound now holds simultaneously for all the loss sequences $v_{1:T}$ of *arbitrary* size. The price to pay is the requirement of knowing D, and the multiplying factor on the path length P is slightly worse, *i.e.*, $(1-\beta)^{-1/2} \to (1-\beta)^{-1}$. A sneak peek into the details: such a slightly worse factor is due to the projection step breaking the *self-bounding* property of β-FTRL, which says the discounted gradient sum $\sum_{s=1}^{t} \beta^{t-s} v_s$ can be controlled by the maximum update magnitude $\sup |\Delta_t|$ times the empirical variance of gradients, *i.e.*, $\sqrt{\sum_{s=1}^{t} (\beta^{t-s} v_s)^2}$. Interested readers may compare Subsections B.4 and B.5 for the subtleties.

Moving forward, Theorems 3.1 and 3.2 constitute our main results characterizing the dynamic regret of β-FTRL. However, there is still one missing piece. The earlier informal result (Theorem 2.2) states that

β-FTRL is a "good" dynamic online learner.

However, we have never explained which dynamic regret is good. Actually, the "goodness" criterion in dynamic online learning could be a bit subtle, as the typical sublinear-in-T metric in static online learning becomes vacuous. Next, we briefly provide this important background.

3.1. Basics of Dynamic Online Learning

Dynamic online learning is intrinsically challenging. It is well-known that regardless of the algorithm, there exist loss and comparator sequences such that the dynamic regret is at least $\Omega(T)$. This is in stark contrast to static regret bounds in OLO, where the standard minimax optimal rate is the sublinear in T, *e.g.*, $\mathcal{O}(\sqrt{T})$.

To bypass this issue, the typical approach is through *instance adaptivity*. Each combination of the loss and comparator sequences can be associated to a *complexity measure*; the larger it is, the harder regret minimization becomes. Although it is impossible to guarantee sublinear-in-T regret bounds against the hardest problem instance, one can indeed guarantee a regret bound that *depends on* such a complexity measure. From this perspective, the study of dynamic online learning centers around finding suitable complexity measures and designing adaptive algorithms.

Only considering the comparator sequence $\mathbf{u}_{0:T-1}$, the predominant complexity measure is the *path length* $P := \sum_{t=1}^{T-1} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|$ (Zinkevich, 2003), whose 1D special case is considered in Theorem 3.1 and Theorem 3.2. On a bounded domain with diameter D, the optimal dynamic regret bound is $\mathcal{O}(G\sqrt{DPT})$, which can be achieved through the classical result for OGD (3.1) with the P-*dependent* learning rate $\eta = G^{-1}\sqrt{DP/T}$. On top of that, one could use a model selection approach (Zhang et al., 2018a) to avoid the infeasible *oracle tuning* (*i.e.*, η depends on the unknown P), at the expense of increased computation. Recent works (Jacobsen and Cutkosky, 2022; Zhang et al., 2023) further extend such results to unbounded domains.

The essential "goodness" of this $\mathcal{O}(G\sqrt{DPT})$ bound is due to $P \le DT$. In the worst case the bound is trivially $\mathcal{O}(DGT)$, but if the comparator is *easy* (*i.e.*, $P = \mathcal{O}(D)$), then it becomes $\mathcal{O}(DG\sqrt{T})$, recovering the well-known optimal static regret bound. In general, the goodness of a dynamic regret bound is usually measured by the exponents of both P and T (*e.g.*, $\frac{1}{2}$ and $\frac{1}{2}$ in $\mathcal{O}(G\sqrt{DPT})$).

Given this background, we now use the dynamic regret results of β-FTRL so far to interpret the role of two key components of Adam, namely the **momentum** (*i.e.*, aggregating past gradients) and the **discounting factor** β (*i.e.*, exponential moving average).

3.2. Benefits of Momentum and Discounting Factor

Recall that a particular strength of the OLU framework is that it establishes **one-to-one correspondence** between optimizers and their online learning counterparts. Thus we can compare a variety of optimizers by **comparing their corresponding online learners**, from the perspective of dynamic regret. Notice that β-FTRL from our analysis corresponds to the scaled parameterization of (Adam). Through that, our ultimate goal is to shed light on Adam's algorithmic components — the momentum and the discounting factor.

We first discuss the baseline online learners for this problem:

- To understand the momentum, we pick the baselines as a family of "degenerate" online learners that induce *non-momentum optimizers*, such as SGD and AdaGrad (Duchi et al., 2011). Concretely, for the loss sequence $\mathbf{v}_{1:T}$, this family of LEARNER in OLU has the following generic update rule: for some coordinate-wise learning rate $\boldsymbol{\alpha}_t[i] > 0 \; \forall i$, it outputs

$$\boldsymbol{\Delta}_t[i] = -\boldsymbol{\alpha}_t[i]\mathbf{v}_t[i]\,. \tag{3.2}$$

For example, given a scalar $\alpha_t > 0$, SGD chooses the coordinate-wise learning rate $\boldsymbol{\alpha}_t[i]$ independently of the coordinates, *i.e.*,

$$\boldsymbol{\alpha}_t[i] = \alpha_t\,, \tag{SGD}$$

while AdaGrad further employs a variance-based preconditioning, *i.e.*,

$$\boldsymbol{\alpha}_t[i] = \frac{\alpha_t}{\sqrt{\sum_{s=1}^{t} \mathbf{v}_s[i]^2}}\,. \tag{AdaGrad}$$

The important observation is that compared to the (FTRL) update rule, **the coordinate-wise update $\Delta_t[i]$ in (3.2) only scales linearly with the most recent observation** $\mathbf{v}_t[i]$, instead of using the entire history $\mathbf{v}_{1:t}[i]$. In other words, from the optimization perspective, this family of algorithms does not make use of the past history of gradients to decide the update direction.

- To understand the discounting factor, we pick the baseline as β-FTRL with $\beta = 1$ (*i.e.*, no discounting). Alternatively, if the domain is bounded, then we use the clipped version of β-FTRL$_D$ with $\beta = 1$ instead. In other words, such baselines correspond to scale-free FTRL (2.2). In light of Jacobsen and Cutkosky (2022), the case of $\beta = 1$ is not a good dynamic online learner, which we discuss more formally below.

The following lower bound, inspired by Jacobsen and Cutkosky (2022), shows that the above baselines fail to achieve sublinear dynamic regret for a very benign example of $P = \mathcal{O}(1)$. See Subsection C.1 for a proof.

Theorem 3.3 (Lower bounds for baselines)**.** *Consider a 2D online linear optimization problem with the bounded domain $[-1, 1]^2$. For any given T, there exist (i) a loss sequence $\mathbf{v}_1, \ldots, \mathbf{v}_T \in \mathbb{R}^2$ with $\|\mathbf{v}_t\| = 1$ for all t, and (ii) a comparator sequence $\mathbf{u}_0, \ldots, \mathbf{u}_{T-1} \in [-1, 1]^2$ with the coordinate-wise path length $\sum_{t=1}^{T-1} |\mathbf{u}_t[i] - \mathbf{u}_{t-1}[i]| \le 1$ for both $i = 1, 2$, such that the following holds:*

- *For all t, $\mathbf{u}_{t-1} \in \arg\min_{\mathbf{u}\in[-1,1]^2} \langle \mathbf{v}_t, \mathbf{u}\rangle$.*
- *Any "non-momentum" online learner of the form (3.2) has the dynamic regret at least $T - 3$.*
- *β-FTRL$_D$ with $\beta = 1$ and $D = 1$ has the dynamic regret at least $(T - 3)/2$.*

The first bullet point says that the constructed comparator sequence $\mathbf{u}_{0:T-1}$ is the best ones *w.r.t.* the loss sequence $\mathbf{v}_{1:T}$, therefore the dynamic regret against such $\mathbf{u}_{0:T-1}$ is a good metric to measure the strength of dynamic online learners. Then, the rest of the theorem shows that the two baselines above (corresponding to optimizers without momentum or discounting factor under OLU) cannot guarantee low regret against $\mathbf{u}_{0:T-1}$, thus are not good dynamic online learners.

In contrast, Theorem 3.2 shows that β-FTRL$_D$ with $\beta < 1$ is a better dynamic online learner, and we make this very concrete through the following corollary. Again, it suffices to consider the 1D setting.

Corollary 3.4. *For $D > 0$, consider any comparator sequence $u_{0:T-1}$ such that $|u_t| \le D$ for all t. Then, given any constant $c > 0$, β-FTRL$_D$ with $\beta = 1 - cT^{-2/3} > 0$ achieves the dynamic regret bound*

$$R_T(u_{0:T-1}) \le \mathcal{O}\left(DGT^{2/3}c^{1/2}(1 + c^{-3/2}P/D)\right),$$

which enjoys a $T^{2/3}$ dependency on T. In particular, with the optimal tuning $c = \Theta\left((P/D)^{2/3}\right)$, the bound becomes $\mathcal{O}(D^{2/3}GP^{1/3}T^{2/3})$.

The proof is deferred to Subsection C.2. We emphasize that in the lower bound example of Theorem 3.3, we have $P = \mathcal{O}(D)$, so the $\beta < 1$ case achieves a sublinear dynamic regret bound of $\mathcal{O}(DGT^{2/3})$. This suggests that in order to design a better dynamic online learner **both momentum and discounting factor are necessary**.

A similar result can be developed for the case of unbounded domain under an assumption regarding the 1D OLO environment that generates the losses $v_{1:T}$.

Corollary 3.5. *Assume the environment is well-behaved in the sense that $M_\beta \le M$ for all $\beta \in (0, 1]$. Consider any comparator sequence $u_{0:T-1}$ such that $|u_t| \le \alpha M$ for all t. Then, given any constant $c > 0$, β-FTRL with parameters α and $\beta = 1 - cT^{-1} > 0$ achieves the dynamic regret bound*

$$R_T(u_{0:T-1}) \le \mathcal{O}\left(\alpha M^2 G\sqrt{T}c^{1/2}\left(1 + \frac{c^{-1}P}{\alpha M}\right)\right),$$

which enjoys a $\sqrt{T}$ dependency on T. In particular, with the optimal tuning $c = \Theta\big(P/(\alpha M)\big)$, the bound becomes $\mathcal{O}\big(\alpha^{1/2}M^{3/2}G\sqrt{PT}\big)$.

In contrast, β-FTRL with the same α but the different $\beta = 1$ achieves a dynamic regret bound which is linear in P:

$$R_T(u_{0:T-1}) \le \mathcal{O}\left(MGP\sqrt{T}\right).$$

Essentially, without the discounting factor we can show an $\mathcal{O}\big(P\sqrt{T}\big)$ dynamic regret bound, but with discounting the bound can be improved to the optimal rate $\mathcal{O}\big(\sqrt{PT}\big)$, under suitable tuning. This provides another evidence that discounting is helpful for designing a dynamic online learner. We remark that without discounting, the dynamic regret of $\mathcal{O}\big(P\sqrt{T}\big)$ is unimprovable in light of the lower bound result (Jacobsen and Cutkosky, 2022, Theorem 3).

3.3. Proof Sketch of Theorem 3.1

Finally, we briefly sketch the proof of Theorem 3.1, the dynamic regret bound of β-FTRL. The proof of Theorem 3.2 mostly follows the same analysis. Our analysis of the dynamic regret relies on the following "discounted" regret.

Definition 3.6 (β-discounted regret)**.** *For any discounting factor $\beta \in (0, 1]$, the β-discounted regret is defined as*

$$R_{T;\beta}(u) \coloneqq \sum_{t=1}^{T} \beta^{T-t} v_t(\Delta_{t-1} - u)\,.$$

It is noteworthy that the discounted regret has been considered in the concurrent works (Zhang et al., 2024; Jacobsen and Cutkosky, 2024) to adapt online learners to dynamic environments. Moreover, the discounted regret has found to be useful in designing nonconvex optimization algorithms, as shown in (Zhang and Cutkosky, 2024; Ahn and Cutkosky, 2024). In our work, we propose a generic conversion approach to analyzing the dynamic regret using the discounted regret, called **discounted-to-dynamic conversion** (Theorem B.3), which could be of independent interest. At a high level, our analysis follows the following steps.

$$\underbrace{\text{(Theorem B.2)}}_{\text{Discounted reg. of } \beta\text{-FTRL}} \xRightarrow[\text{Discount-to-dynamic}]{\text{(Theorem B.3)}} \underbrace{\text{(Theorem 3.1)}}_{\text{Dynamic reg. of } \beta\text{-FTRL}}$$

1. β-FTRL is the discounted version of scale-free FTRL, and the latter is associated to a static regret bound (Theorem A.1). Utilizing this relation, we naturally arrive at a discounted regret bound of β-FTRL (Theorem B.2). Intuitively, it measures the performance of β-FTRL on an exponentially weighted, "local" look-back window that ends at time T. A particular strength is that the bound is *anytime*, *i.e.*, it holds for all T simultaneously.
2. Next, consider the dynamic regret over the entire time horizon $[1, T]$. We can imagine partitioning $[1, T]$ into concatenating subintervals, and on each of them the dynamic regret can be approximately upper-bounded by suitable *aggregations* of the above discounted regret bound (modulo certain approximation error) — this is because the discounted regret bound mostly concerns the few recent rounds, so the dynamic regret can be controlled by the sum of such local metrics. Formalizing this argument results in the discounted-to-dynamic conversion in Theorem B.3: the dynamic regret of an algorithm is expressed using its discounted regret as an equality.

Both Theorem 3.1 and 3.2 are obtained by combining these two elements, with slightly different ways of relaxation.

4. Implications for Optimization

In this section, we provides the details of Theorem 2.1, which justifies the importance of the dynamic regret guarantee of LEARNER in OLU, based on its implications for (non-convex) optimization.

Recall that in OLU, for the function F we want to minimize, we choose the update $\mathbf{\Delta}_t$ by the output of LEARNER on the loss sequence $\mathbf{g}_{1:t}$ that are stochastic gradients of F. Formally, we assume $F : \mathbb{R}^d \to \mathbb{R}$ is differentiable but not necessarily convex. Following the notations of Cutkosky et al. (2023), given an iterate $\mathbf{w}$ and a random variable z, let $\mathrm{GRAD}(\mathbf{w}, z)$ be the standard *stochastic gradient oracle* of F at $\mathbf{w}$, satisfying $\mathbb{E}_z[\mathrm{GRAD}(\mathbf{w}, z)] = \nabla F(\mathbf{w})$.

We first introduce (Cutkosky et al., 2023, Theorem 7) that crucially connects the dynamic regret guarantee to the optimization guarantee, justifying the importance of the dynamic regret of LEARNER.

Theorem 4.1 (**Importance of dynamic regret in OLU**)**.** *Consider the optimization algorithm* (1.1) *where*

- *the update $\mathbf{\Delta}_t$ is chosen by* LEARNER *based on $\mathbf{g}_{1:t}$;*
- *the gradient $\mathbf{g}_t = \mathrm{GRAD}(\mathbf{w}_{t-1} + s_t\mathbf{\Delta}_t, z_t)$, where the i.i.d. samples $s_t \sim$ Unif$([0, 1])$ and $z_t \sim z$.*

Then, for all $T \geq 0$ and any comparator sequence $\mathbf{u}_{0:T-1}$, the iterate $\mathbf{w}_T$ generated by (1.1) *satisfies*

$$\mathbb{E}\left[F(\mathbf{w}_T)\right] - F(\mathbf{w}_0) = \mathbb{E}\left[\sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{u}_{t-1}\rangle + R_T(\mathbf{u}_{0:T-1})\right],$$

where $R_T(\mathbf{u}_{0:T-1})$ is the regret bound of LEARNER.

The proof is provided in Subsection D.1. The insight is that the nonconvexity can be handled by randomization (through s_t) at the gradient query, and if F is convex, it suffices to set $s_t = 1$ (which is more aligned with practice). Regarding the quantitative result, Theorem 4.1 precisely captures the informal claim from Theorem 2.1:

Improving the dynamic regret of LEARNER *directly leads to better optimization guarantee.*

Hence, the effectiveness of β-FTRL as a dynamic online learner (discussed in Section 3) supports the success of its optimizer counterpart, namely the update (Adam).

To make this concrete, we revisit the lower bound examples from Theorem 3.3, and discuss the implications.

4.1. Revisiting lower bound example

Consider an abstract scenario where we fix the stochastic gradient sequence $\mathbf{g}_{1:T}$ to be given the lower bound example of Theorem 3.3. We compare the guarantees in Theorem 4.1,

$$\sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{u}_{t-1} \rangle + R_T(\mathbf{u}_{0:T-1}) . \tag{4.1}$$

Then, the following is a direct corollary of Subsection 3.2.

Corollary 4.2. *Suppose $\mathbf{g}_{1:T}$ and $\mathbf{u}_{0:T-1}$ are chosen as in the 2D example of Theorem 3.3. Then, the following statements hold:*

- ***Adam.*** *If* LEARNER *is β-FTRL$_D$ with $1-\beta = \Theta(T^{-2/3})$ and $D = 1$, then (4.1) $= -T + o(T)$.*
- ***No momentum (*e.g.,* SGD/AdaGrad).*** *If* LEARNER *has the form (3.2), then (4.1) ≥ -3.*
- ***No discounting.*** *If* LEARNER *is β-FTRL$_D$ with $\beta = 1$ and $D = 1$, then (4.1) $\geq -\frac{1}{2}T - \frac{3}{2}$.*

Essentially, this result specializes the key insight from Subsection 3.2, *i.e.*, the benefits of the momentum and the discounting factor, to the corresponding optimizers.

On the other hand, we acknowledge that the abstract scenario of fixing $\mathbf{g}_{1:T}$ to be some desired sequence is not entirely practical, because (i) they should be stochastic gradients of F and (ii) $\mathbf{g}_{1:T}$ depends on the LEARNER of choice. Below, we build on the intuitions of this abstract example and present a concrete classification problem where Adam is more beneficial than the other two baselines.

4.2. Adam Could Be Effective for Sparse and Nonstationary Gradients

Inspired by Duchi et al. (2011), we propose a concrete classification scenario for which we see the performance gap described in Corollary 4.2. At a high level, it is designed such that the associated gradient sequence imitates the one from our lower bound construction, Theorem 3.3.

Classification of sparse data with small η. Consider the classification of $(\mathbf{z}_i, y_i)$ where $\mathbf{z}_i \in \mathbb{R}^d$ is the data vector with its label $y_i \in \{\pm 1\}$. We assume that each data vector $\mathbf{z}_i$ is **sparse**. Concretely, we focus on the setting where the dataset consists of coordinate vectors and positive labels, *i.e.*, $\{(\mathbf{z}_i, y_i)\}_{i=1}^d$ where $\mathbf{z}_i = \mathbf{e}_i$ and $y_i = 1$ for $i = 1, \ldots, d$. We consider the following regularized hinge loss

$$\ell(x) = \max(0, 1-x) + \lambda |x| \quad \text{for } \lambda < 1 ,$$

which prevents the classifier from becoming over-confident. See Figure 1 for the landscape of this hinge loss. Then, the training loss is given as

$$F(\mathbf{w}) = \frac{1}{d} \sum_{i=1}^{d} \ell(y_i \langle \mathbf{z}_i, \mathbf{w} \rangle) ,$$

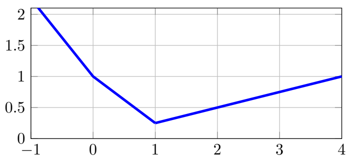

Figure 1. 1D illustration of the regularized hinge loss $\ell(x) = \max(0, 1-x) + \lambda|x|$. We illustrate the case $\lambda = 1/4$.

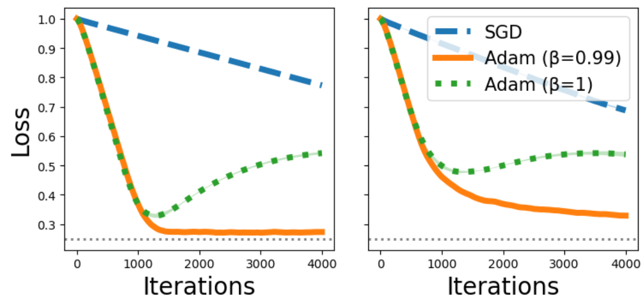

Figure 2. Experimental results for the hinge loss classification. (**Left**) the case of $\mathbf{z}_i = \mathbf{e}_i$. (**Right**) the case of $\mathbf{z}_i = c_i \mathbf{e}_i$ where $c_i \sim \text{Unif}[0, 2]$. The horizontal dotted line indicates the optimum value of F. All experiments are run for five different random seeds, and we plot the error shades (they are quite small and not conspicuous).

and during iteration t, assume the algorithm receives a single data $(\mathbf{z}_{i(t)}, y_{i(t)})$, where $i(t)$ is sampled from $\{1, \ldots, d\}$ uniformly at random.

For experiments, we initialize at $\mathbf{w}_0 = \mathbf{0}$ and use **a small learning rate**, $\eta = 0.01$, so that each coordinate takes multiple steps to approach the minimum $w = 1$. Besides, we choose $d = 100$ and $\lambda = 1/4$. Two different settings are considered:

1. **Left plot of Figure 2:** $\mathbf{z}_i = \mathbf{e}_i$ for $i = 1, \ldots, d$.
2. **Right plot of Figure 2:** $\mathbf{z}_i = c_i \mathbf{e}_i$, where $c_i \sim \text{Unif}[0, 2]$ for $i = 1, \ldots, d$. This setting allows the data vectors to have different magnitudes.

From Figure 2, SGD exhibits sluggish progress owing to the sparse nature of stochastic gradients——that is, only one coordinate is updated at each step. Moreover, setting $\beta = 1$ in Adam also results in suboptimal performance once the coordinate-wise iterate $\mathbf{w}_t[i]$ exceeds 1 (for some coordinate i)—after that, the stochastic gradients point toward other directions. In contrast, adopting Adam with $\beta < 1$ effectively addresses these issues, adeptly managing **both the sparsity of updates and the non-stationarity of gradients**.

Next, we provide a possible qualitative explanation of this gap, from the dynamic regret perspective.

Qualitative dynamic regret analysis. Since $i(t)$ is sampled uniformly, it suffices to focus on the first coordinate and consider the 1D setting for simplicity. Then, since **learning rate η is chosen small**, starting from $w_0 = 0$, the above setting could be abstractly thought as generating the following sparse stochastic gradient sequence

$$g_t = \begin{cases} (1-\lambda) \cdot \mathbb{I}_{[i(t)=1]} & \text{if } t \lesssim \tau \\ -\lambda \cdot \mathbb{I}_{[i(t)=1]} & \text{if } t \gtrsim \tau \end{cases}, \tag{4.2}$$

where τ denotes the first iteration such that $w_\tau > 1$. This can be seen as one of the simplest setting of a **sparse** and **non-stationary** gradient sequence, mirroring the construction from Theorem 3.3. Now we compare the 1D version of the guarantee (4.1), *i.e.*, the *total loss* of the LEARNER

$$\sum_{t=1}^{T} g_t u_{t-1} + R_T(u_{0:T-1}) = \sum_{t=1}^{T} g_t \Delta_{t-1} \tag{4.3}$$

for each LEARNER akin to Corollary 4.2:

- **No momentum.** Due to the sparsity in gradient sequence (4.2), having no momentum incurs a large dynamic regret. More specifically, since $\mathbb{E}\,|g_t g_{t-1}| \lesssim \frac{1}{d^2}$, we have

  $$\mathbb{E}\,|g_t \Delta_{t-1}| \lesssim \frac{1}{d^2},$$

  for "non-momentum" LEARNER of the form (3.2). Therefore, we have (4.3) $= \sum_t g_t \Delta_{t-1} \gtrsim -\frac{1}{d^2} T$.
- **No discounting factor.** Due to the non-stationarity in gradient sequence (4.2), having no discounting factor also incurs a large dynamic regret. In particular, β-FTRL with $\beta = 1$ generates the update Δ_t with the same sign as $-\sum_t g_t$. In a typical run, the sign of $-\sum_t g_t$ remains unchanged throughout, but the sign of g_t flips once $t \gtrsim \tau$. Hence, roughly speaking, the update Δ_t does not have the "correct" sign (corresponding to the *descent* direction) after $t \gtrsim \tau$, leading to (4.3) $\gtrsim -(1-\lambda)\tau$.

In contrast, following Corollary 4.2, Adam achieves (4.3) $\leq -(1-\lambda)\tau - \lambda(T-\tau) + o(T) = -\lambda T - (1-2\lambda)\tau + o(T)$, improving the above when τ is small.

5. Conclusion and Discussion

This work presents a new perspective on the popular Adam optimizer, based on the framework of online learning of updates (OLU) (Cutkosky et al., 2023). Under OLU, our main observation is that Adam corresponds to choosing the dynamic version of FTRL that utilizes the discounting factor. We find this perspective quite advantageous as it gives new insights into the role of Adam's algorithmic components, such as momentum and the exponential moving average.

In fact, our perspective has already inspired a follow-up work by Ahn and Cutkosky (2024), where they show the optimal iteration complexity of Adam for finding stationary points under nonconvex and nonsmooth functions. Their analysis crucially utilizes our perspective that Adam corresponds to β-FTRL under OLU.

In addition to (Ahn and Cutkosky, 2024), the findings in this work unlock several other important future directions. Below, we list a few of them.

- **Role of two discounting factors.** As an initial effort, this work considers the case of $\beta_1 = \beta_2$. Given that the default choice in practice is $\beta_1 = 0.9$ and $\beta_2 = 0.999$, it would be important to understand the precise effect of choosing $\beta_1 < \beta_2$.
- **Other algorithms based on OLU.** The framework OLU unlocks a new way to analyze optimization algorithms. As we highlighted in Subsection 3.2, OLU establishes the *one-to-one correspondence* between other optimizers and their online learning counterparts. The main scope of this work is to provide a better understanding of Adam specifically, and extending our framework to other popular optimizers, such as RMSProp (Tieleman and Hinton, 2012), AdaDelta (Zeiler, 2012), Lion (Chen et al., 2023) etc, is an interesting future direction. We believe that understand them based on our framework would offer new insights for them.
- **Algorithm design based on OLU.** Our current dynamic regret analysis of β-FTRL requires knowledge of the environment. Developing a version of β-FTRL that automatically adapts to the environment without prior knowledge might lead to more practical algorithms. Moreover, whether one can design practical optimizers based on recent advancements in dynamic online learning (*e.g.* Jacobsen and Cutkosky (2022); Zhang et al. (2023)) would be an important future direction.
- **Fine-grained analysis of Adam for practical settings.** As discussed earlier, Adam has gained significant attention due to its effectiveness in training language models. Recently, Kunstner et al. (2024) investigate key characteristics of the language modeling datasets that might have caused the difficulties in training. In particular, they identify the *heavy-tailed imbalance* property, where there are a lot more infrequent words/tokens than frequent ones in most language modeling datasets. Further, they demonstrate this property as a main reason why Adam is particularly effective at language modeling tasks (Zhang et al., 2020b). We find their main insights consistent with our claim in Subsection 4.2. The infrequent words in the dataset would likely lead to sparse and non-stationary gradients. Formally investigating this would be also an interesting future direction.

Acknowledgements

Kwangjun Ahn is indebted to Ashok Cutkosky for several inspiring conversations that led to this project.

Kwangjun Ahn was supported by the ONR grant (N00014-23-1-2299), MIT-IBM Watson, a Vannevar Bush fellowship from Office of the Secretary of Defense, and NSF CAREER award (1846088). Yunbum Kook was supported in part by NSF awards CCF-2007443 and CCF-2134105. Zhiyu Zhang was supported by the funding from Heng Yang.

Impact Statement

This paper provides a new perspective of understanding the Adam optimizer. This work is theoretical, and we do not see any immediate potential societal consequences.

References

Jacob D. Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Conference on Learning Theory (COLT)*, pages 263–274. Omnipress, 2008.

Kwangjun Ahn and Ashok Cutkosky. Adam with model exponential moving average is effective for nonconvex optimization, 2024.

Kwangjun Ahn, Xiang Cheng, Minhak Song, Chulhee Yun, Ali Jadbabaie, and Suvrit Sra. Linear attention is (maybe) all you need (to understand Transformer optimization). In *International Conference on Learning Representations (ICLR)*, 2024.

Ahmet Alacaoglu, Yura Malitsky, Panayotis Mertikopoulos, and Volkan Cevher. A new regret analysis for Adam-type algorithms. In *International Conference on Machine Learning (ICML)*, pages 202–210. PMLR, 2020.

Leon Bottou. Online learning and stochastic approximations. *On-Line Learning in Neural Networks*, 17(9):142, 1998.

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *Information Theory, IEEE Transactions on*, 50 (9):2050–2057, 2004.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of A class of Adam-type algorithms for non-convex optimization. In *International Conference on Learning Representations (ICLR)*, 2019.

Michael Crawshaw, Mingrui Liu, Francesco Orabona, Wei Zhang, and Zhenxun Zhuang. Robustness to unbounded smoothness of generalized SignSGD. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 9955–9968, 2022.

Ashok Cutkosky. Artificial constraints and hints for unbounded online learning. In *Conference on Learning Theory (COLT)*, pages 874–894, 2019.

Ashok Cutkosky, Harsh Mehta, and Francesco Orabona. Optimal stochastic non-smooth non-convex optimization through online-to-non-convex conversion. In *International Conference on Machine Learning (ICML)*, volume 202, pages 6643–6670. PMLR, 2023.

Amit Daniely, Alon Gonen, and Shai Shalev-Shwartz. Strongly adaptive online learning. In *International Conference on Machine Learning (ICML)*, pages 1405–1411. PMLR, 2015.

Alexandre Défossez, Leon Bottou, Francis Bach, and Nicolas Usunier. A simple convergence proof of Adam and AdaGrad. *Transactions on Machine Learning Research (TMLR)*, 2022.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research (JMLR)*, 12(61):2121–2159, 2011.

Geoffrey J. Gordon. Regret bounds for prediction problems. In *Conference on Learning Theory (COLT)*, pages 29–40. ACM, 1999.

Zhishuai Guo, Yi Xu, Wotao Yin, Rong Jin, and Tianbao Yang. A novel convergence analysis for algorithms of the Adam family. *arXiv preprint arXiv:2112.03459*, 2021.

Eric C Hall and Rebecca M Willett. Online convex optimization in dynamic environments. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):647–662, 2015.

Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. *Machine Learning*, 80:165–188, 2010.

Mark Herbster and Manfred K Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research (JMLR)*, 1(281-309):10–1162, 2001.

Andrew Jacobsen and Ashok Cutkosky. Parameter-free mirror descent. In *Conference on Learning Theory (COLT)*, pages 4160–4211. PMLR, 2022.

Andrew Jacobsen and Ashok Cutkosky. Online linear regression in dynamic environments via discounting. In *International Conference on Machine Learning*. PMLR, 2024.

Kaiqi Jiang, Dhruv Malik, and Yuanzhi Li. How does adaptive optimization impact local neural network geometry? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2014.

Frederik Kunstner, Jacques Chen, Jonathan Wilder Lavington, and Mark Schmidt. Noise is not the main factor behind the gap between SGD and Adam on Transformers, but sign descent might be. In *International Conference on Learning Representations (ICLR)*, 2023.

Frederik Kunstner, Robin Yadav, Alan Milligan, Mark Schmidt, and Alberto Bietti. Heavy-tailed class imbalance and why adam outperforms gradient descent on language models. *arXiv preprint arXiv:2402.19449*, 2024.

Haochuan Li, Alexander Rakhlin, and Ali Jadbabaie. Convergence of Adam under relaxed assumptions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Xiaoyu Li and Francesco Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 983–992. PMLR, 2019.

H Brendan McMahan. A survey of algorithms and analysis for adaptive online learning. *The Journal of Machine Learning Research (JMLR)*, 18(1):3117–3166, 2017.

H. Brendan McMahan and Matthew J. Streeter. Adaptive bound optimization for online convex optimization. In *Conference on Learning Theory (COLT)*, pages 244–256. Omnipress, 2010.

Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, 2009.

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

Francesco Orabona and Dávid Pál. Scale-free online learning. *Theoretical Computer Science*, 716:50–69, 2018.

Yan Pan and Yuanzhi Li. Toward understanding why Adam converges faster than SGD for Transformers. *arXiv preprint arXiv:2306.00204*, 2023.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. In *International Conference on Learning Representations (ICLR)*, 2018.

Shai Shalev-Shwartz and Yoram Singer. Online learning meets optimization in the dual. In *Conference on Learning Theory (COLT)*, volume 4005, pages 423–437. Springer, 2006.

Tijmen Tieleman and Geoffrey Hinton. RMSProp: Divide the gradient by a running average of its recent magnitude. *COURSERA Neural Networks Mach. Learn*, 17, 2012.

Tim van Erven. Why FTRL is better than Online Mirror Descent. *Personal Blog Post (Mathematics of Machine learning)*, 2021. URL https://www.timvanerven.nl/blog/ftrl-vs-omd/#fn:unbounded.

Bohan Wang, Jingwen Fu, Huishuai Zhang, Nanning Zheng, and Wei Chen. Closing the gap between the upper bound and lower bound of Adam's iteration complexity. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Rachel Ward, Xiaoxia Wu, and Leon Bottou. AdaGrad stepsizes: Sharp convergence over nonconvex landscapes. In *International Conference on Machine Learning (ICML)*, pages 6677–6686. PMLR, 2019.

Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations (ICLR)*, 2020a.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 15383–15393, 2020b.

Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive online learning in dynamic environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1330–1340, 2018a.

Lijun Zhang, Tianbao Yang, and Zhi-Hua Zhou. Dynamic regret of strongly adaptive methods. In *International Conference on Machine Learning (ICML)*, pages 5882–5891. PMLR, 2018b.

Qinzi Zhang and Ashok Cutkosky. Random scaling and momentum for non-smooth non-convex optimization. In *International Conference on Machine Learning*. PMLR, 2024.

Yushun Zhang, Congliang Chen, Naichen Shi, Ruoyu Sun, and Zhi-Quan Luo. Adam can converge without any modification on update rules. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 28386–28399, 2022.

Zhiyu Zhang, Ashok Cutkosky, and Ioannis Ch Paschalidis. Unconstrained dynamic regret via sparse coding. *arXiv preprint arXiv:2301.13349*, 2023.

Zhiyu Zhang, David Bombara, and Heng Yang. Discounted adaptive online prediction. In *International Conference on Machine Learning (ICML)*. PMLR, 2024.

Shuai Zheng and James T Kwok. Follow the moving leader in deep learning. In *International Conference on Machine Learning (ICML)*, pages 4110–4119. PMLR, 2017.

Zhiming Zhou, Qingru Zhang, Guansong Lu, Hongwei Wang, Weinan Zhang, and Yong Yu. Adashift: Decorrelation and convergence of adaptive learning rate methods. In *International Conference on Learning Representations (ICLR)*, 2019.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML)*, pages 928–936, 2003.

Fangyu Zou, Li Shen, Zequn Jie, Weizhong Zhang, and Wei Liu. A sufficient condition for convergences of Adam and RMSProp. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11127–11135, 2019.

A. Analysis of scale-free FTRL

Recall from Section 2 that our construction starts with a gradient adaptive FTRL algorithm called *scale-free FTRL* (Orabona and Pál, 2018). This section presents a self-contained proof of its undiscounted static regret bound.

Formally, we consider the 1D OLO problem introduced at the beginning of Section 2. Scale-free FTRL is defined as FTRL with the step size $\eta_t = \alpha/\sqrt{\sum_{s=1}^t v_s^2}$, where $\alpha > 0$ is a scaling factor. Equivalently, it has the update rule

$$\Delta_t = \arg\min_x \left[\frac{1}{\eta_t}|x|^2 + \sum_{s=1}^t v_s x\right] = -\eta_t \sum_{s=1}^t v_s = -\alpha \frac{\sum_{s=1}^t v_s}{\sqrt{\sum_{s=1}^t v_s^2}}.$$

For well-posedness, if the denominator $\sqrt{\sum_{s=1}^t v_s^2} = 0$, then we set the update to be $\Delta_t = 0$.

Theorem A.1 (Static regret of scale-free FTRL)**.** *For all $T > 0$, loss sequence $v_{1:T}$ and comparator $u \in \mathbb{R}$, scale-free FTRL guarantees the following static regret bound*

$$\sum_{t=1}^T v_t(\Delta_{t-1} - u) \le \left(\frac{u^2}{2\alpha} + \sqrt{2}\alpha\right)\sqrt{\sum_{t=1}^T v_t^2} + 2\left(\max_{t\in[1,T]} |\Delta_t|\right)\left(\max_{t\in[1,T]} |v_t|\right).$$

We remark that van Erven (2021) directly applies the clipping technique from (Cutkosky, 2019) to obtain a similar regret bound as Theorem A.1, but in this way the associated algorithm is *scale-free FTRL on the "clipped" gradients*, rather than scale-FTRL itself. In contrast, we analyze the original scale-free FTRL algorithm for the purpose of explaining Adam (since in practice, Adam does not use the *Cutkosky-style clipping* on the stochastic gradients). This requires a slightly more involved analysis.

Comparison with (Orabona and Pál, 2018). Before proving this theorem, we compare our regret bound with that of scale-free FTRL from (Orabona and Pál, 2018, Theorem 1). Their regret bound in the unconstrained domain setting (which means the domain diameter D defined in their Theorem 1 is infinite) is

$$\sum_{t=1}^T v_t(\Delta_{t-1} - u) \le \mathcal{O}\left((u^2+1)\sqrt{\sum_{t=1}^T v_t^2} + \sqrt{T}\max_{t\in[1,T]} |v_t|\right).$$

Our bound replaces the $\sqrt{T}$-factor by the maximum output magnitude (*i.e.*, $\max_{t\in[1,T]} |\Delta_t|$), and our is better since

$$|\Delta_t| = \alpha \frac{\left|\sum_{s=1}^t v_s\right|}{\sqrt{\sum_{s=1}^t v_s^2}} \le \alpha\sqrt{t},$$

which follows from the Cauchy-Schwarz inequality. We need such an improvement because in the discounted setting, the scaled loss sequence will have rapidly growing magnitude, which means this Cauchy-Schwarz step would be quite loose.

Our proof makes a nontrivial use of the gradient clipping technique from (Cutkosky, 2019), which is also different from (Orabona and Pál, 2018, Theorem 1) and could be of independent interest. However, we acknowledge that directly modifying the argument of (Orabona and Pál, 2018) might achieve a similar goal.

A.1. Proof of Theorem A.1

On the high level, the proof carefully combines the standard FTRL analysis, *e.g.*, (Orabona, 2019, Lemma 7.1), and the gradient clipping technique of (Cutkosky, 2019).

Step 1. We start with a preparatory step. Let τ be the index such that $v_t = 0$ for all $t \le \tau$, and $v_{\tau+1} \neq 0$. Without loss of generality, assume $T > \tau$. Then,

$$\sum_{t=1}^T v_t(\Delta_{t-1} - u) = \sum_{t=\tau+1}^T v_t(\Delta_{t-1} - u).$$

On the RHS we have $\Delta_\tau = 0$, and for all $t > \tau$, Δ_t is now well-defined by the "nice" gradient adaptive update rule (*i.e.*, the denominator does not cause a problem)

$$\Delta_t = -\eta_t \sum_{s=1}^{t} v_s = -\alpha \frac{\sum_{s=1}^t v_s}{\sqrt{\sum_{s=1}^t v_s^2}} .$$

To proceed, for all $t > \tau$, we define $F_t(x) = \frac{1}{2\eta_t}|x|^2 + \sum_{s=1}^t v_s x$, which means that $\Delta_t = \arg\min_x F_t(x)$. Trivially, at the time index τ, we define $F_\tau(x) = 0$ for all x.

Step 2. The main part of the proof starts from the standard FTRL equality (Orabona, 2019, Lemma 7.1),

$$\begin{aligned}
\sum_{t=1}^{T} v_t(\Delta_{t-1} - u) &= \sum_{t=1}^{T} v_t \Delta_{t-1} - \left(F_T(u) - \frac{1}{\eta_T}|u|^2\right) \\
&= \frac{1}{\eta_T}|u|^2 + \sum_{t=\tau}^{T-1} \left[F_t(\Delta_t) - F_{t+1}(\Delta_{t+1}) + v_{t+1}\Delta_t\right] + F_T(\Delta_T) - F_T(u) \\
&\le \frac{1}{\eta_T}|u|^2 + \sum_{t=\tau}^{T-1} \left[F_t(\Delta_t) - F_{t+1}(\Delta_{t+1}) + v_{t+1}\Delta_t\right] ,
\end{aligned}$$

where the last inequality follows since $\Delta_T = \arg\min_x F_T(x)$.

Consider the terms $F_t(\Delta_t) - F_{t+1}(\Delta_{t+1}) + v_{t+1}\Delta_t$ in the above sum. Let us define the clipped gradient

$$\widetilde{v}_t := \operatorname{clip}_{\sqrt{\sum_{s=1}^{t-1} v_s^2}}(v_t) ,$$

where for any $D \ge 0$, $\operatorname{clip}_D(x) := x \min(\frac{D}{|x|}, 1)$.

- For all $t > \tau$, we have

$$\begin{aligned}
& F_t(\Delta_t) - F_{t+1}(\Delta_{t+1}) + v_{t+1}\Delta_t \\
&= F_t(\Delta_t) + v_{t+1}\Delta_t - F_t(\Delta_{t+1}) - v_{t+1}\Delta_{t+1} + \frac{1}{2\eta_t}\left|\Delta_{t+1}\right|^2 - \frac{1}{2\eta_{t+1}}\left|\Delta_{t+1}\right|^2 \\
&\le F_t(\Delta_t) + v_{t+1}\Delta_t - F_t(\Delta_{t+1}) - v_{t+1}\Delta_{t+1} \\
&\le F_t(\Delta_t) + \widetilde{v}_{t+1}\Delta_t - F_t(\Delta_{t+1}) - \widetilde{v}_{t+1}\Delta_{t+1} + |v_{t+1} - \widetilde{v}_{t+1}|\left(|\Delta_t| + |\Delta_{t+1}|\right) .
\end{aligned}$$

  Following a standard fact of convex functions, e.g., (Orabona, 2019, Lemma 7.8), since $F_t(\Delta) + \widetilde{v}_{t+1}\Delta$ is $\frac{1}{\eta_t}$-strongly convex, it holds that

$$F_t(\Delta_t) + \widetilde{v}_{t+1}\Delta_t - F_t(\Delta_{t+1}) - \widetilde{v}_{t+1}\Delta_{t+1} \le F_t(\Delta_t) + \widetilde{v}_{t+1}\Delta_t - \min_{\Delta}\left[F_t(\Delta) + \widetilde{v}_{t+1}\Delta\right] \le \frac{\eta_t}{2}\widetilde{v}_{t+1}^2 .$$

- As for the case of $t = \tau$, since $F_\tau(\Delta_\tau) = 0$ and $\Delta_\tau = 0$,

$$\begin{aligned}
F_\tau(\Delta_\tau) - F_{\tau+1}(\Delta_{\tau+1}) + v_{\tau+1}\Delta_\tau = -F_{\tau+1}(\Delta_{\tau+1}) &= -\frac{1}{\eta_{\tau+1}}|\Delta_{\tau+1}|^2 - \sum_{s=1}^{\tau+1} v_s \Delta_{\tau+1} \le -v_{\tau+1}\Delta_{\tau+1} \\
&\le |v_{\tau+1} - \widetilde{v}_{\tau+1}|\,|\Delta_{\tau+1}| . \qquad (\widetilde{v}_{\tau+1} = 0)
\end{aligned}$$

Thus, we obtain the following bound:

$$\sum_{t=1}^{T} v_t(\Delta_{t-1} - u) \le \frac{u^2}{2\alpha}\sqrt{\sum_{t=1}^{T} v_t^2} + \frac{\alpha}{2}\sum_{t=\tau+1}^{T-1} \frac{\widetilde{v}_{t+1}^2}{\sqrt{\sum_{s=1}^t v_s^2}} + \sum_{t=1}^{T} |v_t - \widetilde{v}_t|\left(|\Delta_{t-1}| + |\Delta_t|\right) .$$

Step 3. Finally, consider the two summation terms on the RHS one-by-one. We begin with the first term.

$$\frac{\widetilde{v}_{t+1}^2}{\sqrt{\sum_{s=1}^t v_s^2}} = \sqrt{2}\frac{\widetilde{v}_{t+1}^2}{\sqrt{2\sum_{s=1}^t v_s^2}} \le \sqrt{2}\frac{\widetilde{v}_{t+1}^2}{\sqrt{\widetilde{v}_{t+1}^2 + \sum_{s=1}^t v_s^2}} = 2\sqrt{2}\frac{\widetilde{v}_{t+1}^2}{2\sqrt{\widetilde{v}_{t+1}^2 + \sum_{s=1}^t v_s^2}}$$
$$\le 2\sqrt{2}\frac{\widetilde{v}_{t+1}^2}{\sqrt{\widetilde{v}_{t+1}^2 + \sum_{s=1}^t v_s^2} + \sqrt{\sum_{s=1}^t v_s^2}} = 2\sqrt{2}\left(\sqrt{\widetilde{v}_{t+1}^2 + \sum_{s=1}^t v_s^2} - \sqrt{\sum_{s=1}^t v_s^2}\right).$$

Thus, it follows that

$$\sum_{t=\tau+1}^{T-1} \frac{\widetilde{v}_{t+1}^2}{\sqrt{\sum_{s=1}^t v_s^2}} \le 2\sqrt{2}\sum_{t=\tau+1}^{T-1}\left(\sqrt{\widetilde{v}_{t+1}^2 + \sum_{s=1}^t v_s^2} - \sqrt{\sum_{s=1}^t v_s^2}\right) \le 2\sqrt{2}\sum_{t=\tau+1}^{T-1}\left(\sqrt{\sum_{s=1}^{t+1} v_s^2} - \sqrt{\sum_{s=1}^t v_s^2}\right)$$
$$\le 2\sqrt{2}\sqrt{\sum_{t=1}^T v_t^2}\,.$$

As for the second summation, we handle it similarly to (Cutkosky, 2019, Theorem 2). Defining $G_t = \max_{s\in[1,t]}|v_s|$, since $|\Delta_0| = 0$, we have

$$\sum_{t=1}^T |v_t - \widetilde{v}_t|\left(|\Delta_{t-1}| + |\Delta_t|\right) \le 2\left(\max_{t\in[1,T]}|\Delta_t|\right)\sum_{t=1}^T |v_t - \widetilde{v}_t| = 2\max\left(0, \max_{t\in[1,T]}|\Delta_t|\right)\sum_{t=1}^T\left(|v_t| - \sqrt{\sum_{s=1}^{t-1} v_s^2}\right)$$
$$\le 2\max\left(0, \max_{t\in[1,T]}|\Delta_t|\right)\sum_{t=1}^T (G_t - G_{t-1})$$
$$\le 2\left(\max_{t\in[1,T]}|\Delta_t|\right)G_T\,.$$

Combining the two upper bounds above completes the proof.

B. Analysis of β-FTRL

As discussed in Section 2, Adam corresponds to β-FTRL, the discounted version of scale-free FTRL, through the OLU framework. Thus, quantifying Adam's performance comes down to analyzing the dynamic regret of β-FTRL.

We now present the complete version of Theorem 3.1 (the dynamic regret bound of β-FTRL), fleshing out the proof sketch in Subsection 3.3. A main proof ingredient is our *discounted-to-dynamic conversion*. As a quick reminder, the formal setting considered is still the 1D OLO problem, where the output of the algorithm is denoted by $\Delta_t \in \mathbb{R}$, and the loss function is denoted by $\ell_t(x) = v_{t+1}x$ with $v_{t+1} \in \mathbb{R}$.

Step 1: Discounted regret. Our analysis starts with a concept called *discounted regret*, formalized in Definition 3.6. We recall the definition below for reader's convenience.

Definition B.1 (β-discounted regret)**.** *For any discounting factor $\beta \in (0, 1]$, the β-discounted regret is defined as*

$$R_{T;\beta}(u) \coloneqq \sum_{t=1}^T \beta^{T-t} v_t(\Delta_{t-1} - u)\,.$$

When $\beta = 1$, the β-discounted regret recovers the standard static regret $R_T(u)$. The notational difference is simply an extra subscript β in the β-discounted regret, *i.e.*, $R_{\cdot;\beta}$.

Intuitively, β-FTRL should achieve good β-discounted regret, as long as scale-free FTRL achieves good static regret. This intuition follows from observations that β-FTRL is just scale-free FTRL with the "discounted losses" $v_t \leftarrow \beta^{-t}v_t$, and that the β-discounted regret considers the loss sequence $\beta^{-t}v_t$ instead of v_t. We formalize this with the proof in Subsection B.1.

Theorem B.2 (Discounted regret of β-FTRL)**.** *For all $T > 0$, loss sequence $v_{1:T}$ and comparator $u \in \mathbb{R}$, β-FTRL guarantees the β-discounted regret bound*

$$R_{T;\beta}(u) \leq \left(\frac{u^2}{2\alpha} + \sqrt{2}\alpha\right)\sqrt{\sum_{t=1}^{T}(\beta^{T-t}v_t)^2} + 2\left(\max_{t\in[1,T]}|\Delta_t|\right)\left(\max_{t\in[1,T]}|\beta^{T-t}v_t|\right).$$

Step 2: Discounted-to-dynamic conversion. The remaining task is to convert this discounted regret bound to a dynamic regret bound. We accomplish this via a general discounted-to-dynamic conversion, which is of independent interest. The idea is to partition the entire time horizon into subintervals, and then consider static comparators on each of them. To be precise, we consider a *partition* of $[1, T]$ denoted by $\bigcup_{i=1}^{N}[a_i, b_i]$, where $a_1 = 1$, $b_i + 1 = a_{i+1}$ for all $1 \leq i < N - 1$, and $b_N = T$. Then each partitioned interval $[a_i, b_i]$ is coupled with an arbitrary fixed comparator $\bar{u}_i$.

We remark that this conversion is independent of the algorithm. For an algorithm $\mathcal{A}$, $R^{\mathcal{A}}_{T;\beta}(u)$ and $R^{\mathcal{A}}_{T}(u_{0:T-1})$ denote its β-discounted regret (Definition 3.6) and its dynamic regret (2.1), respectively. See Subsection B.2 for the proof.

Theorem B.3 (**Discounted-to-dynamic conversion**)**.** *Consider an arbitrary 1D OLO algorithm $\mathcal{A}$. For all $T > 0$, loss sequence $v_{1:T}$ and comparator sequence $u_{0:T-1}$, the dynamic regret of $\mathcal{A}$ satisfies*

$$\begin{aligned}R^{\mathcal{A}}_{T}(u_{0:T-1}) = \beta R^{\mathcal{A}}_{T;\beta}(\bar{u}_N) + (1-\beta)\sum_{i=1}^{N}\sum_{t\in[a_i,b_i]} R^{\mathcal{A}}_{t;\beta}(\bar{u}_i) \\ + \beta\sum_{i=1}^{N-1}\left[\left(\sum_{t=1}^{b_i}\beta^{b_i-t}v_t\right)(\bar{u}_{i+1}-\bar{u}_i)\right] + \sum_{i=1}^{N}\sum_{t\in[a_i,b_i]} v_t(\bar{u}_i - u_{t-1}),\end{aligned}$$

where $\bigcup_{i=1}^{N}[a_i, b_i]$ is an arbitrary partition of of $[1, T]$, and $\bar{u}_1, \dots \bar{u}_N \in \mathbb{R}$ are also arbitrary.

The remarkable aspect of this result is that it is an *equality*. That is, we do not lose anything through the conversion. Given a discounted regret bound of $\mathcal{A}$, we can make use of this conversion by substituting $R^{\mathcal{A}}_{T;\beta}(\bar{u}_N)$ and $R^{\mathcal{A}}_{t;\beta}(\bar{u}_i)$ with their discounted regret bounds, and then taking the infimum on the RHS *w.r.t.* the partition $\cup_{i\in[N]}[a_i, b_i]$ and the choice of the "approximated comparator sequence" $\bar{u}_1, \dots, \bar{u}_N$.

Step 3: Plugging in β-FTRL. Now we set $\mathcal{A}$ in the conversion to β-FTRL. See Subsection B.3 for the proof.

Theorem B.4 (**Dynamic regret of β-FTRL**)**.** *Consider β-FTRL with a fixed $\alpha > 0$. Consider any loss sequence $v_{1:T}$ and any comparator sequence $u_{0:T-1}$ s.t. $|u_t| \leq U$. The dynamic regret (2.1) of the β-FTRL is bounded as*

$$\begin{aligned}R_T(u_{0:T-1}) \leq \left(\frac{U^2}{2\alpha} + \sqrt{2}\alpha\right)\left[\beta\sqrt{V_\beta(v_{1:T})} + (1-\beta)\sum_{t=1}^{T}\sqrt{V_\beta(v_{1:t})}\right] \\ + 2\beta\left(\max_{t\in[1,T]}|\Delta_t|\cdot\max_{t\in[1,T]}|\beta^{T-t}v_t|\right) + 2(1-\beta)\sum_{t=1}^{T}\left(\max_{s\in[1,t]}|\Delta_s|\cdot\max_{s\in[1,t]}|\beta^{t-s}v_s|\right) \\ + \textsc{Variation},\end{aligned}$$

where $V_\beta(v_{1:t}) \coloneqq \sum_{s=1}^{t}(\beta^{t-s}v_s)^2$ is the discounted variance of the losses *and*

$$\textsc{Variation} \coloneqq \inf\left\{\beta\sum_{i=1}^{N-1}\left(\sum_{t=1}^{b_i}\beta^{b_i-t}v_t\right)(\bar{u}_{i+1}-\bar{u}_i) + \sum_{i=1}^{N}\sum_{t\in[a_i,b_i]} v_t(\bar{u}_i - u_{t-1})\right\},$$

and the infimum in Variation *is taken over all partitions $\bigcup_{i=1}^{N}[a_i, b_i]$ of $[1, T]$ and all choices of $\{\bar{u}_i\}_{i\in N}$ satisfying $|\bar{u}_i| \leq U$.*

In Theorem B.4, the variation term VARIATION consists of two terms. The first part

$$\beta \sum_{i=1}^{N-1}\left(\sum_{t=1}^{b_i}\beta^{b_i-t}v_t\right)(\bar{u}_{i+1}-\bar{u}_i)$$

measures how fast the representative comparators $\bar{u}_i$'s change across different subintervals, and we hence call it the "inter-partition variation". The second term

$$\sum_{i=1}^{N}\sum_{t\in[a_i,b_i]} v_t(\bar{u}_i-u_{t-1})$$

measures how different u_t's are from the representative comparators $\bar{u}_i$'s within each subinterval, and we call it the "intra-partition variation". A notable strength of the variation term is that it is the infimum over all partitions and $\bar{u}_i$'s. In other words, the upper bound will **automatically adapt** to the best choice of partitions and $\bar{u}_i$'s without knowing them explicitly. For instance, choosing

$$\bar{u}_i = \frac{\sum_{t\in[a_i,b_i]} v_t u_{t-1}}{\sum_{t\in[a_i,b_i]} v_t}$$

would make the intra-partition variation term zero.

Referring to Theorem B.4, we immediately obtain its simplified version in the main text, Theorem 3.1 with the proof in Subsection B.4. For the case of bounded comparators (β-FTRL$_D$), the dynamic regret can be analyzed using almost the same strategy, which leads us to state Theorem 3.2 with the proof in Subsection B.5.

B.1. Proof of Theorem B.2

β-FTRL is equivalent to scale-free FTRL with $v_t \leftarrow \beta^{-t}v_t$. Therefore, applying Theorem A.1 with $v_t \leftarrow \beta^{-t}v_t$ leads to

$$\sum_{t=1}^{T}\beta^{-t}v_t(\Delta_{t-1}-u) \le \left(\frac{u^2}{2\alpha}+\sqrt{2}\alpha\right)\sqrt{\sum_{t=1}^{T}(\beta^{-t}v_t)^2+2\left(\max_{t\in[1,T]}|\Delta_t|\right)\left(\max_{t\in[1,T]}\left|\beta^{-t}v_t\right|\right)}\,.$$

Multiplying both sides by β^T completes the proof.

B.2. Proof of Theorem B.3

Overall, the proof draws inspiration from (Zhang et al., 2018b), where a similar partitioning argument was used to prove a dynamic regret guarantee of a *strongly adaptive* online learner (Daniely et al., 2015). Throughout the proof, we will omit the superscript $\mathcal{A}$ for brevity, since our argument is independent of specific algorithms.

We start with a simple fact that connects the dynamic regret to the subinterval static regret. For any partition $\bigcup_{i=1}^{N}[a_i,b_i]$ of $[1,T]$ and any choices of $\{\bar{u}_i\}_{i\in N}$,

$$R_T(u_{0:T-1}) = \sum_{i=1}^{N}\sum_{t=a_i}^{b_i} v_t(\Delta_{t-1}-\bar{u}_i) + \sum_{i=1}^{N}\sum_{t=a_i}^{b_i} v_t(\bar{u}_i-u_{t-1})\,. \tag{B.1}$$

To handle the static regret on the RHS, we use the following result.

Lemma B.5. *On any subinterval $[a,b]\subset[1,T]$, with any $u\in\mathbb{R}$,*

$$\sum_{t=a}^{b} v_t(\Delta_{t-1}-u) = (1-\beta)\sum_{t=a}^{b} R_{t;\beta}(u) + \beta\left(R_{b;\beta}(u)-R_{a-1;\beta}(u)\right)\,.$$

Proof. For all t, notice that

$$R_{t;\beta}(u) = \sum_{s=1}^{t}\beta^{t-s}v_s(\Delta_{s-1}-u) \quad\text{and}\quad R_{t-1;\beta}(u) = \sum_{s=1}^{t-1}\beta^{t-s}v_s(\Delta_{s-1}-u)\,,$$

and thus

$$R_{t;\beta}(u) - \beta R_{t-1;\beta}(u) = v_t(\Delta_{t-1} - u)\,.$$

Summing over $t \in [a, b]$,

$$\begin{aligned}\sum_{t=a}^{b} v_t(\Delta_{t-1} - u) &= \sum_{t=a}^{b} R_{t;\beta}(u) - \beta \sum_{t=a}^{b} R_{t-1;\beta}(u) \\ &= (1-\beta)\sum_{t=a}^{b} R_{t;\beta}(u) - \beta R_{a-1;\beta}(u) + \beta R_{b;\beta}(u)\,. \end{aligned}$$

□

Next, applying Lemma B.5 to each $[a_i, b_i]$ in (B.1) yields:

$$R_T(u_{0:T-1}) = (1-\beta)\sum_{i=1}^{N}\sum_{t\in[a_i,b_i]} R_{t;\beta}(\bar{u}_i) + \beta\sum_{i=1}^{N}[R_{b_i;\beta}(\bar{u}_i) - R_{a_i-1;\beta}(\bar{u}_i)] + \sum_{i=1}^{N}\sum_{t\in[a_i,b_i]} v_t(\bar{u}_i - u_{t-1})\,.$$

Since $a_i - 1 = b_{i-1}$, the second term on the RHS can be rewritten as

$$\begin{aligned}\sum_{i=1}^{N}[R_{b_i;\beta}(\bar{u}_i) - R_{a_i-1;\beta}(\bar{u}_i)] &= R_{T;\beta}(\bar{u}_N) + \sum_{i=1}^{N-1}[R_{b_i;\beta}(\bar{u}_i) - R_{b_i;\beta}(\bar{u}_{i+1})] \\ &= R_{T;\beta}(\bar{u}_N) + \sum_{i=1}^{N-1}\left[\left(\sum_{t=1}^{b_i}\beta^{b_i-t}v_t\right)(\bar{u}_{i+1} - \bar{u}_i)\right]\,.\end{aligned}$$

Combining everything above completes the proof.

B.3. Proof of Theorem B.4

Due to Theorem B.3, for any "approximated comparator sequence" $\bar{u}_1, \dots, \bar{u}_N \in \mathbb{R}$ with $|\bar{u}_i| \le U$ for $i \in [N]$, we have

$$\begin{aligned}R_T^{\mathcal{A}}(u_{0:T-1}) = \beta R_{T;\beta}^{\mathcal{A}}(\bar{u}_N) + (1-\beta)\sum_{i=1}^{N}\sum_{t\in[a_i,b_i]} R_{t;\beta}^{\mathcal{A}}(\bar{u}_i) \\ + \beta\sum_{i=1}^{N-1}\left[\left(\sum_{t=1}^{b_i}\beta^{b_i-t}v_t\right)(\bar{u}_{i+1} - \bar{u}_i)\right] + \sum_{i=1}^{N}\sum_{t\in[a_i,b_i]} v_t(\bar{u}_i - u_{t-1})\,.\end{aligned}$$

Using Theorem B.2 and $|\bar{u}_i| \le U$,

$$\begin{aligned}R_{T;\beta}^{\mathcal{A}}(\bar{u}_N) &\le \left(\frac{\bar{u}_N^2}{2\alpha} + \sqrt{2}\alpha\right)\sqrt{\sum_{t=1}^{T}(\beta^{T-t}v_t)^2} + 2\left(\max_{t\in[1,T]}|\Delta_t|\right)\left(\max_{t\in[1:T]}|\beta^{T-t}v_t|\right) \\ &\le \left(\frac{U^2}{2\alpha} + \sqrt{2}\alpha\right)\sqrt{V_\beta(v_{1:T})} + 2\max_{t\in[1,T]}|\Delta_t| \cdot \max_{t\in[1:T]}|\beta^{T-t}v_t|\,,\end{aligned}$$

and similarly, for any t and $\bar{u}_i$,

$$R_{t;\beta}^{\mathcal{A}}(\bar{u}_i) \le \left(\frac{U^2}{2\alpha} + \sqrt{2}\alpha\right)\sqrt{V_\beta(v_{1:t})} + 2\max_{s\in[1,t]}|\Delta_s| \cdot \max_{s\in[1:t]}|\beta^{t-s}v_s|\,.$$

Putting these bounds into the equality and taking the infimum on the RHS (over the partition and the $\{\bar{u}_i\}_{i\in N}$ sequence satisfying $|\bar{u}_i| \le U$) complete the proof.

B.4. Simplification for unbounded domain: Theorem 3.1

Theorem 3.1 follows as a corollary of Theorem B.4 with the partition $\bigcup_{t=1}^{T}\{t\}$, $U=\alpha D_\beta$ and $\bar{u}_t=u_t$ for all t. With such choices, we have

$$\textsc{Variation} \le \beta \sum_{t=1}^{T-1}\left(\sum_{s=1}^{t}\beta^{t-s}v_s\right)(u_t-u_{t-1})\ .$$

Recall that $M_\beta \coloneqq \max_{t\in[1,T]}\frac{\left|\sum_{s=1}^{t}\beta^{t-s}v_s\right|}{\sqrt{\sum_{s=1}^{t}(\beta^{t-s}v_s)^2}}$. Hence, it follows that

$$\left|\sum_{s=1}^{t}\beta^{t-s}v_s\right| \le M_\beta\sqrt{V_\beta(v_{1:t})}\,.$$

Therefore, the $\textsc{Variation}$ term in Theorem B.4 is reduced to

$$\textsc{Variation} \le \beta M_\beta \sum_{t=1}^{T-1}\sqrt{V_\beta(v_{1:t})}\,|u_t-u_{t+1}|\ ,$$

and thus the bound becomes (notice that $U=\alpha M_\beta$ and $\beta<1$)

$$\begin{aligned}R_T(u_{0:T-1}) \le{}& \left(\frac{1}{2}\alpha M_\beta^2+\sqrt{2}\alpha\right)\left[\sqrt{V_\beta(v_{1:T})}+(1-\beta)\sum_{t=1}^{T}\sqrt{V_\beta(v_{1:t})}\right]+2\alpha M_\beta G\left[1+(1-\beta)T\right]\\&+M_\beta\sum_{t=1}^{T-1}\sqrt{V_\beta(v_{1:t})}\,|u_t-u_{t-1}|\ .\end{aligned}\tag{B.2}$$

For all t, using $\beta<1$

$$V_\beta(v_{1:t})=\sum_{s=1}^{t}(\beta^{t-s}v_s)^2\le G^2\sum_{i=0}^{\infty}\beta^{2i}=\frac{G^2}{1-\beta^2}<\frac{G^2}{1-\beta}\,.$$

Therefore,

$$\begin{aligned}R_T(u_{0:T-1}) &\le \left(\frac{1}{2}\alpha M_\beta^2+\sqrt{2}\alpha\right)\left[\frac{G}{\sqrt{1-\beta}}+\sqrt{1-\beta}GT\right]+2\alpha M_\beta G\left[1+(1-\beta)T\right]+\frac{M_\beta GP}{\sqrt{1-\beta}}\\&=\mathcal{O}\left(\left(\alpha+\alpha M_\beta^2+M_\beta P\right)\frac{G}{\sqrt{1-\beta}}+\left(\alpha+\alpha M_\beta^2\right)\sqrt{1-\beta}GT\right)\,.\end{aligned}$$

Hence, we arrive at Theorem 3.1 presented in the main paper.

B.5. Simplification for bounded domain Theorem 3.2

For the case of bounded comparators, *i.e.*, $|u_t|\le D$, we consider β-FTRL$_D$, the D-clipped version of β-FTRL:

$$\Delta_t=-\mathrm{clip}_D\left(\alpha\frac{\sum_{s=1}^{t}\beta^{t-s}v_s}{\sqrt{\sum_{s=1}^{t}(\beta^{t-s}v_s)^2}}\right)\ ,\tag{β-FTRL$_D$}$$

where $\mathrm{clip}_D(x)=x\min(\frac{D}{|x|},1)$. With β-FTRL$_D$, since $|\Delta_t|\le D$ at each step, the following regret bound holds. The proof boils down to verifying that the entire proof strategy of Theorem B.2 goes through even with projection.

Theorem B.6 (Discounted regret of β-FTRL$_D$)**.** *For all $T>0$, loss sequence $v_{1:T}$ and comparator $|u|\le D$, the discounted regret bound of β-FTRL$_D$ is*

$$R_{T;\beta}(u)\le\left(\frac{u^2}{2\alpha}+\sqrt{2}\alpha\right)\sqrt{\sum_{t=1}^{T}(\beta^{T-t}v_t)^2}+2D\left(\max_{t\in[1:T]}|\beta^{T-t}v_t|\right)\,.$$

Now based on this discounted regret bound, we prove the claimed dynamic regret bound in Theorem 3.2. Similar to the proof of Theorem 3.1, from Theorem B.4, we choose the partition to be $\cup_{t=1}^{T}\{t\}$, and let $U = D$ and $\bar{u}_t = u_t$ for all t. With such choices, we have

$$\textsc{Variation} \le \beta \sum_{t=1}^{T-1} \left| \sum_{s=1}^{t} \beta^{t-s} v_s \right| |u_t - u_{t-1}| \,.$$

Therefore, the upper bound in Theorem B.4 becomes (notice that $\alpha = U = D$ and $\beta < 1$)

$$\begin{aligned} R_T(u_{0:T-1}) &\le 2D \left[\sqrt{V_\beta(v_{1:T})} + (1-\beta) \sum_{t=1}^{T} \sqrt{V_\beta(v_{1:t})} \right] + 2DG\left[1 + (1-\beta)T\right] + \beta \sum_{t=1}^{T-1} \left| \sum_{s=1}^{t} \beta^{t-s} v_s \right| |u_t - u_{t-1}| \\ &\le 4D \left[\sqrt{V_\beta(v_{1:T})} + (1-\beta) \sum_{t=1}^{T} \sqrt{V_\beta(v_{1:t})} \right] + \sum_{t=1}^{T-1} \left| \sum_{s=1}^{t} \beta^{t-s} v_s \right| |u_t - u_{t-1}| \,. \end{aligned}$$

Now notice that for all t, since $\beta < 1$,

$$\begin{aligned} V_\beta(v_{1:t}) &= \sum_{s=1}^{t} (\beta^{t-s} v_s)^2 \le G^2 \sum_{i=0}^{\infty} \beta^{2i} = \frac{G^2}{1-\beta^2} < \frac{G^2}{1-\beta}\,, \quad \text{and} \\ \left| \sum_{s=1}^{t} \beta^{t-s} v_s \right| &\le G \sum_{t=0}^{\infty} \beta^t \le \frac{G}{1-\beta}\,. \end{aligned}$$

Substituting these bounds back to the bound on $R_T(u_{0:T-1})$, we obtain

$$R_T(u_{0:T-1}) \le 4DG \left(\frac{1}{\sqrt{1-\beta}} + \sqrt{1-\beta} \cdot T \right) + \frac{GP}{1-\beta}\,.$$

Therefore, we arrive at Theorem 3.2 presented in the main paper.

C. Benefits of momentum and discounting factor

This section presents omitted details from Subsection 3.2. The goal is to justify the benefits of Adam's algorithmic components, namely the momentum and the discounting factor.

C.1. Proof of lower bounds (Theorem 3.3)

For simplicity, we start by assuming T is a multiple of 4. Consider the following loss sequence: For $1 \le t \le T/2$,

$$\mathbf{v}_t = \begin{cases} (1,0) & \text{for } t \text{ even,} \\ (0,1) & \text{for } t \text{ odd.} \end{cases}$$

For $T/2 < t \le T$,

$$\mathbf{v}_t = \begin{cases} (-1,0) & \text{for } t \text{ even,} \\ (0,-1) & \text{for } t \text{ odd.} \end{cases}$$

The comparator sequence $\mathbf{u}_{0:T-1}$ is given as $\mathbf{u}_t = (-1,-1)$ for $0 \le t \le T/2 - 1$ and $\mathbf{u}_t = (1,1)$ for $t \ge T/2$. Then, we have $\sum_{t=1}^{T} \langle \mathbf{v}_t, \mathbf{u}_{t-1} \rangle = -T$.

As for the total loss,

- Consider the baseline (3.2). Since $\mathbf{v}_t[i]\mathbf{v}_{t+1}[i] = 0$ for all $t \ge 1$ and $i = 1, 2$, we have

$$\sum_{t=1}^{T} \langle \mathbf{v}_t, \boldsymbol{\Delta}_{t-1} \rangle = \sum_{t=1}^{T} \sum_{i=1}^{2} \mathbf{v}_t[i] \boldsymbol{\Delta}_{t-1}[i] = -\sum_{t=1}^{T} \sum_{i=1}^{2} \boldsymbol{\alpha}_{t-1}[i] \mathbf{v}_{t-1}[i] \mathbf{v}_t[i] = 0.$$

- Consider β-FTRL$_D$ with $\beta = 1$ and $D = 1$. Recall its coordinate-wise update rule,

$$\mathbf{\Delta}_t[i] = -\mathrm{clip}_1\left(\alpha \frac{\sum_{s=1}^t \mathbf{v}_s[i]}{\sqrt{\sum_{s=1}^t \mathbf{v}_s[i]^2}}\right).$$

From the loss sequence, it follows that $\sum_{s=1}^t \mathbf{v}_s[i] \geq 0$ for all t, and hence, we have $-1 \leq \mathbf{\Delta}_t[i] \leq 0$ for all $t \geq 0$ and $i = 1, 2$. Hence, $\sum_{t=1}^T \langle \mathbf{v}_t, \mathbf{\Delta}_{t-1}\rangle \geq -T/2$.

This completes the proof under the assumption that T is a multiple of 4.

For general T, let $\widehat{T}$ be the largest integer less or equal to T which is a multiple of 4. Then, we define $\mathbf{v}_{1:\widehat{T}}$ and $\mathbf{u}_{1:\widehat{T}-1}$ as the aforementioned loss and comparator sequences (with T replaced by $\widehat{T}$), and this yields lower bounds on $\sum_{t=1}^{\widehat{T}} \langle \mathbf{v}_t, \mathbf{\Delta}_{t-1} - \mathbf{u}_{t-1}\rangle$. As for the time index satisfying $\widehat{T} < t \leq T$, we define $\mathbf{v}_t = (0, 0)$ and $\mathbf{u}_{t-1} = \mathbf{u}_{\widehat{T}-1}$. In this way, altogether, $\sum_{t=1}^T \langle \mathbf{v}_t, \mathbf{\Delta}_{t-1} - \mathbf{u}_{t-1}\rangle = \sum_{t=1}^{\widehat{T}} \langle \mathbf{v}_t, \mathbf{\Delta}_{t-1} - \mathbf{u}_{t-1}\rangle$, and the lower bounds for the latter can be applied.

C.2. Proof of Corollary 3.4

Using Theorem 3.2 with $\beta = 1 - cT^{-2/3}$,

$$\begin{aligned} R_T(u_{0:T-1}) &\lesssim \frac{DG}{\sqrt{1-\beta}} + \frac{DG}{1-\beta} + \sqrt{1-\beta}DGT && \text{(Theorem 3.2)} \\ &= \frac{DGT^{1/3}}{\sqrt{c}} + \frac{PGT^{2/3}}{c} + \sqrt{c}DGT^{2/3} \\ &\lesssim DGT^{2/3}c^{1/2}\left(1 + \frac{c^{-3/2}P}{D}\right). \end{aligned}$$

With the optimal tuning $c = \Theta\big((P/D)^{2/3}\big)$, it becomes $\mathcal{O}(GD^{2/3}P^{1/3}T^{2/3})$.

C.3. Proof of Corollary 3.5

Consider $\beta < 1$ first. Since the environment is well-behaved with constant M, we can invoke Theorem 3.1 with M_β there replaced by M. Notice that M is independent of β, therefore at the end we may tune β using M. Concretely, using Theorem 3.1 with $\beta = 1 - cT^{-1}$,

$$\begin{aligned} R_T(u_{0:T-1}) &\lesssim \frac{\big(\alpha M^2 + MP\big)G}{\sqrt{1-\beta}} + \sqrt{1-\beta}\alpha M^2 GT && \text{(Theorem 3.1)} \\ &= MG\sqrt{T}c^{1/2}\left(\frac{\alpha M + P}{c} + \alpha M\right) \\ &\lesssim \alpha M^2 G\sqrt{T}c^{1/2}\left(1 + \frac{c^{-1}P}{\alpha M}\right). \end{aligned}$$

With the optimal tuning $c = \Theta\big(P/(\alpha M)\big)$, it becomes $\mathcal{O}(\alpha^{1/2}M^{3/2}GP^{1/2}T^{1/2})$.

Next, consider $\beta = 1$. We follow the same analysis in Subsection B.4 until (B.2), before plugging in any β. Then, instead of using $\beta < 1$ there, we plug in $\beta = 1$, which yields

$$\begin{aligned} R_{[0,T-1]}(u_{0:T-1}) &\leq \left(\frac{1}{2}\alpha M^2 + \sqrt{2}\alpha\right)\sqrt{V_1(v_{1:T})} + 2\alpha MG + M\sum_{t=0}^{T-2}\sqrt{V_1(v_{1:t+1})}\,|u_t - u_{t+1}| \\ &\lesssim \left(\alpha M^2 + \sqrt{2}\alpha\right)G\sqrt{T} + MG\sqrt{T}\sum_{t=0}^{T-2}|u_t - u_{t+1}| \\ &\lesssim MGP\sqrt{T}. && (T \gg 1, \text{ and } P \gg \alpha M) \end{aligned}$$

D. Details on optimization

D.1. Proof of Theorem 4.1

Since F is differentiable, the fundamental theorem of calculus implies that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, $F(\mathbf{y}) - F(\mathbf{x}) = \int_0^1 \langle \nabla F(\mathbf{x} + t(\mathbf{y} - \mathbf{x})), \mathbf{y} - \mathbf{x} \rangle \, \mathrm{d}t$. Hence, we have

$$F(\mathbf{w}_{t+1}) - F(\mathbf{w}_t) = \int_0^1 \langle \nabla F(\mathbf{w}_t + s\boldsymbol{\Delta}_t), \boldsymbol{\Delta}_t \rangle \, \mathrm{d}s = \mathbb{E}_{s \sim \mathrm{Unif}([0,1])} \langle \nabla F(\mathbf{w}_t + s\boldsymbol{\Delta}_t), \boldsymbol{\Delta}_t \rangle$$

Now, summing over t and telescoping yield the desired equality.