Proceedings of Machine Learning Research vol 178:1–34, 2022 35th Annual Conference on Learning Theory

# New Projection-free Algorithms for Online Convex Optimization with Adaptive Regret Guarantees

**Dan Garber** DANGAR@TECHNION.AC.IL
**Ben Kretzu** BENKRETZU@CAMPUS.TECHNION.AC.IL
*Technion - Israel Institute of Technology*

**Editors:** Po-Ling Loh and Maxim Raginsky

## Abstract

We present new efficient *projection-free* algorithms for online convex optimization (OCO), where by projection-free we refer to algorithms that avoid computing orthogonal projections onto the feasible set, and instead relay on different and potentially much more efficient oracles. While most state-of-the-art projection-free algorithms are based on the *follow-the-leader* framework, our algorithms are fundamentally different and are based on the *online gradient descent* algorithm with a novel and efficient approach to computing so-called *infeasible projections*. As a consequence, we obtain the first projection-free algorithms which naturally yield *adaptive regret* guarantees, i.e., regret bounds that hold w.r.t. any sub-interval of the sequence. Concretely, when assuming the availability of a linear optimization oracle (LOO) for the feasible set, on a sequence of length $T$, our algorithms guarantee $O(T^{3/4})$ adaptive regret and $O(T^{3/4})$ adaptive expected regret, for the full-information and bandit settings, respectively, using only $O(T)$ calls to the LOO. These bounds match the current state-of-the-art regret bounds for LOO-based projection-free OCO, which are *not adaptive*. We also consider a new natural setting in which the feasible set is accessible through a separation oracle. We present algorithms which, using overall $O(T)$ calls to the separation oracle, guarantee $O(\sqrt{T})$ adaptive regret and $O(T^{3/4})$ adaptive expected regret for the full-information and bandit settings, respectively.

**Keywords:** projection-free methods, online convex optimization, online learning, Frank-Wolfe, linear optimization oracle

## 1. Introduction

In this paper we consider the problem of Online Convex Optimization (OCO) Hazan (2019); Shalev-Shwartz et al. (2012) with a particular focus on so-called *projection-free* algorithms. Such algorithms are motivated by high-dimensional problems in which the feasible decision set admits a non-trivial structure and thus, computing orthogonal projections onto it, as required by standard methods, is often computationally prohibitive. Instead, projection-free methods access the decision set through a conceptually simpler oracle which in many cases of interest admits a much more efficient implementation than that of an orthogonal projection oracle. Indeed, for this reason such algorithms have drawn significant interest in recent years, see for instance Hazan and Kale (2012); Garber and Hazan (2013); Chen et al. (2019); Garber and Kretzu (2020); Kretzu and Garber (2021); Hazan and Minasyan (2020); Levy and Krause (2019); Wan and Zhang (2021); Ene et al. (2021); Chen et al. (2018); Zhang et al. (2017).

Let us introduce some formalism before moving on. Throughout the paper we assume without loosing much generality that the underlying vector space is $\mathbb{R}^n$. We recall that in OCO, a decision maker (DM) is required throughout $T$ iterations (we will assume throughout that $T$ is known in advanced for ease of presentation), to pick on each iteration $t \in [T]$, a decision in the form of a point $\mathbf{x}_t$ from some fixed convex and compact decision set $\mathcal{K} \subset \mathbb{R}^n$. After choosing $\mathbf{x}_t \in \mathcal{K}$, the DM incurs a loss given by $f_t(\mathbf{x}_t)$, where $f_t : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex[1]. We will make the standard distinction between the *full-information* setting, in which after incurring the loss, the DM gets to observe the loss function $f_t(\cdot)$, and the *bandit* setting, in which the DM only learns the value $f_t(\mathbf{x}_t)$. In the full-information setting we shall assume that the sequence of losses $f_1, \dots, f_T$ is arbitrary, and may even depend on the plays of the DM, while in the bandit setting we shall make a standard simplifying assumption that $f_1, \dots, f_T$ are chosen in *oblivious* fashion, i.e., before the DM has made his first step (and thus are in particular independent of any randomness introduced by the DM). We recall that the standard measure of performance in OCO, which is also the objective that the DM usually strives to minimize, is the *regret* (or its expectation in the bandit setting) which, given the entire history $\{\mathbf{x}_t, f_t\}_{t=1}^T$, is given by

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

Most projection-free OCO algorithms are based on a combination of the *Follow-The-Leader* (FTL) meta-algorithm, and in particular its deterministically regularized variant known as *Regularized-Follow-The-Leader* (RFTL) Hazan (2019), and the use of a linear optimization oracle (LOO) to access the feasible set, e.g., Hazan and Kale (2012); Chen et al. (2019); Garber and Kretzu (2020). We shall refer to these as RFTL-LOO algorithms. Indeed, for many feasible sets of interest and in high-dimensional settings, implementing the LOO can be much more efficient than implementing an orthogonal projection oracle, see many examples in Jaggi (2013); Hazan and Kale (2012). For arbitrary (convex and compact) feasible set and nonsmooth convex losses, the current best regret bound for both the full-information and bandit settings obtainable by these RFTL-LOO algorithms is $O(T^{3/4})$, using overall $O(T)$ calls to the LOO, due to Hazan and Kale (2012) and Garber and Kretzu (2020).

However, the RFTL approach for constructing online algorithms has well known inherent limitations. While the regret, as given in (1), can in principle be negative — due to the ability of the online algorithm to change decisions from iteration to iteration while the benchmark's decision is fixed, it is known that RFTL-type algorithms *always* suffer non-negative regret Gofer and Mansour (2016). As a consequence, such algorithms are also inherently *non-adaptive* in a sense that we now detail. It is often the case that there is no fixed decision in hindsight that has reasonable performance w.r.t. the entire data (i.e., the sequence of loss functions) and thus, the standard regret measure becomes insufficient. In such cases, *adaptive* performance measures which, on different parts of the data, allow to be competitive against different actions, are much more preferable. Such standard adaptive performance measure introduced in Hazan and Seshadhri (2009) is called *adaptive regret* and is given by

$$\text{Adaptive Regret} = \sup_{[s,e] \subseteq [T]} \left\{ \sum_{t=s}^{e} f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{K}} \sum_{t=s}^{e} f_t(\mathbf{x}) \right\}, \quad (2)$$

where for any two integers $1 \leq s \leq e \leq T$, we denote $[s, e] = \{s, s+1, \dots, e\}$. In words, the adaptive regret is the supremum over all standard regrets w.r.t. all sub-intervals of the sequence of loss functions. We refer the interested reader to Hazan and Seshadhri (2009); Daniely et al. (2015) for many useful discussions on the adaptive regret and its connection to other notions of adaptivity in the literature.

1. In fact, it suffices that $f_t$ is convex on a certain Euclidean ball containing the set $\mathcal{K}$.

Unfortunately, due to their inherent non-negative regret property, RFTL-based algorithms cannot guarantee non-trivial adaptive regret bounds. Thus, it is natural to ask:

*Is it possible to design efficient projection-free algorithms for OCO with non-trivial adaptive regret bounds?*

One attempt towards this goal could be to instantiate the *strongly adaptive online learner* of Daniely et al. (2015) with the non-adaptive state-of-the-art RFTL based algorithm for the full-information setting of Hazan and Kale (2012), known as *Online Frank-Wolfe* (OFW), which will result in an adaptive algorithm with $O(T^{3/4})$ adaptive regret.[2] However, this approach is somewhat artificial and will require to run in parallel $O(\log T)$ copies of OFW, which will require $\log T$-fold memory and calls to the LOO. Moreover, this approach is not applicable to the bandit setting.

Another possibility is to design new projection-free algorithms which are not based on the FTL approach, but instead on the Online Mirror Descent meta-algorithm, and in particular its Euclidean variant — Online Gradient Descent (OGD) Zinkevich (2003), which naturally yields an $O(\sqrt{T})$ adaptive regret bound Hazan (2019). While OGD requires to compute on each iteration an orthogonal projection onto the feasible set, a naive approach to making it projection-free using a LOO, is to only approximate the projection on each iteration via the well known Frank-Wolfe method for *offline* constrained minimization of a smooth and convex function, which only uses the LOO Jaggi (2013); Frank and Wolfe (1956). However, as recently noted in Garber (2021), such an approach strikes an highly suboptimal tradeoff between regret and number of calls to the LOO. Instead, Garber (2021) considered using OGD with so-called *infeasible projections*, which on one hand can be computed efficiently with a LOO (at least in terms of the model in Garber (2021) which is significantly different than ours), and on the other-hand could be translated into feasible points, without loosing too much in the regret. Our approach in this paper is inspired by Garber (2021), however, our technique for computing such infeasible projections will be very different (in particular, the setting in Garber (2021) is not concerned with the dimension and thus the Ellipsoid method is used, which is not suitable for our setting, due to its polynomial dependence on the dimension).

**Two projection-free oracles:** While our discussion so far has focused on the assumption that the feasible set is accessible through a linear optimization oracle, which is indeed the most popular assumption in the literature on projection-free methods, in this paper we introduce an additional new natural projection-free setting in which the feasible decision set $\mathcal{K}$ is given by separation oracle (SO). Given some $\mathbf{x} \in \mathcal{K}$, the SO either verifies that $\mathbf{x}$ is feasible, in case it indeed holds that $\mathbf{x} \in \mathcal{K}$ or, returns a hyperplane separating $\mathbf{x}$ from $\mathcal{K}$, in case $\mathbf{x} \notin \mathcal{K}$. For instance, a setting in which the SO model arrises naturally is when the feasible set is given by a functional constraint of the form $\mathcal{K} = \{\mathbf{x} \in \mathbb{R}^n \mid g(\mathbf{x}) \leq 0\}$, where $g(\cdot)$ is convex. Implementing the SO in this setting simply amounts to calling the first-order oracle of $g(\cdot)$ (i.e., computing $g(\mathbf{x})$ and some $\mathbf{g}_{\mathbf{x}} \in \partial g(\mathbf{x})$, for a given input point $\mathbf{x} \in \mathbb{R}^n$). In particular, when $g(\cdot)$ has the following max structure: $g(\mathbf{x}) := \max_{1\leq i\leq m} g_i(\mathbf{x})$, where $m$ is not very large and $g_1, \dots, g_m$ are convex functions which admit simple structure, implementing the SO can be very efficient, while orthogonal projections can still be prohibitive. One such example is a polytope given by the intersection of $m$ halfspaces for moderately-large $m$. Importantly, the SO model allows to efficiently handle the intersection of several simple convex sets, each given by a SO. Note that in the LOO setting there is no simple approach to implement a LOO for a convex set given as the intersection of several sets, each given by a LOO.

2. In fact, such an algorithm will have for any interval $I$, regret bounded by $O(|I|^{3/4} + |I|^{1/2} \log T)$ w.r.t. the interval.

The following example demonstrates the complementing nature of the LOO and SO oracles. Consider the following two, dual to each other, unit balls of matrices which are common in several applications: $\mathcal{B}_* = \{\mathbf{X} \in \mathbb{R}^{m\times n} \mid \|\mathbf{X}\|_* \leq 1\}$, $\mathcal{B}_2 = \{\mathbf{X} \in \mathbb{R}^{m\times n} \mid \|\mathbf{X}\|_2 \leq 1\}$, where for a real matrix $\mathbf{X}$ we let $\|\mathbf{X}\|_*$ denote its nuclear/trace norm, i.e., the sum of singular values, and we let $\|\mathbf{X}\|_2$ denote its spectral norm, i.e., its largest singular value. Euclidean projection onto either $\mathcal{B}_*$ or $\mathcal{B}_2$ requires in general a full-rank singular value decomposition (SVD), which is computationally prohibitive when both $m, n$ are very large. Linear optimization over $\mathcal{B}_*$ is quite efficient and only requires a rank-one SVD (leading singular vectors computations) however, linear optimization over $\mathcal{B}_2$ requires again a full-rank SVD Jaggi (2013). On the other-hand, denoting $g_*(\mathbf{X}) := \|\mathbf{X}\|_* - 1, g_2(\mathbf{X}) := \|\mathbf{X}\|_2 - 1$, we have that implementing the SO for $\mathcal{B}_*$, which requires to compute a subgradient of the nuclear norm, also requires in worst case a full-rank SVD. However, implementing the SO w.r.t. $\mathcal{B}_2$, requires to compute a subgradient of the spectral norm, which is w.l.o.g. a rank-one matrix (corresponding to a top singular vectors pair of $\mathbf{X}$), and thus requires only a rank-one SVD which is far more efficient. Thus, while a LOO is efficient to implement for $\mathcal{B}_*$, the SO is efficient to implement for $\mathcal{B}_2$.

**Contributions:** Our main contributions, stated only informally at this stage, and treating all quantities except for $T$ and the dimension $n$ as constants, are as follows (see also a summary in Table 1).

1. Assuming the feasible set is accessible through a LOO, we present an OGD-based algorithm for the full-information setting with adaptive regret of $O(T^{3/4})$ using overall $O(T)$ calls to the LOO. This improves over the previous state-of-the-art (RFTL-based) *not-adaptive* regret bound of $O(T^{3/4})$ due to Hazan and Kale (2012). We give a similar algorithm for the bandit setting which guarantees $O(\sqrt{n}T^{3/4})$ adaptive expected regret using $O(T)$ calls to the LOO in expectation, which improves upon the previous best bound of $O(\sqrt{n}T^{3/4})$ due to Garber and Kretzu (2020) which only applies to the standard regret.

2. Assuming the feasible set is accessible through a LOO and all loss functions are strongly convex, we show that a projection-free OGD-based algorithm can recover the state-of-the-art $O(T^{2/3})$ (standard) regret bound using $O(T)$ calls to the LOO, which matches that of the RFTL-based method due to Kretzu and Garber (2021).

3. Assuming the feasible set is accessible through a SO, we present an OGD-based algorithm for the full-information setting with adaptive regret of $O(\sqrt{T})$ using overall $O(T)$ calls to the SO. In the bandit setting, we give a similar algorithm with $O(T^{3/4})$ adaptive expected regret using overall $O(T)$ calls to the SO.

We remark that aside of standard subgradient computations of the loss functions observed, and calls to either the LOO or SO, all of our algorithms require only $O(n)$ space, and $O(nT)$ additional runtime (over all $T$ iterations).

We acknowledge a parallel work Mhammedi (2021), in which the author proves that given a separation oracle, it is possible to guarantee a $O(\sqrt{T})$ regret bound for general Lipschitz convex losses, and the techniques could be readily used to also give adaptive regret guarantees in the full information setting (but not in the bandit setting). However, the approach of Mhammedi (2021), which uses substantially different techniques than ours, requires overall $O(T \log T)$ calls to the separation oracle to guarantee $O(\sqrt{T})$ regret, while our result only requires $O(T)$ calls in order to achieve this regret bound.

| | Theorem 9 | Theorem 11 | Theorem 10 | Theorem 14 | Theorem 15 |
|---|---|---|---|---|---|
| Objective | adaptive regret | adaptive expected regret | regret | adaptive regret | adaptive expected regret |
| Losses | convex | convex | strongly convex | convex | convex |
| Feedback | full | bandit | full | full | bandit |
| Oracle | LOO | LOO | LOO | SO | SO |
| Regret | $T^{3/4}$ | $T^{3/4}$ | $T^{2/3}$ | $\sqrt{T}$ | $T^{3/4}$ |

**Table 1:** Summary of results. For clarity, in the regret bounds we treat all quantities except for $T$ as constants.

| Feedback | Objective | Oracle | Reference | Regret |
|---|---|---|---|---|
| Full Information | adaptive regret | projection | Zinkevich (2003) | $\sqrt{T}$ |
| | adaptive regret | SO | This work (Thm. 14) | $\sqrt{T}$ |
| | regret | LOO | Hazan and Kale (2012) | $T^{3/4}$ |
| | adaptive regret | LOO | This work (Thm. 9) | $T^{3/4}$ |
| Bandit | adaptive regret | projection | Flaxman et al. (2005) | $T^{3/4}$ |
| | adaptive regret | SO | This work (Thm. 15) | $T^{3/4}$ |
| | regret | LOO | Garber and Kretzu (2020) | $T^{3/4}$ |
| | adaptive regret | LOO | This work (Thm. 11) | $T^{3/4}$ |

**Table 2:** Comparison of results to previous works. This is a non-exhaustive list. Here we only list the most relevant works which are suitable for arbitrary convex and compact sets and convex and nonsmooth losses, make overall $O(T)$ calls to the oracle of the set, and use $O(n)$ memory and $O(nT)$ additional runtime. For clarity, in the regret bounds we treat all quantities except for $T$ as constants.

We note that due to lack of space some of the results and proofs are deferred to the appendix.

## 2. Preliminaries

### 2.1. Additional notation, assumptions and definitions

Throughout this work we assume without loss of generality that the feasible set $\mathcal{K}$ contains the origin, i.e., $\mathbf{0} \in \mathcal{K}$ and we denote by $R > 0$ a radius such that $\mathcal{K} \subseteq R\mathcal{B}$, where $\mathcal{B}$ denotes the unit Euclidean ball centered at the origin. We also denote by $\mathcal{S}$ the unit sphere centered at the origin, and we write $\mathbf{u} \sim \mathcal{B}$ and $\mathbf{u} \sim \mathcal{S}$ to denote a random vector $\mathbf{u}$ sampled uniformly from $\mathcal{B}$ and $\mathcal{S}$, respectively. We assume the loss functions are bounded by $M$ in $\ell_\infty$ norm and are $G_f$-Lipschitz over $R\mathcal{B}$, that is, for all $t \in [T]$, $\mathbf{x} \in R\mathcal{B}$ and $\mathbf{g} \in \partial f_t(\mathbf{x})$, $|f_t(\mathbf{x})| \leq M$ and $\|\mathbf{g}\|_2 \leq G_f$.

In our results for the bandit feedback setting and when assuming the feasible set is accessible through a SO we shall make the following additional standard assumption.

**Assumption 1** *The feasible set fully contains the ball of radius $r$ around $\mathbf{0}$, for some $r > 0$, i.e., $r\mathcal{B} \subseteq \mathcal{K}$.*

For every $\delta \in (0, 1)$ we define the $\delta$-squeezed version of $\mathcal{K}$ as $\mathcal{K}_\delta = (1-\delta)\mathcal{K} = \{(1-\delta)\mathbf{x} \mid \mathbf{x} \in \mathcal{K}\}$. Note that if Assumption 1 holds, then for all $\mathbf{x} \in \mathcal{K}_{\delta/r}$, it holds that $\mathbf{x}+\delta\mathcal{B} \subseteq \mathcal{K}$ (see Hazan (2019)).

### 2.2. Basic algorithmic tools

#### 2.2.1. THE FRANK-WOLFE ALGORITHM WITH LINE SEARCH

The Frank-Wolfe algorithm Frank and Wolfe (1956); Jaggi (2013) is a well known first-order method for minimizing a smooth and convex function over a convex and compact set, accessible through a LOO. In this work we use the Frank-Wolfe with exact line-search variant, see Algorithm 8 in the appendix.

**Theorem 1** *[Primal convergence of FW Jaggi (2013)] Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be convex and $\beta$-smooth over a convex and compact set $\mathcal{K} \subset \mathbb{R}^n$ with Euclidean diameter $2R$, and denote $\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} f(\mathbf{x})$. Algorithm 8 guarantees that $\forall i \geq 1 : f(\mathbf{x}_i) - f(\mathbf{x}^*) \leq 2\beta(2R)^2/(i+2)$.*

**Theorem 2** *[Dual convergence of FW Jaggi (2013)] Under the same assumptions of Theorem 1, Algorithm 8 guarantees that for every number of iterations $K \geq 2$, there exists an iteration $i$, $K \geq i \geq 2$, such that $\max_{\mathbf{v}\in\mathcal{K}}(\mathbf{x}_i - \mathbf{v})^\top \nabla f(\mathbf{x}_i) \leq 6.75\beta(2R)^2/(K+2)$.*

Note that for a convex function $f(\cdot)$ and a feasible point $\mathbf{x} \in \mathcal{K}$, the dual gap in Theorem 2 serves as an easy-to-compute certificate for the optimality gap of $\mathbf{x}$ w.r.t. any optimal solution $\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{y}\in\mathcal{K}} f(\mathbf{y})$, since from the convexity of $f(\cdot)$ it follows that, $f(\mathbf{x}) - f(\mathbf{x}^*) \leq (\mathbf{x} - \mathbf{x}^*)^\top \nabla f(\mathbf{x}) \leq \max_{\mathbf{v}\in\mathcal{K}}(\mathbf{x} - \mathbf{v})^\top \nabla f(\mathbf{x})$.

#### 2.2.2. Online Gradient Descent Without Feasibility

As discussed, our online algorithms are based on the well known *Online Gradient Descent* method (OGD) Zinkevich (2003), which applies the following updates:

$$\forall t > 1 : \quad \mathbf{y}_{t+1} \leftarrow \mathbf{x}_t - \eta_t \mathbf{g}_t,\ \mathbf{g}_t \in \partial f_t(\mathbf{x}_t),\ \ \mathbf{x}_{t+1} \leftarrow \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} \|\mathbf{x} - \mathbf{y}_{t+1}\|^2.$$

Where $\{\eta_t\}_{t=1}^T$ are the step-sizes and $\mathbf{x}_1$ is an arbitrary feasible point. However, motivated by Garber (2021), instead of considering exact projections on the feasible set, which may be computationally prohibitive, we consider using only *infeasible projections*, as we now define.

**Definition 3** *We say $\tilde{\mathbf{y}} \in \mathbb{R}^n$ is an infeasible projection of some $\mathbf{y} \in \mathbb{R}^n$ onto a convex set $\mathcal{K}$, if $\forall \mathbf{z} \in \mathcal{K}$ it holds that $\|\tilde{\mathbf{y}} - \mathbf{z}\|^2 \leq \|\mathbf{y} - \mathbf{z}\|^2$. We say a function $\mathcal{O}_{IP}(\mathbf{y}, \mathcal{K})$ is an infeasible projection oracle for the set $\mathcal{K}$, if for every input point $\mathbf{y}$, it returns some $\tilde{\mathbf{y}} \leftarrow \mathcal{O}_{IP}(\mathbf{y}, \mathcal{K})$ which is an infeasible projection of $\mathbf{y}$ onto $\mathcal{K}$.*

This definition gives rise to the online gradient descent without feasibility algorithm — Algorithm 1, and its corresponding regret bounds captured in Lemma 4. While this algorithm will play a central role in our projection-free online algorithms, clearly, another central piece, which we will detail later on, will be to transform such infeasible projections into feasible points without loosing too much in the regret bound.

---

**Algorithm 1:** Online Gradient Descent Without Feasibility

---

**Data:** horizon $T$, feasible set $\mathcal{K}$, step-sizes $\{\eta_t\}_{t=1}^T$, infeasible projection oracle $\mathcal{O}_{IP}(\mathcal{K}, \cdot)$
$\tilde{\mathbf{y}}_1 \leftarrow$ arbitrary point in $\mathcal{K}$
**for** $t = 1, \ldots, T$ **do**
  Play $\tilde{\mathbf{y}}_t$, observe $f_t(\tilde{\mathbf{y}}_t)$, and set $\nabla_t \in \partial f_t(\tilde{\mathbf{y}}_t)$
  Update $\mathbf{y}_{t+1} = \tilde{\mathbf{y}}_t - \eta_t \nabla_t$, and set $\tilde{\mathbf{y}}_{t+1} \leftarrow \mathcal{O}_{IP}(\mathcal{K}, \mathbf{y}_{t+1})$
**end**

---

**Lemma 4** *Let $\mathcal{O}_{IP}$ an infeasible projection oracle (Definition 3).*

1. *Suppose all loss functions are convex. Fix some $\eta > 0$ and let $\eta_t = \eta$ for all $t \geq 1$. Algorithm 1 guarantees that the adaptive regret is upper-bounded as follows:*

$$\forall I = [s, e] \subseteq [T]: \quad \sum_{t=s}^{e} f_t(\tilde{\mathbf{y}}_t) - \min_{\mathbf{x}_I \in \mathcal{K}} \sum_{t=s}^{e} f_t(\mathbf{x}_I) \leq \frac{\|\tilde{\mathbf{y}}_s - \mathbf{x}_I\|^2}{2\eta} + \frac{\eta}{2} \sum_{s=1}^{e} \|\nabla_t\|^2.$$

2. *Suppose all loss functions are $\alpha$-strongly convex for some $\alpha > 0$. Let $\eta_t = \frac{1}{\alpha t}$ for all $t \geq 1$. Algorithm 1 guarantees that the (static) regret is upper-bounded as follows:*

$$\sum_{t=1}^{T} f_t(\tilde{\mathbf{y}}_t) - \min_{\mathbf{x} \in \mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}) \leq \sum_{t=1}^{T} \|\nabla_t\|^2 / 2\alpha t.$$

#### 2.2.3. INFEASIBLE PROJECTIONS VIA SEPARATING HYPERPLANES

Continuing the discussion on infeasible projections, our approach for transforming such infeasible projections into feasible points without sacrificing the regret bounds too much, will be to design infeasible projection oracles that always return points that are sufficiently close to the feasible set. The following simple lemma will be instrumental to all of our constructions of such oracles, and shows how using a separating hyperplane we can "pull" an infeasible point closer to the feasible set.

**Lemma 5** *Let $\mathcal{K} \subset \mathbb{R}^n$ be convex and compact, let $\mathbf{y}$ be infeasible w.r.t. $\mathcal{K}$, i.e., $\mathbf{y} \notin \mathcal{K}$, and let $\mathbf{g} \in \mathbb{R}^n$ be a separating hyperplane such that for all $\mathbf{z} \in \mathcal{K}$: $(\mathbf{y} - \mathbf{z})^\top \mathbf{g} \geq Q$, for some $Q \geq 0$. Consider the point $\tilde{\mathbf{y}} = \mathbf{y} - \gamma \mathbf{g}$, for $\gamma = Q/C^2$, where $C \geq \|\mathbf{g}\|$. It holds that $\forall \mathbf{z} \in \mathcal{K}$: $\|\tilde{\mathbf{y}} - \mathbf{z}\|^2 \leq \|\mathbf{y} - \mathbf{z}\|^2 - (Q/C)^2$.*

**Proof** Fix some $\mathbf{z} \in \mathcal{K}$. It holds that

$$\|\tilde{\mathbf{y}} - \mathbf{z}\|^2 = \|\mathbf{y} - \mathbf{z} - \gamma \mathbf{g}\|^2 \leq \|\mathbf{y} - \mathbf{z}\|^2 - 2\gamma(\mathbf{y} - \mathbf{z})^\top \mathbf{g} + \gamma^2 C^2.$$

Since $(\mathbf{y} - \mathbf{z})^\top \mathbf{g} \geq Q$, we indeed obtain

$$\|\tilde{\mathbf{y}} - \mathbf{z}\|^2 \leq \|\mathbf{y} - \mathbf{z}\|^2 - 2\gamma Q + \gamma^2 C^2 \leq \|\mathbf{y} - \mathbf{z}\|^2 - Q^2/C^2,$$

where the last inequality follows from plugging-in the value of $\gamma$. ■

## 3. Projection-free Algorithms via a Linear Optimization Oracle

In this section we present and analyze our LOO-based algorithms.

### 3.1. LLO-based computation of (close) infeasible projections

The main step towards obtaining our novel algorithms will be to construct an efficient LOO-based infeasible projection oracle (Definition 3). A first step towards this goal will be to show how an LOO could be efficiently used to construct separating hyperplanes for the feasible set $\mathcal{K}$. This will be achieved via the Frank-Wolfe algorithm, when applied to computing the Euclidean projection of the given point onto $\mathcal{K}$, i.e., to solve $\min_{\mathbf{x} \in \mathcal{K}} \|\mathbf{x} - \mathbf{y}\|^2$, where $\mathbf{y}$ is the point which should be separated from $\mathcal{K}$. See Algorithm 2 and the corresponding Lemma 6.

**Lemma 6** *Fix $\epsilon > 0$. Algorithm 2 terminates after at most $\lceil (27R^2/\epsilon) - 2 \rceil$ iterations, and returns a point $\tilde{\mathbf{x}} \in \mathcal{K}$ satisfying:*

**Algorithm 2:** Separating hyperplane via Frank-Wolfe

**Data:** feasible set $\mathcal{K}$, error tolerance $\epsilon$, initial vector $\mathbf{x}_1 \in \mathcal{K}$, target vector $\mathbf{y}$.
**for** $i = 1, \ldots$ **do**
  $\mathbf{v}_i \in \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}}\{(\mathbf{x}_i - \mathbf{y})^\top \mathbf{x}\}$;  /* call to LOO of $\mathcal{K}$ */
  **if** $(\mathbf{x}_i - \mathbf{y})^\top(\mathbf{x}_i - \mathbf{v}_i) \leq \epsilon$ *or* $\|\mathbf{x}_i - \mathbf{y}\|^2 \leq 3\epsilon$ **then**
    **return** $\tilde{\mathbf{x}} \leftarrow \mathbf{x}_i$
  $\sigma_i = \operatorname{argmin}_{\sigma\in[0,1]}\{\|\mathbf{y} - \mathbf{x}_i - \sigma(\mathbf{v}_i - \mathbf{x}_i))\|^2\}$ and $\mathbf{x}_{i+1} = \mathbf{x}_i + \sigma_i(\mathbf{v}_i - \mathbf{x}_i)$
**end**

1. $\|\tilde{\mathbf{x}} - \mathbf{y}\|^2 \leq \|\mathbf{x}_1 - \mathbf{y}\|^2$.
2. *At least one of the following holds:* $\|\tilde{\mathbf{x}} - \mathbf{y}\|^2 \leq 3\epsilon$ *or* $\forall \mathbf{z} \in \mathcal{K} : (\mathbf{y} - \mathbf{z})^\top(\mathbf{y} - \tilde{\mathbf{x}}) > 2\epsilon$.
3. *If dist*$^2(\mathbf{y}, \mathcal{K}) < \epsilon$ *then* $\|\tilde{\mathbf{x}} - \mathbf{y}\|^2 \leq 3\epsilon$.

**Proof** Since Algorithm 2 is simply the Frank-Wolfe method with line-search (Algorithm 8) when applied to the function $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2$, which is 1-smooth and with gradient vector $\nabla f(\mathbf{x}) = \mathbf{x} - \mathbf{y}$, the upper-bound on the number of iterations follows directly from Theorem 2, which guarantees that the stopping condition of the algorithm will be met within the prescribed number of iterations.

Similarly, Item 1 in the theorem follows directly since the line-search guarantees that the function value $f(\mathbf{x}_i) = \frac{1}{2}\|\mathbf{x}_i - \mathbf{y}\|^2$ does not increase when moving from iterate $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$.

Item 2 follows from the stopping condition of the algorithm and by noting that in case for some iteration $i$ it holds both that $(\mathbf{x}_i - \mathbf{y})^\top(\mathbf{x}_i - \mathbf{v}_i) \leq \epsilon$ and $\|\mathbf{x}_i - \mathbf{y}\|^2 > 3\epsilon$ (in which case the algorithm will return $\tilde{\mathbf{x}} = \mathbf{x}_i$), then for all $\mathbf{z} \in \mathcal{K}$ it holds

$$(\mathbf{z} - \mathbf{y})^\top (\mathbf{x}_i - \mathbf{y}) = (\mathbf{z} - \mathbf{x}_i)^\top (\mathbf{x}_i - \mathbf{y}) + \|\mathbf{x}_i - \mathbf{y}\|^2 > (\mathbf{v}_i - \mathbf{x}_i)^\top (\mathbf{x}_i - \mathbf{y}) + 3\epsilon \geq 2\epsilon,$$

where the first inequality is due to the definition of $\mathbf{v}_i$. Finally, to prove Item 3, denote $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} \|\mathbf{x} - \mathbf{y}\|^2$. Suppose by contradiction that $\mathrm{dist}^2(\mathbf{y}, \mathcal{K}) = \|\mathbf{x}^* - \mathbf{y}\|^2 < \epsilon$ and that $\|\tilde{\mathbf{x}} - \mathbf{y}\|^2 > 3\epsilon$. Denote the function $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2$ and its gradient vector $\nabla f(\mathbf{x}) = \mathbf{x} - \mathbf{y}$. According to the assumption and by the stopping condition of the algorithm, on the last iteration executed $i$ it must hold that $(\tilde{\mathbf{x}} - \mathbf{y})^\top(\tilde{\mathbf{x}} - \mathbf{v}_i) = \max_{\mathbf{v}\in\mathcal{K}}(\tilde{\mathbf{x}} - \mathbf{v})^\top \nabla f(\tilde{\mathbf{x}}) \leq \epsilon$, which means that

$$\|\tilde{\mathbf{x}} - \mathbf{y}\|^2 - \mathrm{dist}^2(\mathbf{y}, \mathcal{K}) = 2f(\tilde{\mathbf{x}}) - 2f(\mathbf{x}^*) \leq 2(\tilde{\mathbf{x}} - \mathbf{x}^*)^\top \nabla f(\tilde{\mathbf{x}}) \leq 2\max_{\mathbf{v}\in\mathcal{K}}(\tilde{\mathbf{x}} - \mathbf{v})^\top \nabla f(\tilde{\mathbf{x}}) \leq 2\epsilon,$$

where the first inequality is due to the gradient inequality and the convexity of $f(\cdot)$. Thus, we have that $\|\tilde{\mathbf{x}} - \mathbf{y}\|^2 \leq 2\epsilon + \mathrm{dist}^2(\mathbf{y}, \mathcal{K}) \leq 3\epsilon$, which contradicts the assumption that $\|\tilde{\mathbf{x}} - \mathbf{y}\|^2 > 3\epsilon$. ■

We can now use Algorithm 2 as a subroutine in an iterative algorithm which takes as input some infeasible point $\mathbf{y} \notin \mathcal{K}$, and returns an infeasible projection of it w.r.t. the feasible set $\mathcal{K}$ that is also guaranteed to be at a bounded distance for $\mathcal{K}$. In a nutshell, as long as the infeasible point is too far from the set, Algorithm 3 iteratively calls Algorithm 2 to obtain a separating hyperplane which is then used to "pull" the point closer to the set while maintaining the infeasible projection property.

**Lemma 7** *Fix* $\epsilon > 0$. *Setting* $\gamma = \frac{2\epsilon}{\|\mathbf{x}_0 - \mathbf{y}_0\|^2}$, *Algorithm 3 stops after at most* $\max\left\{\frac{\|\mathbf{x}_0 - \mathbf{y}_0\|^2(\|\mathbf{x}_0 - \mathbf{y}_0\|^2 - \epsilon)}{4\epsilon^2} + 1, 1\right\}$ *iterations, and returns* $(\mathbf{x}, \mathbf{y}) \in \mathcal{K} \times R\mathcal{B}$ *scuh that*

**Algorithm 3:** Close infeasible projection via a linear optimization oracle

```
Data: feasible set 𝒦, feasible point x_0 ∈ 𝒦, initial point y_0, error tolerance ϵ, step size γ
y_1 ← y_0/ max{1, ‖y‖/R} ;              /* y_1 is projection of y_0 over RB */
if ‖x_0 − y_0‖^2 ≤ 3ϵ then
 | Return x ← x_0, y ← y_1
end
for i = 1 . . . do
    x_i ← Output of Alg. 2 with set 𝒦, feasible point x_{i−1}, initial vector y_i, and tolerance ϵ.
    if ‖x_i − y_i‖^2 > 3ϵ then
     | y_{i+1} = y_i − γ (y_i − x_i) ;          /* (y_i − x_i) separates y_i from 𝒦 */
    else
     | Return x ← x_i, y ← y_i
    end
end
```

$$\forall \mathbf{z} \in \mathcal{K}: \ \|\mathbf{y} - \mathbf{z}\|^2 \leq \|\mathbf{y}_0 - \mathbf{z}\|^2 \quad \textit{and} \quad \|\mathbf{x} - \mathbf{y}\|^2 \leq 3\epsilon.$$

*Furthermore, if the for loop has completed overall $k$ iterations, then the point $\mathbf{y}$ satisfies*

$$\textit{dist}^2(\mathbf{y}, \mathcal{K}) \leq \min\left\{R, \textit{dist}^2(\mathbf{y}_0, \mathcal{K}) - (k-1)4\epsilon^2/\|\mathbf{x}_0 - \mathbf{y}_0\|^2\right\}.$$

Before proving Lemma 7 we require an additional auxiliary lemma.

**Lemma 8** *Consider Algorithm 3 and fix some $\epsilon$ such that $0 < 3\epsilon < \|\mathbf{x}_0 - \mathbf{y}_0\|^2$. Setting $\gamma = \frac{2\epsilon}{\|\mathbf{x}_0 - \mathbf{y}_0\|^2}$, we have that on every iteration $i$ of Algorithm 3 it holds that $\|\mathbf{x}_i - \mathbf{y}_i\| \leq \|\mathbf{x}_0 - \mathbf{y}_0\|$.*

**Proof** [Proof of Lemma 7] First, we note that since $\mathbf{y}_1$ is the projection of $\mathbf{y}_0$ onto $R\mathcal{B}$ and $\mathcal{K} \subseteq R\mathcal{B}$, it holds that $\forall \mathbf{z} \in \mathcal{K} : \|\mathbf{y}_1 - \mathbf{z}\|^2 \leq \|\mathbf{y}_0 - \mathbf{z}\|^2$. When $\|\mathbf{x}_0 - \mathbf{y}_0\|^2 \leq 3\epsilon$ or $\|\mathbf{x}_1 - \mathbf{y}_1\|^2 \leq 3\epsilon$ the lemma holds trivially.

For the remaining of the proof we shall assume that $\|\mathbf{x}_1 - \mathbf{y}_1\|^2 > 3\epsilon$. Let us denote by $k > 1$ the overall number of iterations of Algorithm 3, i.e. $\|\mathbf{y}_k - \mathbf{x}_k\|^2 \leq 3\epsilon$ and $\|\mathbf{y}_i - \mathbf{x}_i\|^2 > 3\epsilon$ for all $i < k$. Using Lemma 6, we have that for all $i < k$ it holds that $(\mathbf{y}_i - \mathbf{z})^\top (\mathbf{y}_i - \mathbf{x}_i) \geq 2\epsilon$ for every $\mathbf{z} \in \mathcal{K}$. Using Lemma 8 we also have that $\|\mathbf{y}_i - \mathbf{x}_i\| \leq \|\mathbf{y}_0 - \mathbf{x}_0\|$ for all $i < k$. Thus, using Lemma 5 with $\mathbf{g} = (\mathbf{y}_i - \mathbf{x}_i), C = \|\mathbf{y}_0 - \mathbf{x}_0\|$, and $Q = 2\epsilon$ , we have that for every $1 \leq i < k$,

$$\forall \mathbf{z} \in \mathcal{K}: \quad \|\mathbf{y}_{i+1} - \mathbf{z}\|^2 \leq \|\mathbf{y}_i - \mathbf{z}\|^2 - 4\epsilon^2/\|\mathbf{y}_0 - \mathbf{x}_0\|^2. \tag{3}$$

This already guarantees that indeed forall $\mathbf{z} \in \mathcal{K}$, the returned point $\mathbf{y}$ satisfies: $\|\mathbf{y} - \mathbf{z}\|^2 \leq \|\mathbf{y}_1 - \mathbf{z}\|^2 \leq \|\mathbf{y}_0 - \mathbf{z}\|^2$. Since $\mathbf{0} \in \mathcal{K}$, it in particular follows that $\|\mathbf{y}\| \leq \|\mathbf{y}_1\| \leq R$, i.e. $\mathbf{y} \in R\mathcal{B}$. Note also that $\mathbf{x} \in \mathcal{K}$ since it is the output of Algorithm 2.

Now we continue to upper-bound the number of iterations until Algorithm 3 stops and $\text{dist}^2(\mathbf{y}, \mathcal{K})$. Denote $\mathbf{x}_i^* = \text{argmin}_{\mathbf{x} \in \mathcal{K}} \|\mathbf{y}_i - \mathbf{x}\|^2$ for every iteration $i < k$. Using Eq. (3), for every iteration $i < k$ it holds that

$$\begin{aligned}\text{dist}^2(\mathbf{y}_{i+1}, \mathcal{K}) &= \|\mathbf{y}_{i+1} - \mathbf{x}_{i+1}^*\|^2 \leq \|\mathbf{y}_{i+1} - \mathbf{x}_i^*\|^2 \\ &\leq \|\mathbf{y}_i - \mathbf{x}_i^*\|^2 - 4\epsilon^2/\|\mathbf{y}_0 - \mathbf{x}_0\|^2 = \text{dist}^2(\mathbf{y}_i, \mathcal{K}) - 4\epsilon^2/\|\mathbf{y}_0 - \mathbf{x}_0\|^2.\end{aligned}$$

Unrolling the recursion and using $\text{dist}^2(\mathbf{y}_1, \mathcal{K}) \leq \text{dist}^2(\mathbf{y}_0, \mathcal{K})$, we have

$$\begin{aligned}\text{dist}^2(\mathbf{y}_{i+1}, \mathcal{K}) &\leq \text{dist}^2(\mathbf{y}_1, \mathcal{K}) - i4\epsilon^2/\|\mathbf{y}_0 - \mathbf{x}_0\|^2 \leq \text{dist}^2(\mathbf{y}_0, \mathcal{K}) - i4\epsilon^2/\|\mathbf{y}_0 - \mathbf{x}_0\|^2 \\ &\leq \|\mathbf{y}_0 - \mathbf{x}_0\|^2 - i4\epsilon^2/\|\mathbf{y}_0 - \mathbf{x}_0\|^2,\end{aligned}$$

Then, after at most $k-1 = \left(\|\mathbf{y}_0 - \mathbf{x}_0\|^2(\|\mathbf{y}_0 - \mathbf{x}_0\|^2 - \epsilon)\right)/4\epsilon^2$ iterations, we obtain $\text{dist}^2(\mathbf{y}_k, \mathcal{K}) \leq \epsilon$, which by using Lemma 6, implies that the next iteration will be the last one, and the returned points $\mathbf{x}, \mathbf{y}$ will indeed satisfy $\|\mathbf{x} - \mathbf{y}\|^2 \leq 3\epsilon$, as required. ■

### 3.2. LOO-based algorithms for the full-information setting

We are now ready to fully detail our algorithm for the full information setting using a LOO, Algorithm 4, and analyze its regret and oracle complexity. The algorithm combines the OGD without feasibility algorithm, Algorithm 1, and the LOO-based infeasible projection oracle given in Algorithm 3. Since each invokation of Algorithm 3 may call through Algorithm 2 the LOO several times, Algorithm 4 considers the iterations in blocks of $K$ disjoint iterations ($K$ is a parameter to be determined in the analysis), and uses the same prediction for the entire block. Thus, a single call to the infeasible projection oracle, Algorithm 3, is made on each block. Finally, we note that for more practical considerations, the update to predictions of the algorithm is delayed in such a way that, at the end of each block $m$, the algorithm does not need to wait until the prediction for the next block $m + 1$ will be computed but, it is already computed during the course of block $m$.

---

**Algorithm 4:** Blocked Online Gradient Descent with LOO (LOO-BOGD)

---

**Data:** horizon $T$, feasible set $\mathcal{K}$, block size $K$, update step $\eta$, error tolerance $\epsilon$.
$\mathbf{x}_0, \mathbf{x}_1 \leftarrow$ arbitrary points in $\mathcal{K}$.
$\tilde{\mathbf{y}}_0 \leftarrow \mathbf{x}_0, \mathbf{y}_1 \leftarrow \tilde{\mathbf{y}}_0, \tilde{\mathbf{y}}_1 \leftarrow \mathbf{x}_1$.
**for** $t = 1, \ldots, K$ **do**
  Play $\mathbf{x}_0$ and observe $f_t(\mathbf{x}_0)$
  Set $\nabla_t \in \partial f_t(\mathbf{x}_0)$ and update $\mathbf{y}_{t+1} = \mathbf{y}_t - \eta\nabla_t$
**end**
**for** $m = 2, \ldots, \frac{T}{K}$ **do**
  Let $(\mathbf{x}_m, \tilde{\mathbf{y}}_m) \in \mathcal{K} \times R\mathcal{B}$ be the output of Algorithm 3 when called with set $\mathcal{K}$, feasible point $\mathbf{x}_{m-2}$, initial point $\mathbf{y}_{(m-1)K+1}$, and tolerance $\epsilon$ (execute **in parallel** to the following **for** loop over $s$)
  Set $\mathbf{y}_{(m-1)K+1} = \tilde{\mathbf{y}}_{m-1}$
  **for** $s = 1, \ldots, K$ **do**
    Play $\mathbf{x}_{m-1}$ and observe $f_t(\mathbf{x}_{m-1})$ ; /* $t = (m-1)K + s$ */
    Set $\nabla_t \in \partial f_t(\mathbf{x}_{m-1})$ and update $\mathbf{y}_{t+1} = \mathbf{y}_t - \eta\nabla_t$
  **end**
  **Note:** $\mathbf{y}_{mK+1} = \tilde{\mathbf{y}}_{m-1} - \eta\sum_{t=(m-1)K+1}^{mK} \nabla_t$.
**end**

---

**Theorem 9** *Setting* $\eta = (R/G_f)T^{-\frac{3}{4}}, \epsilon = 60R^2T^{-\frac{1}{2}}, K = 5T^{\frac{1}{2}}$ *in Algorithm 4 guarantees that the adaptive regret is upper bounded by*

$$\sup_{I=[s,e]\subseteq[T]} \left\{\sum_{t=s}^{e} f_t(\mathbf{x}_t) - \min_{\mathbf{x}_I\in\mathcal{K}} \sum_{t=s}^{e} f_t(\mathbf{x}_I)\right\} \leq 20G_fRT^{\frac{1}{2}} + 20G_fRT^{\frac{3}{4}},$$

*and that the overall number of calls to the LOO is upper bounded by* $N_{calls} \leq T$

### 3.3. (standard) Regret bound for strongly convex losses

We now consider the case in which all loss functions are $\alpha$-strongly convex, for some known $\alpha > 0$. In this setting, vanilla OGD does not yield adaptive regret guarantees, and the same goes for our OGD-based approach for constructing new LOO-based projection-free algorithms. Instead, here we show that our approach can recover the state-of-the-art (standard) regret bound for this setting of $O(T^{2/3})$ Kretzu and Garber (2021). The algorithm, Algorithm 5, is given below. We note that in the strongly convex setting (as opposed to the case of convex, but not strongly convex losses), we do not require to consider the iterations in blocks, which slightly simplifies the algorithm.

**Algorithm 5:** Online Gradient descent with Linear Optimization Oracle (LOO-OGD)

**Data:** horizon $T$, feasible set $\mathcal{K}$, update steps $\{\eta_t\}_{t=1}^T$, error tolerances $\{\epsilon_t\}_{t=1}^T$
$\mathbf{x}_1 \leftarrow$ arbitrary points in $\mathcal{K}$
$\tilde{\mathbf{y}}_1 \leftarrow \mathbf{x}_1$
**for** $t = 1, \ldots, T$ **do**
  Play $\mathbf{x}_t$ and observe $f_t(\mathbf{x}_t)$
  Set $\nabla_t \in \partial f_t(\mathbf{x}_t)$ and update $\mathbf{y}_{t+1} = \tilde{\mathbf{y}}_t - \eta_t\nabla_t$
  $\mathbf{x}_{t+1}, \tilde{\mathbf{y}}_{t+1} \leftarrow$ Outputs of Algorithm 3 with set $\mathcal{K}$, feasible point $\mathbf{x}_t$, initial vector $\mathbf{y}_{t+1}$, and tolerance $\epsilon_{t+1}$
**end**

**Theorem 10** *Suppose all loss functions* $\{f_t\}_{t=1}^T$ *are* $\alpha$*-strongly convex, for some* $\alpha > 0$*. Setting* $\epsilon_t = 130\left(4G_f^2R^2/\alpha t\right)^{\frac{2}{3}}, \eta_t = \frac{1}{\alpha t}$ *in Algorithm 5, guarantees that the (static) regret is upper bounded by*

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}) \leq 50(R^2G_f^4/\alpha)^{\frac{1}{3}}T^{\frac{2}{3}} + (G_f^2/2\alpha)(1+\ln(T)),$$

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

$$N_{calls} \leq 0.9T + \left(\frac{R\alpha}{G_f}\right)^{\frac{2}{3}} T^{\frac{2}{3}} + \left(\frac{G_f}{\alpha R}\right)^{\frac{2}{3}} \left(\frac{1}{4R^{\frac{4}{3}}} + 1\right) T^{\frac{1}{3}} + \frac{1}{2}\left(\frac{G_f}{2R\alpha}\right)^{\frac{4}{3}} + \frac{1}{2}\left(\frac{G_f}{2R\alpha}\right)^2.$$

### 3.4. LOO-based algorithm for the bandit setting

Our algorithm for the bandit information setting using a LOO, Algorithm 9 (given in the appendix), follows from a simple combination of Algorithm 4 and the standard technique for bandit optimization pioneered in Flaxman et al. (2005), which generates unbiased estimators for gradients of smoothed versions of the original (unknown) loss functions via random sampling in a small neighbourhood of the feasible point. For this reason, Algorithm 9 applies the full-information Algorithm 4 on a slightly squeezed version of the feasible set — the set $\mathcal{K}_{\delta/r} = (1-\delta/r)\mathcal{K}$, so that the sampled points will remain feasible. We remind the reader that in the bandit setting we make the standard assumption that the loss functions are chosen obliviously, i.e., they are independent of any randomness introduced by the algorithm.

**Theorem 11** *Suppose Assumption 1 holds. For all $c > 0$ such that $\frac{cT^{-1/4}}{r} < 1$, Setting $\eta = \frac{R}{\sqrt{nM}}T^{-\frac{3}{4}}, K = 6nMT^{\frac{1}{2}}, \delta = cT^{-\frac{1}{4}}$ in Algorithm 9 guarantees that the adaptive expected regret is upper-bounded as follows*

$$AER_T = \sup_{I=[s,e]\subseteq[T]} \left\{ \mathbb{E}\left[\sum\nolimits_{t=s}^{e} f_t(\mathbf{z}_t)\right] - \min_{\mathbf{x}_I\in\mathcal{K}} \sum\nolimits_{t=s}^{e} f_t(\mathbf{x}_I) \right\} \leq$$

$$\leq \left(4+\frac{R}{r}\right) G_f c T^{\frac{3}{4}} + \sqrt{nM}\left(4R+\frac{1}{\sqrt{6}}+3RG_f^2+\frac{RnM}{2c^2}\right)T^{\frac{3}{4}} + 24RnM\left(\frac{\sqrt{nM}}{c\sqrt{6}}+G_f\right)T^{\frac{1}{2}},$$

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

$$\mathbb{E}[N_{calls}] \leq \frac{27R^2}{2nMc^2}\left(\frac{6^5R^4\left(nM\right)^4}{4c^8}+\frac{6^6R^4\left(nM\right)^3G_f^2}{2c^6}+\frac{6^6R^4(nM)^2G_f^4}{3c^4}+19\right)T.$$

*In particular, if $\left(\frac{20R\sqrt{nM}}{r}\right)^4 \leq T$ then, setting $c = 20R\sqrt{nM}$, we have*

$$AER_T \leq R\sqrt{nM}\left(80G_f+20G_f\frac{R}{r}+4+\frac{1}{2R}+3G_f^2+\frac{1}{4R^2}\right)T^{\frac{3}{4}} + 24nM\left(\frac{1}{2}+RG_f\right)T^{\frac{1}{2}},$$

*and $\mathbb{E}[N_{calls}] \leq \frac{\tilde{c}}{(nM)^2}\left(\frac{1}{R^4}+\frac{G_f^2}{R^2}+G_f^4+1\right)T$, where $0<\tilde{c}<1$ is an universal constant.*

## 4. Projection-free Algorithms with a Separation Oracle

In this section we discuss our SO-based algorithms. Similarly to our LOO-based algorithms, here also we will begin by showing how to efficiently compute infeasible projections using the SO, and then we will combine it with the OGD without feasibility approach (Algorithm 1), to obtain our algorithms. More concretely, our SO-based algorithms will be based on the following idea, which is slightly different than the one used for our LOO-based algorithms. Note that under Assumption 1, for any $\delta \in [0,1]$ it holds that $\mathcal{K}_\delta = (1-\delta)\mathcal{K} \subseteq \mathcal{K}$. Thus, our approach will be to fix some $\delta \in (0,1]$ and to treat $\mathcal{K}_\delta$ as if it was the feasible set, and compute infeasible projections w.r.t. it, while ensuring that at all times, the points played by the algorithms remain within the enclosing feasible set $\mathcal{K}$.

For clarity, throughout this section we introduce the notation $\mathcal{K}_{\delta_1,\delta_2} = (1-\delta_1)(1-\delta_2)\mathcal{K} = \{(1-\delta_1)(1-\delta_2)\mathbf{x} \mid \mathbf{x}\in\mathcal{K}\}$, for any $(\delta_1,\delta_2)\in[0,1]^2$.

### 4.1. Efficient (close) infeasible projection via a SO

We now turn to detail the main ingredient in our SO-based online algorithms — efficient infeasible projections onto the set $\mathcal{K}_{\delta,\delta'/r}$, for any given $(\delta,\delta')\in[0,1]\times[0,r]$, using the SO.

As in our LOO-based construction, the first step will be to show how the SO of $\mathcal{K}$ can be used to construct separating hyperplanes w.r.t. $\mathcal{K}_{\delta,\delta'/r}$, which will in turn be used to "pull" infeasible points closer to the set, while maintaining the infeasible projection property.

**Lemma 12** *Suppose Assumption 1 holds. Fix $(\delta,\delta') \in (0,1)\times[0,r)$, and let $\mathbf{y}\in\mathbb{R}^n$ such that $\frac{\mathbf{y}}{1-\delta'/r}\notin\mathcal{K}$. Let $\mathbf{g}\in\mathbb{R}^n$ be the output of the SO of $\mathcal{K}$ w.r.t. $\frac{\mathbf{y}}{1-\delta'/r}$, i.e., for all $\mathbf{x}\in\mathcal{K}$, $\left(\frac{\mathbf{y}}{1-\delta'/r}-\mathbf{x}\right)^\top\mathbf{g}>0$. Then, it holds that, $\forall\mathbf{z}\in\mathcal{K}_{\delta,\delta'/r}:\quad (\mathbf{y}-\mathbf{z})^\top\mathbf{g}>\delta(r-\delta')\|\mathbf{g}\|$.*

We can now present our SO-based infeasible projection oracle, see Algorithm 6.

**Algorithm 6:** Infeasible projection via a separation oracle

**Data:** feasible set $\mathcal{K}$, radius $r$, squeeze parameters $(\delta,\delta')\in[0,1]\times[0,r]$, initial vector $\mathbf{y}_0$.
$\mathbf{y}_1\leftarrow\mathbf{y}_0/\max\{1,\|\mathbf{y}\|/R\}$ ; /* $\mathbf{y}_1$ is projection of $\mathbf{y}_0$ over $R\mathcal{B}$ */
**for** $i=1\ldots$ **do**
  Call $\text{SO}_\mathcal{K}$ with input $\frac{\mathbf{y}_i}{1-\delta'/r}$
  **if** $\frac{\mathbf{y}_i}{1-\delta'/r}\notin\mathcal{K}$ **then**
    Set $\mathbf{g}_i\leftarrow$ hyperplane outputted by $\text{SO}_\mathcal{K}$ ; /* $\forall\mathbf{x}\in\mathcal{K}\ \left(\frac{\mathbf{y}_i}{1-\delta'/r}-\mathbf{x}\right)^\top\mathbf{g}_i>0$ */
    Update $\mathbf{y}_{i+1}=\mathbf{y}_i-\gamma_i\mathbf{g}_i$
  **else**
    **Return** $\mathbf{y}\leftarrow\mathbf{y}_i$
  **end**
**end**

**Lemma 13** *Suppose Assumption 1 holds. Let $0<\delta<1$, and $0\le\delta'<r$. Setting $\gamma_i=\delta(r-\delta')/\|\mathbf{g}_i\|$, Algorithm 6 stops after at most $\frac{dist^2(\mathbf{y}_0,\mathcal{K}_{\delta,\delta'/r})-dist^2(\mathbf{y},\mathcal{K}_{\delta,\delta'/r})}{\delta^2(r-\delta')^2}+1$, iterations, and returns $\mathbf{y}\in\mathcal{K}_{\delta'}=(1-\delta')\mathcal{K}$ such that $\forall\mathbf{z}\in\mathcal{K}_{\delta,\delta'/r}:\quad\|\mathbf{y}-\mathbf{z}\|^2\le\|\mathbf{y}_0-\mathbf{z}\|^2$.*

### 4.2. SO-based algorithm for the full-information setting

Our SO-based algorithm for the full-information setting, Algorithm 7, is given below.

**Algorithm 7:** Online gradient descent via a separation oracle (SO-OGD)

**Data:** horizon $T$, feasible set $\mathcal{K}$, update step $\eta$, squeeze parameter $\delta$.
$\tilde{\mathbf{y}}_1\leftarrow\mathbf{0}\in\mathcal{K}_\delta$.
**for** $t=1,\ldots,T$ **do**
  Play $\tilde{\mathbf{y}}_t$ and observe $f_t(\tilde{\mathbf{y}}_t)$.
  Set $\nabla_t\in\partial f_t(\tilde{\mathbf{y}}_t)$ and update $\mathbf{y}_{t+1}=\tilde{\mathbf{y}}_t-\eta\nabla_t$.
  Set $\tilde{\mathbf{y}}_{t+1}\leftarrow$ Outputs of Algorithm 6 with set $\mathcal{K}$, radius $r$, initial vector $\mathbf{y}_{t+1}$, and squeeze parameters $(\delta,0)$.
**end**

**Theorem 14** *Suppose Assumption 1 holds. Fix $c > 0$ such that $\delta = cT^{-\frac{1}{2}} \in (0,1)$, and set $\eta = \frac{c_1 r}{2G_f} T^{-\frac{1}{2}}$. Algorithm 7 guarantees that the adaptive regret is upper bounded by*

$$\sup_{I=[s,e]\subseteq[T]} \left\{ \sum_{t=s}^{e} f_t(\tilde{\mathbf{y}}_t) - \min_{\mathbf{x}_I\in\mathcal{K}} \sum_{t=r}^{s} f_t(\mathbf{x}_I) \right\} \leq \left( G_f Rc + \frac{rG_f}{4} + \frac{4R^2 G_f}{r} \right) \sqrt{T},$$

*and that the overall number of calls to the SO is upper bounded by $N_{calls} \leq \left(\frac{R}{rc} + \frac{1}{4c^2} + 1\right) T$ .*

*In particular, if $\frac{4R}{r} \leq \sqrt{T}$, then setting $c = \frac{4R}{r}$, we have that*

$$\sup_{[s,e]\subseteq[T]} \left\{ \sum_{t=s}^{e} f_t(\tilde{\mathbf{y}}_t) - \min_{\mathbf{x}_I\in\mathcal{K}} \sum_{t=r}^{s} f_t(\mathbf{x}_I) \right\} \leq G_f \left( \frac{r}{4} + \frac{8R^2}{r} \right) \sqrt{T}, \text{ and } N_{calls} \leq \left( \frac{5}{4} + \frac{r^2}{64R^2} \right) T,$$

### 4.3. SO-based algorithm for the bandit setting

Similarly to our LLO-based algorithm for the bandit setting, our SO-based bandit algorithm follows from combining our SO-based algorithm for the full-information setting together with the use of unbiased estimators for the gradients of smoothed versions of the loss functions, as pioneered in Flaxman et al. (2005). Our algorithm for the bandit feedback, Algorithm 10, is given in the appendix. As opposed to the full-information setting which used a single squeeze parameter (i.e., we set $\delta' = 0$ when considering the squeezed set $\mathcal{K}_{\delta,\delta'/r}$), in the bandit setting, due to the ball-sampling technique which is used to construct the unbiased gradient estimators, in order to keep the iterates feasible, we set $\delta'$ to be strictly positive.

**Theorem 15** *Suppose Assumption 1 holds. Fix some $c', c > 0$ such that $2c'T^{-1/4} < r$ and $cT^{-1/4} < 1$. Setting $\eta = \frac{r}{4\sqrt{nM}} T^{-\frac{3}{4}}, \delta = cT^{-1/4}, \delta' = c'T^{-\frac{1}{4}}$ in Algorithm 10, guarantees that the adaptive expected regret is upper bounded as follows*

$$AER_T = \sup_{I=[s,e]\subseteq[T]} \left\{ \mathbb{E}\left[ \sum_{t=s}^{e} f_t(\mathbf{z}_t) \right] - \min_{\mathbf{x}_I\in\mathcal{K}} \sum_{t=s}^{e} f_t(\mathbf{x}_I) \right\} \leq$$
$$\leq G_f R \left( \frac{3c'}{R} + \frac{c'}{r} + c + \frac{4\sqrt{nM}}{rG_f} + \frac{(nM)^{\frac{3}{2}}}{8G_f R} \frac{r}{c'^2} \right) T^{\frac{3}{4}} + G_f R \frac{cc'}{r} T^{\frac{1}{2}},$$

*and the overall number of calls to SO is upper bounded by $N_{calls} \leq T + \frac{2R\sqrt{nM}}{rcc'} T^{\frac{3}{4}} + \frac{nM}{4c^2c'^2} T^{\frac{1}{2}}$. In particular, if $T^{1/4} > \max\{\frac{2\sqrt{nM}}{r}, \frac{8}{r}\}$, then setting $c = \frac{8}{r}$ and $c' = \sqrt{nM}$, we have*

$$AER_T \leq R\sqrt{nM} \left( \frac{4G_f}{r} + \frac{4}{r} + \frac{r}{8R} \right) T^{\frac{3}{4}} + \frac{8G_f R}{r} T^{\frac{3}{4}} + G_f R \frac{8\sqrt{nM}}{r^2} T^{\frac{1}{2}},$$

*and $N_{calls} \leq T + (R/4)\, T^{\frac{3}{4}} + (r/16)^2\, T^{\frac{1}{2}}$.*

## References

Dimitri P Bertsekas. Stochastic optimization problems with nondifferentiable cost functionals. *Journal of Optimization Theory and Applications*, 12(2):218–231, 1973.

Lin Chen, Christopher Harshaw, Hamed Hassani, and Amin Karbasi. Projection-free online optimization with stochastic gradient: From convexity to submodularity. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 814–823. PMLR, 10–15 Jul 2018.

Lin Chen, Mingrui Zhang, and Amin Karbasi. Projection-free bandit convex optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2047–2056. PMLR, 2019.

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

Alina Ene, Huy L Nguyen, and Adrian Vladu. Projection-free bandit optimization with privacy guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7322–7330, 2021.

Abraham D Flaxman, Adam Tauman Kalai, Adam Tauman Kalai, and H Brendan McMahan. 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, 2005.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:149–154, 1956.

Dan Garber. Efficient online linear optimization with approximation algorithms. *Mathematics of Operations Research*, 46(1):204–220, 2021.

Dan Garber and Elad Hazan. Playing non-linear games with linear oracles. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 420–428. IEEE, 2013.

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

Eyal Gofer and Yishay Mansour. Lower bounds on individual sequence regret. *Machine Learning*, 103(1):1–26, 2016.

Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

Elad Hazan and Edgar Minasyan. Faster projection-free online learning. In *Conference on Learning Theory*, pages 1877–1893. PMLR, 2020.

Elad Hazan and Comandur Seshadhri. Efficient learning algorithms for changing environments. In *Proceedings of the 26th annual international conference on machine learning*, pages 393–400, 2009.

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

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

Ben Kretzu and Dan Garber. Revisiting projection-free online learning: the strongly convex case. In *International Conference on Artificial Intelligence and Statistics*, pages 3592–3600. PMLR, 2021.

Kfir Levy and Andreas Krause. Projection free online learning over smooth sets. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1458–1466, 2019.

Zakaria Mhammedi. Efficient projection-free online convex optimization with membership oracle. *arXiv preprint arXiv:2111.05818*, 2021.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Yuanyu Wan and Lijun Zhang. Projection-free online learning over strongly convex sets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10076–10084, 2021.

Wenpeng Zhang, Peilin Zhao, Wenwu Zhu, Steven CH Hoi, and Tong Zhang. Projection-free distributed online learning in networks. In *International Conference on Machine Learning*, pages 4054–4062. PMLR, 2017.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003.

## Appendix A. Details Missing from Section 2

### A.1. The Frank-Wolfe algorithm with line-search

**Algorithm 8:** Frank-Wolfe with line-search

**Data:** feasible set $\mathcal{K}$, initial point $\mathbf{x}_0 \in \mathcal{K}$, objective function $f(\cdot)$.

**for** $i = 0, \ldots$ **do**

  $\mathbf{v}_i \in \mathrm{argmin}_{\mathbf{x}\in\mathcal{K}}\{\nabla f(\mathbf{x}_i)^\top \mathbf{x}\}$ ; /* call to LOO of $\mathcal{K}$ */

  $\sigma_i = \mathrm{argmin}_{\sigma\in[0,1]}\{f(\mathbf{x}_i + \sigma(\mathbf{v}_i - \mathbf{x}_i))\}$

  $\mathbf{x}_{i+1} = \mathbf{x}_i + \sigma_i(\mathbf{v}_i - \mathbf{x}_i)$

**end**

### A.2. Smoothed loss functions for bandit optimization

A standard component of bandit algorithms Flaxman et al. (2005); Chen et al. (2018); Garber and Kretzu (2020); Kretzu and Garber (2021), is the use of smoothed versions of the loss functions and their unbiased estimators. We define the $\delta$-smoothing of a loss function $f$ by $\widehat{f}_\delta(\mathbf{x}) = \mathbb{E}_{\mathbf{u}\sim\mathcal{B}}\left[f(\mathbf{x}+\delta\mathbf{u})\right]$ . We now cite several standard useful lemmas regarding such smoothed functions.

**Lemma 16 (Lemma 2.1 in Hazan (2019))** *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be convex and $G_f$-Lipschitz over a convex and compact set $\mathcal{K} \subset \mathbb{R}^n$. Then $\widehat{f}_\delta$ is convex and $G_f$-Lipschitz over $\mathcal{K}_\delta$, and $\forall \mathbf{x} \in \mathcal{K}_\delta$ it holds that $|\widehat{f}_\delta(\mathbf{x}) - f(\mathbf{x})| \leq \delta G_f$.*

**Lemma 17 (Lemma 6.5 in Hazan (2019))** *$\widehat{f}_\delta(\mathbf{x})$ is differentiable and $\nabla \widehat{f}_\delta(\mathbf{x}) = \mathbb{E}_{\mathbf{u}\sim \mathcal{S}^n}\left[\frac{n}{\delta} f(\mathbf{x} + \delta\mathbf{u})\mathbf{u}\right]$, where $\mathbf{u}$ is sampled uniformly from $S^n$.*

**Lemma 18 (see Bertsekas (1973))** *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be convex and suppose that all subgradients of $f$ are upper-bounded by $G_f$ in $\ell_2$-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 $\|\nabla \widehat{f}_\delta(\mathbf{x})\| \leq G_f$.*

## Appendix B. Proof of Lemma 4

**Proof** [Proof of Lemma 4] Fix some iteration $t$ of Algorithm 1. Since $\tilde{\mathbf{y}}_{t+1}$ is an infeasible projection of $\mathbf{y}_{t+1}$, and $\mathbf{y}_{t+1} = \tilde{\mathbf{y}}_t - \eta_t \nabla_t$, we have that

$$\begin{aligned}\forall \mathbf{x} \in \mathcal{K}: \; \|\tilde{\mathbf{y}}_{t+1} - \mathbf{x}\|^2 \leq \|\mathbf{y}_{t+1} - \mathbf{x}\|^2 &= \|\tilde{\mathbf{y}}_t - \eta_t\nabla_t - \mathbf{x}\|^2 \\ &\leq \|\tilde{\mathbf{y}}_t - \mathbf{x}\|^2 + \eta_t^2\|\nabla_t\|^2 - 2\eta_t\nabla_t^\top(\tilde{\mathbf{y}}_t - \mathbf{x}).\end{aligned}$$

Rearranging, then we have

$$\forall \mathbf{x} \in \mathcal{K}: \; \nabla_t^\top(\tilde{\mathbf{y}}_t - \mathbf{x}) \leq \frac{\|\tilde{\mathbf{y}}_t - \mathbf{x}\|^2}{2\eta_t} - \frac{\|\tilde{\mathbf{y}}_{t+1} - \mathbf{x}\|^2}{2\eta_t} + \frac{\eta_t\|\nabla_t\|^2}{2}.$$

Fix some positive integers $1 \leq s \leq e \leq T$. Summing over the interval $[s, e]$, we have that

$$\forall \mathbf{x} \in \mathcal{K}: \; \sum_{t=s}^{e} \nabla_t^\top(\tilde{\mathbf{y}}_t - \mathbf{x}) \leq \frac{\|\tilde{\mathbf{y}}_s - \mathbf{x}\|^2}{2\eta_s} + \sum_{t=s+1}^{e}\left(\frac{1}{2\eta_t} - \frac{1}{2\eta_{t-1}}\right)\|\tilde{\mathbf{y}}_t - \mathbf{x}\|^2 + \sum_{t=s}^{e}\frac{\eta_t}{2}\|\nabla_t\|^2. \tag{4}$$

Using the convexity of each $f_t(\cdot)$ and plugging-in $\eta_t = \eta$ for all $t \geq 1$, we have that

$$\forall \mathbf{x} \in \mathcal{K}: \; \sum_{t=s}^{e} f_t(\tilde{\mathbf{y}}_t) - f_t(\mathbf{x}) \leq \frac{\|\tilde{\mathbf{y}}_s - \mathbf{x}\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=s}^{e}\|\nabla_t\|^2,$$

which yields the first guarantee of the lemma.

In case all loss function $f_t(\cdot), 1 \leq t \leq T$, are $\alpha$-strongly convex, using the inequality $f_t(\mathbf{y}) - f_t(\mathbf{x}) \leq \nabla f_t(\mathbf{y})^\top(\mathbf{y} - \mathbf{x}) - \frac{\alpha}{2}\|\mathbf{y} - \mathbf{x}\|^2, (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^n$, and setting $(s, e) = (1, T)$ in Eq.(4), for every $\mathbf{x} \in \mathcal{K}$ we have that,

$$\begin{aligned}\sum_{t=1}^{T} f_t(\tilde{\mathbf{y}}_t) - f_t(\mathbf{x}) \leq &\sum_{t=1}^{T}\frac{\eta_t\|\nabla_t\|^2}{2} + \left(\frac{1}{2\eta_1} - \frac{\alpha}{2}\right)\|\tilde{\mathbf{y}}_1 - \mathbf{x}\|^2 \\ &+ \sum_{t=2}^{T}\left(\frac{1}{2\eta_t} - \frac{1}{2\eta_{t-1}} - \frac{\alpha}{2}\right)\|\tilde{\mathbf{y}}_t - \mathbf{x}\|^2.\end{aligned}$$

Plugging in $\eta_t = \frac{1}{\alpha t}$, we obtain the second guarantee of the lemma. ■

## Appendix C. Proof of Lemma 8

**Proof** [Proof of Lemma 8] Using the update step of the algorithm, for every iteration $i > 1$ we have that $\mathbf{y}_i = \mathbf{y}_{i-1} - \gamma(\mathbf{y}_{i-1} - \mathbf{x}_{i-1})$, and thus, for every $i > 1$ we have that

$$\|\mathbf{x}_{i-1} - \mathbf{y}_i\| = \|\mathbf{x}_{i-1} - \mathbf{y}_{i-1} + \gamma(\mathbf{y}_{i-1} - \mathbf{x}_{i-1})\| = (1-\gamma)\|\mathbf{x}_{i-1} - \mathbf{y}_{i-1}\| \leq \|\mathbf{x}_{i-1} - \mathbf{y}_{i-1}\|,$$

where the last inequality holds since our choice of $\gamma$ satisfies $\gamma \in [0, 1)$.

From Lemma 6 we also have that for all $i \geq 1$, $\mathbf{x}_i$ satisfies $\|\mathbf{x}_i - \mathbf{y}_i\| \leq \|\mathbf{x}_{i-1} - \mathbf{y}_i\|$. Combining this with the inequality above gives

$$\|\mathbf{x}_i - \mathbf{y}_i\| \leq \|\mathbf{x}_{i-1} - \mathbf{y}_{i-1}\| \leq \cdots \leq \|\mathbf{x}_1 - \mathbf{y}_1\| \leq \|\mathbf{x}_0 - \mathbf{y}_1\| \leq \|\mathbf{x}_0 - \mathbf{y}_0\|,$$

where the last inequality follows since $\mathbf{x}_0 \in \mathcal{K}$ and $\mathbf{y}_1 \leftarrow \mathbf{y}_0 / \max\{1, \|\mathbf{y}_0\|/R\}$, i.e. $\mathbf{y}_1$ is the projection of $\mathbf{y}_0$ over the set $R\mathcal{B}$ ($\mathcal{K} \subseteq R\mathcal{B}$), and thus $\|\mathbf{x}_0 - \mathbf{y}_1\| \leq \|\mathbf{x}_0 - \mathbf{y}_0\|$. ■

## Appendix D. Proof of Theorem 9

Before proving the theorem we need an additional lemma.

**Lemma 19** *Let $\{\tilde{\mathbf{y}}_m\}_{m=2}^{\frac{T}{K}-1} \subset R\mathcal{B}$ be as in Algorithm 4 when ran with some block size $K$, for some positive integer $K$, and step-size $\eta > 0$. It holds that*

$$\sup_{I=[s,e]\subseteq[T]} \left\{ \sum_{t=s}^{e} f_t(\tilde{\mathbf{y}}_t) - \min_{\mathbf{x}_I \in \mathcal{K}} \sum_{t=s}^{e} f_t(\mathbf{x}_I) \right\} \leq \frac{\eta}{2} K G_f^2 T + 4RKG_f + \frac{4R^2}{\eta}.$$

**Proof** Denote $\mathcal{T}_m = \{(m-1)K+1, \ldots, mK\}$ for every $m \in [T/K]$. Since for every $m \in [T/K]$, $\tilde{\mathbf{y}}_{m+1}$ is the output of Algorithm 3 when called with the input $\mathbf{y}_{mK+1}$, we have from Lemma 7 that $\forall \mathbf{x} \in \mathcal{K}: \ \|\tilde{\mathbf{y}}_{m+1} - \mathbf{x}\|^2 \leq \|\mathbf{y}_{mK+1} - \mathbf{x}\|^2$. Note also that $\mathbf{y}_{mK+1} = \tilde{\mathbf{y}}_{m-1} - \eta \sum_{t \in \mathcal{T}_m} \tilde{\nabla}_t$, where $\tilde{\nabla}_t \in \partial f_t(\tilde{\mathbf{y}}_{m-1})$. Thus, we have that

$$\begin{aligned}\forall \mathbf{x} \in \mathcal{K}: \quad \|\tilde{\mathbf{y}}_{m+1} - \mathbf{x}\|^2 &\leq \|\mathbf{y}_{mK+1} - \mathbf{x}\|^2 = \left\| \tilde{\mathbf{y}}_{m-1} - \eta \sum_{t \in \mathcal{T}_m} \tilde{\nabla}_t - \mathbf{x} \right\|^2 \\ &\leq \|\tilde{\mathbf{y}}_{m-1} - \mathbf{x}\|^2 + \eta^2 K^2 G_f^2 - 2\eta \sum_{t \in \mathcal{T}_m} \tilde{\nabla}_t^\top (\tilde{\mathbf{y}}_{m-1} - \mathbf{x}),\end{aligned}$$

where in the last inequality we have used the assumption that for all $t \in [T]$ and $\mathbf{x} \in R\mathcal{B}$ it holds $\|\nabla f_t(\mathbf{x})\| \leq G_f$.

Rearranging, we have for every block $m$ that

$$\sum_{t \in \mathcal{T}_m} \tilde{\nabla}_t^\top (\tilde{\mathbf{y}}_{m-1} - \mathbf{x}) \leq \frac{\|\tilde{\mathbf{y}}_{m-1} - \mathbf{x}\|^2}{2\eta} - \frac{\|\tilde{\mathbf{y}}_{m+1} - \mathbf{x}\|^2}{2\eta} + \frac{\eta}{2} K^2 G_f^2. \tag{5}$$

Fix some interval $[s, e], 1 \leq s \leq e \leq T$. We define two scalars $m_s$ and $m_e$ which are set to the smallest block index and the largest block index that are fully contained in the interval $[s, e]$, respectively. Recall that for a certain block $m$, all iterations $t \in \mathcal{T}_m$ share the same prediction $\tilde{\mathbf{y}}_{m-1}$.Thus, for every $\mathbf{x} \in \mathcal{K}$ we have that

$$\sum_{t=s}^{e} \tilde{\nabla}_t^\top \left(\tilde{\mathbf{y}}_{m(t)-1} - \mathbf{x}\right) \leq \sum_{t=s}^{m_{s-1}K} \tilde{\nabla}_t^\top (\tilde{\mathbf{y}}_{m_{s-2}} - \mathbf{x}) + \sum_{m=m_s}^{m_e} \sum_{t \in \mathcal{T}_m} \tilde{\nabla}_t^\top (\tilde{\mathbf{y}}_{m-1} - \mathbf{x}) + \sum_{t=m_eK+1}^{e} \tilde{\nabla}_t^\top (\tilde{\mathbf{y}}_{m_e} - \mathbf{x}).$$

Using the Cauchy-Schwarz inequality, recalling that $\tilde{\mathbf{y}}_m \in R\mathcal{B}$ for all $m$, and $\|\nabla f_t(\mathbf{z})\| \leq G_f$ for all $t \geq 1$ and $\mathbf{z} \in \mathcal{RB}$, we have that $\tilde{\nabla}_t^\top (\tilde{\mathbf{y}}_{m(t)} - \mathbf{x}) \leq 2G_fR$ for every $t \geq 1$ and $\mathbf{x} \in \mathcal{K}$. Using Eq.(5), and this last observation, we have that for every $\mathbf{x} \in \mathcal{K}$,

$$\sum_{t=s}^{e} \tilde{\nabla}_t^\top \left(\tilde{\mathbf{y}}_{m(t)-1} - \mathbf{x}\right) \leq \sum_{m=m_s}^{m_e} \left( \frac{\|\tilde{\mathbf{y}}_{m-1} - \mathbf{x}\|^2}{2\eta} - \frac{\|\tilde{\mathbf{y}}_{m+1} - \mathbf{x}\|^2}{2\eta} + \frac{K^2 \eta G_f^2}{2} \right) + 4KG_fR.$$

Since for every $t \in [T]$, $f_t(\cdot)$ is convex in $R\mathcal{B}$, we have that

$$\forall \mathbf{x} \in \mathcal{K}: \sum_{t=s}^{e} f_t\left(\tilde{\mathbf{y}}_{m(t)-1}\right) - f_t(\mathbf{x}) \leq \frac{4R^2}{\eta} + \frac{\eta}{2} KG_f^2 T + 4KG_fR,$$

and thus the lemma follows. ■

**Proof** [Proof of Theorem 9] Denote $m(t) = \lceil \frac{t}{K} \rceil$. Fix some interval $[s, e], 1 \leq s \leq e \leq T$, and fix some minimizer $\mathbf{x}_I^* \in \operatorname{argmin}_{\mathbf{x} \in \mathcal{K}} \sum_{t=s}^{e} f_t(\mathbf{x})$. From the convexity of $f_t(\cdot)$ for every $t \in [T]$, we have that for every $\mathbf{x} \in R\mathcal{B}$ it holds that

$$\begin{aligned}\sum_{t=s}^{e} f_t\left(\mathbf{x}_{m(t)-1}\right) - f_t(\mathbf{x}) &= \sum_{t=s}^{e} f_t\left(\mathbf{x}_{m(t)-1}\right) - f_t\left(\tilde{\mathbf{y}}_{m(t)-1}\right) + f_t\left(\tilde{\mathbf{y}}_{m(t)-1}\right) - f_t(\mathbf{x}) \\ &\leq \sum_{t=s}^{e} \nabla_t^\top \left(\mathbf{x}_{m(t)-1} - \tilde{\mathbf{y}}_{m(t)-1}\right) + \sum_{t=s}^{e} f_t\left(\tilde{\mathbf{y}}_{m(t)-1}\right) - f_t(\mathbf{x}). \end{aligned} \tag{6}$$

Using Lemma 7, for every block $m$, Algorithm 3 returns points $(\mathbf{x}_m, \tilde{\mathbf{y}}_m) \in \mathcal{K} \times R\mathcal{B}$ such that $\|\mathbf{x}_m - \tilde{\mathbf{y}}_m\|^2 \leq 3\epsilon$. Since for every $t \geq 1$, $f_t(\cdot)$ is $G_f-$Lipschitz over $R\mathcal{B}$, from both observations and using the Cauchy-Schwarz inequality, for every $t \in [T]$ we have that,

$$\nabla_t^\top \left(\mathbf{x}_{m(t)-1} - \tilde{\mathbf{y}}_{m(t)-1}\right) \leq G_f \left\|\mathbf{x}_{m(t)-1} - \tilde{\mathbf{y}}_{m(t)-1}\right\| \leq G_f\sqrt{3\epsilon}. \tag{7}$$

Since Eq.(6) holds for any interval $[s, e]$, using Eq.(7) and the fact that $\sup_x\{f_1(\mathbf{x}) + f_2(\mathbf{x})\} \leq \sup_x\{f_1(\mathbf{x})\} + \sup_x\{f_2(\mathbf{x})\}$, we have that

$$\sup_{I=[s,e]\subseteq[T]} \left\{ \sum_{t=s}^{e} f_t\left(\mathbf{x}_{m(t)-1}\right) - \sum_{t=s}^{e} f_t(\mathbf{x}_I^*) \right\} \leq \sup_{I=[s,e]\subseteq[T]} \left\{ \sum_{t=s}^{e} f_t\left(\tilde{\mathbf{y}}_{m(t)-1}\right) - \sum_{t=s}^{e} f_t(\mathbf{x}_I^*) \right\} + G_f\sqrt{3\epsilon}T.$$

Using Lemma 19, we have that

$$\sup_{I=[s,e]\subseteq[T]}\left\{\sum_{t=s}^{e} f_t(\tilde{\mathbf{y}}_{m(t)-1}) - \min_{\mathbf{x}_I\in\mathcal{K}}\sum_{t=s}^{e} f_t(\mathbf{x}_I)\right\} \leq 4RG_fK + \frac{4R^2}{\eta} + \frac{G_f^2}{2}K\eta T.$$

Combining the last two equations and plugging-in the values of $\epsilon, \eta, K$ stated in the theorem, we obtain the adaptive regret bound stated in the theorem.

We now move on to upper-bound the overall number of calls to the linear optimization oracle. Recall that on each block $m \in [2, \ldots, T/K]$, the call to Algorithm 3 returns points $(\mathbf{x}_m, \tilde{\mathbf{y}}_m) \in \mathcal{K}\times R\mathcal{B}$ which satisfy $\|\mathbf{x}_m - \tilde{\mathbf{y}}_m\|^2 \leq 3\epsilon$, and Algorithm 4 updates $\mathbf{y}_{mK+1} = \tilde{\mathbf{y}}_{m-1} - \eta\sum_{t=(m-1)K+1}^{mK}\nabla_t$. Thus, the points $\mathbf{x}_{m-1}, \mathbf{y}_{mK+1}$ which are the input sent to Algorithm 3 on the following block $m+1$ satisfy:

$$\|\mathbf{x}_{m-1} - \mathbf{y}_{mK+1}\| \leq \|\mathbf{x}_{m-1} - \tilde{\mathbf{y}}_{m-1}\| + \|\tilde{\mathbf{y}}_{m-1} - \mathbf{y}_{mK+1}\| \leq \sqrt{3\epsilon} + K\eta G_f.$$

Using $(a+b)^2 \leq 2a^2 + 2b^2$, we have that for any block $m$,

$$\|\mathbf{x}_{m-1} - \mathbf{y}_{mK+1}\|^2 \leq 6\epsilon + 2K^2\eta^2G_f^2.$$

Using Lemma 7, each call to Algorithm 3 on some block $m$ makes at most

$$\max\left\{\frac{\|\mathbf{x}_{m-1} - \mathbf{y}_{mK+1}\|^2(\|\mathbf{x}_{m-1} - \mathbf{y}_{mK+1}\|^2 - \epsilon)}{4\epsilon^2} + 1, 1\right\}$$

iterations. On each iteration of Algorithm 3 it calls Algorithm 2, which according to Lemma 6, makes at most $\left\lceil \frac{27R^2}{\epsilon} - 2\right\rceil$ calls to a linear optimization oracle. Thus, Algorithm 4 on block $m$ makes

$$\begin{aligned} n_m &\leq \max\left\{\frac{\|\mathbf{x}_{m-1} - \mathbf{y}_{mK+1}\|^2(\|\mathbf{x}_{m-1} - \mathbf{y}_{mK+1}\|^2 - \epsilon)}{4\epsilon^2} + 1, 1\right\}\frac{27R^2}{\epsilon} \\ &\leq \left(8.5 + 5.5\frac{K^2\eta^2G_f^2}{\epsilon} + \frac{K^4\eta^4G_f^4}{\epsilon^2}\right)\frac{27R^2}{\epsilon} \end{aligned}$$

calls to linear optimization oracle. Thus, the overall number of calls to a linear optimization oracle is

$$N_{calls} = \sum_{m=1}^{T/K} n_m \leq \frac{T}{K}\left(8.5 + 5.5\frac{K^2\eta^2G_f^2}{\epsilon} + \frac{K^4\eta^4G_f^4}{\epsilon^2}\right)\frac{27R^2}{\epsilon}.$$

■

## Appendix E. Proof of Theorem 10

Before proving the theorem we need an additional observation.

**Observation 1** *For all $t \geq 1$ the followings hold*

$$1.\ \frac{(t+2)^{\frac{4}{3}}}{(t+1)^{\frac{2}{3}}} - \frac{(t+1)^{\frac{4}{3}}}{t^{\frac{2}{3}}} \leq \frac{2}{t^{\frac{1}{3}}}, \ \textit{and}\ \ 2.\ \frac{(t+2)^{\frac{4}{3}}}{(t+1)^2} - \frac{(t+1)^{\frac{4}{3}}}{t^2} \leq \frac{3}{t^{\frac{2}{3}}}.$$

**Proof** [Proof of Theorem 10] Recall that according to Lemma 7, Algorithm. 3 is an infeasible projection oracle. Denote $\tilde{\nabla}_t \in \partial f_t(\tilde{\mathbf{y}}_t)$. Since for every $t \in [T]$ we have that $\eta_t = \frac{1}{\alpha t}$, $f_t(\cdot)$ is $\alpha$-strongly convex, and $\tilde{\mathbf{y}}_t$ is an infeasible projection of $\mathbf{y}_t$ over $\mathcal{K}$, then, from Lemma 4, it follows that for every $\mathbf{x} \in \mathcal{K}$,

$$\sum_{t=1}^{T} f_t(\tilde{\mathbf{y}}_t) - f_t(\mathbf{x}) \leq \sum_{t=1}^{T} \frac{\|\tilde{\nabla}_t\|^2}{2\alpha t}.$$

Since for every $t \in [T]$, $f_t(\cdot)$ is also $G_f-$Lipschitz, using Lemma 7, we have that

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - f_t(\tilde{\mathbf{y}}_t) \leq \sum_{t=1}^{T} \nabla_t^\top(\mathbf{x}_t - \tilde{\mathbf{y}}_t) \leq \sqrt{3}G_f \sum_{t=1}^{T} \sqrt{\epsilon_t}.$$

Denote $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x})$. Since for every $t \in [T]$, $\tilde{\mathbf{y}}_t \in R\mathcal{B}$, it holds that $\|\tilde{\nabla}_t\| \leq G_f$. Since $\sum_{t=1}^{T} t^{-1} \leq 1 + \ln(T)$, and combining the two last equations, we have

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - f_t(\mathbf{x}^*) \leq \sqrt{3}G_f \sum_{t=1}^{T} \sqrt{\epsilon_t} + \frac{G_f^2}{2\alpha}(1 + \ln(T)).$$

Plugging-in the value of $\epsilon_t$ listed in the theorem, and using the facts that $\sum_{t=1}^{T} t^{-\frac{1}{3}} \leq \int_1^T t^{-\frac{1}{3}} dt \leq \frac{3}{2}T^{\frac{2}{3}}$, we obtain the regret bound.

We turn to upper-bound the number of calls to the linear optimization oracle. Recall that for every $t \in [T]$, Algorithm 3 returns points $\mathbf{x}_t, \tilde{\mathbf{y}}_t$ such that $\|\mathbf{x}_t - \tilde{\mathbf{y}}_t\|^2 \leq 3\epsilon_t$, and that Algorithm 5 updates $\mathbf{y}_{t+1} = \tilde{\mathbf{y}}_t - \eta_t\nabla_t$. Thus, for every $t \geq 1$, we have that

$$\|\mathbf{x}_t - \mathbf{y}_{t+1}\| \leq \|\mathbf{x}_t - \tilde{\mathbf{y}}_t + \tilde{\mathbf{y}}_t - \mathbf{y}_{t+1}\| \leq \sqrt{3\epsilon_t} + \eta_t G_f.$$

Since $(a+b)^2 \leq 2a^2 + 2b^2$, for any $t \geq 1$ we have that

$$\mathrm{dist}^2(\mathbf{y}_{t+1}, \mathcal{K}) \leq \|\mathbf{x}_t - \mathbf{y}_{t+1}\|^2 \leq 6\epsilon_t + 2\eta_t^2 G_f^2. \tag{8}$$

For every $t \geq 1$, let us denote by $n_t$ the number of iterations executed by Algorithm 3, when called from Algorithm 5 on iteration $t$. Using lemma 7, for every $t \in [T]$, after $n_t$ iterations of Algorithm 3, we have that

$$\begin{aligned}
\mathrm{dist}^2(\tilde{\mathbf{y}}_t, \mathcal{K}) &\leq \mathrm{dist}^2(\mathbf{y}_t, \mathcal{K}) - (n_t - 1)\frac{4\epsilon_t^2}{\|\mathbf{x}_{t-1} - \mathbf{y}_t\|^2} \\
&\leq \mathrm{dist}^2(\tilde{\mathbf{y}}_{t-1} - \eta_{t-1}\nabla_t, \mathcal{K}) - (n_t - 1)\frac{4\epsilon_t^2}{\|\mathbf{x}_{t-1} - \mathbf{y}_t\|^2} \\
&\leq \mathrm{dist}^2(\tilde{\mathbf{y}}_{t-1}, \mathcal{K}) + \mathrm{dist}(\tilde{\mathbf{y}}_{t-1}, \mathcal{K})\eta_{t-1}G_f + \eta_{t-1}^2 G_f^2 - (n_t - 1)\frac{4\epsilon_t^2}{\|\mathbf{x}_{t-1} - \mathbf{y}_t\|^2} \\
&\leq \mathrm{dist}^2(\tilde{\mathbf{y}}_{t-1}, \mathcal{K}) + 2\sqrt{3}\eta_{t-1}G_f\sqrt{\epsilon_{t-1}} + \eta_{t-1}^2 G_f^2 - (n_t - 1)\frac{4\epsilon_t^2}{\|\mathbf{x}_{t-1} - \mathbf{y}_t\|^2},
\end{aligned}$$

where the last inequality is due to $\mathrm{dist}(\tilde{\mathbf{y}}_{t-1}, \mathcal{K}) \leq \|\mathbf{x}_{t-1} - \tilde{\mathbf{y}}_{t-1}\| \leq \sqrt{3\epsilon_{t-1}}$.

Summing over all $T$ iterations, we have that

$$\sum_{t=1}^{T} n_t \leq T + \sum_{t=1}^{T} \frac{\|\mathbf{x}_t - \mathbf{y}_{t+1}\|^2}{4\epsilon_{t+1}^2}\left(\text{dist}^2(\tilde{\mathbf{y}}_t, \mathcal{K}) - \text{dist}^2(\tilde{\mathbf{y}}_{t+1}, \mathcal{K}) + 2\sqrt{3}\eta_t G_f \sqrt{\epsilon_t} + \eta_t^2 G_f^2\right).$$

Using Eq.(8), and since $\text{dist}^2(\tilde{\mathbf{y}}_1, \mathcal{K}) = 0$, and for every $t \in [T]$ $\text{dist}^2(\tilde{\mathbf{y}}_t, \mathcal{K}) \leq 3\epsilon_t$, we have that

$$\begin{aligned}\sum_{t=1}^{T} n_t &\leq T + \sum_{t=1}^{T} \frac{6\epsilon_t + 2\eta_t^2 G_f^2}{4\epsilon_{t+1}^2}\left(\text{dist}^2(\tilde{\mathbf{y}}_t, \mathcal{K}) - \text{dist}^2(\tilde{\mathbf{y}}_{t+1}, \mathcal{K}) + 2\sqrt{3}\eta_t G_f \sqrt{\epsilon_t} + \eta_t^2 G_f^2\right) \\ &\leq T + 3\sum_{t=1}^{T} \epsilon_{t+1}\left(\frac{3\epsilon_{t+1} + \eta_{t+1}^2 G_f^2}{2\epsilon_{t+2}^2} - \frac{3\epsilon_t + \eta_t^2 G_f^2}{2\epsilon_{t+1}^2}\right) + \frac{G_f^4 \eta_t}{\epsilon_{t+1}^2}\left(\frac{\sqrt{3}\epsilon_t^{\frac{3}{2}}}{G_f^3} + \frac{\eta_t \epsilon_t}{2G_f^2} + \frac{\eta_t^2 \sqrt{\epsilon_t}}{\sqrt{3}G_f} + \frac{\eta_t^3}{6}\right).\end{aligned}$$

Since according to Lemma 6 every iteration of Algorithm 3, when calling Algorithm 2, results in at most $\left\lceil \frac{27R^2}{\epsilon_{t+1}} - 2 \right\rceil$ calls to the linear optimization oracle, the overall number of calls to linear optimization oracle is

$$\begin{aligned}N_{calls} \leq &\sum_{t=1}^{T}\left(\frac{9\epsilon_{t+1}^2}{2\epsilon_{t+2}^2} - \frac{9\epsilon_t}{2\epsilon_{t+1}} + \frac{3\epsilon_{t+1}\eta_{t+1}^2 G_f^2}{2\epsilon_{t+2}^2} - \frac{3\eta_t^2 G_f^2}{2\epsilon_{t+1}}\right)\frac{27R^2}{\epsilon_{t+1}} \\ &+ 3G_f \sum_{t=1}^{T}\left(\frac{\sqrt{3}\eta_t \epsilon_t^{\frac{3}{2}}}{\epsilon_{t+1}^2} + \frac{G_f \eta_t^2 \epsilon_t}{2\epsilon_{t+1}^2} + \frac{\sqrt{3}G_f^2 \eta_t^3 \sqrt{\epsilon_t}}{3\epsilon_{t+1}^2} + \frac{G_f^3 \eta_t^4}{6\epsilon_{t+1}^2}\right)\frac{27R^2}{\epsilon_{t+1}}.\end{aligned}$$

Plugging-in the values of $\{\epsilon_t\}_{t\geq 1}, \{\eta_t\}_{t\geq 1}$ listed in the theorem, and denoting $c_1 = 130^{\frac{3}{2}}\frac{4G_f R^2}{\alpha}$, we have that

$$\begin{aligned}N_{calls} \leq &\frac{122R^2}{c_1^{\frac{2}{3}}}\sum_{t=1}^{T}\left(\frac{(t+2)^{\frac{4}{3}}}{(t+1)^{\frac{2}{3}}} - \frac{(t+1)^{\frac{4}{3}}}{t^{\frac{2}{3}}} + \frac{G_f^2}{3c_1^{\frac{2}{3}}\alpha^2}\left(\frac{(t+2)^{\frac{4}{3}}}{(t+1)^2} - \frac{(t+1)^{\frac{4}{3}}}{t^2}\right)\right) \\ &+ \frac{122G_f R^2}{\sqrt{3}c_1}\sum_{t=1}^{T}\frac{(t+1)^2}{t^2}\left(\frac{6}{\alpha} + \frac{\sqrt{3}G_f}{\alpha^2 c_1^{\frac{1}{3}} t^{\frac{2}{3}}} + \frac{2G_f^2}{\alpha^3 c_1^{\frac{2}{3}} t^{\frac{4}{3}}} + \frac{G_f^3}{\sqrt{3}c_1 \alpha^4 t^2}\right).\end{aligned}$$

Using Observation 1, and the fact that $\frac{(t+1)^2}{t^2} = \left(1 + \frac{1}{t}\right)^2 \leq 4$ for every $t \geq 1$, we have that

$$N_{calls} \leq \frac{122R^2}{c_1^{\frac{2}{3}}}\sum_{t=1}^{T}\left(\frac{2}{t^{\frac{1}{3}}} + \frac{G_f^2}{c_1^{\frac{2}{3}}\alpha^2 t^{\frac{2}{3}}}\right) + \frac{488G_f R^2}{\sqrt{3}c_1}\sum_{t=1}^{T}\left(\frac{6}{\alpha} + \frac{\sqrt{3}G_f}{\alpha^2 c_1^{\frac{1}{3}} t^{\frac{2}{3}}} + \frac{2G_f^2}{\alpha^3 c_1^{\frac{2}{3}} t^{\frac{4}{3}}} + \frac{G_f^3}{\sqrt{3}c_1 \alpha^4 t^2}\right).$$

Then, we have

$$N_{calls} \leq \frac{122R^2}{c_1^{\frac{2}{3}}}\left(3T^{\frac{2}{3}} + \frac{3G_f^2}{c_1^{\frac{2}{3}}\alpha^2}T^{\frac{1}{3}}\right) + \frac{488G_f R^2}{\sqrt{3}c_1}\left(\frac{6}{\alpha}T + \frac{3\sqrt{3}G_f}{\alpha^2 c_1^{\frac{1}{3}}}T^{\frac{1}{3}} + \frac{6G_f^2}{\alpha^3 c_1^{\frac{2}{3}}} + \frac{G_f^3}{\sqrt{3}c_1\alpha^4}\right).$$

Plugging in $c_1 = 130^{\frac{3}{2}}\frac{4G_f R^2}{\alpha}$, we obtain the lemma. ■

**Algorithm 9:** Blocked Bandit Gradient Descent using Linear Optimization Oracle (LOO-BBGD)

**Data:** horizon $T$, feasible set $\mathcal{K}$ with parameters $r, R$, block size $K$, step size $\eta$, smoothing parameter $\delta \in (0, r]$
$\mathbf{x}_0, \mathbf{x}_1 \leftarrow$ arbitrary points in $\mathcal{K}_{\delta/r}$
$\tilde{\mathbf{y}}_0 \leftarrow \mathbf{x}_0$, $\mathbf{y}_1 \leftarrow \tilde{\mathbf{y}}_0, \tilde{\mathbf{y}}_1 \leftarrow \mathbf{x}_1$.
**for** $t = 1, \ldots, K$ **do**
  Set $\mathbf{u}_t \sim S^n$ and play $\mathbf{z}_t = \tilde{\mathbf{x}}_0 + \delta \mathbf{u}_t$.
  Observe $f_t(\mathbf{z}_t)$, set $\mathbf{g}_t = \frac{n}{\delta} f_t(\mathbf{z}_t)\mathbf{u}_t$ and update $\mathbf{y}_{t+1} = \mathbf{y}_t - \eta \mathbf{g}_t$.
**end**
**for** $m = 2, \ldots, \frac{T}{K}$ **do**
  Let $(\mathbf{x}_m, \tilde{\mathbf{y}}_m)$ be the output of Algorithm 3 with set $\mathcal{K}_{\delta/r}$, feasible point $\mathbf{x}_{m-2}$, initial vector $\mathbf{y}_{(m-1)K+1}$, and tolerance $\frac{\delta^2}{3}$ (execute **in parallel** to following **for** loop over $s$)
  Set $\mathbf{y}_{(m-1)K+1} = \tilde{\mathbf{y}}_{m-1}$
  **for** $s = 1, \ldots, K$ **do**
    Set $\mathbf{u}_t \sim S^n$ and play $\mathbf{z}_t = \tilde{\mathbf{x}}_{m-1} + \delta \mathbf{u}_t$. ; /* $t = (m-1)K + s$ */
    Observe $f_t(\mathbf{z}_t)$, set $\mathbf{g}_t = \frac{n}{\delta} f_t(\mathbf{z}_t)\mathbf{u}_t$ and update $\mathbf{y}_{t+1} = \mathbf{y}_t - \eta \mathbf{g}_t$.
  **end**
  **Note:** $\mathbf{y}_{mK+1} = \tilde{\mathbf{y}}_{m-1} - \eta \sum_{t=(m-1)K+1}^{mK} \mathbf{g}_t$.
**end**

## Appendix F. LLO-based Algorithm for the Bandit Setting and Proof of Theorem 11

Our LLO-based algorithm for the bandit setting is given in Algorithm 9. Before proving Theorem 11, we need an additional lemma.

**Lemma 20** *Fix some interval $\mathcal{T} = \{\tau+1, \ldots, \tau+L\}$ of size L, a set of i.i.d. samples $\{\mathbf{u}_t\}_{t\in\mathcal{T}}$, $\mathbf{u}_t \sim \mathcal{S}^n$, and some $\mathbf{y} \in (1-\delta/r)\mathcal{K} = \mathcal{K}_{\delta/r}$, for some $\delta \in (0, r)$. Define $\mathbf{g}_t = \frac{n}{\delta} f_t(\mathbf{y} + \delta \mathbf{u}_t)\mathbf{u}_t, t \in \mathcal{T}$, and let $\widehat{\mathbf{g}}_{\mathcal{T}} = \sum_{t\in\mathcal{T}} \mathbf{g}_t$. Then, it holds that*

1. $\mathbb{E}\left[\|\widehat{\mathbf{g}}_{\mathcal{T}}\|\right]^2 \leq \mathbb{E}\left[\|\widehat{\mathbf{g}}_{\mathcal{T}}\|^2\right] \leq L\left(\frac{nM}{\delta}\right)^2 + L^2 G_f^2$.

2. $\mathbb{E}\left[\|\widehat{\mathbf{g}}_{\mathcal{T}}\|^4\right] \leq 3L^2\left(\frac{nM}{\delta}\right)^4 + 6L^3\left(\frac{nM}{\delta}\right)^2 G_f^2 + L^4 G_f^4$.

**Proof** We start with the first item. It holds that

$$\mathbb{E}\left[\|\widehat{\mathbf{g}}_{\mathcal{T}}\|^2\right] = \mathbb{E}\left[\left\|\sum_{t\in\mathcal{T}} \mathbf{g}_t\right\|^2\right] = \mathbb{E}\left[\sum_{t\in\mathcal{T}} \|\mathbf{g}_t\|^2 + \sum_{(i,j)\in\mathcal{T}^2, i\neq j} \mathbf{g}_i^\top \mathbf{g}_j\right]$$
$$= \mathbb{E}\left[\sum_{t\in\mathcal{T}} \|\mathbf{g}_t\|^2\right] + \sum_{(i,j)\in\mathcal{T}^2, i\neq j} \mathbb{E}\left[\mathbf{g}_i^\top \mathbf{g}_j\right]. \quad (9)$$

Since $\max_{\mathbf{x}\in\mathcal{K}} |f_t(\mathbf{x})| \leq M$, we have that $\|\mathbf{g}_t\| \leq \frac{n}{\delta}|f_t(\mathbf{y}+\delta\mathbf{u}_t)|\|\mathbf{u}_t\| \leq \frac{nM}{\delta}$, and thus,

$$\sum_{t\in\mathcal{T}} \|\mathbf{g}_t\|^2 \leq L\left(\frac{nM}{\delta}\right)^2. \tag{10}$$

Using Lemma 18 we have that for all $t \in \mathcal{T}$, $\|\mathbb{E}[\mathbf{g}_t|\mathbf{y}]\| = \left\|\nabla \widehat{f}_{t,\delta}(\mathbf{y})\right\| \leq G_f$. Furthermore, since, conditioned on $\mathbf{y}$, $\forall i \neq j$, $\mathbf{g}_i$, $\mathbf{g}_j$ are independent random vectors, we have that

$$\sum_{(i,j)\in\mathcal{T}^2, i\neq j} \mathbb{E}\left[\mathbf{g}_i^\top \mathbf{g}_j\right] = \sum_{(i,j)\in\mathcal{T}^2, i\neq j} \mathbb{E}\left[\mathbb{E}[\mathbf{g}_i|\mathbf{y}]^\top \mathbb{E}[\mathbf{g}_j|\mathbf{y}]\right] \leq \left(L^2 - L\right) G_f^2. \tag{11}$$

Combining Equations (9), (10), and (11), we obtain the first part of the lemma:

$$\mathbb{E}\left[\|\widehat{\mathbf{g}}_{\mathcal{T}}\|\right]^2 \leq \mathbb{E}\left[\|\widehat{\mathbf{g}}_{\mathcal{T}}\|^2\right] \leq L\left(\frac{nM}{\delta}\right)^2 + \left(L^2 - L\right) G_f^2,$$

where the first inequality follows from Jensen's inequality.

We move on to prove the second part of the lemma. It holds that

$$\begin{aligned}
\mathbb{E}\left[\|\widehat{\mathbf{g}}_{\mathcal{T}}\|^4\right] &= \mathbb{E}\left[\left\|\sum_{t\in\mathcal{T}_m} \mathbf{g}_t\right\|^4\right] = \mathbb{E}\left[\left(\sum_{t\in\mathcal{T}} \|\mathbf{g}_t\|^2 + \sum_{(i,j)\in\mathcal{T}^2, i\neq j} \mathbf{g}_i^\top \mathbf{g}_j\right)^2\right] \\
&= \mathbb{E}\left[\left(\sum_{t\in\mathcal{T}} \|\mathbf{g}_t\|^2\right)^2\right] + 2\mathbb{E}\left[\left(\sum_{t\in\mathcal{T}} \|\mathbf{g}_t\|^2\right)\left(\sum_{(i,j)\in\mathcal{T}^2, i\neq j} \mathbf{g}_i^\top \mathbf{g}_j\right)\right] + \mathbb{E}\left[\left(\sum_{(i,j)\in\mathcal{T}^2, i\neq j} \mathbf{g}_i^\top \mathbf{g}_j\right)^2\right].
\end{aligned}$$

Using Eq. (10) and Eq. (11) we have,

$$\begin{aligned}
\mathbb{E}\left[\|\widehat{\mathbf{g}}_{\mathcal{T}}\|^4\right] &\leq L^2\left(\frac{nM}{\delta}\right)^4 + 2L\left(\frac{nM}{\delta}\right)^2 \sum_{(i,j)\in\mathcal{T}^2, i\neq j} \mathbb{E}\left[\mathbf{g}_i^\top \mathbf{g}_j\right] + \mathbb{E}\left[\left(\sum_{(i,j)\in\mathcal{T}^2, i\neq j} \mathbf{g}_i^\top \mathbf{g}_j\right)^2\right] \\
&\leq L^2\left(\frac{nM}{\delta}\right)^4 + 2L\left(\frac{nM}{\delta}\right)^2 (L^2 - L)G_f^2 + \mathbb{E}\left[\left(\sum_{(i,j)\in\mathcal{T}^2, i\neq j} \mathbf{g}_i^\top \mathbf{g}_j\right)^2\right].
\end{aligned}$$

Now we upper-bound the last term in the RHS. Note that, the expectation argument has $(L^2 - L)^2$ summands. Since conditioned on $\mathbf{y}$, for every four indices $i \neq j \neq k \neq l$, the random vectors $\mathbf{g}_i$, $\mathbf{g}_j$, $\mathbf{g}_k$, $\mathbf{g}_l$ are independent, we have that

$$\mathbb{E}\left[\mathbf{g}_i^\top \mathbf{g}_j \mathbf{g}_k^\top \mathbf{g}_l\right] = \mathbb{E}\left[\mathbb{E}[\mathbf{g}_i^\top|\mathbf{y}]\mathbb{E}[\mathbf{g}_j|\mathbf{y}]\mathbb{E}[\mathbf{g}_k^\top|\mathbf{y}]\mathbb{E}[\mathbf{g}_l|\mathbf{y}]\right] \leq G_f^4, \tag{12}$$

where the last inequality follows, as before, from Lemma 18 which yields $\|\mathbb{E}[\mathbf{g}_t|\mathbf{y}]\| \leq G_f$ for all $t \in \mathcal{T}$.

In the case of three different indices $i \neq j \neq k$, we have

$$\begin{aligned}
\mathbb{E}\left[\mathbf{g}_j^\top \mathbf{g}_i \mathbf{g}_k^\top \mathbf{g}_j\right] &= \mathbb{E}\left[\mathbf{g}_j^\top \mathbf{g}_i \mathbf{g}_j^\top \mathbf{g}_k\right] = \mathbb{E}\left[\mathbf{g}_i^\top \mathbf{g}_j \mathbf{g}_k^\top \mathbf{g}_j\right] = \mathbb{E}\left[\mathbf{g}_i^\top \mathbf{g}_j \mathbf{g}_j^\top \mathbf{g}_k\right] \\
&= \mathbb{E}\left[\mathbb{E}[\mathbf{g}_i|\mathbf{y}]^\top \mathbb{E}[\mathbf{g}_j \mathbf{g}_j^\top|\mathbf{y}]\mathbb{E}[\mathbf{g}_k|\mathbf{y}]\right] \leq \mathbb{E}\left[\|\mathbb{E}[\mathbf{g}_i|\mathbf{y}]\|\|\mathbb{E}[\mathbf{g}_j \mathbf{g}_j^\top|\mathbf{y}]\|\|\mathbb{E}[\mathbf{g}_k|\mathbf{y}]\|\right] \\
&\leq \mathbb{E}[\|\mathbf{g}_j\|^2]G_f^2 \leq \frac{n^2 M^2 G_f^2}{\delta^2}.
\end{aligned} \tag{13}$$

There are $L(L-1)(L-2)(L-3)$ summands with four different indices, and $2(L^2-L)$ summands with exactly two different indices. Thus, since there are overall $(L^2-L)^2$ summands, there are $4L^3-12L^2+8L$ summands wiht exactly three different indices. Thus, using Lemma 18, Eq. (12), and Eq. (13), it holds that

$$\mathbb{E}\left[\left(\sum_{(i,j)\in\mathcal{T}^2, i\neq j}\mathbf{g}_i^\top\mathbf{g}_j\right)^2\right]\leq 2L^2\left(\frac{nM}{\delta}\right)^4+4L^3\left(\frac{nM}{\delta}\right)^2G_f^2+L^4G_f^4.$$

Thus, we obtain that

$$\mathbb{E}\left[\|\widehat{\mathbf{g}}_{\mathcal{T}}\|^4\right]\leq 3L^2\left(\frac{nM}{\delta}\right)^4+6L^3\left(\frac{nM}{\delta}\right)^2G_f^2+L^4G_f^4.$$

■

**Proof** [Proof of Theorem 11] First, we establish that Algorithm 3 indeed plays feasible points. Using Lemma 7, for each block $m\in[2,\ldots,T/K]$, Algorithm 3 returns $\mathbf{x}_m\in\mathcal{K}_{\delta/r}=(1-\delta/r)\mathcal{K}$. Thus, for every iteration $t\in[T]$ it indeed holds that $\mathbf{z}_t\in\mathcal{K}$.

We now turn to prove the upper-bound to the adaptive expected regret. Throughout the proof of the regret bound let us fix some interval $I=[s,e], 1\leq s\leq e\leq T$. We start with an upper bound on $\mathbb{E}\left[\sum_{t=s}^{e}\widehat{f}_{\delta,t}\left(\mathbf{x}_{m(t)-1}\right)-\widehat{f}_{\delta,t}(\mathbf{x})\right]$ which holds for every $\mathbf{x}\in\mathcal{K}_{\delta/r}$. We will first take a few preliminary steps. For all $t\in[T]$, denote the history of all predictions and gradient estimates by $\mathcal{F}_t=\{\mathbf{x}_1,\ldots,\mathbf{x}_{m(t-1)},\mathbf{g}_1,\ldots,\mathbf{g}_{t-1}\}$, where $m(t):=\lceil\frac{t}{K}\rceil$. Since for all $t\in[T]$, $\mathbf{g}_t$ is an unbiased estimator of $\nabla\widehat{f}_{t,\delta}\left(\mathbf{x}_{m(t)-1}\right)$, i.e., $\mathbb{E}\left[\mathbf{g}_t|\mathcal{F}_t\right]=\nabla\widehat{f}_{t,\delta}\left(\mathbf{x}_{m(t)-1}\right)$, and $\mathbb{E}\left[\mathbf{x}_{m(t)-1}|\mathcal{F}_t\right]=\mathbf{x}_{m(t)-1}$, we have that for every $t\in[T]$ and $\mathbf{x}\in\mathcal{K}_{\delta/r}$ it holds that,

$$\mathbb{E}\left[\mathbf{g}_t^\top\left(\mathbf{x}_{m(t)-1}-\mathbf{x}\right)\right]=\mathbb{E}\left[\mathbb{E}\left[\mathbf{g}_t|\mathcal{F}_t\right]^\top\left(\mathbf{x}_{m(t)-1}-\mathbf{x}\right)\right]=\mathbb{E}\left[\nabla\widehat{f}_{\delta,t}(\mathbf{x}_{m(t)-1})^\top(\mathbf{x}_{m(t)-1}-\mathbf{x})\right].\tag{14}$$

For every block $m\in[T/K]$, denote $\mathcal{T}_m=\{(m-1)K+1,\ldots,mK\}$. Using Lemma 7, we have that for every block $m\in[T/K]$, the point $\tilde{\mathbf{y}}_{m+2}$ is an infeasible projection of $\mathbf{y}_{(m+1)K+1}$ over $\mathcal{K}_{\delta/r}$. Since $\mathbf{y}_{(m+1)K+1}=\tilde{\mathbf{y}}_m-\eta\sum_{t\in\mathcal{T}_{m+1}}\mathbf{g}_t$, we have that for every block $m\in[T/K]$ and $\mathbf{x}\in\mathcal{K}_{\delta/r}$, it holds that

$$\begin{aligned}\|\tilde{\mathbf{y}}_{m+2}-\mathbf{x}\|^2\leq\|\mathbf{y}_{(m+1)K+1}-\mathbf{x}\|^2&=\left\|\tilde{\mathbf{y}}_m-\eta\sum\nolimits_{t\in\mathcal{T}_{m+1}}\mathbf{g}_t-\mathbf{x}\right\|^2\\&=\|\tilde{\mathbf{y}}_m-\mathbf{x}\|^2+\eta^2\left\|\sum\nolimits_{t\in\mathcal{T}_{m+1}}\mathbf{g}_t\right\|^2-2\eta\sum\nolimits_{t\in\mathcal{T}_{m+1}}\mathbf{g}_t^\top(\tilde{\mathbf{y}}_m-\mathbf{x}).\end{aligned}$$

Rearranging, we have for every block $m$ that,

$$\sum\nolimits_{t\in\mathcal{T}_{m+1}}\mathbf{g}_t^\top(\tilde{\mathbf{y}}_m-\mathbf{x})\leq\frac{\|\tilde{\mathbf{y}}_m-\mathbf{x}\|^2}{2\eta}-\frac{\|\tilde{\mathbf{y}}_{m+2}-\mathbf{x}\|^2}{2\eta}+\frac{\eta}{2}\left\|\sum\nolimits_{t\in\mathcal{T}_{m+1}}\mathbf{g}_t\right\|^2.\tag{15}$$

Denote by $m_s$ and $m_e$ the smallest and the largest index of block that is fully contained in the interval $[s,e]$, respectively, i.e., $\{(m_s-1)K+1,\ldots,m_eK\}=\{\mathcal{T}_{m_s},\ldots,\mathcal{T}_{m_e}\}\subseteq[s,e]$. Recall that all iterations $t \in \mathcal{T}_m$ share the same prediction $\tilde{\mathbf{y}}_{m-1}$. Since $\{s, \ldots, m_{s-1}K\} \subset \mathcal{T}_{m_{s-1}}$ and $\{m_e K + 1, \ldots, e\} \subset \mathcal{T}_{m_{e+1}}$, for every $\mathbf{x} \in \mathcal{K}_{\delta/r}$ we have that

$$\mathbb{E}\left[\sum_{t=s}^{e} \mathbf{g}_t^\top \left(\tilde{\mathbf{y}}_{m(t)-1} - \mathbf{x}\right)\right] = \mathbb{E}\left[\sum_{t=s}^{m_{s-1}K} \mathbf{g}_t^\top (\tilde{\mathbf{y}}_{m_{s-2}} - \mathbf{x})\right] + \mathbb{E}\left[\sum_{m=m_s}^{m_e} \sum_{t\in\mathcal{T}_m} \mathbf{g}_t^\top (\tilde{\mathbf{y}}_{m-1} - \mathbf{x})\right] + \mathbb{E}\left[\sum_{t=m_eK+1}^{e} \mathbf{g}_t^\top (\tilde{\mathbf{y}}_{m_e} - \mathbf{x})\right].$$

Using the Cauchy-Schwarz inequality, Lemma 7 (which yields that $\tilde{\mathbf{y}}_m \in R\mathcal{B}$), and Lemma 20, with the fact that for all $a, b \in \mathbb{R}^+ : \sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, we obtain the bound: $\mathbb{E}\left[\sum_{t=mK+1}^{(m+1)K} \mathbf{g}_t^\top (\tilde{\mathbf{y}}_{m(t)} - \mathbf{x})\right] \leq 2RK\left(\frac{nM}{\delta\sqrt{K}} + G_f\right)$ for every $t \in [T]$, and $\mathbf{x} \in \mathcal{K}_{\delta/r}$. Combining Eq.(15), and this bound, we have that

$$\mathbb{E}\left[\sum_{t=s}^{e} \mathbf{g}_t^\top \left(\tilde{\mathbf{y}}_{m(t)-1} - \mathbf{x}\right)\right] \leq 4RK\left(\frac{nM}{\delta\sqrt{K}} + G_f\right) + \frac{4R^2}{\eta} + \frac{\eta}{2}\sum_{m=m_s}^{m_e} \mathbb{E}\left[\left\|\sum_{t\in\mathcal{T}_m} \mathbf{g}_t\right\|^2\right].$$

Combining Eq.(14) and Lemma 20, we have that for every $\mathbf{x} \in \mathcal{K}_{\delta/r}$ it holds that,

$$\sum_{t=s}^{e} \mathbb{E}\left[\nabla \widehat{f}_{\delta,t}\left(\mathbf{x}_{m(t)-1}\right)^\top \left(\mathbf{x}_{m(t)-1} - \mathbf{x}\right)\right] \leq \mathbb{E}\left[\sum_{t=s}^{e} \mathbf{g}_t^\top \left(\mathbf{x}_{m(t)-1} - \tilde{\mathbf{y}}_{m(t)-1}\right)\right] + \frac{4R^2}{\eta} + 4RK\left(\frac{nM}{\delta\sqrt{K}} + G_f\right) + \frac{\eta}{2}K\left(\frac{n^2M^2}{\delta^2 K} + G_f^2\right)T.$$

From Lemma 7, we have that for every block $m$, $\|\mathbf{x}_{m-1} - \tilde{\mathbf{y}}_{m-1}\| \leq \delta$. Using Lemma 20 with the fact that for all $a, b \in \mathbb{R}^+ : \sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, we have that $\mathbb{E}\left[\|\sum_{t\in\mathcal{T}_m} \mathbf{g}_t\|\right] \leq \sqrt{K}\left(\frac{nM}{\delta}\right) + KG_f$ for every block $m$. Plugging-in these two observations, we have that for every $\mathbf{x} \in \mathcal{K}_{\delta/r}$ it holds that,

$$\sum_{t=s}^{e} \mathbb{E}\left[\nabla \widehat{f}_{\delta,t}\left(\mathbf{x}_{m(t)-1}\right)^\top \left(\mathbf{x}_{m(t)-1} - \mathbf{x}\right)\right] \leq \left(\frac{nM}{\sqrt{K}} + \delta G_f\right)T + 4RK\left(\frac{nM}{\delta\sqrt{K}} + G_f\right) + \frac{4R^2}{\eta} + \frac{\eta}{2}K\left(\frac{n^2M^2}{\delta^2K} + G_f^2\right)T.$$

Since for every $t \in [T]$, $f_t(\cdot)$ is convex in $\mathcal{K}$, using Lemma 16 it holds that the smoothed function $\widehat{f}_{t,\delta}(\cdot)$ is convex in $\mathcal{K}_{\delta/r}$ for all $t \in [T]$. Thus, for every $\mathbf{x} \in \mathcal{K}_{\delta/r}$ we obtain that,

$$\mathbb{E}\left[\sum_{t=s}^{e} \widehat{f}_{\delta,t}\left(\mathbf{x}_{m(t)-1}\right) - \widehat{f}_{\delta,t}(\mathbf{x})\right] \leq \left(\frac{nM}{\sqrt{K}} + \delta G_f\right)T + 4RK\left(\frac{nM}{\delta\sqrt{K}} + G_f\right) + \frac{4R^2}{\eta} + \frac{\eta K}{2}\left(\frac{n^2M^2}{K\delta^2} + G_f^2\right)T. \tag{16}$$

Let us now denote by $\mathbf{x}_I^*$ a feasible minimizer w.r.t. the interval $I = [s, e]$, i.e. $\mathbf{x}_I^* \in \underset{\mathbf{x}\in\mathcal{K}}{\operatorname{argmin}} \sum_{t=s}^{e} f_t(\mathbf{x})$. Denote also $\tilde{\mathbf{x}}_I^* = \left(1 - \frac{\delta}{r}\right)\mathbf{x}_I^* \in \mathcal{K}_{\delta/r}$. It holds that,

$$\mathbb{E}\left[\sum_{t=s}^{e} f_t(\mathbf{z}_t) - f_t(\mathbf{x}_I^*)\right] = \mathbb{E}\left[\sum_{t=s}^{e} f_t(\mathbf{z}_t) - f_t\left(\mathbf{x}_{m(t)-1}\right) + f_t\left(\mathbf{x}_{m(t)-1}\right) - f_t(\tilde{\mathbf{x}}_I^*) + f_t(\tilde{\mathbf{x}}_I^*) - f_t(\mathbf{x}_I^*)\right]. \tag{17}$$

Since for every $t \in [T]$, $\mathbf{z}_t = \mathbf{x}_{m(t)-1} + \delta\mathbf{u}_t$, and $f_t$ is $G_f$-Lipschitz over $\mathcal{K}$, we have that

$$\begin{aligned}\mathbb{E}\left[\sum_{t=s}^{e} f_t(\mathbf{z}_t) - f_t(\mathbf{x}_{m(t)-1})\right] &= \sum_{t=s}^{e} \mathbb{E}\left[f_t(\mathbf{x}_{m(t)-1} + \delta\mathbf{u}_t) - f_t(\mathbf{x}_{m(t)-1})\right] \\ &\leq \sum_{t=s}^{e} \mathbb{E}\left[G_f\delta\|\mathbf{u}_t\|\right] \leq G_f\delta T,\end{aligned}$$

and since $\|\mathbf{x}_I^*\| \leq R$, we have

$$\mathbb{E}\left[\sum_{t=s}^{e} f_t(\tilde{\mathbf{x}}_I^*) - f_t(\mathbf{x}_I^*)\right] = \sum_{t=s}^{e} f_t(\tilde{\mathbf{x}}_I^*) - f_t(\mathbf{x}_I^*) \leq \sum_{t=s}^{e} G_f\|\tilde{\mathbf{x}}_I^* - \mathbf{x}_I^*\| \leq \frac{RG_f}{r}\delta T.$$

Using Lemma 16 and Eq.(16), we have

$$\begin{aligned}\mathbb{E}\left[\sum_{t=s}^{e} f_t\left(\mathbf{x}_{m(t)-1}\right) - f_t(\tilde{\mathbf{x}}_I^*)\right] &= \mathbb{E}\left[\sum_{t=s}^{e} f_t\left(\mathbf{x}_{m(t)-1}\right) - \widehat{f}_{\delta,t}\left(\mathbf{x}_{m(t)-1}\right)\right] \\ &\quad + \mathbb{E}\left[\sum_{t=s}^{e} \widehat{f}_{\delta,t}\left(\mathbf{x}_{m(t)-1}\right) - \widehat{f}_{\delta,t}(\tilde{\mathbf{x}}_I^*)\right] + \mathbb{E}\left[\sum_{t=s}^{e} \widehat{f}_{\delta,t}(\tilde{\mathbf{x}}_I^*) - f_t(\tilde{\mathbf{x}}_I^*)\right] \\ &\leq 2\delta G_f T + \left(\frac{nM}{\sqrt{K}} + \delta G_f\right)T + 4RK\left(\frac{nM}{\delta\sqrt{K}} + G_f\right) + \frac{4R^2}{\eta} + \frac{\eta}{2}K\left(\frac{n^2M^2}{\delta^2 K} + G_f^2\right)T.\end{aligned}$$

Combining the last three equations and Eq.(17), we obtain that

$$\begin{aligned}\mathbb{E}\left[\sum_{t=s}^{e} f_t(\mathbf{z}_t) - f_t(\mathbf{x}_I^*)\right] &\leq \left(3 + \frac{R}{r}\right)G_f\delta T + \left(\frac{nM}{\sqrt{K}} + \delta G_f\right)T + 4RK\left(\frac{nM}{\delta\sqrt{K}} + G_f\right) \\ &\quad + \frac{4R^2}{\eta} + \frac{\eta}{2}\left(\frac{n^2M^2}{\delta^2} + KG_f^2\right)T.\end{aligned}$$

Plugging-in the values of $K, \eta, \delta$ listed in the theorem, we obtain the regret bound of the theorem.

We now turn to prove the upper-bound on the expected overall number of calls to the LOO. We start with find an upper-bound on $\mathbb{E}\left[\|\mathbf{x}_{m-1} - \mathbf{y}_{mK+1}\|^4\right]$. Since $(a+b)^4 \leq 8(a^4 + b^4)$, we have

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{x}_{m-1} - \mathbf{y}_{mK+1}\|^4\right] &= \mathbb{E}\left[\|\mathbf{x}_{m-1} - \tilde{\mathbf{y}}_{m-1} + \tilde{\mathbf{y}}_{m-1} - \mathbf{y}_{mK+1}\|^4\right] \\ &\leq 8\mathbb{E}\left[\|\mathbf{x}_{m-1} - \tilde{\mathbf{y}}_{m-1}\|^4 + \|\tilde{\mathbf{y}}_{m-1} - \mathbf{y}_{mK+1}\|^4\right].\end{aligned}$$

Using Lemma 7, for every block $m$, Algorithm 3 returns points $\mathbf{x}_m, \tilde{\mathbf{y}}_m$ such that $\|\mathbf{x}_m - \tilde{\mathbf{y}}_m\|^2 \leq \delta^2$. Since Algorithm 9 updates $\mathbf{y}_{mK+1} = \tilde{\mathbf{y}}_{m-1} - \eta \sum_{t\in\mathcal{T}_m} \mathbf{g}_t$, using Lemma 20, we have that

$$\mathbb{E}\left[\|\mathbf{x}_{m-1} - \mathbf{y}_{mK+1}\|^4\right] \leq 8\left(\delta^4 + \eta^4 \mathbb{E}\left[\left\|\sum_{t\in\mathcal{T}_m} \mathbf{g}_t\right\|^4\right]\right)$$
$$\leq 8\left(\delta^4 + 3\eta^4 K^2 \left(\frac{nM}{\delta}\right)^4 + 6\eta^4 K^3 \left(\frac{nM}{\delta}\right)^2 G_f^2 + \eta^4 K^4 G_f^4\right).$$

Using Lemma 7, for every block $m$, Algorithm 3 makes at most

$$\max\left\{\frac{\|\mathbf{x}_{m-1} - \mathbf{y}_{mK+1}\|^2 \left(\|\mathbf{x}_{m-1} - \mathbf{y}_{mK+1}\|^2 - \frac{\delta^2}{3}\right)}{4\left(\frac{\delta^2}{3}\right)^2} + 1, 1\right\}$$

iterations, where $\delta^2/3$ is the error tolerance. On each iteration of Algorithm 3, it calls Algorithm 2, which in turn, by Lemma 6, makes at most $\left\lceil \frac{27R^2}{\delta^2/3} - 2 \right\rceil$ calls to a linear optimization oracle. Thus, the call to Algorithm 3 in block $m$ executes

$$\mathbb{E}[n_m] \leq \mathbb{E}\left[\frac{\|\mathbf{x}_{m-1} - \mathbf{y}_{mK+1}\|^2 \left(\|\mathbf{x}_{m-1} - \mathbf{y}_{mK+1}\|^2 - \frac{\delta^2}{3}\right)}{4\left(\frac{\delta^2}{3}\right)^2} + 1\right] \frac{81R^2}{\delta^2}$$
$$\leq \left(\frac{18\eta^4 K^2 \left(3\left(\frac{nM}{\delta}\right)^4 + 6K\left(\frac{nM}{\delta}\right)^2 G_f^2 + K^2 G_f^4\right)}{\delta^4} + 19\right) \frac{81R^2}{\delta^2}$$

calls to linear optimization oracle in expectation. Thus, the expected overall number of calls to a linear optimization oracle is bounded by

$$\mathbb{E}\left[N_{calls}\right] = \sum_{m=1}^{T/K} \mathbb{E}\left[n_m\right] \leq \frac{T}{K}\left(\frac{54\eta^4 K^2 (nM)^4}{\delta^8} + \frac{108\eta^4 K^3 (nM)^2 G_f^2}{\delta^6} + \frac{18\eta^4 K^4 G_f^4}{\delta^4} + 19\right)\frac{81R^2}{\delta^2}.$$

It only remains to plug-in the value of $K, \eta, \delta$ listed in the theorem. ■

## Appendix G. Missing Proofs from Section 4.1

**Lemma 12** Suppose Assumption 1 holds. Fix $(\delta, \delta') \in (0,1) \times [0, r')$, and let $\mathbf{y} \in \mathbb{R}^n$ such that $\frac{\mathbf{y}}{1-\delta'/r} \notin \mathcal{K}$. Let $\mathbf{g} \in \mathbb{R}^n$ be the output of the SO of $\mathcal{K}$ w.r.t. $\frac{\mathbf{y}}{1-\delta'/r}$, i.e., for all $\mathbf{x} \in \mathcal{K}$, $\left(\frac{\mathbf{y}}{1-\delta'/r} - \mathbf{x}\right)^\top \mathbf{g} > 0$. Then, it holds that,

$$\forall \mathbf{z} \in \mathcal{K}_{\delta,\delta'/r}: \quad (\mathbf{y} - \mathbf{z})^\top \mathbf{g} > \delta(r - \delta')\|\mathbf{g}\|.$$

Before proving the lemma we require an additional observation.

**Observation 2** *Suppose Assumption 1 holds and fix some $(\delta, \delta') \in [0,1] \times [0, r]$. Then, for all $\mathbf{z} \in \mathcal{K}_{\delta,\delta'/r} = (1-\delta)(1-\delta'/r)\mathcal{K}$, it holds that $\mathbf{z} + \delta(r - \delta')\mathcal{B} \subseteq \mathcal{K}_{\delta'/r}$.*

**Proof** [Proof of Lemma 12] Note that $\mathcal{K}_\delta = (1-\delta)\mathcal{K} \subseteq \mathcal{K}$, and $\mathcal{K}_{\delta,\delta'/r} = (1-\delta'/r)(1-\delta)\mathcal{K} \subseteq \mathcal{K}_\delta$. Since for all $\mathbf{x} \in \mathcal{K}$, $(\mathbf{y} - (1-\delta'/r)\mathbf{x})^\top \mathbf{g} > 0$, we have that for all $\mathbf{w} \in \mathcal{K}_{\delta'/r}$, $(\mathbf{y} - \mathbf{w})^\top \mathbf{g} > 0$. Fix some $\mathbf{z} \in \mathcal{K}_{\delta,\delta'/r}$, and note that using Observation 2, it holds that $\mathbf{z} + \delta(r-\delta')\hat{\mathbf{g}} \in \mathcal{K}_{\delta'/r}$, where $\hat{\mathbf{g}} = \frac{\mathbf{g}}{\|\mathbf{g}\|}$. Then, we have that,

$$0 < \left(\mathbf{y} - (\mathbf{z} + \delta(r-\delta')\hat{\mathbf{g}})\right)^\top \mathbf{g} = (\mathbf{y} - \mathbf{z})^\top \mathbf{g} - \delta(r-\delta')\|\mathbf{g}\|.$$

Rearranging, we obtain the lemma.

■

**Proof** [Proof of Lemma 13] Denote by $k$ the number of iterations until Algorithm 6 stops. Then, for every iteration $i < k$, it holds that $\frac{\mathbf{y}_i}{1-\delta'/r} \notin \mathcal{K}$, which implies that $\mathbf{y}_i \notin \mathcal{K}_{\delta'/r} = (1-\delta'/r)\mathcal{K}$. Thus, using Lemma 12, we have that for every $i < k$, it holds for all $\mathbf{z} \in \mathcal{K}_{\delta,\delta'/r}$ that $(\mathbf{y}_i - \mathbf{z})^\top \mathbf{g}_i \geq \delta(r-\delta')\|\mathbf{g}_i\|$. From these observations and using Lemma 5 with $\mathbf{g} = \mathbf{g}_i$, $C = \|\mathbf{g}_i\|$, and $Q = \delta(r-\delta')\|\mathbf{g}_i\|$, we have that for every $i < k$,

$$\forall \mathbf{z} \in \mathcal{K}_{\delta,\delta'/r} : \quad \|\mathbf{y}_{i+1} - \mathbf{z}\|^2 \leq \|\mathbf{y}_i - \mathbf{z}\|^2 - \delta^2(r-\delta')^2, \tag{18}$$

Specifically for $i = k-1$, and unrolling the recursion, we obtain that for all $\mathbf{z} \in \mathcal{K}_{\delta,\delta'/r}$, $\|\mathbf{y} - \mathbf{z}\|^2 \leq \|\mathbf{y}_1 - \mathbf{z}\|^2$, and since $\mathbf{y}_1$ is the projection of $\mathbf{y}_0$ onto $R\mathcal{B}$ and $\mathcal{K}_{\delta,\delta'/r} \subseteq R\mathcal{B}$, it holds that for all $\mathbf{z} \in \mathcal{K}_{\delta,\delta'/r}$, $\|\mathbf{y}_1 - \mathbf{z}\|^2 \leq \|\mathbf{y}_0 - \mathbf{z}\|^2$, and we can conclude that indeed for all $\mathbf{z} \in \mathcal{K}_{\delta,\delta'/r}$, $\|\mathbf{y} - \mathbf{z}\|^2 \leq \|\mathbf{y}_0 - \mathbf{z}\|^2$, as needed.

Now, we upper-bound $k$ — the number of iterations until Algorithm 6 stops. Denote $\mathbf{x}_i^* = \operatorname{argmin}_{\mathbf{x} \in \mathcal{K}_{\delta,\delta'/r}} \|\mathbf{y}_i - \mathbf{x}\|^2$. Using Eq. (18) for every iteration $i < k$ it holds that,

$$\begin{aligned}\operatorname{dist}^2(\mathbf{y}_{i+1}, \mathcal{K}_{\delta,\delta'/r}) &= \|\mathbf{y}_{i+1} - \mathbf{x}_{i+1}^*\|^2 \leq \|\mathbf{y}_{i+1} - \mathbf{x}_i^*\|^2 \\ &\leq \|\mathbf{y}_i - \mathbf{x}_i^*\|^2 - \delta^2(r-\delta')^2 = \operatorname{dist}^2(\mathbf{y}_i, \mathcal{K}_{\delta,\delta'/r}) - \delta^2(r-\delta')^2.\end{aligned}$$

Unrolling the recursion, and Since $\mathbf{y}_1$ is the projection of $\mathbf{y}_0$ onto $R\mathcal{B}$ and $\mathcal{K}_{\delta,\delta'/r} \subseteq R\mathcal{B}$, we have

$$\begin{aligned}\operatorname{dist}^2(\mathbf{y}, \mathcal{K}_{\delta,\delta'/r}) &\leq \operatorname{dist}^2(\mathbf{y}_1, \mathcal{K}_{\delta,\delta'/r}) - (k-1)\delta^2(r-\delta')^2 \\ &\leq \operatorname{dist}^2(\mathbf{y}_0, \mathcal{K}_{\delta,\delta'/r}) - (k-1)\delta^2(r-\delta')^2.\end{aligned}$$

Thus, after at most

$$k = \frac{\operatorname{dist}^2(\mathbf{y}_0, \mathcal{K}_{\delta,\delta'/r}) - \operatorname{dist}^2(\mathbf{y}, \mathcal{K}_{\delta,\delta'/r})}{\delta^2(r-\delta')^2} + 1$$

iterations Algorithm 6 must stop. ■

## Appendix H. Proof of Theorem 14

Before proving the theorem we need an additional observation.

**Observation 3** *Fix $\delta \in (0,1)$. For any $\mathbf{y} \in \mathcal{K}$ it holds that dist$(\mathbf{y}, \mathcal{K}_\delta) \leq R\delta$.*

**Proof** [Proof of Theorem 14] First, we note that since for every $t \in [2, T]$, $\tilde{\mathbf{y}}_t$ is output of Algorithm 6, using Lemma 13 with $\delta' = 0$, it follows that $\tilde{\mathbf{y}}_t \in \mathcal{K}$, and thus, Algorithm 7 indeed plays feasible points. Now, we prove the upper-bound on the adaptive regret. Fix an interval $I = [s, e], 1 \leq s \leq e \leq T$, and a feasible minimizer w.r.t. this interval, $\mathbf{x}_I^* \in \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} \sum_{t=s}^{e} f_t(\mathbf{x})$. Define $\tilde{\mathbf{x}}_I = (1-\delta)\mathbf{x}_I^* \in \mathcal{K}_\delta$. Since for every $t \in [T]$, $f_t(\cdot)$ is $G_f-$Lipschitz over $\mathcal{K}$, we have that

$$\begin{aligned}\sum_{t=s}^{e} f_t\left(\tilde{\mathbf{y}}_t\right) - f_t(\mathbf{x}_I^*) &= \sum_{t=s}^{e} f_t\left(\tilde{\mathbf{y}}_t\right) - f_t\left(\tilde{\mathbf{x}}_I\right) + f_t\left(\tilde{\mathbf{x}}_I\right) - f_t(\mathbf{x}_I^*) \\ &\leq G_f R\delta T + \sum_{t=s}^{e} f_t\left(\tilde{\mathbf{y}}_t\right) - f_t\left(\tilde{\mathbf{x}}_I\right).\end{aligned}$$

Using Lemma 13 with $\delta' = 0$ for all $t \geq 1$ we have that, $\tilde{\mathbf{y}}_t \in \mathcal{K}$ is an infeasible projection of $\mathbf{y}_t$ over $\mathcal{K}_\delta$. Thus, from Lemma 4, we have that

$$\sum_{t=s}^{e} f_t(\tilde{\mathbf{y}}_t) - \sum_{t=s}^{e} f_t(\tilde{\mathbf{x}}_I) \leq \frac{\|\tilde{\mathbf{y}}_s - \mathbf{x}\|^2}{2\eta} + \frac{\eta}{2}\sum_{s=1}^{e}\|\nabla_t\|^2 \leq \frac{2R^2}{\eta} + \frac{\eta G_f^2}{2}T.$$

Combining the last two equations, we obtain that

$$\sum_{t=s}^{e} f_t\left(\tilde{\mathbf{y}}_t\right) - f_t(\mathbf{x}_I^*) \leq \left(G_f R\delta + \frac{G_f^2\eta}{2}\right)T + \frac{2R^2}{\eta}.$$

The regret bound in the theorem now follows from plugging-in the values of $\delta, \eta$ listed in the theorem.

We turn to upper-bound the number of calls to the SO. For every $t \geq 1$, denote $\tilde{\mathbf{x}}_t^* = \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}_\delta}\|\mathbf{x} - \tilde{\mathbf{y}}_t\|$. Since $\mathbf{y}_{t+1} = \tilde{\mathbf{y}}_t - \eta\nabla_t$, we have

$$\operatorname{dist}(\mathbf{y}_{t+1}, \mathcal{K}_\delta) \leq \|\tilde{\mathbf{x}}_t^* - \mathbf{y}_{t+1}\| \leq \|\tilde{\mathbf{x}}_t^* - \tilde{\mathbf{y}}_t\| + \|\tilde{\mathbf{y}}_t - \mathbf{y}_{t+1}\| \leq \operatorname{dist}(\tilde{\mathbf{y}}_t, \mathcal{K}_\delta) + \|\eta\nabla_t\|.$$

It follows that, for any iteration $t \geq 1$, Algorithm 7 calls Algorithm 6 with $\mathbf{y}_{t+1}$ such that

$$\operatorname{dist}^2(\mathbf{y}_{t+1}, \mathcal{K}_\delta) \leq \operatorname{dist}^2(\tilde{\mathbf{y}}_t, \mathcal{K}_\delta) + 2\operatorname{dist}(\tilde{\mathbf{y}}_t, \mathcal{K}_\delta)\eta G_f + \eta^2 G_f^2. \tag{19}$$

Using Lemma 13 with initial point $\mathbf{y}_{t+1}$, feasible set $\mathcal{K}$, radius $r$, squeeze parameters $(\delta, \delta' = 0)$, and the returned point $\tilde{\mathbf{y}}_{t+1}$, we have that for every iteration $t \geq 1$, Algorithm 6 makes at most

$$\frac{\operatorname{dist}^2\left(\mathbf{y}_{t+1}, \mathcal{K}_\delta\right) - \operatorname{dist}^2\left(\tilde{\mathbf{y}}_{t+1}, \mathcal{K}_\delta\right)}{\delta^2 r^2} + 1$$

iterations. Thus, using Eq.(19) and Observation 3, the overall number of calls to the SO of $\mathcal{K}$ that Algorithm 6 makes is

$$\begin{aligned}N_{calls} &\leq \sum_{t=1}^{T}\frac{1}{\delta^2 r^2}\left(\operatorname{dist}^2(\tilde{\mathbf{y}}_t, \mathcal{K}_\delta) + 2R\delta\eta G_f + \eta^2 G_f^2 - \operatorname{dist}^2(\tilde{\mathbf{y}}_{t+1}, \mathcal{K}_\delta)\right) + 1 \\ &\leq \frac{2RG_f}{r^2}\frac{\eta}{\delta}T + \frac{G_f^2}{r^2}\frac{\eta^2}{\delta^2}T + T,\end{aligned}$$

where the last inequality is since $\operatorname{dist}^2(\tilde{\mathbf{y}}_1, \mathcal{K}_\delta) = 0$. ■

## Appendix I. SO-based Algorithm for the Bandit Setting and Proof of Theorem 15

Our SO-based algorithm for the bandit setting is given in Algorithm 10.

**Algorithm 10:** Bandit online gradient descent via a separation oracle (SO-BGD)

**Data:** horizon $T$, feasible set $\mathcal{K}$ with parameters $r, R$, update step $\eta$, squeeze parameters $(\delta, \delta')$.
$\tilde{\mathbf{y}}_1 \leftarrow \mathbf{0} \in \mathcal{K}_{\delta,\delta'}$
**for** $t = 1, \ldots, T$ **do**
  Set $\mathbf{u}_t \sim S^n$, play $\mathbf{z}_t = \tilde{\mathbf{y}}_t + \delta' \mathbf{u}_t$, and observe $f_t(\mathbf{z}_t)$.
  Set $\mathbf{g}_t = \frac{n}{\delta'} f_t(\mathbf{z}_t)\mathbf{u}_t$ and update $\mathbf{y}_{t+1} = \tilde{\mathbf{y}}_t - \eta \mathbf{g}_t$.
  Set $\tilde{\mathbf{y}}_{t+1} \leftarrow$ output of Algorithm 6 with set $\mathcal{K}$, radius $r$, initial vector $\mathbf{y}_{t+1}$, and squeeze parameters $(\delta, \delta')$.
**end**

**Proof** [Proof of Theorem 15] First, we establish that Algorithm 10 indeed plays feasible points, meaning $\mathbf{z}_t \in \mathcal{K}$ for all $t \in [T]$. Since for every $t \geq 1$, Algorithm 6 returns $\tilde{\mathbf{y}}_t \in \mathcal{K}_{\delta'/r} = (1 - \delta'/r)\mathcal{K}$ and $r\mathcal{B} \subseteq \mathcal{K}$, it follows that indeed $\mathbf{z}_t = \tilde{\mathbf{y}}_t + \delta' \mathbf{u}_t \in \mathcal{K}$ for every $t \in [T]$.

Now, we turn to prove the upper-bound on the adaptive expected regret. Let us fix some interval $I = [s, e], 1 \leq s \leq e \leq T$. We start with an upper bound on $\mathbb{E}\left[\sum_{t=s}^{e} \widehat{f}_{t,\delta'}(\tilde{\mathbf{y}}_t) - \widehat{f}_{t,\delta'}(\mathbf{x})\right]$ for every $\mathbf{x} \in \mathcal{K}_{\delta,\delta'/r} = (1 - \delta'/r)(1 - \delta)\mathcal{K}$. We will first take a few preliminary steps. For every $t \in [T]$, denote by $\mathcal{F}_t = \{\tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_{t-1}, \mathbf{g}_1, \ldots, \mathbf{g}_{t-1}\}$ the history of all predictions and gradient estimates up to time $t$. Since for all $t \geq 1$, $\mathbf{g}_t$ is an unbiased estimator of $\nabla \widehat{f}_{t,\delta'}(\tilde{\mathbf{y}}_t)$, i.e., $\mathbb{E}[\mathbf{g}_t | \mathcal{F}_t] = \nabla \widehat{f}_{t,\delta'}(\tilde{\mathbf{y}}_t)$, we have that for all $t \in [T]$ and $\mathbf{x} \in \mathcal{K}_{\delta,\delta'/r}$, it holds that

$$\mathbb{E}\left[\mathbf{g}_t^\top (\tilde{\mathbf{y}}_t - \mathbf{x})\right] = \mathbb{E}\left[\mathbb{E}[\mathbf{g}_t | \mathcal{F}_t]^\top (\tilde{\mathbf{y}}_t - \mathbf{x})\right] = \mathbb{E}\left[\nabla \widehat{f}_{t,\delta'}(\tilde{\mathbf{y}}_t)^\top (\tilde{\mathbf{y}}_t - \mathbf{x})\right]. \tag{20}$$

From Lemma 13 with $\delta' \neq 0$, we have that for every $t \in [T]$, the point $\tilde{\mathbf{y}}_t \in \mathcal{K}_{\delta'/r}$, and is an infeasible projection of $\mathbf{y}_t$ over $\mathcal{K}_{\delta,\delta'/r}$. Thus, we have that

$$\forall t \in [T]\ \forall \mathbf{x} \in \mathcal{K}_{\delta,\delta'/r}: \ \ \|\tilde{\mathbf{y}}_{t+1} - \mathbf{x}\|^2 \leq \|\mathbf{y}_{t+1} - \mathbf{x}\|^2.$$

Since $\mathbf{y}_{t+1} = \tilde{\mathbf{y}}_t - \eta \mathbf{g}_t$, for every $t \in [T]$ and $\mathbf{x} \in \mathcal{K}_{\delta,\delta'/r}$ we have that

$$\|\tilde{\mathbf{y}}_{t+1} - \mathbf{x}\|^2 \leq \|\tilde{\mathbf{y}}_t - \eta \mathbf{g}_t - \mathbf{x}\|^2 = \|\tilde{\mathbf{y}}_t - \mathbf{x}\|^2 + \eta^2 \|\mathbf{g}_t\|^2 - 2\eta \mathbf{g}_t^\top (\tilde{\mathbf{y}}_t - \mathbf{x}).$$

Rearranging, we obtain that for every $t \in [T]$ and $\mathbf{x} \in \mathcal{K}_{\delta,\delta'/r}$, it holds that

$$\mathbf{g}_t^\top (\tilde{\mathbf{y}}_t - \mathbf{x}) \leq \frac{\|\tilde{\mathbf{y}}_t - \mathbf{x}\|^2}{2\eta} - \frac{\|\tilde{\mathbf{y}}_{t+1} - \mathbf{x}\|^2}{2\eta} + \frac{\eta}{2} \|\mathbf{g}_t\|^2.$$

Summing over the interval $[s, e]$ and taking expectation, we have that

$$\mathbb{E}\left[\sum_{t=s}^{e} \mathbf{g}_t^\top (\tilde{\mathbf{y}}_t - \mathbf{x})\right] \leq \mathbb{E}\left[\sum_{t=s}^{e} \frac{\|\tilde{\mathbf{y}}_t - \mathbf{x}\|^2}{2\eta} - \frac{\|\tilde{\mathbf{y}}_{t+1} - \mathbf{x}\|^2}{2\eta}\right] + \frac{\eta}{2} \sum_{t=s}^{e} \mathbb{E}\left[\|\mathbf{g}_t\|^2\right].$$

Since $\tilde{\mathbf{y}}_t \in \mathcal{K}_{\delta'/r}$ for every $t \in [T]$, then $\|\tilde{\mathbf{y}}_t - \mathbf{x}\| \leq 2R$ for every $\mathbf{x} \in \mathcal{K}_{\delta,\delta'/r}$, and thus,

$$\mathbb{E}\left[\sum_{t=s}^{e} \mathbf{g}_t^\top (\tilde{\mathbf{y}}_t - \mathbf{x})\right] \leq \frac{R}{\eta} + \frac{\eta}{2}\sum_{t=s}^{e}\mathbb{E}\left[\|\mathbf{g}_t\|^2\right].$$

Using Eq. (20), for every $\mathbf{x} \in \mathcal{K}_{\delta,\delta'/r}$ we have that,

$$\sum_{t=s}^{e}\mathbb{E}\left[\nabla \widehat{f}_{t,\delta'}(\tilde{\mathbf{y}}_t)^\top (\tilde{\mathbf{y}}_t - \mathbf{x})\right] = \sum_{t=s}^{e}\mathbb{E}\left[\mathbf{g}_t^\top (\tilde{\mathbf{y}}_t - \mathbf{x})\right] \leq \frac{R}{\eta} + \frac{\eta}{2}\sum_{t=s}^{e}\mathbb{E}\left[\|\mathbf{g}_t\|^2\right].$$

Since for every $t \in [T]$, $f_t(\cdot)$ is convex in $\mathcal{K}$, using Lemma 16, it holds that $\widehat{f}_{t,\delta'}(\cdot)$ is convex in $\mathcal{K}_{\delta'/r}$. Thus, for every $\mathbf{x} \in \mathcal{K}_{\delta,\delta'/r}$ we obtain that,

$$\mathbb{E}\left[\sum_{t=s}^{e} \widehat{f}_{t,\delta'}(\tilde{\mathbf{y}}_t) - \widehat{f}_{t,\delta'}(\mathbf{x})\right] \leq \frac{R}{\eta} + \frac{\eta}{2}\sum_{t=s}^{e}\mathbb{E}\left[\|\mathbf{g}_t\|^2\right]. \tag{21}$$

Let us denote by $\mathbf{x}_I^*$ a feasible minimizer w.r.t. to interval $I = [s,e]$, i.e., $\mathbf{x}_I^* \in \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} \sum_{t=s}^{e} f_t(\mathbf{x})$, and define accordingly $\tilde{\mathbf{x}}_I^* = (1-\delta'/r)(1-\delta)\mathbf{x}_I^* \in \mathcal{K}_{\delta,\delta'/r}$. It holds that,

$$\mathbb{E}\left[\sum_{t=s}^{e} f_t(\mathbf{z}_t)\right] - \sum_{t=s}^{e} f_t(\mathbf{x}_I) = \mathbb{E}\left[\sum_{t=s}^{e} f_t(\mathbf{z}_t) - f_t(\tilde{\mathbf{y}}_t) + \sum_{t=s}^{e} f_t(\tilde{\mathbf{y}}_t) - f_t(\tilde{\mathbf{x}}_I^*)\right] + \sum_{t=s}^{e} f_t(\tilde{\mathbf{x}}_I^*) - \sum_{t=s}^{e} f_t(\mathbf{x}_I^*). \tag{22}$$

Since for every $t \in [T]$ $f_t(\cdot)$ is $G_f$-Lipschitz, we have that

$$\mathbb{E}\left[\sum_{t=s}^{e} f_t(\mathbf{z}_t) - f_t(\tilde{\mathbf{y}}_t)\right] = \sum_{t=s}^{e}\mathbb{E}\left[f_t(\tilde{\mathbf{y}}_t + \delta'\mathbf{u}_t) - f_t(\tilde{\mathbf{y}}_t)\right] \leq G_f\delta' T,$$

and

$$\sum_{t=s}^{e} f_t(\tilde{\mathbf{x}}_I^*) - f_t(\mathbf{x}_I^*) \leq \sum_{t=s}^{e} \nabla f_t(\tilde{\mathbf{x}}_I^*)^\top (\tilde{\mathbf{x}}_I^* - \mathbf{x}_I^*) \leq \sum_{t=s}^{e} G_f\|\tilde{\mathbf{x}}_I^* - \mathbf{x}_I^*\| \leq G_f R\left(\frac{\delta'}{r} + \delta + \frac{\delta\delta'}{r}\right)T.$$

Using Lemma 16 and Eq. (21), we have

$$\begin{aligned}\mathbb{E}\left[\sum_{t=s}^{e} f_t(\tilde{\mathbf{y}}_t) - f_t(\tilde{\mathbf{x}}_I^*)\right] =& \mathbb{E}\left[\sum_{t=s}^{e} f_t(\tilde{\mathbf{y}}_t) - \widehat{f}_{t,\delta'}(\tilde{\mathbf{y}}_t)\right] + \mathbb{E}\left[\sum_{t=s}^{e} \widehat{f}_{t,\delta'}(\tilde{\mathbf{x}}_I^*) - f_t(\tilde{\mathbf{x}}_I^*)\right] \\ &+ \mathbb{E}\left[\sum_{t=s}^{e} \widehat{f}_{t,\delta'}(\tilde{\mathbf{y}}_t) - \widehat{f}_{t,\delta'}(\tilde{\mathbf{x}}_I^*)\right] \leq 2\delta' G_f T + \frac{R}{\eta} + \frac{\eta}{2}\sum_{t=s}^{e}\mathbb{E}\left[\|\mathbf{g}_t\|^2\right].\end{aligned}$$

Combining the last three equations and Eq. (22), and using the fact that $\|\mathbf{g}_t\| \leq \frac{nM}{\delta'}$, we obtain that

$$\mathbb{E}\left[\sum_{t=s}^{e} f_t(\mathbf{z}_t) - f_t(\mathbf{x}_I^*)\right] \leq G_f\left(3\delta' + R\left(\frac{\delta'}{r} + \delta + \frac{\delta\delta'}{r}\right)\right)T + \frac{R}{\eta} + \frac{n^2M^2}{2}\frac{\eta}{\delta'^2}T.$$

Plugging-in the values of $\eta, \delta, \delta'$ listed in the theorem, we obtain the adaptive expected regret bound in the theorem.

We now move on to upper-bound the overall number of calls to the SO of $\mathcal{K}$. For every $t \in [T]$, let us denote $\mathbf{x}_t^* = \mathrm{argmin}_{\mathbf{x}\in\mathcal{K}_{\delta,\delta'/r}} \|\mathbf{x} - \tilde{\mathbf{y}}_t\|$. Since Algorithm 10 updates $\mathbf{y}_{t+1} = \tilde{\mathbf{y}}_t - \eta \mathbf{g}_t$, we have

$$\mathrm{dist}\left(\mathbf{y}_{t+1}, \mathcal{K}_{\delta,\delta'/r}\right) \leq \|\tilde{\mathbf{x}}_t^* - \mathbf{y}_{t+1}\| \leq \|\tilde{\mathbf{x}}_t^* - \tilde{\mathbf{y}}_t\| + \|\tilde{\mathbf{y}}_t - \mathbf{y}_{t+1}\| = \mathrm{dist}\left(\tilde{\mathbf{y}}_t, \mathcal{K}_{\delta,\delta'/r}\right) + \|\eta \mathbf{g}_t\|,$$

which, by plugging-in the upper-bound on $\|\mathbf{g}_t\|$, gives

$$\mathrm{dist}^2\left(\mathbf{y}_{t+1}, \mathcal{K}_{\delta,\delta'/r}\right) \leq \mathrm{dist}^2\left(\tilde{\mathbf{y}}_t, \mathcal{K}_{\delta,\delta'/r}\right) + 2\mathrm{dist}\left(\tilde{\mathbf{y}}_t, \mathcal{K}_{\delta,\delta'/r}\right)\eta\frac{nM}{\delta'} + \eta^2\frac{(nM)^2}{\delta'^2}. \tag{23}$$

For any $t \in [T]$, using Lemma 13 with initial point $\mathbf{y}_{t+1}$, feasible set $\mathcal{K}$, radius $r$, squeeze parameters $(\delta, \delta')$, and the returned point $\tilde{\mathbf{y}}_{t+1}$, we have that Algorithm 6 makes at most

$$\frac{\mathrm{dist}^2\left(\mathbf{y}_{t+1}, \mathcal{K}_{\delta,\delta'/r}\right) - \mathrm{dist}^2\left(\tilde{\mathbf{y}}_{t+1}, \mathcal{K}_{\delta,\delta'/r}\right)}{\delta^2\left(r-\delta'\right)^2} + 1$$

iterations. Since $\tilde{\mathbf{y}}_t \in \mathcal{K}_{\delta'/r} \subseteq \mathcal{K}$ and $(1-\delta)\mathcal{K}_{\delta'/r} = \mathcal{K}_{\delta,\delta'/r}$, using Observation 3 it holds that $\mathrm{dist}\left(\tilde{\mathbf{y}}_t, \mathcal{K}_{\delta,\delta'/r}\right) \leq R\delta$. Thus, using this observation and Eq.(23), the overall number of calls to the SO of $\mathcal{K}$ that Algorithm 6 makes is

$$\begin{aligned} N_{calls} &\leq T + \frac{1}{\delta^2\left(r-\delta'\right)^2}\sum_{t=1}^{T}\left(\mathrm{dist}^2\left(\tilde{\mathbf{y}}_t, \mathcal{K}_{\delta,\delta'/r}\right) + 2R\delta\eta\frac{nM}{\delta'} + \eta^2\frac{(nM)^2}{\delta'^2} - \mathrm{dist}^2\left(\tilde{\mathbf{y}}_{t+1}, \mathcal{K}_{\delta,\delta'/r}\right)\right) \\ &\leq \left(1 + \frac{8RnM}{r^2}\frac{\eta}{\delta\delta'} + \frac{4(nM)^2}{r^2}\frac{\eta^2}{\delta^2\delta'^2}\right)T, \end{aligned}$$

where last inequality follows since $\mathrm{dist}^2\left(\tilde{\mathbf{y}}_1, \mathcal{K}_{\delta,\delta'/r}\right) = 0$, and $\delta' \leq r/2$. ■

## Appendix J. Proofs of Additional Observations

**Proof** [Proof of Observation 1] For every $t \geq 1$ it holds that,

$$(t+1)(t+3)^2 - (t+2)^3 = t^2 + 3t + 1 > 0.$$

Since both sides are non-negative, we obtain

$$(t+1)^{\frac{1}{3}}(t+3)^{\frac{2}{3}} \geq \left((t+2)^3\right)^{\frac{1}{3}} = t+2.$$

Rearranging, we have that fort any $t \geq 1$,

$$\frac{(t+1)^{1/3}(t+3)^{2/3}}{t+2} \geq 1. \tag{24}$$

Thus, we have that

$$\frac{(t+2)^{\frac{4}{3}}}{(t+1)^{\frac{2}{3}}} \underset{(a)}{\leq} \frac{(t+2)^{\frac{1}{3}}(t+3)^{\frac{2}{3}}}{(t+1)^{\frac{1}{3}}} < \frac{t+3}{(t+1)^{\frac{1}{3}}} = (t+1)^{\frac{2}{3}} + \frac{2}{(t+1)^{\frac{1}{3}}},$$

where (a) follows from (24).

Thus, we can write

$$\frac{(t+2)^{\frac{4}{3}}}{(t+1)^{\frac{2}{3}}} \leq (t+1)^{\frac{2}{3}} + \frac{2}{(t+1)^{\frac{1}{3}}} \leq \frac{(t+1)^{\frac{4}{3}}}{t^{\frac{2}{3}}} + \frac{2}{t^{\frac{1}{3}}}.$$

The first item of the observation follows from rearranging the equation.

Now, we prove the second item. It holds for all $t \geq 1$ that,

$$\begin{aligned}\frac{(t+2)^{\frac{4}{3}}}{(t+1)^2} = \frac{(t+2)^2}{(t+1)^2}\frac{1}{(t+2)^{\frac{2}{3}}} &= \frac{(t+1)^2 + 2(t+1) + 1}{(t+1)^2}\frac{1}{(t+2)^{\frac{2}{3}}} \\ &= \left(1 + \frac{2}{t+1} + \frac{1}{(t+1)^2}\right)\frac{1}{(t+2)^{\frac{2}{3}}} \leq \frac{3}{(t+2)^{\frac{2}{3}}} \leq \frac{3}{t^{\frac{2}{3}}}.\end{aligned}$$

Thus, we indeed obtain

$$\frac{(t+2)^{\frac{4}{3}}}{(t+1)^2} - \frac{(t+1)^{\frac{4}{3}}}{t^2} \leq \frac{(t+2)^{\frac{4}{3}}}{(t+1)^2} \leq \frac{3}{t^{\frac{2}{3}}}.$$

■

**Proof** [Proof of Observation 2] First we prove that $(r-\delta')\mathcal{B} \subseteq \mathcal{K}_{\delta'/r} = (1-\delta'/r)\mathcal{K}$. Fix some $\mathbf{u} \in (r-\delta')\mathcal{B}$, i.e., $\|\mathbf{u}\| \leq r-\delta'$. Since $r\mathcal{B} \subseteq \mathcal{K}$, it holds that $\mathbf{u}\frac{1}{(1-\delta'/r)} = \mathbf{u}\frac{r}{r-\delta'} \in r\mathcal{B} \subseteq \mathcal{K}$. This in turn implies that $\mathbf{u} = (1-\delta'/r)\mathbf{u}\frac{1}{1-\delta'/r} \in (1-\delta'/r)\mathcal{K} = \mathcal{K}_{\delta'/r}$.

Now, we recall that if a convex set $\mathcal{P} \subset \mathbb{R}^n$ satisfies that $p\mathcal{B} \subseteq \mathcal{P}$, for some $p > 0$, then for any $\gamma \in [0,p]$ and any $\mathbf{z} \in (1-\gamma/p)\mathcal{P}$, it holds that $\mathbf{z} + \gamma\mathcal{B} \subseteq \mathcal{P}$ (see for instance the chapter on bandit algorithms in Hazan (2019). Applying this with $\mathcal{P} = \mathcal{K}_{\delta'/r}$, $p = (r-\delta')$, and $\gamma = \delta(r-\delta')$, we have that for any $\mathbf{z} \in (1-\delta(r-\delta')/(r-\delta'))\mathcal{K}_{\delta'/r} = (1-\delta)\mathcal{K}_{\delta'/r} = \mathcal{K}_{\delta,\delta'/r}$, it holds that $\mathbf{z} + \delta(r-\delta')\mathcal{B} \subseteq \mathcal{K}_{\delta'/r}$, as needed. ■

**Proof** [Proof of Observation 3] Denote $\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{K}_\delta}{\operatorname{argmin}} \|\mathbf{x}-\mathbf{y}\|^2$ and $\mathbf{y}_\delta = (1-\delta)\mathbf{y}$. Since $\mathbf{y} \in \mathcal{K}$ and $\mathbf{y}_\delta \in \mathcal{K}_\delta$ it holds that

$$\operatorname{dist}(\mathbf{y}, \mathcal{K}_\delta) = \|\mathbf{x}^* - \mathbf{y}\| \leq \|\mathbf{y}_\delta - \mathbf{y}\| = \|\delta\mathbf{y}\| \leq \delta R.$$

■