
Revisiting Projection-free Online Learning: the Strongly Convex Case

Dan Garber

Technion - Israel Institute of Technology

Ben Kretzu

Technion - Israel Institute of Technology

Abstract

Projection-free optimization algorithms, which are mostly based on the classical Frank-Wolfe method, have gained significant interest in the machine learning community in recent years due to their ability to handle convex constraints that are popular in many applications, but for which computing projections is often computationally impractical in high-dimensional settings, and hence prohibit the use of most standard projection-based methods. In particular, a significant research effort was put on projection-free methods for online learning. In this paper we revisit the Online Frank-Wolfe (OFW) method suggested by Hazan and Kale (2012) and fill a gap that has been left unnoticed for several years: OFW achieves a faster rate of $O(T^{2/3})$ on strongly convex functions (as opposed to the standard $O(T^{3/4})$ for convex but not strongly convex functions), where T is the sequence length. This is somewhat surprising since it is known that for offline optimization, in general, strong convexity does not lead to faster rates for Frank-Wolfe. We also revisit the bandit setting under strong convexity and prove a similar bound of $\tilde{O}(T^{2/3})$ (instead of $O(T^{3/4})$ without strong convexity). Hence, in the current state-of-affairs, the best projection-free upper-bounds for the full-information and bandit settings with strongly convex and nonsmooth functions match up to logarithmic factors in T .

1 INTRODUCTION

Computing projections onto convex sets is a fundamental computational primitive in most popular optimization methods such as projected gradient methods which are at the heart of numerous machine learning tasks. However, many machine learning applications involve optimization with structural constraints for which computing projections (e.g. Euclidean projection) is impractical in high-dimensional settings. It is for this reason that so-called *projection-free* optimization methods, which replace the potentially computationally-expensive projection primitive with a different more efficient primitive, have attracted significant interest within the machine learning community in recent years. These projection-free methods are mostly based on the classical Frank-Wolfe method for constrained convex optimization (aka the conditional gradient method) (Jaggi, 2013; Frank and Wolfe, 1956; Levitin and Polyak, 1966) which replaces the projection operation with a linear optimization step over the constraints. Indeed in many important cases such as constraints arising from combinatorial structure (e.g., paths/matchings/spanning trees in combinatorial graphs, or matroids), or from a low-rank matrix structure, linear optimization over the feasible set can be carried out very efficiently, sometimes by orders of magnitude faster than the projection operation, see for instance discussions in Jaggi (2013); Hazan and Kale (2012). Developing efficient projection-free methods, and in particular Frank-Wolfe-based methods for various optimization paradigms central to machine learning as thus become a popular research area where efforts have been focused on traditional offline optimization (Garber and Hazan, 2016; Lacoste-Julien and Jaggi, 2015; Garber and Hazan, 2015), stochastic optimization (Hazan and Luo, 2016; Goldfarb et al., 2017; Garber and Kaplan, 2019), online learning (Hazan and Kale, 2012; Garber and Hazan, 2013; Chen et al., 2019; Garber and Kretzu, 2020), and distributed computation (Bellet et al., 2015; Wang et al., 2016).

In online learning (Cesa-Bianchi and Lugosi, 2006; Hazan, 2016), which deals with sequential prediction

over a (large) number of rounds, projection-free methods are of particular interest since naturally in such a setting, the response time of the online algorithms, i.e., the time it takes to compute a new prediction on each iteration, is of major importance in many applications. Hazan and Kale (2012) introduced the Online Frank-Wolfe method (OFW) for online convex optimization and proved that it attains $O(T^{3/4})$ regret, where T is the number of prediction rounds, for convex loss functions in the full-information setting (i.e., after each round the loss function is fully observable to the learner), using a single linear optimization step over the feasible set per iteration. This is contrast to the Online Gradient Descent method (OGD) which attains $O(\sqrt{T})$ regret, however requires T projection steps. For the special case in which the feasible set is a polytope, Garber and Hazan (2013) gave a modified Online Frank-Wolfe method with regret $O(\sqrt{nT})$, where n is the dimension. Very recently Hazan and Minasyan (2020) presented a randomized online algorithm that also uses only a single linear optimization step per iteration, however one that is not based on the Frank-Wolfe method. Their algorithm attains $(\sqrt{n}T^{2/3})$ expected regret under the additional assumption that the loss functions are also smooth.

In the bandit setting, in which only the scalar loss incurred by the online learner is observed after each round and not the loss function, Chen et al. (2019) gave the first projection-free algorithm, which combines OFW and the gradient estimation idea of Flaxman et al. (2005), to obtain an algorithm with $O(nT^{4/5})$ expected regret. This was very recently improved by Garber and Kretzu (2020) by taking advantage of the special structure of the gradient estimator of Flaxman et al. (2005) and by considering a similar algorithm to that of Chen et al. (2019) with the difference of considering the prediction round in blocks in order to obtain higher accuracy in the computation of the iterates, which led to a $O(\sqrt{n}T^{3/4})$ expected regret bound using overall $O(T)$ linear optimization steps in expectation, which matches (in terms of T) the current best upper-bound for the full-information setting (without additional assumptions such as smoothness) of Hazan and Kale (2012). We also note that, besides the theoretical contributions, both Hazan and Kale (2012) and Chen et al. (2019) have demonstrated the practical appeal of online Frank-Wolfe methods via extensive empirical studies.

While Hazan and Minasyan (2020) have managed to obtain a faster rate (at least in T) by leveraging the *smoothness* of the loss functions, a property that is well known to allow for faster rates in offline and even stochastic optimization but is much less useful in online optimization, *strong convexity*, another prop-

erty of loss functions that is also well known to allow faster rates in convex optimization paradigms and in particular in online convex optimization (Hazan et al., 2006), has remained unexplored in the context of projection-free methods. This is perhaps not surprising, since as discussed, most projection-free algorithms are based on the Frank-Wolfe method, and it is well known that, in general, the convergence rate of Frank-Wolfe for offline convex optimization does not improve in the presence of strong convexity (Jaggi, 2013; Garber and Hazan, 2016)¹. Nevertheless, in this work we show, to the best of our knowledge, for the first time, that both in the full-information and bandit online settings, strong convexity does in fact lead to faster rates for the OFW algorithm (Hazan and Kale, 2012) and its bandit variant with blocks (Garber and Kretzu, 2020). In particular we prove a $O(T^{2/3})$ regret bound for the full-information setting and a $O((nT)^{2/3}(\ln T)^{1/3} + T^{2/3}\ln T)$ expected regret bound for the bandit setting.

Quite pleasingly, our results do not hinge on any new particular technique, but mostly adapt those of Hazan and Kale (2012) to leverage the strong convexity of the losses. With this respect, beyond formally deriving the improved rates, we view our contribution as mainly of conceptual importance: observing that, as opposed to the offline setting, in the online setting strong convexity is indeed beneficial for Frank-Wolfe-based methods.

On a slightly more technical level, an intuition to why strong convexity helps is as follows: in OFW, Frank-Wolfe is used to iteratively approximate (using one linear optimization step per iteration) the steps of the regret-optimal Regularized Follow the Leader method (RFTL), which on each iteration seeks to minimize the aggregated loss up to the current round plus an additional strongly convex regularization term. The reason for the slow rate of OFW is that the RFTL objective (i.e., subproblem that needs to be solved each iteration) drifts too much from one iteration to the next which does not allow for a good enough approximation. To control this drift one must increase the regularization beyond the optimal level which in turn leads to sub-optimal regret of $O(T^{3/4})$. Nevertheless, in the strongly convex case, due to the strong convexity, it can be shown that the drift from one round to the next in the RFTL objective is milder, which in turn allows for better approximation via a single Frank-Wolfe step. The same intuition, coupled with

¹Obtaining faster rates for Frank-Wolfe variants under strong convexity-like properties is an active research effort in recent years, however it mostly requires focussing on specific types of feasible sets, e.g., polytopes (Garber and Hazan, 2016) or strongly convex sets (Garber and Hazan, 2015), etc., while here we are mainly interested in generic methods that apply to arbitrary convex and compact sets.

Table 1: Comparison of regret bounds for state-of-the-art linear optimization-based online algorithms.

REFERENCE	FEEDBACK	SMOOTHNESS?	STRONG CONVEXITY?	REGRET
Hazan and Kale (2012)	Full	x	x	$T^{3/4}$
Garber and Kretzu (2020)	Bandit	x	x	$\sqrt{n}T^{3/4}$
Hazan and Minasyan (2020)	Full	✓	x	$\sqrt{n}T^{2/3}$
This work (Thm. 1)	Full	x	✓	$T^{2/3}$
This work (Thm. 2)	Bandit	x	✓	$(nT)^{2/3}(\ln T)^{1/3} + T^{2/3} \ln T$

the recent result of Garber and Kretzu (2020), also leads to the improved rate in the bandit setting.

2 PRELIMINARIES

2.1 Online Convex Optimization

In the online convex optimization with full information setting (Hazan, 2016), an online learner is required to iteratively choose actions from a fixed feasible set $\mathcal{K} \subseteq \mathbb{R}^n$ which is assumed convex and compact. After choosing his action $\mathbf{x}_t \in \mathcal{K}$ on round $t \in [T]$ (T is assumed to be known beforehand), the learner observes a loss function $f_t(\cdot)$, which is chosen by an adversary, and incurs the loss $f_t(\mathbf{x}_t)$, where $f_t : \mathcal{K} \rightarrow \mathbb{R}$ is a convex.

The goal of the learner is to minimize the regret which is given by

$$\mathcal{R}_T := \sum_{t=1}^T f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{K}} \sum_{t=1}^T f_t(\mathbf{x}). \quad (1)$$

In the bandit-information setting, instead of observing the loss function after each iteration, the learner only observes his loss, that is the scalar value $f_t(\mathbf{x}_t)$. In this setting, we assume the adversary is oblivious, i.e., the loss functions f_1, \dots, f_T are chosen beforehand and do not depend on the actions of the learner. The bandit-feedback requires the learner to use random exploration and therefore, the goal is to minimize the expected regret $\mathbb{E}[\mathcal{R}_T]$.

We also make the following standard boundness assumptions, $\forall t \in [T] \forall \mathbf{x} \in \mathcal{K} \forall \mathbf{g} \in \partial f_t(\mathbf{x})$: $\|f_t\|_\infty = \sup_{\mathbf{x} \in \mathcal{K}} |f_t(\mathbf{x})| \leq M$ and $\|\mathbf{g}\|_2 \leq G$, for some $M, G > 0$. The latter implies each $f_t(\cdot)$ is G -Lipschitz over \mathcal{K} .

Following Flaxman et al. (2005), we also assume the feasible set \mathcal{K} is full dimensional, contains the origin, and that there exist scalars $r, R > 0$ such that $r\mathcal{B}^n \subseteq \mathcal{K} \subseteq R\mathcal{B}^n$, where \mathcal{B}^n denotes the unit Euclidean ball centered at the origin in \mathbb{R}^n .

2.2 Additional Notation And Definitions

We denote by \mathcal{S}^n the unit sphere in \mathbb{R}^n , and we write $\mathbf{u} \sim \mathcal{S}^n$ and $\mathbf{u} \sim \mathcal{B}^n$ to denote a random vector \mathbf{u} sampled uniformly from \mathcal{S}^n and \mathcal{B}^n , respectively. We denote by $\|\mathbf{x}\|$ the ℓ_2 norm of the vector \mathbf{x} .

Finally, for a compact and convex set $\mathcal{K} \subset \mathbb{R}^n$, which satisfies the above assumptions (i.e., $r\mathcal{B}^n \subseteq \mathcal{K} \subseteq R\mathcal{B}^n$), and a scalar $0 < \delta \leq r$, we define the set $\mathcal{K}_\delta := (1 - \delta/r)\mathcal{K} = \{(1 - \delta/r)\mathbf{x} \mid \mathbf{x} \in \mathcal{K}\}$. In particular, it holds that $\mathcal{K}_\delta \subseteq \mathcal{K}$ and for all $\mathbf{x} \in \mathcal{K}_\delta$, $\mathbf{x} + \delta\mathcal{B}^n \subseteq \mathcal{K}$.

We recall that a function $f : \mathcal{K} \rightarrow \mathbb{R}$ is α -strongly convex over \mathcal{K} if $\forall \mathbf{x}, \mathbf{y} \in \mathcal{K}$: $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{\alpha}{2} \|\mathbf{y} - \mathbf{x}\|^2$. Let \mathbf{x}^* be the unique minimizer of f , an α -strongly convex function over \mathcal{K} . From the above definition and the first order optimality condition it follows that $\forall \mathbf{x} \in \mathcal{K}$:

$$\frac{\alpha}{2} \|\mathbf{x} - \mathbf{x}^*\|^2 \leq f(\mathbf{x}) - f(\mathbf{x}^*). \quad (2)$$

2.3 Smoothed Loss Functions

As in Flaxman et al. (2005); Chen et al. (2019); Garber and Kretzu (2020), our bandit algorithm is based on a randomized-smoothing of the loss functions technique. We define the δ -smoothing of a loss function f by $\hat{f}_\delta(\mathbf{x}) = \mathbb{E}_{\mathbf{u} \sim \mathcal{B}^n} [f(\mathbf{x} + \delta\mathbf{u})]$. We now cite several useful lemmas regarding smoothed functions.

Lemma 1 (Lemma 2.6 in Hazan (2016)). Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be α -strongly convex and G -Lipschitz over a convex and compact set $\mathcal{K} \subset \mathbb{R}^n$. Then \hat{f}_δ is α -strongly convex and G -Lipschitz over \mathcal{K}_δ , and $\forall \mathbf{x} \in \mathcal{K}_\delta$ it holds that $|\hat{f}_\delta(\mathbf{x}) - f(\mathbf{x})| \leq \delta G$.

Lemma 2 (Lemma 6.5 in Hazan (2016)). $\hat{f}_\delta(\mathbf{x})$ is differentiable and $\nabla \hat{f}_\delta(\mathbf{x}) = \mathbb{E}_{\mathbf{u} \sim \mathcal{B}^n} \left[\frac{n}{\delta} f(\mathbf{x} + \delta\mathbf{u}) \mathbf{u} \right]$.

Lemma 3 (see Bertsekas (1973)). Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be convex and suppose that all subgradients of f are upper-bounded by G in ℓ_2 -norm over a convex and compact set $\mathcal{K} \subset \mathbb{R}^n$. Then, for any $\mathbf{x} \in \mathcal{K}_\delta$ it holds that $\|\nabla \hat{f}_\delta(\mathbf{x})\| \leq G$.

3 FULL- INFORMATION SETTING

In this section we present and prove our main result - an improved $O(T^{2/3})$ regret bound for Online Frank-Wolfe (Algorithm 1 below), in case all loss functions are α -strongly convex for some $\alpha > 0$.

Algorithm 1: Online Frank-Wolfe (see also Hazan and Kale (2012))

Data: T, α, T_0
 $\mathbf{x}_1 \leftarrow$ arbitrary point in \mathcal{K}
for $t = 1, \dots, T$ **do**
 Play \mathbf{x}_t and observe $f_t(\mathbf{x}_t)$
 Set $\nabla_t \leftarrow \nabla f_t(\mathbf{x}_t)$
 Define $F_{t+1}(\mathbf{x}) := \sum_{i=1}^t \mathbf{x}^\top \nabla_i + \frac{\alpha}{2} \|\mathbf{x} - \mathbf{x}_1\|^2 + T_0 \frac{\alpha}{2} \|\mathbf{x} - \mathbf{x}_1\|^2$
 $\mathbf{v}_t \in \operatorname{argmin}_{\mathbf{x} \in \mathcal{K}} \{\nabla F_{t+1}(\mathbf{x}_t)^\top \cdot \mathbf{x}\}$
 $\sigma_t = \operatorname{argmin}_{\sigma \in [0,1]} \{F_{t+1}(\mathbf{x}_t + \sigma(\mathbf{v}_t - \mathbf{x}_t))\} \triangleright$ Line-search
 $\mathbf{x}_{t+1} = \mathbf{x}_t + \sigma_t(\mathbf{v}_t - \mathbf{x}_t)$
end

Theorem 1. Suppose all loss functions are α -strongly convex for some $\alpha > 0$. Setting $T_0 = \max \left\{ 1, \frac{2(G+2R\alpha)\sqrt{b}}{\alpha R^2} \right\}$ with $b = \max \left\{ \left(\frac{G+2R\alpha}{\alpha}\right)^2, \frac{8(2R)^2(G+2R\alpha)}{\alpha} \right\}$ in Algorithm 1, guarantees that the regret is upper-bounded by

$$\begin{aligned} \mathcal{R}_T &\leq 4 \frac{(G+2R\alpha)^2}{\alpha} \ln(T) + 2\alpha R^2 \\ &\quad + 10 \frac{(G+2R\alpha)^2}{\alpha} + 16(G+2R\alpha)^{\frac{4}{3}} \frac{R^{\frac{2}{3}}}{\alpha^{\frac{1}{3}}} \\ &\quad + 4G \left(\frac{G+2R\alpha}{\alpha} \right) T^{\frac{2}{3}} + 8\sqrt{2} \frac{(G+2R\alpha)^{\frac{1}{3}} R^{\frac{2}{3}}}{\alpha^{\frac{1}{3}}} T^{\frac{2}{3}}, \end{aligned}$$

and the overall number of calls to the linear optimization oracle is T (one per iteration).

3.1 Proof of Theorem 1

For the regret analysis we need a well known lemma known as the "Follow-the-leader-Be-the-leader" lemma, which we state here in a slightly modified version. A proof is given in the appendix for completeness.

Lemma 4. Let $\mathcal{K} \subseteq \mathbb{R}^d$ a convex and compact set, $\{g_m(\mathbf{x})\}_{m=1}^T$ a set of convex functions, $\mathbf{x}_1 \in \mathbb{R}^d$ and $c_1 \in \mathbb{R}^+$. Denote $\mathbf{x}_\tau^* = \operatorname{argmin}_{\mathbf{x} \in \mathcal{K}} \left\{ \sum_{m=1}^{\tau-1} g_m(\mathbf{x}) + c_1 \|\mathbf{x} - \mathbf{x}_1\|^2 \right\}$ for every $\tau \in [1, T+1]$. Then for every $\mathbf{x} \in \mathcal{K}$

we have that

$$\begin{aligned} \sum_{m=1}^T (g_m(\mathbf{x}_m^*) - g_m(\mathbf{x})) &\leq \sum_{m=1}^T (g_m(\mathbf{x}_m^*) - g_m(\mathbf{x}_{m+1}^*)) \\ &\quad + c_1 \|\mathbf{x} - \mathbf{x}_1\|^2. \end{aligned}$$

For the purpose of the analysis of Algorithm 1, we define the auxiliary sequence $\{\mathbf{x}_t^*\}_{t=1}^T$ as $\mathbf{x}_t^* = \operatorname{argmin}_{\mathbf{x} \in \mathcal{K}} F_t(\mathbf{x})$, where $F_t(\cdot)$ is as defined in Algorithm 1. Note \mathbf{x}_t^* is simply the point played by the well-known (exact) Regularized Follow the Leader (RFTL) method (Hazan, 2016).

The following lemma upper-bounds the regret of Algorithm 1 in terms of how well does the iterate \mathbf{x}_t , which is obtained by applying a single Frank-Wolfe step to the RFTL objective $F_t(\cdot)$, approximates the optimal value of the RFTL objective, whose minimizer is the RFTL iterate \mathbf{x}_t^* . We recall this sequence of approximation errors is captured by the sequence $\{\epsilon_t\}_{t \geq 1}$.

Lemma 5. Let $\{\epsilon_t\}_{t=1}^T > 0$ and $T_0 \geq 1$. Suppose that throughout the run of Algorithm 1, for all time steps $t = 1, \dots, T$ it holds that $F_t(\mathbf{x}_t) - F_t(\mathbf{x}_t^*) \leq \epsilon_t$. Then, the regret of Algorithm 1 is upper-bounded by

$$\begin{aligned} \mathcal{R}_T &\leq \frac{2(G+2R\alpha)^2}{\alpha} (1 + \ln(T)) + 2\alpha R^2 T_0 \\ &\quad + G \sum_{t=1}^T \sqrt{\frac{2\epsilon_t}{\alpha(t-1+T_0)}}. \end{aligned}$$

Proof. Using the definition of the regret, Eq.(1), and the α -strongly convexity of each $f_t(\mathbf{x})$, we have that

$$\begin{aligned} \mathcal{R}_T &\leq \sum_{t=1}^T \nabla_t^\top (\mathbf{x}_t - \mathbf{x}^*) - \frac{\alpha}{2} \|\mathbf{x}_t - \mathbf{x}^*\|^2 \\ &= \sum_{t=1}^T \nabla_t^\top (\mathbf{x}_t - \mathbf{x}_t^* + \mathbf{x}_t^* - \mathbf{x}^*) - \frac{\alpha}{2} \|\mathbf{x}_t - \mathbf{x}^*\|^2. \quad (3) \end{aligned}$$

Using Cauchy-Schwarz inequality, $\|\nabla_t\| \leq G$, the fact that $F_t(\mathbf{x})$ is $(t-1+T_0)\alpha$ -strongly convex, Eq.(2), and the assumption that $F_t(\mathbf{x}_t) - F_t(\mathbf{x}_t^*) \leq \epsilon_t$, we have

$$\begin{aligned} \sum_{t=1}^T \nabla_t^\top (\mathbf{x}_t - \mathbf{x}_t^*) &\leq G \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}_t^*\| \\ &\leq G \sum_{t=1}^T \sqrt{\frac{2(F_t(\mathbf{x}_t) - F_t(\mathbf{x}_t^*))}{\alpha(t-1+T_0)}} \\ &\leq G \sum_{t=1}^T \sqrt{\frac{2\epsilon_t}{\alpha(t-1+T_0)}}. \quad (4) \end{aligned}$$

Now, we need to obtain an upper bound on $\nabla_t^\top (\mathbf{x}_t^* - \mathbf{x}^*) - \frac{\alpha}{2} \|\mathbf{x}_t - \mathbf{x}^*\|^2$. We will start with a

few preliminary steps.

For all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{K}$, the following holds

$$\begin{aligned} \|\mathbf{x} - \mathbf{z}\|^2 - \|\mathbf{y} - \mathbf{z}\|^2 &\leq \|\mathbf{x} + \mathbf{y} - 2\mathbf{z}\| \|\mathbf{x} - \mathbf{y}\| \\ &\leq 4R\|\mathbf{x} - \mathbf{y}\|. \end{aligned} \quad (5)$$

Define $g_t(\mathbf{x}) = \mathbf{x}^\top \nabla_t + \frac{\alpha}{2}\|\mathbf{x} - \mathbf{x}_t\|^2$, using Cauchy-Schwarz inequality and Eq.(5), we have for all $\mathbf{x}, \mathbf{y} \in \mathcal{K}$,

$$g_t(\mathbf{x}) - g_t(\mathbf{y}) \leq (G + 2R\alpha) \|\mathbf{x} - \mathbf{y}\|. \quad (6)$$

Since for every t , $F_{t+1}(\mathbf{x})$ is $(t + T_0)\alpha$ -strongly convex, using Eq. (2), we have that

$$\begin{aligned} \frac{(t + T_0)\alpha}{2}\|\mathbf{x}_t^* - \mathbf{x}_{t+1}^*\|^2 &\leq F_{t+1}(\mathbf{x}_t^*) - F_{t+1}(\mathbf{x}_{t+1}^*) \\ &= F_t(\mathbf{x}_t^*) - F_t(\mathbf{x}_{t+1}^*) + g_t(\mathbf{x}_t^*) - g_t(\mathbf{x}_{t+1}^*) \\ &\leq g_t(\mathbf{x}_t^*) - g_t(\mathbf{x}_{t+1}^*). \end{aligned}$$

The last inequality is since for every t , $F_t(\mathbf{x}_t^*) \leq F_t(\mathbf{x}_{t+1}^*)$. From the two last equations, we obtain

$$\|\mathbf{x}_t^* - \mathbf{x}_{t+1}^*\| \leq \frac{2(G + 2R\alpha)}{(t + T_0)\alpha}. \quad (7)$$

From Eq. (6) and Eq. (7), we have that

$$\begin{aligned} \sum_{t=1}^T g_t(\mathbf{x}_t^*) - g_t(\mathbf{x}_{t+1}^*) &\leq \sum_{t=1}^T \frac{2(G + 2R\alpha)^2}{(t + T_0)\alpha} \\ &\leq \frac{2(G + 2R\alpha)^2}{\alpha} (1 + \ln(T)). \end{aligned} \quad (8)$$

The last inequality is since $\sum_{t=1}^T \frac{1}{t+T_0} \leq 1 + \ln(T)$. Using Lemma 4, for all $\mathbf{x} \in \mathcal{K}$, we have that

$$\begin{aligned} \sum_{t=1}^T g_t(\mathbf{x}_t^*) - \sum_{t=1}^T g_t(\mathbf{x}) &\leq \sum_{t=1}^T g_t(\mathbf{x}_t^*) - g_t(\mathbf{x}_{t+1}^*) \\ &\quad + \frac{T_0\alpha}{2}\|\mathbf{x} - \mathbf{x}_1\|^2 \end{aligned} \quad (9)$$

By definition of $g_t(\mathbf{x})$, we have

$$\begin{aligned} \sum_{t=1}^T \nabla_t^\top (\mathbf{x}_t^* - \mathbf{x}^*) - \frac{\alpha}{2}\|\mathbf{x}_t - \mathbf{x}^*\|^2 &\leq \\ &\leq \sum_{t=1}^T \nabla_t^\top \mathbf{x}_t^* + \frac{\alpha}{2}\|\mathbf{x}_t^* - \mathbf{x}_t\|^2 - \nabla_t^\top \mathbf{x}^* - \frac{\alpha}{2}\|\mathbf{x}^* - \mathbf{x}_t\|^2 \\ &= \sum_{t=1}^T g_t(\mathbf{x}_t^*) - g_t(\mathbf{x}^*). \end{aligned}$$

Thus, applying Eq.(9) and Eq.(8), we obtain that

$$\begin{aligned} \sum_{t=1}^T \nabla_t^\top (\mathbf{x}_t^* - \mathbf{x}^*) - \frac{\alpha}{2}\|\mathbf{x}_t - \mathbf{x}^*\|^2 &\leq \\ &\leq \sum_{t=1}^T g_t(\mathbf{x}_t^*) - g_t(\mathbf{x}_{t+1}^*) + \frac{T_0\alpha}{2}\|\mathbf{x}^* - \mathbf{x}_1\|^2 \\ &\leq \frac{2(G + 2R\alpha)^2}{\alpha} (1 + \ln(T)) + 2\alpha R^2 T_0. \end{aligned}$$

Plugging-in the above bound together with Eq.(4) in Eq.(3), we obtain the result of the lemma:

$$\begin{aligned} \mathcal{R}_T &\leq \frac{2(G + 2R\alpha)^2}{\alpha} (1 + \ln(T)) + 2\alpha R^2 T_0 \\ &\quad + G \sum_{t=1}^T \sqrt{\frac{2\epsilon_t}{\alpha(t-1+T_0)}}. \end{aligned}$$

□

In order to complement Lemma 5, the following lemma sets an upper-bound on the sequence of approximation errors $\{\epsilon_t\}_{t \geq 1}$ which can be guaranteed throughout the run of Algorithm 1 using a single Frank-Wolfe step on each iteration (i.e., single call to the linear optimization oracle per iteration).

Lemma 6. Let $\{\epsilon_t = b\alpha(t + T_0)^{1/3}\}_{t=1}^T$, when $T_0 = \max\left\{1, \frac{2(G+2R\alpha)\sqrt{b}}{\alpha R^2}\right\}$ and $b = \max\left\{2\left(\frac{G+2R\alpha}{\alpha}\right)^2, \left(\frac{64(G+2R\alpha)R^2}{\alpha}\right)^{\frac{2}{3}}\right\}$. Then, throughout the run of Algorithm 1, for all time steps $t = 1, \dots, T$ it holds that $F_t(\mathbf{x}_t) - F_t(\mathbf{x}_t^*) \leq \epsilon_t$.

Proof. We will prove this lemma by induction. We will first start with a few preliminary steps.

Since the step-size $\sigma_t \in [0, 1]$ is chosen via line-search in Algorithm 1, we have that

$$F_{t+1}(\mathbf{x}_{t+1}) = F_{t+1}(\mathbf{x}_t + \sigma_t(\mathbf{v}_t - \mathbf{x}_t)) \leq F_{t+1}(\mathbf{x}_t).$$

Also, from the convexity of $F_t(\cdot)$ and the choice of \mathbf{v}_t , it follows that

$$\begin{aligned} F_{t+1}(\mathbf{x}_t) - F_{t+1}(\mathbf{x}_{t+1}^*) &\leq \nabla F_{t+1}(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{x}_{t+1}^*) \\ &\leq \nabla F_{t+1}(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{v}_t). \end{aligned}$$

Then, from the last two observations, we obtain

$$F_{t+1}(\mathbf{x}_{t+1}) - F_{t+1}(\mathbf{x}_{t+1}^*) \leq \nabla F_{t+1}(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{v}_t) \quad (10)$$

Since for every t , $F_t(\mathbf{x})$ is $\alpha(t-1+T_0)$ -strongly convex, using Eq.(2), we have that for any $\mathbf{x} \in \mathcal{K}$,

$$\|\mathbf{x} - \mathbf{x}_t^*\|^2 \leq \frac{2(F_t(\mathbf{x}) - F_t(\mathbf{x}_t^*))}{\alpha(t-1+T_0)} \quad (11)$$

Now we start the proof using induction.

Induction base: for $\tau = 1$, we have

$$F_1(\mathbf{x}_1) - F_1(\mathbf{x}_1^*) = 0 - T_0 \frac{\alpha}{2} \|\mathbf{x}_1^* - \mathbf{x}_1\|^2 \leq 0 \leq \epsilon_1.$$

Induction assumption: for $\tau = t$ it holds that

$$F_t(\mathbf{x}_t) - F_t(\mathbf{x}_t^*) \leq \epsilon_t.$$

Induction step: Let $\tau = t+1$ and we need to show that $F_{t+1}(\mathbf{x}_{t+1}) - F_{t+1}(\mathbf{x}_{t+1}^*) \leq \epsilon_{t+1}$. We start by bounding the initial gap, $F_{t+1}(\mathbf{x}_t) - F_{t+1}(\mathbf{x}_{t+1}^*)$ and then by bounding the improvement step $F_{t+1}(\mathbf{x}_{t+1}) - F_{t+1}(\mathbf{x}_t)$. Finally, we will combine them.

Since $F_t(\mathbf{x}_{t+1}^*) \geq F_t(\mathbf{x}_t^*)$, using Eq.(6), we have

$$\begin{aligned} F_{t+1}(\mathbf{x}_t) - F_{t+1}(\mathbf{x}_{t+1}^*) &= \\ &= F_t(\mathbf{x}_t) - F_t(\mathbf{x}_{t+1}^*) + g_t(\mathbf{x}_t) - g_t(\mathbf{x}_{t+1}^*) \\ &\leq F_t(\mathbf{x}_t) - F_t(\mathbf{x}_t^*) + (G + 2R\alpha) \|\mathbf{x}_t - \mathbf{x}_{t+1}^*\|. \end{aligned}$$

Using the induction assumption and the Cauchy-Schwarz inequality, we have

$$\begin{aligned} F_{t+1}(\mathbf{x}_t) - F_{t+1}(\mathbf{x}_{t+1}^*) &\leq \\ &\leq \epsilon_t + (G + 2R\alpha) \|\mathbf{x}_t - \mathbf{x}_t^*\| + (G + 2R\alpha) \|\mathbf{x}_t^* - \mathbf{x}_{t+1}^*\| \\ &\leq \epsilon_t + \frac{(G + 2R\alpha) \sqrt{2\epsilon_t}}{\sqrt{\alpha(t-1+T_0)}} + \frac{2(G + 2R\alpha)^2}{(t+T_0)\alpha}. \end{aligned} \quad (12)$$

The last inequality follows using Eq.(7), Eq.(11), and the induction assumption.

We now show that on the RHS of Eq.(12), the second term is larger than the third term. Since $\epsilon_t = b\alpha(t+T_0)^{1/3}$, for every t it holds that

$$\begin{aligned} \sqrt{\frac{2\epsilon_t}{\alpha(t-1+T_0)}} &= \sqrt{\frac{2b(t+T_0)^{1/3}}{(t-1+T_0)}} \geq \frac{\sqrt{2b}}{(t-1+T_0)^{1/3}} \\ &\geq \frac{2(G + 2R\alpha)}{(t+T_0)\alpha} \end{aligned}$$

The last inequality is since $b \geq 2 \left(\frac{G+2R\alpha}{\alpha}\right)^2$, $T_0 \geq 1$.

Then, plugging-in the value of ϵ_t in Eq.(12), using the last result, and since $\frac{(t+T_0)^{1/3}}{(t-1+T_0)} \leq \frac{2^{1/3}}{(t-1+T_0)^{2/3}}$, we obtain an upper bound for the initial gap:

$$F_{t+1}(\mathbf{x}_t) - F_{t+1}(\mathbf{x}_{t+1}^*) \leq \epsilon_t + \frac{4(G + 2R\alpha) \sqrt{b}}{(t-1+T_0)^{1/3}}. \quad (13)$$

Now we will analyze the improvement step. For our analysis we define the optimal step-size $\hat{\sigma}_t = \min \left\{ \frac{\nabla F_{t+1}(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{v}_t)}{\alpha(t+T_0)(2R)^2}, 1 \right\}$. Since σ_t is chosen via line-search, we have that

$$F_{t+1}(\mathbf{x}_{t+1}) \leq F_{t+1}(\mathbf{x}_t + \hat{\sigma}_t(\mathbf{v}_t - \mathbf{x}_t)).$$

Since $F_{t+1}(\mathbf{x})$ is $\alpha(t+T_0)$ -smooth, it holds that

$$\begin{aligned} F_{t+1}(\mathbf{x}_{t+1}) - F_{t+1}(\mathbf{x}_t) &\leq \hat{\sigma}_t \nabla F_{t+1}(\mathbf{x}_t)^\top (\mathbf{v}_t - \mathbf{x}_t) \\ &\quad + \frac{\alpha(t+T_0)}{2} \hat{\sigma}_t^2 \|\mathbf{v}_t - \mathbf{x}_t\|^2. \end{aligned}$$

We now consider several cases.

Case 1: If $\nabla F_{t+1}(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{v}_t) \leq \epsilon_{t+1}$, then from Eq.(10) we have

$$F_{t+1}(\mathbf{x}_{t+1}) - F_{t+1}(\mathbf{x}_{t+1}^*) \leq \epsilon_{t+1}.$$

Case 2: Else, $\nabla F_{t+1}(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{v}_t) \geq \epsilon_{t+1}$, then we have two sub cases:

Case 2.1: If $\alpha(t+T_0)(2R)^2 \leq \nabla F_{t+1}(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{v}_t)$, then $\hat{\sigma}_t = 1$ and we have

$$F_{t+1}(\mathbf{x}_{t+1}) - F_{t+1}(\mathbf{x}_t) \leq -\frac{\alpha(t+T_0)(2R)^2}{2}. \quad (14)$$

Case 2.2: Else, $\alpha(t+T_0)(2R)^2 \geq \nabla F_{t+1}(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{v}_t)$, then $\hat{\sigma}_t = \frac{\nabla F_{t+1}(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{v}_t)}{\alpha(t+T_0)(2R)^2}$ and we have

$$\begin{aligned} F_{t+1}(\mathbf{x}_{t+1}) - F_{t+1}(\mathbf{x}_t) &\leq -\frac{(\nabla F_{t+1}(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{v}_t))^2}{2\alpha(t+T_0)(2R)^2} \\ &\leq -\frac{\epsilon_{t+1}^2}{2\alpha(t+T_0)(2R)^2} \end{aligned} \quad (15)$$

Combining Eq.(13), Eq.(14), Eq.(15) and plugging-in the value of ϵ_{t+1} , we have

$$\begin{aligned} F_{t+1}(\mathbf{x}_{t+1}) - F_{t+1}(\mathbf{x}_{t+1}^*) &\leq \epsilon_t + \frac{4(G + 2R\alpha) \sqrt{b}}{(t-1+T_0)^{1/3}} \\ &\quad - \min \left\{ 2\alpha(t+T_0)R^2, \frac{b^2\alpha(t+1+T_0)^{2/3}}{8(t+T_0)R^2} \right\} \end{aligned} \quad (16)$$

Since $T_0 \geq \frac{2(G+2R\alpha)}{\alpha R^2} \sqrt{b}$, we have $2\alpha(t+T_0)R^2 \geq \frac{4(G+2R\alpha)\sqrt{b}}{(t-1+T_0)^{1/3}}$ and also, since $b \geq \left(\frac{64(G+2R\alpha)R^2}{\alpha}\right)^{2/3}$, we have $\frac{b^2\alpha(t+1+T_0)^{2/3}}{8(t+T_0)R^2} \geq \frac{4(G+2R\alpha)\sqrt{b}}{(t-1+T_0)^{1/3}}$. From both, we have that

$$\frac{4(G + 2R\alpha) \sqrt{b}}{(t-1+T_0)^{1/3}} \leq \min \left\{ 2\alpha(t+T_0)R^2, \frac{b^2\alpha(t+1+T_0)^{2/3}}{8(t+T_0)R^2} \right\}.$$

Then, from Eq.(16) and since $\epsilon_t \leq \epsilon_{t+1}$, we finally obtain $F_{t+1}(\mathbf{x}_{t+1}) - F_{t+1}(\mathbf{x}_{t+1}^*) \leq \epsilon_t \leq \epsilon_{t+1}$. \square

With all technical ingredients set in place, we can now prove Theorem 1.

Proof of Theorem 1. Using Lemma 5, Lemma 6, and plugging-in $T_0 = \max \left\{ 1, \frac{2(G+2R\alpha)\sqrt{b}}{\alpha R^2} \right\}$ and $\{\epsilon_t = b\alpha(t+T_0)^{1/3}\}_{t=1}^T$, we have

$$\begin{aligned} \mathcal{R}_T &\leq \frac{2(G+2R\alpha)^2}{\alpha} (1 + \ln(T)) + 4(G+2R\alpha)\sqrt{b} \\ &\quad + 2\alpha R^2 + \sqrt{2}G\sqrt{b} \sum_{t=1}^T \sqrt{\frac{(t+T_0)^{1/3}}{(t-1+T_0)}} \\ &\leq \frac{2(G+2R\alpha)^2}{\alpha} (1 + \ln(T)) + 4(G+2R\alpha)\sqrt{b} \\ &\quad + 2\alpha R^2 + 2\sqrt{2}G\sqrt{b}T^{2/3} \end{aligned}$$

Plugging in $b = \max \left\{ 2 \left(\frac{G+2R\alpha}{\alpha} \right)^2, \left(\frac{64(G+2R\alpha)R^2}{\alpha} \right)^{2/3} \right\}$, we obtain the regret bound in the theorem. \square

4 BANDIT SETTING

In this section we present our improved bound for the bandit-information setting. Our algorithm is simply that of Garber and Kretzu (2020) and is presented below as Algorithm 2. Due to lack of space, we refer the interested reader to Garber and Kretzu (2020) for more details.

Theorem 2. Suppose all loss functions are α -strongly convex for some $\alpha > 0$. For all $c > 0$ such that $\frac{cT^{-1/3}}{r} \leq 1$, setting $\delta = cT^{-1/3}$, $K = T^{2/3}$, $T_0 = \max \{4T^{2/3}, \frac{8}{\alpha}\}$, $\epsilon_m = 16R^2\beta_m^{1/3}$, when $\beta_m = \alpha(mK + T_0)$, in Algorithm 2, guarantees that the expected regret is upper-bounded by

$$\begin{aligned} \mathbb{E}[\mathcal{R}_T] &\leq \left(\frac{4cRG}{r} + 2R^2\alpha + \frac{4G}{\alpha^{1/3}} \right) T^{2/3} \\ &\quad + \left(\frac{2nM}{c\sqrt{\alpha}} + \frac{2G}{\sqrt{\alpha}} + 4R\sqrt{\alpha} \right)^2 T^{2/3} (1 + \ln(T)), \end{aligned}$$

and that the expected overall number of calls to the linear optimization oracle is upper-bounded by

$$\begin{aligned} \mathbb{E} \left[\sum_{m=1}^{\bar{K}} L_m \right] &\leq \left(\frac{nM}{c2R} + \frac{G}{2R} + \alpha \right) T \\ &\quad + \frac{1}{\alpha^{1/3}} \left(\frac{nM}{c2R} + \frac{G}{2R} + \alpha \right)^2 T. \end{aligned}$$

Algorithm 2: Online Bandit Frank-Wolfe (see Garber and Kretzu (2020))

Data: $T, r, R, K, \alpha, \delta \in (0, r]$, $\{\epsilon_m\}_{m=1}^{T/K}$, T_0
 $\mathbf{x}_0 \leftarrow$ arbitrary point in \mathcal{K}_δ , $\mathbf{x}_1 \leftarrow \mathbf{x}_0$
for $m = 1, \dots, \frac{T}{K}$ **do**
 define
 $\hat{F}_m(\mathbf{x}) := \sum_{i=1}^{m-1} \mathbf{x}^\top \hat{\mathbf{g}}_i + K \frac{\alpha}{2} \|\mathbf{x} - \mathbf{x}_{i-1}\|^2$
 $+ T_0 \frac{\alpha}{2} \|\mathbf{x} - \mathbf{x}_1\|^2$
 if $m > 1$ **then**
 run Algorithm 3 with set \mathcal{K}_δ , tolerance ϵ_m ,
 initial vector \mathbf{x}_{m-1} , and function $\hat{F}_m(\mathbf{x})$.
 Execute **in parallel** to the following **for**
 loop over s
 end
 for $s = 1, \dots, K$ **do**
 $\mathbf{u}_t \sim S^n$ $\triangleright t = (m-1)K + s$
 play $\mathbf{y}_t \leftarrow \mathbf{x}_{m-1} + \delta \mathbf{u}_t$ and observe $f_t(\mathbf{y}_t)$
 $\mathbf{g}_t \leftarrow \frac{n}{\delta} f_t(\mathbf{y}_t) \mathbf{u}_t$
 end
 $\hat{\mathbf{g}}_m \leftarrow \sum_{s=1}^K \mathbf{g}_{(m-1)K+s}$
 if $m > 1$ **then**
 $\mathbf{x}_m \leftarrow$ output of Algorithm 3
 end
end

Algorithm 3: Frank-Wolfe with Stopping Condition

Data: $\epsilon, \mathbf{x}_{in}, F_m(\mathbf{x})$
 $\mathbf{z}_1 \leftarrow \mathbf{x}_{in}$, $\tau \leftarrow 0$
do
 $\tau \leftarrow \tau + 1$
 $\mathbf{v}_\tau \in \operatorname{argmin}_{\mathbf{x} \in \mathcal{K}} \{\nabla F_m(\mathbf{z}_\tau)^\top \cdot \mathbf{x}\}$
 $\sigma_\tau = \operatorname{argmin}_{\sigma \in [0, 1]} \{F_m(\mathbf{z}_\tau + \sigma(\mathbf{v}_\tau - \mathbf{z}_\tau))\}$ \triangleright Line-search
 $\mathbf{z}_{\tau+1} = \mathbf{z}_\tau + \sigma_\tau(\mathbf{v}_\tau - \mathbf{z}_\tau)$ $\triangleright \mathbf{z}_{\tau+1} \in \mathcal{K}$
while $\nabla F_m(\mathbf{z}_\tau)^\top (\mathbf{z}_\tau - \mathbf{v}_\tau) > \epsilon$;
 $\mathbf{x}_{out} \leftarrow \mathbf{z}_\tau$

In particular, if $\frac{(nM)^2 \ln T}{r^2 G R \alpha} \leq T$ then, setting $c = (r(nM)^2 \ln T / G R \alpha)^{1/3}$, we have

$$\begin{aligned} \mathbb{E}[\mathcal{R}_T] &\leq \left(\frac{4(nMGR)^{2/3} (\ln T)^{1/3}}{\alpha^{1/3} r^{2/3}} + 2R^2\alpha + \frac{4G}{\alpha^{1/3}} \right) T^{2/3} \\ &\quad + \left(\frac{2(nMGR)^{1/3}}{\alpha^{1/6} r^{1/3} (\ln T)^{1/3}} + \frac{2G}{\sqrt{\alpha}} + 4R\sqrt{\alpha} \right)^2 T^{2/3} (1 + \ln(T)), \end{aligned}$$

and

$$\begin{aligned} \mathbb{E} \left[\sum_{m=1}^{\frac{T}{K}} L_m \right] &\leq \left(\frac{(nMG\alpha)^{\frac{1}{3}}}{2r^{\frac{1}{3}}R^{\frac{2}{3}}(\ln T)^{\frac{1}{3}}} + \frac{G}{2R} + \alpha \right) T \\ &\quad + \frac{1}{\alpha^{\frac{2}{3}}} \left(\frac{(nMG\alpha)^{\frac{1}{3}}}{2r^{\frac{1}{3}}R^{\frac{2}{3}}(\ln T)^{\frac{1}{3}}} + \frac{G}{2R} + \alpha \right)^2 T. \end{aligned}$$

4.1 Proof of Theorem 2

The following lemma is a key ingredient for obtaining improved bounds for the bandit setting. At a high-level it can be used to show that the ‘‘drift’’ in objective $\hat{F}_m(\cdot)$ from one round to the next, which is due to the gradients on each new block, as a first approximation, scales like \sqrt{K} (the second term on the RHS K^2G^2 will typically be smaller) and not linear in K . On the other-hand, during such a block of length K we can squeeze $O(K)$ linear optimization steps. This intuitively explains why the optimal tradeoff is to take block-length $K > 1$ (as opposed to the full-information setting).

Lemma 7. [Lemma 5 in Garber and Kretzu (2020)] For any iteration (block) m of the outer-loop in Algorithm 2 it holds that

$$\mathbb{E} [\|\hat{\mathbf{g}}_m\|^2] \leq \mathbb{E} [\|\hat{\mathbf{g}}_m\|^2] \leq K(nm\delta^{-1})^2 + K^2G^2.$$

The following lemma is analogous to Lemma 5 in the full-information setting, and upper-bounds the regret of Algorithm 2 in terms of the approximation error each iterate \mathbf{x}_m guarantees with respect to the corresponding objective $\hat{F}_m(\cdot)$ (which is captured by the scalar ϵ_m). The proof is given in the appendix.

Lemma 8. Let $\{\epsilon_m\}_{m=1}^{T/K} > 0$. Suppose that throughout the run of Algorithm 2, for all blocks $m = 1, \dots, \frac{T}{K}$ it holds that $\hat{F}_m(\mathbf{x}_m) - \hat{F}_m(\mathbf{x}_m^*) \leq \epsilon_m$. Then, the expected regret of the algorithm is upper-bounded by

$$\begin{aligned} \mathbb{E}[\mathcal{R}_T] &\leq \frac{4}{\alpha} \left(\frac{nM}{\delta} + \sqrt{K}(G + 2R\alpha) \right)^2 (1 + \ln T) \\ &\quad + 2\alpha R^2 T_0 + 3\delta GT + \delta \frac{R}{r} GT \\ &\quad + \sqrt{\frac{2}{\alpha}} GK \sum_{m=1}^{\frac{T}{K}} \sqrt{\frac{\epsilon_m}{(m-1)K + T_0}}. \end{aligned}$$

The following lemma is analogues to the use of Lemma 6 in the full-information setting, and is used to upper-bound the number of iterations required by the Frank-Wolfe method, Algorithm 3, to terminate on each invocation from Algorithm 2. A proof is given in the appendix.

Lemma 9. Let $\epsilon > 0$. Given a function $F(\mathbf{x})$, 2β -smooth, and $\mathbf{x}_1 \in \mathcal{K}_\delta$ such that $h_1 := F(\mathbf{x}_1) - F(\mathbf{x}^*)$, where $\mathbf{x}^* = \underset{\mathbf{x} \in \mathcal{K}_\delta}{\operatorname{argmin}} F(\mathbf{x})$, Algorithm 3 produces a point $\mathbf{x}_{L+1} \in \mathcal{K}_\delta$ such that $F(\mathbf{x}_{L+1}) - F(\mathbf{x}^*) \leq \epsilon$ after at most $L = \max \left\{ \frac{4\beta(2R)^2}{\epsilon^2}(h_1 - \epsilon), \frac{2}{\epsilon}(h_1 - \epsilon) \right\}$ iterations.

Proof of Theorem 2. We first upper bound the expected overall number of calls to the linear optimization oracle throughout the run of the algorithm, and then we upper-bound the expected regret.

Let $\mathbf{z}_{m,\tau}$ be the iterate of Algorithm 3 after completing $\tau - 1$ iterations of the do-while loop, when invoked on iteration (block) m of Algorithm 2. Also, for all m, τ , define $h_{m,\tau} := \hat{F}_m(\mathbf{z}_{m,\tau}) - \hat{F}_m(\mathbf{x}_m^*)$. Recall that for any iteration m of Algorithm 3, we have $\mathbf{z}_{m,1} = \mathbf{x}_{m-1}$.

Using the fact $\hat{F}_m(\mathbf{x}_{m+1}^*) \geq \hat{F}_m(\mathbf{x}_m^*)$, we have

$$\begin{aligned} h_{m+1,1} &= \hat{F}_{m+1}(\mathbf{z}_{m+1,1}) - \hat{F}_{m+1}(\mathbf{x}_{m+1}^*) \\ &\leq \hat{F}_m(\mathbf{x}_m) - \hat{F}_m(\mathbf{x}_m^*) + g_m(\mathbf{x}_m) - g_m(\mathbf{x}_{m+1}^*) \\ &\leq \epsilon_m + (\|\hat{\mathbf{g}}_m^\top\| + 2RK\alpha) \|\mathbf{x}_m - \mathbf{x}_{m+1}^*\|. \end{aligned}$$

The last inequality is since, $h_{m,L_m} = F_m(\mathbf{x}_m) - F_m(\mathbf{x}_m^*) \leq \epsilon_m$ and Eq.(22).

Using triangle inequality, we have

$$\begin{aligned} \mathbb{E}[h_{m+1,1}] &\leq \epsilon_m + \mathbb{E} [(\|\hat{\mathbf{g}}_m\| + 2RK\alpha) \|\mathbf{x}_m - \mathbf{x}_m^*\|] \\ &\quad + \mathbb{E} [(\|\hat{\mathbf{g}}_m\| + 2RK\alpha) \|\mathbf{x}_m^* - \mathbf{x}_{m+1}^*\|]. \end{aligned}$$

Since $h_{m,L_m} \leq \epsilon_m$, using Eq. (2), we have that $\|\mathbf{x}_m - \mathbf{x}_m^*\| \leq \sqrt{\frac{2\epsilon_m}{((m-1)K+T_0)\alpha}}$. Also, from Eq. (23), we have $\|\mathbf{x}_m^* - \mathbf{x}_{m+1}^*\| \leq \frac{2}{(mK+T_0)\alpha} (\|\hat{\mathbf{g}}_m\| + 2RK\alpha)$. Thus, we have

$$\begin{aligned} \mathbb{E}[h_{m+1,1}] &\leq \epsilon_m + \frac{\sqrt{2\epsilon_m} (\mathbb{E} [\|\hat{\mathbf{g}}_m\|] + 2RK\alpha)}{\sqrt{\alpha((m-1)K+T_0)}} \\ &\quad + \frac{2\mathbb{E} [\|\hat{\mathbf{g}}_m\| + 2RK\alpha]^2}{\alpha(mK+T_0)}. \end{aligned} \tag{17}$$

Using Lemma 9 with $h_1 = h_{m,1}$, then we have

$$\begin{aligned} \mathbb{E}[h_1] &\leq \epsilon_{m-1} + \frac{\sqrt{2\epsilon_{m-1}} (\mathbb{E} [\|\hat{\mathbf{g}}_{m-1}\|] + 2RK\alpha)}{\sqrt{\alpha((m-2)K+T_0)}} \\ &\quad + \frac{2\mathbb{E} [\|\hat{\mathbf{g}}_{m-1}\| + 2RK\alpha]^2}{\alpha((m-1)K+T_0)}. \end{aligned}$$

for $m = 1, \dots, \frac{T}{K}$, we have that on each iteration (block) m , the number of calls to the linear optimization oracle is $L_m \leq \max \left\{ \frac{2\beta_m(2R)^2}{\epsilon_m^2}(h_{m,1} - \epsilon_m) \right\}$

$\epsilon_m), \frac{2}{\epsilon_m}(h_{m,1} - \epsilon_m)\}$, when $\hat{F}_m(\mathbf{x})$ is $2\beta_m$ -Smooth,

note that $\beta_m = \alpha(mK + T_0)$. Since $\epsilon_m = 16R^2\beta_m^{\frac{1}{3}}$ and $T_0 \geq \frac{8}{\alpha}$ for any m we have $\beta_m(2R)^2 > \epsilon_m$, then we have $L_m \leq \frac{2\beta_m(2R)^2}{\epsilon_m^2}(h_{m,1} - \epsilon_m)$. Following Eq. (17), Lemma 7 and plugging in ϵ_m , we have

$$\begin{aligned} \mathbb{E}[L_m] &\leq \frac{1}{8R^2} \frac{\beta_m^{\frac{1}{3}}}{4} (\mathbb{E}[h_{m,1}] - \epsilon_m) \\ &\leq \frac{\sqrt{2}}{8R} \frac{\beta_m^{\frac{1}{2}}}{\beta_{m-2}^{\frac{1}{2}}} \left(\frac{\sqrt{KnM}}{\delta} + KG + 2RK\alpha \right) \\ &\quad + \frac{\beta_m^{\frac{1}{3}}}{16R^2\beta_{m-1}} \left(\frac{\sqrt{KnM}}{\delta} + KG + 2RK\alpha \right)^2. \end{aligned}$$

Since for every $m = 0, \dots, \frac{T}{K}$, $\beta_m = \frac{\alpha(mK + T_0)}{2}$ and $T_0 \geq 4K$, it holds that $\sum_{m=1}^{\frac{T}{K}} (\beta_m^{\frac{1}{2}}/\beta_{m-2}^{\frac{1}{2}}) \leq \sqrt{2} \frac{T}{K}$ and $\sum_{m=2}^{\frac{T}{K}} \frac{(\beta_m)^{\frac{1}{3}}}{\beta_{m-1}^{\frac{1}{2}}} \leq \frac{4}{\alpha^{\frac{2}{3}}} \frac{T^{\frac{1}{3}}}{K}$. Then, overall on all blocks, we obtain

$$\begin{aligned} \mathbb{E} \left[\sum_{m=1}^{\frac{T}{K}} L_m \right] &\leq \frac{1}{4R} \frac{T}{K} \left(\frac{\sqrt{KnM}}{\delta} + KG + 2RK\alpha \right) \\ &\quad + \frac{1}{4R^2\alpha^{\frac{2}{3}}} \frac{T^{\frac{1}{3}}}{K} \left(\frac{\sqrt{KnM}}{\delta} + KG + 2RK\alpha \right)^2 \\ &\stackrel{(d)}{\leq} \left(\frac{nM}{c4R} + \frac{G}{4R} + \frac{\alpha}{2} \right) T + \frac{1}{\alpha^{\frac{2}{3}}} \left(\frac{nM}{c2R} + \frac{G}{2R} + \alpha \right)^2 T. \end{aligned}$$

Inequality (d) is due to plugging-in δ, K .

We now turn to upper-bound the expected regret of the algorithm. using Lemma 8, plugging-in ϵ_m and the fact that $\sum_{m=1}^{\frac{T}{K}} \sqrt{\frac{(\beta_m)^{\frac{1}{3}}}{\beta_{m-1}}} \leq \frac{2}{\alpha^{\frac{2}{3}}} \frac{T^{\frac{2}{3}}}{K}$, we obtain

$$\begin{aligned} \mathbb{E}[\mathcal{R}_T] &\leq 3\delta GT + \delta \frac{R}{r} GT + 2\alpha R^2 T_0 + \frac{2\sqrt{2}}{\alpha^{\frac{1}{3}}} GT^{\frac{2}{3}} \\ &\quad + \frac{4}{\alpha} \left(\frac{nM}{\delta} + \sqrt{K}(G + 2R\alpha) \right)^2 (1 + \ln(T)) \\ &\stackrel{(d)}{\leq} \left(3cG + c \frac{R}{r} G + 2R^2\alpha + \frac{2\sqrt{2}G}{\alpha^{\frac{1}{3}}} \right) T^{\frac{2}{3}} \\ &\quad + \frac{4}{\alpha} \left(\frac{nM}{c} + G + 2R\alpha \right)^2 T^{\frac{2}{3}} (1 + \ln(T)). \end{aligned}$$

Equality (d) is due to plugging-in δ, K, T_0 . \square

5 DISCUSSION

We have proved strong convexity leads to faster rates for projection-free online learning both in the full-

information and bandit settings. This is obtained via the standard Online Frank-Wolfe method (Hazan and Kale, 2012) and its bandit variant with blocks (Garber and Kretzu, 2020). In particular, this improvement is in stark contrast to the state-of-affairs in offline convex optimizaiton, in which, in general, strong convexity does not lead to faster rates for the Frank-Wolfe method.

In light of this current work and the recent work, Hazan and Minasyan (2020), it is interesting if the combination of smoothness and strong convexity can lead to a faster rate than $O(T^{2/3})$. Also, given the recent interest in projection-free online learning, and in particular with a linear optimization oracle, it would be interesting to derive lower bounds on the regret of such algorithms, perhaps with online variants of standard constructions used in the offline setting (see for instance Jaggi (2013); Garber and Hazan (2016)).

6 Acknowledgments

This research was supported by the ISRAEL SCIENCE FOUNDATION (grant No. 1108/18).

References

- Bellet, A., Liang, Y., Garakani, A. B., Balcan, M.-F., and Sha, F. (2015). A distributed frank-wolfe algorithm for communication-efficient sparse learning. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 478–486. SIAM.
- Bertsekas, D. P. (1973). Stochastic optimization problems with nondifferentiable cost functionals. *Journal of Optimization Theory and Applications*, 12(2):218–231.
- Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.
- Chen, L., Zhang, M., and Karbasi, A. (2019). Projection-free bandit convex optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, pages 2047–2056.
- Flaxman, A. D., Kalai, A. T., Kalai, A. T., and McMahan, H. B. (2005). Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394. Society for Industrial and Applied Mathematics.
- Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:149–154.
- Garber, D. and Hazan, E. (2013). Playing non-linear games with linear oracles. In *54th Annual IEEE*

Symposium on Foundations of Computer Science, FOCS.

Garber, D. and Hazan, E. (2015). Faster rates for the frank-wolfe method over strongly-convex sets. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 541–549.

Garber, D. and Hazan, E. (2016). A linearly convergent variant of the conditional gradient algorithm under strong convexity, with applications to online and stochastic optimization. *SIAM Journal on Optimization*, 26(3):1493–1528.

Garber, D. and Kaplan, A. (2019). Fast stochastic algorithms for low-rank and nonsmooth matrix problems. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 286–294. PMLR.

Garber, D. and Kretzu, B. (2020). Improved regret bounds for projection-free bandit convex optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2196–2206. PMLR.

Goldfarb, D., Iyengar, G., and Zhou, C. (2017). Linear convergence of stochastic frank wolfe variants. In *Artificial Intelligence and Statistics*, pages 1066–1074.

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

Hazan, E., Kalai, A., Kale, S., and Agarwal, A. (2006). Logarithmic regret algorithms for online convex optimization. In Lugosi, G. and Simon, H. U., editors, *Learning Theory, 19th Annual Conference on Learning Theory, COLT 2006, Pittsburgh, PA, USA, June 22-25, 2006, Proceedings*, volume 4005 of *Lecture Notes in Computer Science*, pages 499–513. Springer.

Hazan, E. and Luo, H. (2016). Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, pages 1263–1271.

Hazan, E. and Minasyan, E. (2020). Faster projection-free online learning. In Abernethy, J. D. and Agarwal, S., editors, *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pages 1877–1893. PMLR.

Hazan, E. E. and Kale, S. (2012). Projection-free online learning. In *29th International Conference on Machine Learning, ICML 2012*, pages 521–528.

Jaggi, M. (2013). Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML (1)*, pages 427–435.

Lacoste-Julien, S. and Jaggi, M. (2015). On the global linear convergence of Frank-Wolfe optimization variants. In *Advances in Neural Information Processing Systems*, pages 496–504.

Levitin, E. S. and Polyak, B. T. (1966). Constrained minimization methods. *USSR Computational mathematics and mathematical physics*, 6:1–50.

Wang, Y.-X., Sadhanala, V., Dai, W., Neiswanger, W., Sra, S., and Xing, E. (2016). Parallel and distributed block-coordinate frank-wolfe algorithms. In *International Conference on Machine Learning*, pages 1548–1557.