Optimistic Safety for Online Convex Optimization with Unknown Linear Constraints

Spencer Hutchinson
UC Santa Barbara

Tianyi Chen
Rensselaer Polytechnic Institute

Mahnoosh Alizadeh
UC Santa Barbara

Abstract

We study the problem of online convex optimization (OCO) under unknown linear constraints that are either static, or stochastically time-varying. For this problem, we introduce an algorithm that we term *Optimistically Safe OCO* (OSOCO) and show that it enjoys $\tilde{\mathcal{O}}(\sqrt{T})$ regret and *no constraint violation*. In the case of static linear constraints, this improves on the previous best known $\tilde{\mathcal{O}}(T^{2/3})$ regret under the same assumptions. In the case of stochastic time-varying constraints, our work supplements existing results that show $\mathcal{O}(\sqrt{T})$ regret and $\mathcal{O}(\sqrt{T})$ cumulative violation under more general convex constraints and a different set of assumptions. In addition to our theoretical guarantees, we also give numerical results that further validate the effectiveness of our approach.

1 INTRODUCTION

Online convex optimization (OCO), formalized by Zinkevich (2003), is a setting where, in each round $t \in [T]$, a player chooses a vector action x_t, suffers the cost of that action according to an adversarially-chosen cost function $f_t(x_t)$, and subsequently observes the cost function f_t. This setting has found broad applicability in various fields including online advertising (McMahan et al., 2013), network routing (Awerbuch and Kleinberg, 2008), portfolio management (Cover, 1991) and resource allocation (Yu and Neely, 2019). However, despite its significance, this fundamental setting assumes that any constraints on the player's actions are known a priori, which is not the case in many real-world applications. As a result, the problem of OCO under unknown constraints has received considerable attention in the last decade, e.g. (Mannor et al., 2009; Yu et al., 2017; Cao and Liu, 2018; Guo et al., 2022).

Within the broader literature on OCO with unknown constraints, this paper focuses on two important research directions: (1) *OCO with stochastic constraints*, where there are different constraints in each round $g_t(x_t) \leq 0 \ \forall t$ that are independent and identically-distributed (i.i.d.) and the player receives feedback on g_t after choosing each action x_t, and (2) *OCO with static constraints*, where there is a static and deterministic constraint $g(x_t) \leq 0 \ \forall t$ for which the player receives noisy feedback $g(x_t) + \epsilon_t$ after choosing each action x_t. In OCO with stochastic constraints, existing works typically leverage primal-dual algorithms to ensure both sublinear regret and sublinear cumulative violation $\sum_{t=1}^{T} g_t(x_t)$. These algorithms operate by jointly iterating over both the actions (primal variables) and constraint penalties (dual variables), and have been shown to ensure $\mathcal{O}(\sqrt{T})^{[1]}$ expected regret and $\mathcal{O}(\sqrt{T})$ expected cumulative violation (Yu et al., 2017; Wei et al., 2020; Castiglioni et al., 2022; Yu et al., 2023). On the other hand, the works on OCO with static constraints ensure *no violation* $g(x_t) \leq 0 \ \forall t$ with high probability, by explicitly estimating the constraints and then only choosing actions that are verifiably-safe (Chaudhary and Kalathil, 2022; Chang et al., 2023). However, the approach in Chaudhary and Kalathil (2022) and Chang et al. (2023) requires a pure-exploration phase of $\Theta(T^{2/3})$ rounds and therefore incurs a less favorable regret of $\tilde{\mathcal{O}}(T^{2/3})$.

In this work, we introduce an algorithm for both static and stochastic constraints that *directly estimates the constraints,* and therefore ensures no violation, while avoiding the costly exploration phase used in prior works (Chaudhary and Kalathil, 2022; Chang et al., 2023). Precisely, our algorithm uses *optimistic* estimates of the constraints (outer-approximations of the feasible set) to identify actions that are low-regret, but

---

[1]For ease of presentation in the introduction, we use $\mathcal{O}(\cdot)$ to hide non-T quantities, and $\tilde{\mathcal{O}}(\cdot)$ to further hide *log* factors.

potentially unsafe, and then *pessimistic* estimates of the constraints (inner-approximations of the feasible set) to find the scalings on these actions that will ensure constraint satisfaction. This approach, which we call *optimistic safety*, allows for the constraint to be learned *simultaneous* to the regret minimization and therefore does not require a pure-exploration phase as in prior works. Ultimately, our approach yields the following results:

- In OCO with static linear constraints and noisy constraint feedback, we show that our algorithm enjoys $\tilde{\mathcal{O}}(\sqrt{T})$ regret and no violation, in either high probability (Theorem 1) or in expectation (Proposition 1). Our high probability guarantees improve on the $\tilde{\mathcal{O}}(T^{2/3})$ regret attained by prior work on this problem, under the same assumptions and violation guarantees (Chaudhary and Kalathil, 2022; Chang et al., 2023).

- We then extend our results to OCO with stochastic linear constraints g_t, where the player receives bandit feedback on the constraint $g_t(x_t)$ (Section 6). We show $\tilde{\mathcal{O}}(\sqrt{T})$ expected regret and no violation in expectation $\mathbb{E}[g_t(x_t)] \leq 0 \ \forall t$ in this setting (Corollary 1). This supplements existing results on OCO with stochastic convex constraints that show $\mathcal{O}(\sqrt{T})$ expected regret and $\mathcal{O}(\sqrt{T})$ expected cumulative violation $\mathbb{E}[\sum_t g_t(x_t)]$, although with a different set of assumptions (see Section 6.1) (Yu et al., 2017; Wei et al., 2020; Castiglioni et al., 2022; Yu et al., 2023).

1.1 Overview

In Section 2, we give the problem setup and assumptions. Then, we design a meta-algorithm in Section 3 that leverages the idea of optimistic safety. In Section 4, we show that this meta-algorithm ensures $\tilde{\mathcal{O}}(d\sqrt{T})$ regret and no constraint violation in either high-probability or expectation. In Section 5, we improve the computational efficiency of our original algorithm by giving a version that only requires $2d$ convex projections in each round, although with a slightly larger dependence on the dimension in the regret bound, i.e. $\tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$. In Section 6, we extend our approach to the setting of OCO with stochastic linear constraints and bandit constraint feedback and show matching regret bounds in expectation and no expected violation. Lastly, we give simulation results in Section 7 that showcase the empirical performance of our algorithm.

1.2 Related work

In the following, we discuss prior work on OCO with unknown static constraints, OCO with time-varying

constraints, and safe learning. Lastly, we discuss a concurrent work. In Appendix A, we provide additional discussion on OCO with long-term constraints, projection-free OCO, and online control.

OCO with Unknown Static Constraints The problem of OCO with unknown linear constraints and noisy constraint feedback was first studied by Chaudhary and Kalathil (2022) who gave an algorithm with $\tilde{\mathcal{O}}(dT^{2/3})$ regret and no violation $g(x_t) \leq 0 \ \forall t$, with high probability. In the same setting, we give an algorithm that enjoys $\tilde{\mathcal{O}}(d\sqrt{T})$ regret and no violations with high probability, and a more efficient variant that enjoys $\tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$ regret. With a different choice of algorithm parameters, we also show matching bounds on the expected regret and show no expected violation $\mathbb{E}[g_t(x_t)] \leq 0$. The approach taken by Chaudhary and Kalathil (2022) was then extended to the distributed and nonconvex setting in Chang et al. (2023), where an $\tilde{\mathcal{O}}(dT^{2/3} + T^{1/3}C_T)$ high-probability regret bound was given with C_T as the path length of the best minimizer sequence. Another variant of OCO with unknown static constraints was considered by Hutchinson and Alizadeh (2024), although they assume that the constraint function is strongly-convex, and thus their setting is distinct from the one we consider here.

OCO with Time-varying Constraints For OCO with time-varying constraints that are chosen adversarially, Mannor et al. (2009) showed that no algorithm can simultaneously achieve both sublinear cumulative violation and sublinear regret against the best action that satisfies the constraints on average. Following this negative result, considerable effort has been focused on ways to weaken the setting to guarantee sublinear regret and cumulative violation, e.g. (Neely and Yu, 2017; Sun et al., 2017; Chen et al., 2017; Liakopoulos et al., 2019; Cao and Liu, 2018; Yi et al., 2022; Guo et al., 2022; Castiglioni et al., 2022; Kolev et al., 2023). Most relevant to this paper, several works study the setting in which the constraints are convex functions that are stochastic and i.i.d., and give algorithms with $\mathcal{O}(\sqrt{T})$ expected regret and $\mathcal{O}(\sqrt{T})$ expected cumulative violation when the player receives full feedback (Yu et al., 2017; Wei et al., 2020; Castiglioni et al., 2022) and $\mathcal{O}(d\sqrt{T})$ expected regret and violation when the player receives two-point bandit feedback (Yu et al., 2023). In the present work, we supplement this literature by consider a setting (in Section 6) where the constraints are i.i.d. linear functions and the player receives one-point bandit feedback on the constraint (but full feedback on the cost). In this setting, we guarantee $\tilde{\mathcal{O}}(d\sqrt{T})$ expected regret (or $\tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$ for our more efficient version) and *no expected violation*. We additionally assume that the player knows an action

that is strictly feasible in expectation and its expected constraint value, as discussed in Section 6.1.

Safe Learning Our setting falls under the classification of "safe learning," which often deals with learning problems that have uncertain constraints. Some examples of such problems are constrained MDPs e.g. (Achiam et al., 2017; Wachi and Sui, 2020; Liu et al., 2021; Amani et al., 2021; Ghosh et al., 2024), safe Gaussian process bandits e.g. (Sui et al., 2015, 2018; Losalka and Scarlett, 2024), safe optimization with uncertain constraints e.g. (Usmanova et al., 2019; Fereydounian et al., 2020) and safe stochastic linear bandits e.g. (Amani et al., 2019; Moradipari et al., 2021; Pacchiano et al., 2021; Hutchinson et al., 2024). The most relevant to the present work is the problem of safe stochastic linear bandits (Moradipari et al., 2021; Pacchiano et al., 2021; Hutchinson et al., 2024) as it has the same constraint formulation as OCO with static linear constraints and both concern regret minimization. However, the safe stochastic linear bandit setting has stochastic linear costs with bandit feedback, while OCO with static linear constraints has adversarial convex costs with full cost feedback. Note that our approach draws inspiration from the safe stochastic linear bandit work Hutchinson et al. (2024), although we emphasize that OCO with static linear constraints has fundamental challenges that are not present in the safe stochastic linear bandit problem. Most notably, OCO with static linear constraints has the challenges posed by time-varying and nonconvex action sets in OCO, for which we use a phased-based approach and a novel OCO algorithm for nonconvex action sets.

Concurrent Work A concurrent work (Sridharan and Yoo, 2024) appeared simultaneous to the arXiv version of the present work. For OCO with static linear constraints and noisy constraint feedback, they give an algorithm with $\tilde{\mathcal{O}}(d\sqrt{T})$ regret and no violation in high probability for the special case of linear cost functions, i.e. when $f_t(x) = \theta_t^\top x$.

1.3 Notation

We use $\mathcal{O}(\cdot)$ to refer to big-O notation and $\tilde{\mathcal{O}}(\cdot)$ to refer to the same ignoring log factors. Also, we denote the 2-norm by $\|\cdot\|$. For a natural number n, we define $[n] := \{1, 2, ..., n\}$. For a matrix M, the transpose of M is denoted by $M^\top$. We also use $\mathbf{1}$ and $\mathbf{0}$ to refer to a vector of ones and zeros, respectively. Given $p \in [1, \infty]$, we denote the p-norm ball as $\mathbb{B}_p$ (where $\mathbb{B}$ is the 2-norm ball). For vectors $x, y \in \mathbb{R}^d$, we use $x \leq y$ to refer to the component-wise partial ordering, i.e. $x \leq y \iff x_i \leq y_i \ \forall i \in [d]$. Lastly, we use $\mathbb{E}_t$ to refer to the expectation conditioned on the randomness up to and including round $t - 1$.

2 PROBLEM SETUP

The problem is played over T rounds, where in each round $t \in [T]$, a player chooses an action x_t from the convex action set $\mathcal{X} \subseteq \mathbb{R}^d$ and, simultaneously, an adversary chooses the convex cost function $f_t : \mathcal{X} \to \mathbb{R}$ according to the actions chosen by the player in previous rounds. The player then suffers the cost $f_t(x_t)$, observes the cost function f_t and receives noisy constraint feedback $y_t := Ax_t + \epsilon_t$. The matrix $A \in \mathbb{R}^{n \times d}$ is the unknown and deterministic constraint matrix, and the vector $\epsilon_t \in \mathbb{R}^n$ is a zero-mean random noise. Additionally, there is a deterministic constraint limit $b \in \mathbb{R}^n$ that is known to the player.[2]

We aim either to ensure that there is *no constraint violation in high-probability* or *no constraint violation in expectation*. These two types of guarantees are summarized as follows:

- **High-probability:** $Ax_t \leq b$ for all $t \in [T]$, with high probability.

- **Expectation:** $\mathbb{E}[Ax_t] \leq b$ for all $t \in [T]$.

Note that these guarantees are with respect to the noise $\epsilon_1, ..., \epsilon_T$ and any randomization used by the algorithm.

In either case, we additionally aim to minimize the regret with respect to the best constraint-satisfying action in hindsight, i.e.

$$R_T := \sum_{t=1}^T f_t(x_t) - \sum_{t=1}^T f_t(x^\star),$$

where $x^\star \in \arg\min_{x \in \mathcal{Y}} \sum_{t=1}^T f_t(x)$ and $\mathcal{Y} = \{x \in \mathcal{X} : Ax \leq b\}$.

Our approach to this setting relies on the following assumptions. We first assume that the costs have bounded gradients and the action set is compact, which are standard in OCO (e.g. Zinkevich (2003); Hazan et al. (2016)).

Assumption 1 (Bounded gradients). *For all $t \in [T]$, f_t is differentiable and $\|\nabla f_t(x)\| \leq G$ for all $x \in \mathcal{X}$.*

Assumption 2 (Compact action set). *The action set $\mathcal{X}$ is closed and $\|x\| \leq D/2$ for all $x \in \mathcal{X}$.*

We further assume that elements of b are positive, the noise is conditionally sub-gaussian, and that the constraint parameter A is bounded. These match the assumptions used in Chaudhary and Kalathil (2022) and Chang et al. (2023).

---

[2]The assumption that b is known is also used in prior work on this problem (Chaudhary and Kalathil, 2022; Chang et al., 2023).

Assumption 3 (Initial safe action). *It holds that* $\mathbf{0} \in \mathcal{X}$ *and* $b_{\min} := \min_{i \in [n]} b_i > 0$ *where* b_i *is the* i*th element of* b.[3]

Assumption 4 (Conditionally-subgaussian noise). *The noise terms* $(\epsilon_t)_{t \in [T]}$ *are element-wise conditionally* ρ*-subgaussian, i.e. it holds for all* $t \in [T]$ *and* $i \in [n]$ *that* $\mathbb{E}[\epsilon_{t,i}|x_1, \epsilon_1, ..., \epsilon_{t-1}, x_t] = 0$ *and* $\mathbb{E}[\exp(\lambda \epsilon_{t,i})|x_1, \epsilon_1, ..., \epsilon_{t-1}, x_t] \leq \exp(\frac{\lambda^2 \rho^2}{2}), \forall \lambda \in \mathbb{R}$ *where* $\epsilon_{t,i}$ *is the* i*th element of* ϵ_t.

Assumption 5 (Bounded constraint). *Each row of* A *is bounded, i.e.* $\|a_i\| \leq S$ *for all* $i \in [n]$ *where* $a_i^\top$ *is the* i*th row of* A.

3 ALGORITHM

To address the stated problem, we propose the algorithm *Optimistically Safe OCO* (OSOCO) given in Algorithm 1. This algorithm operates over phases and invokes a generic online optimization algorithm $\mathcal{A}$ in each phase. We give the details on OSOCO in Section 3.1. Then, in Section 3.2, we give the requirements on the online optimization algorithm $\mathcal{A}$.

3.1 Details on OSOCO

The OSOCO algorithm operates over a number of phases. We describe the initialization of each phase, the update in each round, and then the phase termination criteria in the following.

Phase Initialization (lines 3-7) At the beginning of each phase j, the regularized least-squares estimator of the constraint $\hat{A}_j$ is calculated (line 4), which is then used to construct the pessimistic action set $\mathcal{Y}_j^p$ (line 5) and the optimistic action set $\mathcal{Y}_j^o$ (line 6). Then, the online optimization algorithm $\mathcal{A}$ is initialized with the optimistic action set (line 7). To appropriately construct the pessimistic and optimistic action sets, we use bounds on the constraint function of the form,

$$\hat{A}_j x - \bar{\beta}_j \|x\|_{\bar{V}_j^{-1}} \mathbf{1} \leq Ax \leq \hat{A}_j x + \bar{\beta}_j \|x\|_{\bar{V}_j^{-1}} \mathbf{1} \quad (1)$$

which were shown in Abbasi-Yadkori et al. (2011) to hold for all rounds with high probability given an appropriate choice of confidence radius $\bar{\beta}_j$. It is natural that this bound is proportional to the weighted norm $\|x\|_{\bar{V}_j^{-1}}$ because this results in the bound being tighter when x is in a direction that has been played more frequently and therefore in a direction for which the least-squares estimator is more accurate. Additionally,

---

Algorithm 1 Optimistically Safe OCO (OSOCO)

---

input Action set $\mathcal{X}$, Horizon T, Online algorithm $\mathcal{A}$, Confidence set radius $\beta_t > 0$, Regularization $\lambda > 0$, Tightening $\kappa \in [0, b_{\min})$.
1: Initialize: $t = 1, j = 1, V_t = \lambda I, S_t = \mathbf{0}$.
2: **while** $t \leq T$ **do**
3:     Initialize phase: $\bar{\beta}_j = \beta_t, \bar{S}_j = S_t, \bar{V}_j = V_t$.
4:     Estimate constraints: $\hat{A}_j = \bar{S}_j \bar{V}_j^{-1}$.
5:     Update pessimistic action set:
    $\mathcal{Y}_j^p := \left\{ x \in \mathcal{X} : \hat{A}_j x + \bar{\beta}_j \|x\|_{\bar{V}_j^{-1}} \mathbf{1} \leq b - \kappa \mathbf{1} \right\}$
6:     Update optimistic action set:
    $\mathcal{Y}_j^o := \left\{ x \in \mathcal{X} : \hat{A}_j x - \bar{\beta}_j \|x\|_{\bar{V}_j^{-1}} \mathbf{1} \leq b - \kappa \mathbf{1} \right\}$
7:     Initialize $\mathcal{A}$ with $\mathcal{Y}_j^o$ as action set.
8:     **while** $\det(V_t) \leq 2 \det(\bar{V}_j)$ **and** $t \leq T$ **do**
9:         Receive $\tilde{x}_t$ from $\mathcal{A}$.
10:        Find safe scaling:
        $\gamma_t = \max \left\{ \mu \in [0,1] : \mu \tilde{x}_t \in \mathcal{Y}_j^p \right\}$.
11:        Play $x_t = \gamma_t \tilde{x}_t$ and observe f_t, y_t.
12:        Send f_t to $\mathcal{A}$.
13:        $V_{t+1} = V_t + x_t x_t^\top, S_{t+1} = S_t + y_t x_t^\top$.
14:        $t = t + 1$.
15:     **end while**
16:     Terminate $\mathcal{A}$.
17:     $j = j + 1$.
18: **end while**

---

we incorporate a tightening parameter κ in the optimistic and pessimistic sets which is used to ensure no violation in expectation. Therefore, the bounds on the constraint function in (1) ensure that the optimistic set (resp. pessimistic set) is a superset (resp. subset) of the tightened feasible set $\tilde{\mathcal{Y}} = \{x \in \mathcal{X} : Ax \leq b - \kappa \mathbf{1}\}$ for all phases with high probability, i.e. $\mathcal{Y}_j^p \subseteq \tilde{\mathcal{Y}} \subseteq \mathcal{Y}_j^o$ for all j with high probability.

Update in Each Round (lines 9-14) In each round, the online optimization algorithm $\mathcal{A}$ chooses a point $\tilde{x}_t$ from the optimistic set (line 9), and then the OSOCO algorithm finds the largest scaling γ_t on $\tilde{x}_t$ such that it is in the pessimistic set (line 10), before playing $x_t = \gamma_t \tilde{x}_t$ (line 11). The matrices used for estimation (V_t, S_t) are then updated (lines 13). The reason that we *scale* the optimistic action $\tilde{x}_t$ down into the pessimistic set rather than using another method, such as the euclidean projection, is because scaling this action down ensures that we are still choosing an action in the *direction* of the optimistic action which gives the algorithm more constraint information in this direction. This ultimately leads to effective decision-making because the optimistic actions are updated on the optimistic set and therefore will naturally gravitate towards areas that are uncertain and/or areas that have low regret, effectively balancing exploitation and

---

[3]If a nonzero safe action x^s and its constraint value Ax^s are given (as in Chaudhary and Kalathil (2022)), then the problem can be shifted such that the safe action is at the origin. This will not impact the other problem parameters, except that D may be increased by a factor ≤ 2.

exploration.

Phase Termination The phase ends when the determinant of the Gram matrix (V_{t+1}) has doubled (line 8), after which it terminates $\mathcal{A}$ (line 16) and starts a new phase. This type of "rare update" ensures that the phase only ends when a significant amount of new constraint information has been collected. As such, it results in a small number $\mathcal{O}(d \log(T))$ of phases and therefore there will only be a small cost to restarting the OCO algorithm in each phase. Note that a related form of rare updating was used as early as Abbasi-Yadkori et al. (2011) to reduce the number of determinant calculations in stochastic linear bandits.

3.2 Requirements of Online Algorithm $\mathcal{A}$

The online optimization algorithm $\mathcal{A}$ needs to be chosen from a specific class of algorithms in order for OSOCO to work. In particular, the optimistic action set is not necessarily convex and $\mathcal{A}$ can be terminated at any time, so $\mathcal{A}$ needs to perform well when its action set is nonconvex and when the horizon is unknown. We make this concrete in the following.

Definition 1 (Regret Bound on $\mathcal{A}$). *Consider an OCO setting where the action set $\mathcal{X}$ is not necessarily convex, and the problem can be terminated at any time (and therefore the final round T is not known to the player). Also, we assume Assumptions 1 and 2 hold, and that $\mathcal{A}$ chooses a distribution p_t in each round from which it samples x_t.[4] Then, $C_{\mathcal{A}}$ is a positive real such that,*

$$\sum_{t \in [T]} \mathbb{E}_{x_t \sim p_t}[f_t(x_t)] - \min_{x \in \mathcal{X}} \sum_{t \in [T]} f_t(x) \leq C_{\mathcal{A}} \sqrt{T},$$

if the algorithm $\mathcal{A}$ is played in any such online setting. We let $C_{\mathcal{A}}$ depend on T at most $\mathcal{O}(\mathrm{polylog}(T))$.

Most algorithms that admit such a regret bound are those designed for the online nonconvex optimization setting as they can handle the nonconvex action set (and then the "doubling trick" can be used to handle the unknown horizon). Some examples of applicable algorithms are FTPL (Kalai and Vempala, 2005) and continuous versions of the Hedge algorithm (Freund and Schapire, 1997) which enjoy $\mathcal{O}(d^{3/2}\sqrt{T})$ and $\tilde{\mathcal{O}}(\sqrt{dT})$ regret, respectively. We explicitly give several example algorithms and their regret guarantees in Appendix F. The primary drawback of these algorithms is that they are often computationally inefficient. In Section 5, we provide a more computationally efficient version of OSOCO at the cost of a slightly worse regret bound (i.e. $\tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$ instead of $\tilde{\mathcal{O}}(d\sqrt{T})$).

---

[4]We assume that $\mathcal{A}$ is randomized because an online optimization algorithm needs randomness to achieve sublinear regret when the action set is not convex, see Cesa-Bianchi and Lugosi (2006).

4 ANALYSIS

In this section, we provide regret and safety guarantees for OSOCO. The high-probability guarantees and expectation guarantees are given in Sections 4.1 and 4.2, respectively.

4.1 High-probability Guarantees

The following theorem shows that, with high probability, OSOCO ensures no violation $Ax_t \leq b$ and that the regret is $\tilde{\mathcal{O}}((d + C_{\mathcal{A}}\sqrt{d})\sqrt{T})$, where $C_{\mathcal{A}}$ is the regret constant of the online algorithm $\mathcal{A}$ that is used by OSOCO (see Definition 1). It immediately follows that the regret of OSOCO is $\tilde{\mathcal{O}}(d\sqrt{T})$ when $C_{\mathcal{A}}$ is $\tilde{\mathcal{O}}(\sqrt{d})$, which holds when algorithms such as Hedge (Freund and Schapire, 1997) are used for $\mathcal{A}$ (given in Appendix F).

Theorem 1. *Fix some $\delta \in (0, 1/2)$, and set $\lambda = \max(1, D^2)$, $\beta_t = \rho\sqrt{d \log\left(\frac{1+(t-1)D^2/\lambda}{\delta/n}\right)} + \sqrt{\lambda}S$ and $\kappa = 0$. Then if $T \geq 3$, with probability at least $1 - 2\delta$, the actions of OSOCO (Algorithm 1) satisfy,*

$$R_T \leq 4\frac{DG}{b_{\min}}\beta_T \sqrt{3dT \log(T)} + C_{\mathcal{A}}\sqrt{4dT \log(T)}$$
$$+ 2DG\sqrt{2T \log(1/\delta)}$$

and $Ax_t \leq b$ for all $t \in [T]$.

The proof of Theorem 1 (given in Appendix B), decomposes the regret as,

$$R_T = \underbrace{\sum_{t \in [T]} (f_t(x_t) - f_t(\tilde{x}_t))}_{\text{Term I}} + \underbrace{\sum_{t \in [T]} (f_t(\tilde{x}_t) - f_t(x^\star))}_{\text{Term II}}.$$

Intuitively, Term I is the *cost of safety*, i.e. the cost that the algorithm incurs by maintaining no violation, while Term II is the *optimistic regret*, i.e. the regret that the algorithm would incur if it played the optimistic actions $\tilde{x}_t$ (although this would violate the constraints). We bound each of these in the following lemmas.

Lemma 1 (Cost of Safety). *Under the same conditions as Theorem 1, it holds that*

$$\text{Term I} \leq \frac{4DG}{b_{\min}}\beta_T \sqrt{3dT \log(T)}. \quad (2)$$

The key idea behind the proof of Lemma 1 is that, if we can relate Term I to the estimation error of the constraint given in (1), then we can show that it is $\tilde{\mathcal{O}}(\sqrt{T})$ by the elliptic potential lemma (Dani et al. (2008)). One difficulty in doing so is relating the distance between $\tilde{x}_t$ and x_t to the estimation error. We do so by first establishing a lower bound on the safe scaling γ_t (line 10) in terms of the estimation error at
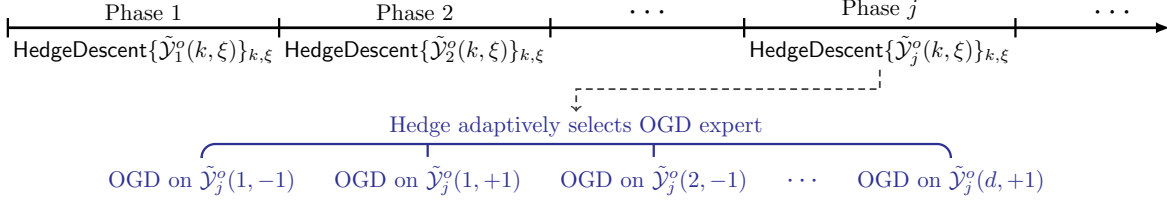
Figure 1: Implementation of HedgeDescent in OSOCO. OSOCO calls HedgeDescent in each phase and passes the family of optimistic sets $\{\tilde{\mathcal{Y}}_j^o(k,\xi)\}_{k,\xi}$ to it. HedgeDescent uses an online gradient descent (OGD) "expert" for each of these sets and then uses Hedge to adaptively select the OGD expert in each round.

the optimistic action $\bar{\beta}_j \|\tilde{x}_t\|_{V_j^{-1}}$ and then relating this to the estimation error at the played actions $\bar{\beta}_j \|x_t\|_{V_j^{-1}}$ using the fact that $\tilde{x}_t$ and x_t are scalar multiples of each other. Then, since all f_t have bounded gradients and thus Lipschitz, we can bound Term I in terms of $\|\tilde{x}_t - x_t\|$ and therefore in terms of $1 - \gamma_t$ and then $\bar{\beta}_j \|x_t\|_{V_j^{-1}}$. Finally, we adapt the rare updating analysis from Abbasi-Yadkori et al. (2011) to account for the fact that the optimistic and pessimistic sets are updated only at the beginning of each phase, and then apply the elliptic potential lemma to get $\tilde{\mathcal{O}}(\sqrt{T})$ bound.

Lemma 2 (Optimistic Regret). *Under the same conditions as Theorem 1, it holds with probability at least $1 - 2\delta$ that*

$$\text{Term II} \leq C_{\mathcal{A}} \sqrt{4dT \log(T)} + DG\sqrt{2T \log(1/\delta)}.$$

The key idea behind the proof of Lemma 2 is that the optimistic set $\mathcal{Y}_j^o$ contains the feasible set $\mathcal{Y}$ with high probability (when $\kappa = 0$), and therefore we can bound Term II in terms of the sum of the regrets that $\mathcal{A}$ incurs in each phase. Concretely, with τ_j denoting the duration of the jth phase and N the number of phases, it holds with high probability,

$$
\begin{aligned}
&\sum_{t=1}^T \mathbb{E}_t[f_t(\tilde{x}_t)] - \sum_{t=1}^T f_t(x^\star) \\
&\leq \sum_{j=1}^N C_{\mathcal{A}}\sqrt{\tau_j} \leq C_{\mathcal{A}}\sqrt{NT} \leq C_{\mathcal{A}}\sqrt{4dT\log(T)},
\end{aligned}
\tag{3}
$$

where the second inequality is Cauchy-Schwarz and the third inequality uses the fact that $N \leq 2d\log(T)$ thanks to the phase termination criteria. Lastly, we use the Azuma inequality to convert (3) to a high probability bound on Term II.

4.2 Expectation Guarantees

The following proposition shows that OSOCO ensures no expected violation $\mathbb{E}[Ax_t] \leq b$ and that the expected regret is $\tilde{\mathcal{O}}((d + C_{\mathcal{A}}\sqrt{d})\sqrt{T})$.

Proposition 1. *Choose $\delta = \min(1/2, b_{\min}/(2SDT))$ and $\kappa = b_{\min}/T$, and set λ and β_t the same as in Theorem 1. Then, if $T \geq 3$, it holds that $\mathbb{E}[Ax_t] \leq b$*

for all $t \in [T]$ and

$$
\begin{aligned}
\mathbb{E}[R_T] \leq{}& \frac{6DG}{b_{\min}}\beta_T\sqrt{2dT\log(T)} + C_{\mathcal{A}}\sqrt{2dT\log(T)} \\
&+ 2DG\sqrt{2T\log(1/\delta)} + GD + \frac{b_{\min}G}{S}.
\end{aligned}
$$

The complete proof of Proposition 1 is given in Appendix C. The key idea behind this proof is that, by tightening the constraint, we can cancel out the violation that might occur if the high probability event in Theorem 1 does not hold. To see this, let $\mathcal{E}$ be the high probability event in Theorem 1 and note that for all $i \in [n]$ and $t \in [T]$,

$$
\begin{aligned}
\mathbb{E}[a_i^\top x_t] &= \mathbb{E}[a_i^\top x_t \mathbb{I}\{\mathcal{E}\}] + \mathbb{E}[a_i^\top x_t \mathbb{I}\{\mathcal{E}^c\}] \\
&\leq b_i - \kappa + 2SD\delta \leq b_i,
\end{aligned}
$$

where the first inequality uses that $a_i^\top x_t \leq b_i - \kappa$ under $\mathcal{E}$, that $a_i^\top x_t \leq SD$ always and that $\mathbb{P}(\mathcal{E}^c) \leq 2\delta$, and the last inequality uses the fact that $2SD\delta \leq \kappa$. Then, to characterize the impact that tightening the constraint has on the regret, we bound the distance between the optimal action $x^\star$ and the "tightened" feasible set $\tilde{\mathcal{Y}} = \{x \in \mathcal{X} : Ax \leq b - \kappa\mathbf{1}\}$. Specifically, we show that this distance is smaller than $\kappa D/b_{\min}$, and therefore the algorithm only incurs a cost of $\mathcal{O}(1/T)$ per round, or $\mathcal{O}(1)$ in total, for tightening the constraint.

5 AN EFFICIENT VERSION

One drawback of the original OSOCO algorithm is that it requires the use of an online nonconvex optimization algorithm, which often does not necessarily allow for computationally efficient implementations. In this section, we modify the OSOCO algorithm to improve its computational efficiency, while only slightly weakening the regret guarantees to $\tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$. The key idea is that we modify the optimistic action set so that it can be represented as the union of a finite number of convex sets. We then give an efficient online optimization algorithm that we call HedgeDescent (to be used for $\mathcal{A}$) that is designed for this type of structured nonconvex action set. The implementation of HedgeDescent in OSOCO is depicted in Figure 1.

Relaxation of Optimistic Action Set In order for the optimistic action set to be amenable to efficient online optimization, we use the idea from Dani et al. (2008) of relaxing the least-squares confidence set. In particular, from the equivalence of norms, the optimistic action set can be relaxed to,

$$\tilde{\mathcal{Y}}_j^o = \left\{ x \in \mathcal{X} : \hat{A}_j x - \sqrt{d}\bar{\beta}_j \|\bar{V}_j^{-1/2} x\|_\infty \mathbf{1} \le b - \kappa \mathbf{1} \right\}$$

Although this set is still nonconvex, it can now be represented as the union of a finite number of convex sets. To explicitly define these convex sets, we use the idea from Cassel et al. (2022) of parameterizing the infinity norm by the active element in the vector and the sign of this element, i.e. $\|z\|_\infty = \max_{k \in [d], \xi \in \{-1,1\}} \xi z_k$ for $z \in \mathbb{R}^d$. As such we can parameterize the optimistic set as,

$$\begin{aligned} \tilde{\mathcal{Y}}_j^o(k, \xi) := \big\{ x \in \mathcal{X} : \\ \hat{A}_j x - \sqrt{d}\bar{\beta}_j \xi (\bar{V}_j^{-1/2})_k x \mathbf{1} \le b - \kappa \mathbf{1} \big\}, \end{aligned}$$

where $(\bar{V}_j^{-1/2})_k$ denotes the kth row of $\bar{V}_j^{-1/2}$. Evidently, $\tilde{\mathcal{Y}}_j^o(k, \xi)$ is convex for any fixed $k, \xi \in [d] \times \{-1, 1\}$ and furthermore the union over k and ξ is the original set, i.e. $\bigcup_{k,\xi \in [d] \times \{-1,1\}} \tilde{\mathcal{Y}}_j^o(k, \xi) = \tilde{\mathcal{Y}}_j^o$. Accordingly, we modify OSOCO such that it sends the family of sets $\{\tilde{\mathcal{Y}}_j^o(k, \xi)\}_{k,\xi}$ to $\mathcal{A}$ in line 7.

HedgeDescent Algorithm In order to appropriately exploit the structure of the relaxed optimistic action set, we propose the online algorithm HedgeDescent (Algorithm 2) to be used for $\mathcal{A}$ in OSOCO. Since the optimistic action set is now the union of a finite number of convex sets, HedgeDescent uses a finite online algorithm (i.e. Hedge (Freund and Schapire, 1997)) to choose which convex set to play in and a continuous online algorithm (i.e. online gradient descent (Zinkevich, 2003)) to ensure low regret within each convex set.[5] As such, HedgeDescent only requires M convex projections (where M is the number of convex sets). When OSOCO is modified such that the optimistic set is $\{\tilde{\mathcal{Y}}_j^o(k, \xi)\}_{k,\xi}$, then $M = 2d$ and thus the combined algorithm requires $2d$ convex projections per round.

Analysis The following proposition (proven in Appendix D) shows that with the aforementioned modifications, OSOCO enjoys $\tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$ regret and no violation (in either the high-probability or expected sense). Although this is slightly worse than the $\tilde{\mathcal{O}}(d\sqrt{T})$ regret attained by the original OSOCO algorithm (Theorem 1 and Proposition 1), the modified algorithm only requires $2d$ convex projections (which is polynomial-time) in each round instead of updating a grid over the action set or performing non-convex optimization

---

[5]This approach is inspired by Cassel et al. (2022), where a similar idea is used to handle nonconvexity in the cost rather than nonconvexity in the constraint.

Algorithm 2 HedgeDescent

input $\{\mathcal{X}_m\}_{m \in [M]}$.
1: Initialize: $x_t(m) = \mathbf{0}$, $p_t(m) = 1/M \; \forall m \in [M]$.
2: **for** $t = 1$ **to** T **do**
3:      Sample Hedge: $m_t \sim p_t$.
4:      Play $x_t(m_t)$ and observe f_t.
5:      Update experts:
         $x_{t+1}(m) = \Pi_{\mathcal{X}_m}(x_t(m) - \eta_t \nabla f_t(x_t(m)))$ for all $m \in [M]$.
6:      Update Hedge:
         $p_{t+1}(m) \propto p_t(m) \exp(-\zeta_t f_t(x_t(m)))$.
7: **end for**

(neither of which is polynomial-time) as required by the original algorithm.

Proposition 2. *If OSOCO is modified such that $\mathcal{Y}_j^o \leftarrow \{\tilde{\mathcal{Y}}_j^o(k, \xi)\}_{k,\xi}$ and HedgeDescent is used for $\mathcal{A}$, then the following hold:*

- *(High-probability) Choosing the algorithm parameters as in Theorem 1 ensures, with probability at least $1 - 2\delta$, that $Ax_t \le b$ and $R_T \le \tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$.*

- *(Expectation) Choosing the algorithm parameters as in Proposition 1 ensures that $\mathbb{E}[Ax_t] \le b$ and $\mathbb{E}[R_T] \le \tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$.*

6 EXTENSION TO OCO WITH STOCHASTIC LINEAR CONSTRAINTS

In this section, we consider the problem of OCO with stochastic linear constraints and bandit constraint feedback, and show that OSOCO guarantees $\tilde{\mathcal{O}}(\sqrt{T})$ expected regret and no expected constraint violation.

6.1 Problem Setup

In each round $t \in [T]$, the player chooses an action x_t from $\mathcal{X}$ and, simultaneously, the adversary choose the convex cost function $f_t : \mathcal{X} \to \mathbb{R}$ according to the actions chosen by the player in previous rounds. The player then receives full cost feedback f_t and bandit constraint feedback $g_t(x_t)$. The constraint function is of the form $g_t(x) = A_t x - b_t$, where $(A_t, b_t)_{t=1}^T$ is an i.i.d. sequence in $\mathbb{R}^{n \times d} \times \mathbb{R}^n$. The player aims to ensure *no expected violation*,

$$\mathbb{E}[g_t(x_t)] \le 0 \quad \forall t \in [T],$$

and to minimize the regret,

$$R_T^{\text{stoch}} := \sum_{t \in [T]} f_t(x_t) - \min_{x \in \mathcal{X} : \bar{g}(x) \le 0} \sum_{t \in [T]} f_t(x),$$
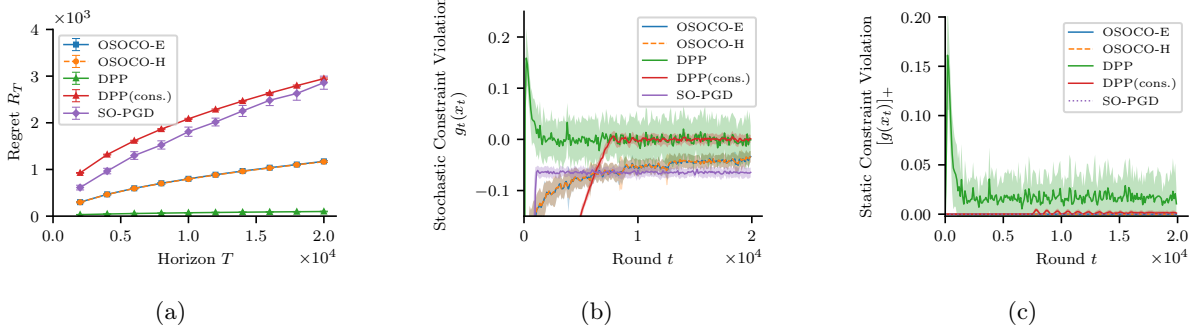
(a)   (b)   (c)

Figure 2: Simulation results for our algorithm OSOCO configured for high-probability (OSOCO-H) and expectation (OSOCO-E) guarantees, alongside DPP from Yu et al. (2017) and SO-PGD from Chaudhary and Kalathil (2022). The points indicate the average over 30 trials and the error bars and shading are ±1 standard deviation.

where $\bar{g}(x) := \mathbb{E}_{A_t,b_t}[g_t(x)] = \mathbb{E}[A_t]x - \mathbb{E}[b_t]$.

Note that we take (A_t, b_t) to be stochastic in this case, whereas we consider deterministic (A, b) in the setting with static constraints (i.e. Section 2).

We use the following the assumptions on the constraint and will assume that the cost functions and action set satisfy Assumptions 1 and 2.

Assumption 6 (Assumptions on Stochastic Constraint). *With $g_{t,i}$ denoting the ith component of g_t, assume that,*

   (i) $\|\nabla g_{t,i}(\mathbf{0})\| \leq G_g$ *for all $i \in [n], t \in [T]$,*

   (ii) $\|g_t(\mathbf{0})\| \leq F$ *for all $t \in [T]$,*

   (iii) $\bar{g}(\mathbf{0}) < 0$,

   (iv) $\bar{g}(\mathbf{0})$ *is known.*

Condition (i) and Condition (ii) are standard in OCO with stochastic constraints, e.g. (Yu et al., 2017, 2023). Condition (iii) requires both the existence of a strictly feasible point in expectation (i.e. Slater's condition) which is commonly used in OCO with stochastic constraints (e.g. Yu et al. (2017); Castiglioni et al. (2022)), as well as knowledge of such a strictly feasible point in expectation. Although the knowledge of a strictly feasible point and its expected function value (i.e. Condition (iv)) is not typically assumed in OCO with stochastic constraints, such an assumption is often made in safe learning problems that ensure no violation, e.g. (Moradipari et al., 2021; Amani et al., 2021).

6.2 Regret and Violation Guarantees

The regret and violation guarantees of OSOCO in this setting in the following corollary to Proposition 1. In particular, Corollary 1 (proven in Appendix E) guarantees no expected violation and $\tilde{\mathcal{O}}((d + C_\mathcal{A}\sqrt{d})\sqrt{T})$

expected regret in general, and $\tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$ expected regret for the more efficient version in Section 5.

Corollary 1. *Suppose that the cost functions and action set satisfy Assumptions 1 and 2. Also, assume that the constraint function satisfies Assumption 6, and let $\rho^2 = (G_g D + 4F)^2/4$. Then playing OSOCO (Algorithm 1) with the algorithm parameters chosen as in Proposition 1 ensures that $\mathbb{E}[g_t(x_t)] \leq 0$ for all $t \in [T]$ and $\mathbb{E}[R_T^{\text{stoch}}] = \tilde{\mathcal{O}}((d + C_\mathcal{A}\sqrt{d})\sqrt{T})$. Furthermore, with modifications in Proposition 2, it holds that $\mathbb{E}[R_T^{\text{stoch}}] = \tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$.*

7 NUMERICAL EXPERIMENTS

In order to further validate the effectiveness of OSOCO, we perform numerical experiments in a toy setting. Due to space limitations, we defer the experiment details to Appendix G and just give a high level explanation here. We study OSOCO configured for both high-probability (OSOCO-H) and expectation (OSOCO-E) guarantees, as well as the algorithm from Yu et al. (2017) with two different choice of parameters (DPP, DPP(cons)) and Chaudhary and Kalathil (2022) (SO-PGD). The results are shown in Figure 2, where average (over 30 trials) is indicated by a point or line and the standard deviation is indicated by an errorbar or shading. DPP has the lowest regret in all experiments, although also has sizeable constraint violation. OSOCO-E and OSOCO-H have the next lowest regret, and do not incur stochastic violation on average, or static violation at all.

8 FUTURE WORK

Some promising directions for future work are (1) considering more general constraint models, such as gaussian process models, and (2) applying our approach to related safe learning problems, such as online control or MDPs with unknown constraints.

Acknowledgements

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. PMLR, 2017.

Sanae Amani, Mahnoosh Alizadeh, and Christos Thrampoulidis. Linear stochastic bandits under safety constraints. *Advances in Neural Information Processing Systems*, 32, 2019.

Sanae Amani, Christos Thrampoulidis, and Lin Yang. Safe reinforcement learning with linear function approximation. In *International Conference on Machine Learning*, pages 243–253. PMLR, 2021.

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of computing*, 8(1):121–164, 2012.

Baruch Awerbuch and Robert Kleinberg. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114, 2008.

Xuanyu Cao and KJ Ray Liu. Online convex optimization with time-varying constraints and bandit feedback. *IEEE Transactions on automatic control*, 64(7):2665–2680, 2018.

Asaf Benjamin Cassel, Alon Peled-Cohen, and Tomer Koren. Rate-optimal online convex optimization in adaptive linear control. *Advances in Neural Information Processing Systems*, 35:7410–7422, 2022.

Matteo Castiglioni, Andrea Celli, Alberto Marchesi, Giulia Romano, and Nicola Gatti. A unifying framework for online optimization with long-term constraints. *Advances in Neural Information Processing Systems*, 35:33589–33602, 2022.

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

Ting-Jui Chang, Sapana Chaudhary, Dileep Kalathil, and Shahin Shahrampour. Dynamic regret analysis of safe distributed online optimization for convex and non-convex problems. *Transactions on Machine Learning Research*, 2023.

Sapana Chaudhary and Dileep Kalathil. Safe online convex optimization with unknown linear safety constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6175–6182, 2022.

Tianyi Chen, Qing Ling, and Georgios B Giannakis. An online convex optimization approach to proactive network resource allocation. *IEEE Transactions on Signal Processing*, 65(24):6350–6364, 2017.

Alexey Chernov and Fedor Zhdanov. Prediction with expert advice under discounted loss. In *International Conference on Algorithmic Learning Theory*, pages 255–269. Springer, 2010.

Alon Cohen, Avinatan Hasidim, Tomer Koren, Nevena Lazic, Yishay Mansour, and Kunal Talwar. Online linear quadratic control. In *International Conference on Machine Learning*, pages 1029–1038. PMLR, 2018.

Alon Cohen, Tomer Koren, and Yishay Mansour. Learning linear-quadratic regulators efficiently with only $\sqrt{T}$ regret. In *International Conference on Machine Learning*, pages 1300–1309. PMLR, 2019.

Thomas M Cover. Universal portfolios. *Mathematical finance*, 1(1):1–29, 1991.

Varsha Dani, Sham M Kakade, and Thomas Hayes. The price of bandit information for online optimization. *Advances in Neural Information Processing Systems*, 20, 2007.

Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. 2008.

Mohammad Fereydounian, Zebang Shen, Aryan Mokhtari, Amin Karbasi, and Hamed Hassani. Safe learning under uncertain objectives and constraints. *arXiv preprint arXiv:2006.13326*, 2020.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

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

Arnob Ghosh, Xingyu Zhou, and Ness Shroff. Towards achieving sub-linear regret and hard constraint violation in model-free rl. In *International Conference on Artificial Intelligence and Statistics*, pages 1054–1062. PMLR, 2024.

Hengquan Guo, Xin Liu, Honghao Wei, and Lei Ying. Online convex optimization with hard constraints: Towards the best of two worlds and beyond. *Advances in Neural Information Processing Systems*, 35:36426–36439, 2022.

Elad Hazan and Satyen Kale. Projection-free online learning. In *Proceedings of the 29th International*

Coference on International Conference on Machine Learning, pages 1843–1850, 2012.

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

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

Spencer Hutchinson and Mahnoosh Alizadeh. Safe online convex optimization with multi-point feedback. In *6th Annual Learning for Dynamics & Control Conference*, pages 168–180. PMLR, 2024.

Spencer Hutchinson, Berkay Turan, and Mahnoosh Alizadeh. Directional optimism for safe linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 658–666. PMLR, 2024.

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

Pavel Kolev, Georg Martius, and Michael Muehlebach. Online learning under adversarial nonlinear constraints. In *Advances in Neural Information Processing Systems*, volume 36, pages 53227–53238, 2023.

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.

Walid Krichene, Maximilian Balandat, Claire Tomlin, and Alexandre Bayen. The hedge algorithm on a continuum. In *International Conference on Machine Learning*, pages 824–832. PMLR, 2015.

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. PMLR, 2019.

Nikolaos Liakopoulos, Apostolos Destounis, Georgios Paschos, Thrasyvoulos Spyropoulos, and Panayotis Mertikopoulos. Cautious regret minimization: Online optimization with long-term budget constraints. In *International Conference on Machine Learning*, pages 3944–3952. PMLR, 2019.

Tao Liu, Ruida Zhou, Dileep Kalathil, Panganamala Kumar, and Chao Tian. Learning policies with zero or bounded constraint violation for constrained mdps. *Advances in Neural Information Processing Systems*, 34:17183–17193, 2021.

Arpan Losalka and Jonathan Scarlett. No-regret algorithms for safe bayesian optimization with monotonicity constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 3232–3240. PMLR, 2024.

Mehrdad Mahdavi, Rong Jin, and Tianbao Yang. Trading regret for efficiency: online convex optimization with long term constraints. *The Journal of Machine Learning Research*, 13(1):2503–2528, 2012.

Odalric-Ambrym Maillard and Rémi Munos. Online learning in adversarial lipschitz environments. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 305–320. Springer, 2010.

Shie Mannor, John N Tsitsiklis, and Jia Yuan Yu. Online learning with sample path constraints. *Journal of Machine Learning Research*, 10(3), 2009.

H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230, 2013.

Zakaria Mhammedi. Efficient projection-free online convex optimization with membership oracle. In *Conference on Learning Theory*, pages 5314–5390. PMLR, 2022.

Ahmadreza Moradipari, Sanae Amani, Mahnoosh Alizadeh, and Christos Thrampoulidis. Safe linear thompson sampling with side information. *IEEE Transactions on Signal Processing*, 69:3755–3767, 2021.

Michael J Neely and Hao Yu. Online convex optimization with time-varying constraints. *arXiv preprint arXiv:1702.04783*, 2017.

Aldo Pacchiano, Mohammad Ghavamzadeh, Peter Bartlett, and Heinrich Jiang. Stochastic bandits with linear constraints. In *International conference on artificial intelligence and statistics*, pages 2827–2835. PMLR, 2021.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

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

Karthik Sridharan and Seung Won Wilson Yoo. Online learning with unknown constraints. *arXiv preprint arXiv:2403.04033v1*, 2024.

Arun Sai Suggala and Praneeth Netrapalli. Online non-convex learning: Following the perturbed leader is optimal. In *Algorithmic Learning Theory*, pages 845–861. PMLR, 2020.

Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with gaus-

sian processes. In *International conference on machine learning*, pages 997–1005. PMLR, 2015.

Yanan Sui, Vincent Zhuang, Joel Burdick, and Yisong Yue. Stagewise safe bayesian optimization with gaussian processes. In *International conference on machine learning*, pages 4781–4789. PMLR, 2018.

Wen Sun, Debadeepta Dey, and Ashish Kapoor. Safety-aware algorithms for adversarial contextual bandit. In *International Conference on Machine Learning*, pages 3280–3288. PMLR, 2017.

Ilnura Usmanova, Andreas Krause, and Maryam Kamgarpour. Safe convex learning under uncertain constraints. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2106–2114. PMLR, 2019.

Akifumi Wachi and Yanan Sui. Safe reinforcement learning in constrained markov decision processes. In *International Conference on Machine Learning*, pages 9797–9806. PMLR, 2020.

Xiaohan Wei, Hao Yu, and Michael J Neely. Online primal-dual mirror descent under stochastic constraints. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(2):1–36, 2020.

Xinlei Yi, Xiuxian Li, Tao Yang, Lihua Xie, Tianyou Chai, and H Karl. Regret and cumulative constraint violation analysis for distributed online constrained convex optimization. *IEEE Transactions on Automatic Control*, 2022.

Hao Yu and Michael J Neely. Learning-aided optimization for energy-harvesting devices with outdated state information. *IEEE/ACM Transactions on Networking*, 27(4):1501–1514, 2019.

Hao Yu and Michael J. Neely. A low complexity algorithm with $O(\sqrt{T})$ regret and $O(1)$ constraint violations for online convex optimization with long term constraints. *Journal of Machine Learning Research*, 21(1):1–24, 2020.

Hao Yu, Michael Neely, and Xiaohan Wei. Online convex optimization with stochastic constraints. *Advances in Neural Information Processing Systems*, 30, 2017.

Jichi Yu, Jueyou Li, and Guo Chen. Online bandit convex optimisation with stochastic constraints via two-point feedback. *International Journal of Systems Science*, 54(10):2089–2105, 2023.

Jianjun Yuan and Andrew Lamperski. Online convex optimization for cumulative constraints. *Advances in Neural Information Processing Systems*, 31, 2018.

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.

Checklist

1. For all models and algorithms presented, check if you include:

   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**/No/Not Applicable]

   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**/No/Not Applicable]

   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**Yes**/No/Not Applicable]

2. For any theoretical claim, check if you include:

   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**/No/Not Applicable]

   (b) Complete proofs of all theoretical results. [**Yes**/No/Not Applicable]

   (c) Clear explanations of any assumptions. [**Yes**/No/Not Applicable]

3. For all figures and tables that present empirical results, check if you include:

   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**/No/Not Applicable]

   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes/No/**Not Applicable**]

   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**/No/Not Applicable]

   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Yes**/No/Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. [Yes/No/**Not Applicable**]

   (b) The license information of the assets, if applicable. [Yes/No/**Not Applicable**]

   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/**Not Applicable**]

   (d) Information about consent from data providers/curators. [Yes/No/**Not Applicable**]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

    (a) The full text of instructions given to participants and screenshots. [Yes/No/**Not Applicable**]

    (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/**Not Applicable**]

    (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/**Not Applicable**]

A   More Related Work

In this section, we discuss the relationship with literature on OCO with long-term constraints, projection-free OCO literature and online control. We also present the literature on OCO with static constraints in Table 1, which is also discussed in Section 1.2.

OCO with Long-term Constraints   The literature on OCO with long-term constraints considers the problem of OCO with fixed constraints where it is not necessary to maintain the feasibility of actions, but instead only to ensure sublinear cumulative violation, e.g. (Mahdavi et al., 2012; Yu and Neely, 2020; Guo et al., 2022). Some methods in this literature only use the constraint function value and gradient at the played action, which is a form of limited feedback on the constraint, e.g. Mahdavi et al. (2012); Yu et al. (2017); Yuan and Lamperski (2018). Nonetheless, this is a richer form of feedback than we consider (noisy function value) and therefore these methods are not applicable to our setting. Furthermore, these methods do not ensure feasibility, and therefore are not applicable for the guarantees that we are interested in.

Projection-free OCO   The problem of OCO under unknown constraints has similar goals to that of projection-free OCO, as both settings aim to develop effective methods for OCO when the action set cannot be accessed through a projection oracle. A large number of projection-free OCO methods use the cheaper linear optimization oracle instead of the projection oracle and often use variants of the Franke-Wolfe method, e.g. Hazan and Kale (2012); Garber and Hazan (2016); Hazan and Minasyan (2020); Kretzu and Garber (2021). Some other approaches use the membership oracle Mhammedi (2022) or the constraint function and gradient at any point Levy and Krause (2019). The literature on projection-free OCO is distinct from OCO under unknown constraints in that they consider different types of oracles that use the constraints, e.g. membership oracle, constraint function value and gradient at any point, or linear optimization oracle.

Online Control   There is an interesting connection between OCO with unknown linear constraints and online control with unknown linear dynamics (e.g. Cohen et al. (2018); Simchowitz et al. (2020); Cassel et al. (2022)). In particular, both settings have unknown linear functions with noisy feedback and adversarial convex costs. However, in online control, the unknown linear function impacts the incurred cost, while in OCO with unknown linear constraints, the unknown linear function forms the constraint. The latter setting poses a fundamentally different challenge that we aim to address in this work.

Table 1: Prior work and our results in OCO with static constraints. The term C_T^* refers to the path length of the minimizer sequence as defined in Chang et al. (2023).

Reference	Regret Type	Setting	Cost Type	Regret Bound	Efficient?
Chaudhary and Kalathil (2022)	Static	Centralized	Convex	$\tilde{\mathcal{O}}(dT^{2/3})$	Yes
Chang et al. (2023)	Dynamic	Distributed	Convex	$\tilde{\mathcal{O}}(dT^{2/3} + C_T^* T^{1/3})$	Yes
Chang et al. (2023)	Dynamic	Distributed	Non-convex	$\tilde{\mathcal{O}}(dT^{2/3} + C_T^* T^{2/3})$	Yes
Theorem 1	Static	Centralized	Convex	$\tilde{\mathcal{O}}(d\sqrt{T})$	No
Proposition 2	Static	Centralized	Convex	$\tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$	Yes

B   Proof of Theorem 1 (High-probability Guarantees)

In this section, we give the proof of Theorem 1. We first give the notation and assumptions in Section B.1 and state the high probability events in Section B.2. Then, we give the bound on Term I (Cost of Safety) in Section B.3 and the bound on Term II (Optimistic Regret) in Section B.4. We then complete the proof of Theorem 1 (Regret of OSOCO) in Section B.5. Lastly, side lemmas are given in Section B.6.

B.1 Notation and Assumptions

In addition to the notation given in the body of the paper, we use N to refer to the total number of phases and t_j to refer to the first round within phase j where $t_{N+1} := T$. Additionally, we use j_t to refer to the phase to which round t belongs, i.e. $j_t = \max\{j \in [N] : t_j \leq t\}$. Lastly, we use the notation $\mathcal{H}_t := \sigma(\epsilon_1, .., \epsilon_t, \tilde{x}_1, ..., \tilde{x}_t)$, i.e. the randomness in the first t rounds (due to both the environment and algorithm), and note that $\mathbb{E}[\cdot | \mathcal{H}_{t-1}] = \mathbb{E}_t[\cdot]$.

In addition to the assumptions made in the body, we will also assume that the assumptions made in Theorem 1 hold, i.e. $\kappa = 0$, $\lambda = \max(1, D^2)$, $\beta_t := \rho\sqrt{d\log\left(\frac{1+(t-1)L^2/\lambda}{\delta/n}\right)} + \sqrt{\lambda}S$, that $\mathcal{A}$ enjoys a regret constant $C_{\mathcal{A}}$ (in the sense of Definition 1) and that $T \geq 3$.

B.2 High probability events

In this section, we show that with an appropriate choice of problem parameters, the events

$$\mathcal{E}_{\text{conf}} := \left\{(A - \hat{A}_j)x \in \bar{\beta}_j \|x\|_{\bar{V}_j^{-1}}\mathbb{B}_\infty, \forall x \in \mathbb{R}^d, \forall j \in [N]\right\},$$

and

$$\mathcal{E}_{\text{azuma}} := \left\{\sum_{t=1}^{T}(f_t(\tilde{x}_t) - \mathbb{E}[f_t(\tilde{x}_t)|\mathcal{H}_{t-1}]) \leq 2DG\sqrt{2T\log(1/\delta)}\right\},$$

occur with high probability. First, the following theorem from Abbasi-Yadkori et al. (2011) tells us that $\mathbb{P}(\mathcal{E}_{\text{conf}}) \geq 1 - \delta$.

Theorem 2 (Theorem 2 in Abbasi-Yadkori et al. (2011)). *Fix $\delta \in (0, 1)$ and let $\check{A}_t = S_t V_t^{-1}$ and,*

$$\beta_t := \rho\sqrt{d\log\left(\frac{1 + (t-1)D^2/\lambda}{\delta/n}\right)} + \sqrt{\lambda}S.$$

Then, with probability at least $1 - \delta$, it holds for all $t \geq 1$ and $x \in \mathbb{R}^d$ that $(\check{A}_t - A)x \in \beta_t\|x\|_{V_t^{-1}}\mathbb{B}_\infty$.

Then, an application of Azuma's inequality yields a bound on the probability of $\mathcal{E}_{\text{azuma}}$ as shown in the following lemma.

Lemma 3 (Probability of $\mathcal{E}_{\text{azuma}}$). *The event $\mathcal{E}_{\text{azuma}}$ occurs with probability at least $1 - \delta$.*

Proof. Let

$$Z_t := \sum_{s=1}^{t}(f_s(\tilde{x}_s) - \mathbb{E}[f_s(\tilde{x}_s)|\mathcal{H}_{s-1}])$$

We prove this lemma by proving that $Z = (Z_t)_t$ is a martingale with respect to $\mathcal{H} = (\mathcal{H}_t)_t$ and has bounded differences. Then, an application of Azuma's inequality gives the statement of the lemma.

First, note that f_t is fully determined by the randomness in rounds $[t - 1]$ and $\tilde{x}_t$ is fully determined by the randomness in rounds $[t]$, so $f_t(\tilde{x}_t)$ is $\mathcal{H}_t$-measurable. Therefore, Z_t is $\mathcal{H}_t$-measurable and we say that Z is adapted to $\mathcal{H}$. Furthermore, it holds that

$$\mathbb{E}[Z_t|\mathcal{H}_{t-1}] = \mathbb{E}\left[\sum_{s=1}^{t}(f_s(\tilde{x}_s) - \mathbb{E}[f_s(\tilde{x}_s)|\mathcal{H}_{s-1}]) \,\middle|\, \mathcal{H}_{t-1}\right]$$

$$= \sum_{s=1}^{t-1}(f_s(\tilde{x}_s) - \mathbb{E}[f_s(\tilde{x}_s)|\mathcal{H}_{s-1}]) + \mathbb{E}\left[f_t(\tilde{x}_t)|\mathcal{H}_{t-1}\right] - \mathbb{E}[f_t(\tilde{x}_t)|\mathcal{H}_{t-1}]$$

$$= Z_{t-1}$$

and therefore Z is a martingale with respect to $\mathcal{H}$. Then, by fixing $y \in \mathcal{X}$, it follows that

$$
\begin{aligned}
|Z_t - Z_{t-1}| &= |f_t(\tilde{x}_t) - \mathbb{E}[f_t(\tilde{x}_t)|\mathcal{H}_{t-1}]| \\
&= |f_t(\tilde{x}_t) - f_t(y) + f_t(y) - \mathbb{E}[f_t(\tilde{x}_t)|\mathcal{H}_{t-1}]| \\
&= |f_t(\tilde{x}_t) - f_t(y) + \mathbb{E}[f_t(y) - f_t(\tilde{x}_t)|\mathcal{H}_{t-1}]| \\
&\leq |f_t(\tilde{x}_t) - f_t(y)| + \mathbb{E}[|f_t(y) - f_t(\tilde{x}_t)||\mathcal{H}_{t-1}] \\
&\leq 2G\|\tilde{x}_t - y\| \\
&\leq 2GD,
\end{aligned}
\tag{4}
$$

where we the first inequality uses Jensen's inequality, the second inequality uses the assumption of bounded gradients as specified in Assumption 1, and the third inequality uses the assumption of bounded action set from Assumption 2. Therefore, we can apply Azuma's inequality to get that

$$
\mathbb{P}(Z_T > \nu) \leq \exp\left(\frac{-2\nu^2}{4TG^2D^2}\right),
$$

for any $\nu > 0$. Setting the right-hand side to δ gives that $\mathbb{P}(\mathcal{E}_{\text{azuma}}) \geq 1 - \delta$. □

B.3 Term I (Cost of Safety)

Before proving the bound on Term I, we first give a key lemma lower bounding γ_t.

Lemma 4 (Bound on γ_t). *It holds for all $t \in [T]$ that*

$$
\gamma_t \geq 1 - \frac{2}{b_{\min}}\bar{\beta}_{j_t}\|x_t\|_{\bar{V}_{j_t}^{-1}}.
$$

Proof. Throughout the proof, we we take $j = j_t$ to simplify notation. Then to establish the proof, we first show that $\alpha\tilde{x}_t \in \mathcal{Y}_j^p$ where

$$
\alpha := \frac{b_{\min}}{b_{\min} + 2\bar{\beta}_j\|\tilde{x}_t\|_{\bar{V}_j^{-1}}}
$$

and then complete the proof by showing that this implies the statement of the lemma. Each of these steps are given in the following sections.

Proof that $\alpha\tilde{x}_t \in \mathcal{Y}_j^p$: It holds that $\alpha\tilde{x}_t \in \mathcal{Y}_j^p$ if both (a) $\alpha\tilde{x}_t \in \mathcal{X}$ and (b) $\hat{a}_{j,i}^\top(\alpha\tilde{x}_t) + \bar{\beta}_j\|\alpha\tilde{x}_t\|_{\bar{V}_j^{-1}} \leq b_i$ for all $i \in [n]$. Point (a) holds because $\alpha \in [0,1]$, $\tilde{x}_t \in \mathcal{X}$, $\mathbf{0} \in \mathcal{X}$ and $\mathcal{X}$ is convex, and therefore

$$
\alpha\tilde{x}_t = \alpha\tilde{x}_t + (1-\alpha)\mathbf{0} \in \mathcal{X}.
$$

Point (b) holds given that for all $i \in [n]$,

$$
\begin{aligned}
\hat{a}_{j,i}^\top(\alpha\tilde{x}_t) + \bar{\beta}_j\|\alpha\tilde{x}_t\|_{\bar{V}_j^{-1}} &= \alpha(\hat{a}_{j,i}^\top\tilde{x}_t + \bar{\beta}_j\|\tilde{x}_t\|_{\bar{V}_j^{-1}}) \\
&= \alpha(\hat{a}_{j,i}^\top\tilde{x}_t - \bar{\beta}_j\|\tilde{x}_t\|_{\bar{V}_j^{-1}} + 2\bar{\beta}_j\|\tilde{x}_t\|_{\bar{V}_j^{-1}}) \\
&\leq \alpha(b_i + 2\bar{\beta}_j\|\tilde{x}_t\|_{\bar{V}_j^{-1}}) & \text{(a)} \\
&= \frac{b_{\min}}{b_{\min} + 2\bar{\beta}_j\|\tilde{x}_t\|_{\bar{V}_j^{-1}}}(b_i + 2\bar{\beta}_j\|\tilde{x}_t\|_{\bar{V}_j^{-1}}) & \text{(b)} \\
&\leq \frac{b_i}{b_i + 2\bar{\beta}_j\|\tilde{x}_t\|_{\bar{V}_j^{-1}}}(b_i + 2\bar{\beta}_j\|\tilde{x}_t\|_{\bar{V}_j^{-1}}) & \text{(c)} \\
&= b_i
\end{aligned}
$$

where (a) uses the fact that $\tilde{x}_t \in \mathcal{Y}_j^o$, (b) uses the definition of α, and (c) uses the fact that $\frac{b_{\min}}{b_{\min} + 2\bar{\beta}_j\|\tilde{x}_t\|_{\bar{V}_j^{-1}}}$ is increasing in $b_{\min}$ and $b_{\min} \leq b_i$.

Completing the proof: From the definition of γ_t, it holds that

$$\gamma_t = \max\{\alpha \in [0,1] : \alpha\tilde{x}_t \in \mathcal{Y}_j^p\} \geq \frac{b_{\min}}{b_{\min} + 2\bar{\beta}_j\|\tilde{x}_t\|_{\bar{V}_j^{-1}}}. \tag{5}$$

Then, since $x_t = \gamma_t\tilde{x}_t$ and $\gamma_t \geq 0$, it also holds that

$$\gamma_t\|\tilde{x}_t\|_{\bar{V}_j^{-1}} = \|\gamma_t\tilde{x}_t\|_{\bar{V}_j^{-1}} = \|x_t\|_{\bar{V}_j^{-1}}.$$

Therefore, we can rearrange (5) to get that

$$\gamma_t \geq 1 - 2\frac{1}{b_{\min}}\bar{\beta}_j\|x_t\|_{\bar{V}_j^{-1}},$$

completing the proof. □

With this, we then give the bound on Term I.

Lemma 5 (Cost of Safety). *It holds that*

$$\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(\tilde{x}_t) \leq 4DG\frac{1}{b_{\min}}\beta_T\sqrt{3dT\log(T)}. \tag{6}$$

Proof. We can study the lefthand side of (6) to get that

$$\sum_{t=1}^{T}(f_t(x_t) - f_t(\tilde{x}_t)) \leq G\sum_{t=1}^{T}\|\tilde{x}_t - x_t\|$$

$$= G\sum_{t=1}^{T}\|\tilde{x}_t - \gamma_t\tilde{x}_t\| \tag{a}$$

$$\leq DG\sum_{t=1}^{T}(1 - \gamma_t)$$

$$\leq 2DG\frac{1}{b_{\min}}\sum_{t=1}^{T}\bar{\beta}_{j_t}\|x_t\|_{\bar{V}_{j_t}^{-1}} \tag{b}$$

$$\leq 2DG\frac{1}{b_{\min}}\beta_T\sum_{t=1}^{T}\|x_t\|_{\bar{V}_{j_t}^{-1}} \tag{c}$$

$$\leq 2DG\frac{1}{b_{\min}}\beta_T\sum_{t=1}^{T}\frac{\det(V_t)}{\det(\bar{V}_{j_t})}\|x_t\|_{V_t^{-1}} \tag{d}$$

$$\leq 4DG\frac{1}{b_{\min}}\beta_T\sum_{t=1}^{T}\|x_t\|_{V_t^{-1}} \tag{e}$$

$$\leq 4DG\frac{1}{b_{\min}}\beta_T\sqrt{T\sum_{t=1}^{T}\|x_t\|_{V_t^{-1}}^2} \tag{f}$$

$$\leq 4DG\frac{1}{b_{\min}}\beta_T\sqrt{2dT\log\left(1 + \frac{T}{\lambda d}\right)} \tag{g}$$

$$\leq 4DG\frac{1}{b_{\min}}\beta_T\sqrt{3dT\log(T)}, \tag{h}$$

where each step is justified in the following:

(a) $x_t = \gamma_t\tilde{x}_t$ as specified in line 11 in the algorithm.

(b) Lemma 4.

(c) β_t is increasing in t.

(d) For pd matrices $B \preceq C$ it holds that $\|z\|_{B^{-1}} \leq \frac{\det(C)}{\det(B)} \|z\|_{C^{-1}}$ for any z (see e.g. Lemma 12 in Abbasi-Yadkori et al. (2011) or Lemma 27 in Cohen et al. (2019)).

(e) Follows from phase termination criteria (line 14) which ensures that $\det(V_t) \leq 2\det(\bar{V}_{j_t})$ for all t.

(f) Cauchy-Schwarz.

(g) Elliptic potential lemma (Lemma 7).

(h) $2d\log\left(1 + \frac{T}{\lambda d}\right) \leq 2d\log\left(1 + T\right) \leq 3d\log\left(T\right)$ when $T \geq 3$ and $\lambda = \max(1, D^2)$.

$\square$

B.4 Term II (Optimistic Regret)

We prove a bound on Term II in the following.

Lemma 6 (Optimistic Regret). *Conditioned on $\mathcal{E}_{\mathrm{conf}}$ and $\mathcal{E}_{\mathrm{azuma}}$, it holds that*

$$\sum_{t=1}^{T} f_t(\tilde{x}_t) - \sum_{t=1}^{T} f_t(x^\star) \leq C_{\mathcal{A}}\sqrt{4dT\log(T)} + DG\sqrt{2T\log(1/\delta)}$$

Proof. Let $x_j^\star$ be an optimal action within the optimistic set in phase j, i.e. $x_j^\star \in \arg\min_{x \in \mathcal{Y}_j^\circ} \sum_{t=t_j}^{t_{j+1}-1} f_t(x)$. Then, consider the decomposition,

$$\sum_{t=1}^{T} f_t(\tilde{x}_t) - \sum_{t=1}^{T} f_t(x^\star) = \underbrace{\sum_{t=1}^{T} f_t(\tilde{x}_t) - \sum_{t=1}^{T} \mathbb{E}[f_t(\tilde{x}_t)|\mathcal{H}_{t-1}]}_{\text{Term I}}$$

$$+ \underbrace{\sum_{t=1}^{T} \mathbb{E}[f_t(\tilde{x}_t)|\mathcal{H}_{t-1}] - \sum_{j=1}^{N} \sum_{t=t_j}^{t_{j+1}-1} f_t(x_j^\star)}_{\text{Term II}}$$

$$+ \underbrace{\sum_{j=1}^{N} \sum_{t=t_j}^{t_{j+1}-1} f_t(x_j^\star) - \sum_{t=1}^{T} f_t(x^\star)}_{\text{Term III}}.$$

We bound each of the terms in the following sections.

Term I: Conditioning on $\mathcal{E}_{\mathrm{azuma}}$, it holds by definition that

$$\text{Term I} = \sum_{t=1}^{T} f_t(\tilde{x}_t) - \sum_{t=1}^{T} \mathbb{E}[f_t(\tilde{x}_t)|\mathcal{H}_{t-1}] \leq 2DG\sqrt{2T\log(1/\delta)}. \tag{7}$$

Term II: We bound Term II as follows,

$$\text{Term II} = \sum_{j=1}^{N} \sum_{t=t_j}^{t_{j+1}-1} \mathbb{E}[f_t(\tilde{x}_t)|\mathcal{H}_{t-1}] - \sum_{j=1}^{N} \sum_{t=t_j}^{t_{j+1}-1} f_t(x_j^\star)$$

$$= \sum_{j=1}^{N} \sum_{t=t_j}^{t_{j+1}-1} \mathbb{E}_{\tilde{x}_t \sim p_t}[f_t(\tilde{x}_t)] - \sum_{j=1}^{N} \sum_{t=t_j}^{t_{j+1}-1} f_t(x_j^\star) \tag{a}$$

$$\leq \sum_{j=1}^{N} C_{\mathcal{A}} \sqrt{t_{j+1} - t_j} \tag{b}$$

$$\leq C_{\mathcal{A}} \sum_{j=1}^{N} \sqrt{t_{j+1} - t_j} \tag{c}$$

$$\leq C_{\mathcal{A}} \sqrt{N \sum_{j=1}^{N} (t_{j+1} - t_j)} \tag{d}$$

$$= C_{\mathcal{A}} \sqrt{NT}$$

$$\leq C_{\mathcal{A}} \sqrt{4d \log(T) T} \tag{e}$$

where each step is justified in the following:

(a) Note that the the distribution from which $\tilde{x}_t$ is sampled (denoted by p_t) is fully determined by the randomness in rounds $[t-1]$ and therefore p_t is the conditional distribution of $\tilde{x}_t$ given $\mathcal{H}_{t-1}$. Additionally, the adversary chooses f_t simultaneous to the player choosing x_t and therefore f_t is determined by the randomness in rounds $[t-1]$. Therefore, the conditional distribution of $f_t(\tilde{x}_t)$ given $\mathcal{H}_{t-1}$ is p_t, and thus $\mathbb{E}[f_t(\tilde{x}_t)|\mathcal{H}_{t-1}] = \mathbb{E}_{\tilde{x}_t \sim p_t}[f_t(\tilde{x}_t)]$.

(b) The regret bound of $\mathcal{A}$ as specified in Definition 1. Note that the duration of each phase $(t_{j+1} - t_j)$ depends on the actions chosen in that phase. Nonetheless, $\mathcal{A}$ is specified to handle a setting that can be terminated at any time and therefore the regret guarantees hold even when the duration of the phase depends on the actions taken.

(c) The fact that $C_{\mathcal{A}}$ is polylogarithmic in $t_{j+1} - t_j$ and therefore (with abuse of notation) pull it out of the summation to mean that it is polylogarithmic in T as $t_{j+1} - t_j \leq T$.

(d) Cauchy-Schwarz.

(e) $N \leq 4d \log(T)$ from Lemma 8.

Term III: Conditioned on $\mathcal{E}_{\mathrm{conf}}$, it holds that

$$\text{Term III} = \sum_{j=1}^{N} \sum_{t=t_j}^{t_{j+1}-1} f_t(x_j^\star) - \sum_{t=1}^{T} f_t(x^\star)$$

$$= \sum_{j=1}^{N} \min_{x \in \mathcal{Y}_j^o} \sum_{t=t_j}^{t_{j+1}-1} f_t(x) - \sum_{t=1}^{T} f_t(x^\star)$$

$$\leq \sum_{j=1}^{N} \min_{x \in \mathcal{Y}} \sum_{t=t_j}^{t_{j+1}-1} f_t(x) - \sum_{t=1}^{T} f_t(x^\star)$$

$$\leq \min_{x \in \mathcal{Y}} \sum_{t=1}^{T} f_t(x) - \sum_{t=1}^{T} f_t(x^\star) = 0,$$

where the first inequality uses $\mathcal{Y} \subseteq \mathcal{Y}_j^o$ for all j which follows from Theorem 2.

Completing the proof: Putting everything together, we get, conditioned on $\mathcal{E}_{\mathrm{conf}}$ and $\mathcal{E}_{\mathrm{azuma}}$ that

$$\sum_{t=1}^{T} f_t(\tilde{x}_t) - \sum_{t=1}^{T} f_t(x^\star) \leq C_{\mathcal{A}} \sqrt{4dT \log(T)} + DG \sqrt{2T \log(1/\delta)},$$

completing the proof. □

B.5 Completing the proof

Finally, we restate Theorem 1 in the following.

Theorem 3 (Restatement of Theorem 1). *Fix some $\delta \in (0, 1/2)$. Then, with probability at least $1 - 2\delta$, the actions of OSOCO (Algorithm 1) satisfy,*

$$R_T \leq 4\frac{DG}{b_{\min}}\beta_T\sqrt{3dT\log(T)} + C_{\mathcal{A}}\sqrt{4dT\log(T)} + 2DG\sqrt{2T\log(1/\delta)}$$

and $Ax_t \leq b$ for all $t \in [T]$.

Proof. We can apply Lemmas 5 and 6 and condition on $\mathcal{E}_{\mathrm{conf}}$ and $\mathcal{E}_{\mathrm{azuma}}$ to get that

$$\begin{aligned}
R_T &= \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^\star) \\
&= \underbrace{\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(\tilde{x}_t)}_{\text{Term I}} + \underbrace{\sum_{t=1}^{T} f_t(\tilde{x}_t) - \sum_{t=1}^{T} f_t(x^\star)}_{\text{Term II}} \\
&\leq 4DG\frac{1}{b_{\min}}\beta_T\sqrt{2dT\log(T)} + 2DG\sqrt{2T\log(1/\delta)} + C_{\mathcal{A}}\sqrt{4d\log(T)T}.
\end{aligned}$$

Also, conditioned on $\mathcal{E}_{\mathrm{conf}}$, it holds that $x_t \in \mathcal{Y}_{j_t}^p \subseteq \mathcal{Y} := \{x \in \mathcal{X} : Ax \leq b\}$ for all $t \in [T]$ and therefore $Ax_t \leq b$ for all $t \in [T]$. To get a bound of the probability of both events holding, we use Theorem 2 and Lemma 3 with the union bound to get that

$$\mathbb{P}(\mathcal{E}_{\mathrm{conf}} \cap \mathcal{E}_{\mathrm{azuma}}) = 1 - \mathbb{P}(\mathcal{E}_{\mathrm{conf}}^c \cup \mathcal{E}_{\mathrm{azuma}}^c) \geq 1 - (\mathbb{P}(\mathcal{E}_{\mathrm{conf}}^c) + \mathbb{P}(\mathcal{E}_{\mathrm{azuma}}^c)) \geq 1 - 2\delta,$$

completing the proof. □

B.6 Side lemmas

In this section, we give side lemmas that were used in the proof. First, we have the well-known elliptic potential lemma. The version that we give is from Abbasi-Yadkori et al. (2011).

Lemma 7 (Elliptic Potential, Lemma 11 in Abbasi-Yadkori et al. (2011)). *Consider a sequence $(w_t)_{t\in\mathbb{N}}$ where $w_t \in \mathbb{R}^d$ and $\|w_t\| \leq L$ for all $t \in \mathbb{N}$. Let $W_t = \lambda I + \sum_{s=1}^{t-1} w_s w_s^\top$ for some $\lambda \geq \max(1, L^2)$. Then, it holds for any $m \in \mathbb{N}$ that*

$$\sum_{t=1}^{m} \|w_t\|_{W_t^{-1}}^2 \leq 2d\log\left(1 + \frac{m}{\lambda d}\right)$$

Then, we give a bound on the number of phases in the following lemma.

Lemma 8 (Number of Phases). *It holds that $N \leq 4d\log(T)$.*

Proof. Because $\det(\bar{V}_{j+1}) \geq 2\det(\bar{V}_j)$ for all $j \in [1, N-1]$ by definition, it holds that

$$\begin{aligned}
&\det(V_T) \geq \det(\bar{V}_N) \geq 2\det(\bar{V}_{N-1}) \geq \cdots \geq 2^{N-1}\det(\bar{V}_1) \\
\implies\quad &2^{N-1} \leq \frac{\det(V_T)}{\det(\bar{V}_1)} \\
\iff\quad &(N-1)\log(2) \leq \log\left(\frac{\det(V_T)}{\det(\bar{V}_1)}\right) \\
\implies\quad &N \leq 1 + 2\log\left(\frac{\det(V_T)}{\det(\bar{V}_1)}\right)
\end{aligned}$$

At the same time, it holds that,

$$\frac{\det(V_T)}{\det(\bar{V}_1)} = \det(\bar{V}_1^{-1/2}V_T\bar{V}_1^{-1/2})$$

$$\leq \|\bar{V}_1^{-1/2} V_T \bar{V}_1^{-1/2}\|^d$$

$$= \left\| \bar{V}_1^{-1/2} \left(\sum_{t=1}^{T} x_t x_t^\top + \bar{V}_1 \right) \bar{V}_1^{-1/2} \right\|^d$$

$$= \left\| \frac{1}{\lambda} \sum_{t=1}^{T} x_t x_t^\top + I \right\|^d$$

$$\leq \left(\frac{1}{\lambda} \sum_{t=1}^{T} \|x_t x_t^\top\| + 1 \right)^d$$

$$= \left(\frac{1}{\lambda} \sum_{t=1}^{T} \|x_t\|^2 + 1 \right)^d$$

$$\leq \left(\frac{TD^2}{\lambda} + 1 \right)^d.$$

Putting everything together,

$$N \leq 1 + 2 \log \left(\frac{\det(V_T)}{\det(\bar{V}_1)} \right) \leq 1 + 2d \log \left(\frac{TD^2}{\lambda} + 1 \right) \leq 1 + 2d \log(T+1) \leq 4d \log(T)$$

where we use $\lambda \geq D^2$ and $T \geq 3$. $\qquad\square$

C   Proof of Proposition 1 (Expectation Guarantees)

In this section, we restate and prove Proposition 1.

Proposition 3 (Restatement of Proposition 1). *Choose $\delta = \min(1/2, b_{\min}/(2SDT))$ and $\kappa = b_{\min}/T$, and set λ and β_t the same as in Thereom 1. Then, if $T \geq 3$, it holds that $\mathbb{E}[Ax_t] \leq b$ for all $t \in [T]$ and*

$$\mathbb{E}[R_T] \leq \frac{6DG}{b_{\min}} \beta_T \sqrt{2dT \log(T)} + C_{\mathcal{A}} \sqrt{2dT \log(T)} + 2DG \sqrt{2T \log(1/\delta)} + GD + \frac{b_{\min} G}{S}.$$

Proof. Let the event that Theorem 1 holds be denoted by $\mathcal{E}$ which occurs with probability at least $1 - 2\delta$. We individually show the safety guarantee and regret guarantee in the following sections.

Safety guarantee: Running OSOCO (with $\kappa > 0$) ensures that $Ax_t \leq b - \kappa \mathbf{1}$ for all $t \in [T]$ conditioned on $\mathcal{E}$. Therefore, it holds for all $i \in [n]$ that

$$\mathbb{E}[a_i^\top x_t] = \mathbb{E}[a_i^\top x_t \mathbb{I}\{\mathcal{E}\}] + \mathbb{E}[a_i^\top x_t \mathbb{I}\{\mathcal{E}^c\}]$$
$$\leq b_i - \kappa + 2SD\delta$$
$$\leq b_i - b_{\min}/T + b_{\min}/T$$
$$= b_i$$

where we use the trivial constraint bound $a_i^\top x_t \leq SD$ in the first inequality and then the choice of κ and δ in the last step. This gives the safety guarantee.

Regret guarantee: Let $\tilde{\mathcal{Y}} = \{x \in \mathcal{X} : Ax \leq b - \mathbf{1}\kappa\}$. First, we show that when $\kappa \in [0, b_{\min}]$, it holds that $\alpha x^\star$ is in $\tilde{\mathcal{Y}}$ where $\alpha := \frac{b_{\min} - \kappa}{b_{\min}}$. To do so, we need that both (a) $\alpha x^\star \in \mathcal{X}$ and (b) $a_i^\top (\alpha x^\star) \leq b_i - \kappa$ for all $i \in [n]$. Point (a) holds because $\alpha \in [0, 1]$, $\mathbf{0} \in \mathcal{X}$, $x^\star \in \mathcal{X}$ and $\mathcal{X}$ is convex, so

$$\alpha x^\star = \alpha x^\star + (1 - \alpha)\mathbf{0} \in \mathcal{X}.$$

Point (b) holds because, for all $i \in [n]$,

$$a_i^\top (\alpha x^\star) = \alpha a_i^\top x^\star \leq \alpha b_i = \frac{b_{\min} - \kappa}{b_{\min}} b_i \leq \frac{b_i - \kappa}{b_i} b_i = b_i - \kappa,$$

where the first inequality uses that $x^\star$ is feasible (i.e. $Ax^\star \le b$) and the second inequality uses the fact that $\frac{b_{\min}-\kappa}{b_{\min}}$ is increasing in $b_{\min}$ and $b_{\min} \le b_i$.

Then, with $\bar{x}^\star = \arg\min_{x \in \tilde{\mathcal{Y}}} \sum_{t=1}^T f_t(x)$ and R_T^{osoco} as the (high probability) regret bound of OSOCO when $\kappa > 0$, it holds conditioned on $\mathcal{E}$ that

$$R_T = \sum_{t=1}^T f_t(x_t) - \sum_{t=1}^T f_t(x^\star) = \sum_{t=1}^T f_t(x_t) - \sum_{t=1}^T f_t(\bar{x}^\star) + \sum_{t=1}^T f_t(\bar{x}^\star) - \sum_{t=1}^T f_t(x^\star)$$

$$\le R_T^{\text{osoco}} + \sum_{t=1}^T f_t(\bar{x}^\star) - \sum_{t=1}^T f_t(x^\star)$$

$$\le R_T^{\text{osoco}} + \sum_{t=1}^T f_t(\alpha x^\star) - \sum_{t=1}^T f_t(x^\star)$$

$$\le R_T^{\text{osoco}} + G \sum_{t=1}^T \|\alpha x^\star - x^\star\|$$

$$\le R_T^{\text{osoco}} + (1-\alpha)GDT$$

$$\le R_T^{\text{osoco}} + \frac{\kappa}{b_{\min}}GDT$$

$$\le R_T^{\text{osoco}} + GD,$$

where the first inequality uses the regret bound on OSOCO given that it plays within $\tilde{\mathcal{Y}}$ and $\bar{x}^\star$ is the optimal action within $\tilde{\mathcal{Y}}$, the second inequality is due to the fact that $\alpha x^\star$ is in $\tilde{\mathcal{Y}}$ and therefore $\sum_{t=1}^T f_t(\bar{x}^\star) = \min_{x \in \tilde{\mathcal{Y}}} \sum_{t=1}^T f_t(x) \le \sum_{t=1}^T f_t(\alpha x^\star)$, and the last inequality uses the choice of κ.

We can then apply this to the expected regret to get that

$$\mathbb{E}[R_T] = \mathbb{E}[R_T \mathbb{I}\{\mathcal{E}\}] + \mathbb{E}[R_T \mathbb{I}\{\mathcal{E}^c\}]$$

$$\le R_T^{\text{osoco}} + GD + 2GDT\delta$$

$$\le R_T^{\text{osoco}} + GD + \frac{b_{\min}G}{S},$$

where we use the trivial bound on the regret given by

$$R_T = \sum_{t=1}^T f_t(x_t) - \sum_{t=1}^T f_t(x^\star) \le G \sum_{t=1}^T \|x_t - x^\star\| \le GDT.$$

Finally, we need to account for the choice of κ in the regret bound R_T^{osoco}. We do so by applying Theorem 1 with $b \leftarrow b - \kappa \mathbf{1}$ to get that

$$R_T^{\text{osoco}} = 4DG\frac{1}{b_{\min}-\kappa}\beta_T\sqrt{3dT\log(T)} + C_{\mathcal{A}}\sqrt{4dT\log(T)} + 2DG\sqrt{2T\log(1/\delta)}$$

$$= 4DG\frac{1}{b_{\min}(1-1/T)}\beta_T\sqrt{3dT\log(T)} + C_{\mathcal{A}}\sqrt{4dT\log(T)} + 2DG\sqrt{2T\log(1/\delta)}$$

$$\le 6DG\frac{1}{b_{\min}}\beta_T\sqrt{3dT\log(T)} + C_{\mathcal{A}}\sqrt{4dT\log(T)} + 2DG\sqrt{2T\log(1/\delta)},$$

where we use $T \ge 3$. $\qquad\square$

D   Proof of Propositions 2 (Improved Efficiency)

We first prove the regret guarantees of HedgeDescent in Section D.1 and then give the proof of Proposition 2 in Section D.2.

D.1 Regret of HedgeDescent

We give the regret guarantees of HedgeDescent (Algorithm 2) in the following proposition.

Proposition 4. *Consider the online optimization setting described in Definition 1 except with action set $\bar{\mathcal{X}} = \bigcup_{m \in [M]} \mathcal{X}_m$ where each $\mathcal{X}_m$ is convex. If HedgeDescent (Algorithm 2) is played in this setting with $\zeta_t = \sqrt{4 \log(M)}/GD\sqrt{t}$ and $\eta_t = D/G\sqrt{t}$, then it holds that*

$$\sum_{t=1}^{T} \mathbb{E}_{m_t \sim p_t}[f_t(x_t(m_t))] - \min_{x \in \bar{\mathcal{X}}} \sum_{t=1}^{T} f_t(x) \leq DG\sqrt{T \log(M)} + 3DG\sqrt{T}.$$

Proof. First, note that we can apply the standard regret guarantees of online gradient descent (e.g. Theorem 3.1 Hazan et al. (2016)) to get that the regret of each expert $m \in [M]$ is

$$\sum_{t=1}^{T} f_t(x_t(m)) - \sum_{t=1}^{T} f_t(x_m^\star) \leq 3DG\sqrt{T},$$

where $x_m^\star = \arg\min_{x \in \mathcal{X}_m} \sum_{t=1}^{T} f_t(x)$. Then, we cast the problem of choosing m as an expert advice problem. To put this into the standard form, we shift and scale the losses by defining the surrogate losses $\ell_t(m) = \frac{f_t(x_t(m)) - \bar{f}_t}{GD}$ where $\bar{f}_t = \min_{y \in \bar{\mathcal{X}}} f_t(y)$. This ensures that the $\ell_t(m) \in [0, 1]$. Then, the Hedge update becomes

$$\begin{aligned} p_{t+1}(m) &\propto p_t(m) \exp(-\zeta_t f_t(x_t(m))) \\ &= p_t(m) \exp(-\zeta_t(GD\ell_t(m) + \bar{f}_t)) \\ &= p_t(m) \exp(-\zeta_t \bar{f}_t) \exp(-\zeta_t GD\ell_t(m)) \\ &\propto p_t(m) \exp(-(\sqrt{4 \log(M)}/\sqrt{t})\ell_t(m)). \end{aligned}$$

Note that this update is exactly the form of the classical Hedge algorithm with step size $\bar{\zeta}_t = \sqrt{4 \log(M)}/\sqrt{t}$. As such, using the standard analysis of Hedge with time-dependent step size, e.g. Chernov and Zhdanov (2010), it holds that

$$\sum_{t=1}^{T} \mathbb{E}_{m_t \sim p_t}[\ell_t(m_t)] - \sum_{t=1}^{T} \ell_t(m^\star) \leq \sqrt{T \log(M)},$$

where $m^\star$ is such that $x^\star \in \mathcal{X}_{m^\star}$. Putting everything together, we get that

$$\begin{aligned} &\sum_{t=1}^{T} \mathbb{E}_{m_t \sim p_t}[f_t(x_t(m_t))] - \sum_{t=1}^{T} f_t(x^\star) \\ &= \sum_{t=1}^{T} \mathbb{E}_{m_t \sim p_t}[f_t(x_t(m_t))] - \sum_{t=1}^{T} f_t(x_t(m^\star)) + \sum_{t=1}^{T} f_t(x_t(m^\star)) - \sum_{t=1}^{T} f_t(x^\star) \\ &= DG \sum_{t=1}^{T} \mathbb{E}_{m_t \sim p_t}[\ell_t(m_t)] - DG \sum_{t=1}^{T} \ell_t(m^\star) + \sum_{t=1}^{T} f_t(x_t(m^\star)) - \sum_{t=1}^{T} f_t(x^\star) \\ &\leq DG\sqrt{T \log(M)} + 3DG\sqrt{T}, \end{aligned}$$

where we use the fact that $\sum_{t=1}^{T} f_t(x^\star) = \sum_{t=1}^{T} f_t(x_{m^\star}^\star)$. $\square$

D.2 Completing the proof

In the following, we restate and prove Proposition 2.

Proposition 5 (Duplicate of Proposition 2). *If OSOCO is modified such that $\mathcal{Y}_j^o \leftarrow \{\tilde{\mathcal{Y}}_j^o(k, \xi)\}_{k, \xi}$ and HedgeDescent is used for $\mathcal{A}$, then the following hold:*

- *(Static constraints) Choosing the algorithm parameters as in Theorem 1 ensures, with probability at least $1 - 2\delta$, that $Ax_t \leq b$ and $R_T \leq \tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$.*

- (Time-varying constraints) *Choosing the algorithm parameters as in Proposition 1 ensures that* $\mathbb{E}[y_t] \leq b$ *and* $\mathbb{E}[R_T] \leq \tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$.

Proof. First note that, following the same process as Lemma 4, it holds that $\gamma_t \geq 1 - \frac{1}{b_{\min}}\sqrt{d}\bar{\beta}_{j_t}\|\bar{V}_{j_t}^{-1/2}x_t\|_\infty$. Then, using the equivalence of norms, it follows that

$$\gamma_t \geq 1 - \frac{1}{b_{\min}}\sqrt{d}\bar{\beta}_{j_t}\|\bar{V}_{j_t}^{-1/2}x_t\|_\infty \geq 1 - \frac{1}{b_{\min}}\sqrt{d}\bar{\beta}_{j_t}\|x_t\|_{\bar{V}_{j_t}^{-1}}$$

Then, following the rest of the proof of Theorem 1 gives that, under the conditions of Theorem 1,

$$R_T \leq 8DG\frac{1}{b_{\min}}\beta_T d\sqrt{2T\log(T)} + C_{\mathcal{A}}\sqrt{2dT\log(T)} + 2DG\sqrt{2T\log(1/\delta)},$$

and $Ax_t \leq b$ $\forall t$ with probability at least $1 - 2\delta$. Then, following the proof of Proposition 1, it holds under the conditions of Proposition 1 that

$$\mathbb{E}[R_T] \leq 8DG\frac{1}{b_{\min}}\beta_T d\sqrt{2T\log(T)} + C_{\mathcal{A}}\sqrt{2dT\log(T)} + 2DG\sqrt{2T\log(1/\delta)} + GD + \frac{b_{\min}G}{S},$$

and $\mathbb{E}[y_t] \leq b$ $\forall t$. Finally, applying the regret guarantees of HedgeDescent by taking $C_{\mathcal{A}} = DG\sqrt{\log(2d)} + 3DG$ gives the result. $\qquad\square$

E   Proof of Corollary 1 (Stochastic Constraints)

In this section, we restate and prove Corollary 1.

Corollary 2. *Suppose that the cost functions and action set satisfy Assumptions 1 and 2. Also, assume that the constraint function satisfies Assumption 6, and let $\rho^2 = (G_g D + 2F)^2$. Then playing OSOCO (Algorithm 1) with the algorithm parameters chosen as in Proposition 1 ensures that $\mathbb{E}[g_t(x_t)] \leq 0$ for all $t \in [T]$ and $\mathbb{E}[R_T^{\mathrm{stoch}}] = \tilde{\mathcal{O}}((d + C_{\mathcal{A}}\sqrt{d})\sqrt{T})$. Furthermore, with modifications in Proposition 2, it holds that $\mathbb{E}[R_T^{\mathrm{stoch}}] = \tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$.*

Proof. To show the claim, we need to show that the setting of OCO with stochastic constraints is subsumed by OCO with static constraints and no violation in expectation, and the guarantees will follow immediately from Proposition 1. We do so by taking $g_t(x_t) = y_t$, $A := \mathbb{E}[A_t]$, $b := \mathbb{E}[b_t]$ and $\epsilon_t = (A_t - A)x_t - b_t + b$.

First, let $\mathcal{F}_t = \sigma(x_1, \epsilon_1, ..., \epsilon_{t-1}, x_t)$. Also, we restate Assumption 6 in terms of A_t and b_t:

   (i) $\|a_{t,i}\| \leq G_g$ for all $i \in [n]$,

  (ii) $\|b_t\| \leq F$,

 (iii) $b > 0$,

  (iv) b is known.

First, we show how the constraint $g_t(x_t) \leq 0$ is subsumed by the model in Section 2. Indeed,

$$y_t = g_t(x_t) = A_t x_t - b_t = Ax_t - b + \underbrace{(A_t - A)x_t - b_t + b}_{\epsilon_t}.$$

It remains to show that Assumption 4 holds, i.e. that $\mathbb{E}[\epsilon_{t,i}|\mathcal{F}_t] = 0$ and $\mathbb{E}[\exp(\lambda\epsilon_{t,i})|\mathcal{F}_t] \leq \exp(\frac{\lambda^2\rho^2}{2})$ $\forall\lambda \in \mathbb{R}$. Indeed,

$$\mathbb{E}[\epsilon_{t,i}|\mathcal{F}_t] = \mathbb{E}[(a_{t,i} - a_i)^\top x_t|\mathcal{F}_t] - \mathbb{E}[b_{t,i}|\mathcal{F}_t] + b_i = \mathbb{E}[(a_{t,i} - a_i)^\top|\mathcal{F}_t]x_t - \mathbb{E}[b_{t,i}] + b_i = \mathbb{E}[(a_{t,i} - a_i)^\top]x_t = 0,$$

where the second equality uses that x_t is $\mathcal{F}_t$-measurable and that b_t is independent of $\mathcal{F}_t$, and the third equality uses that A_t is independent of $\mathcal{F}_t$. Also, it holds almost surely that,

$$|\epsilon_{t,i}| = |(a_{t,i} - a_i)^\top x_t - b_{t,i} + b_i|$$

$$\leq (\|a_{t,i}\| + \|a_i\|)\|x_t\| + |b_{t,i}| + |b_i|$$
$$= (\|a_{t,i}\| + \|\mathbb{E}[a_{t,i}]\|)\|x_t\| + |b_{t,i}| + |\mathbb{E}[b_{t,i}]|$$
$$\leq (\|a_{t,i}\| + \mathbb{E}[\|a_{t,i}\|])\|x_t\| + |b_{t,i}| + \mathbb{E}[|b_{t,i}|]$$
$$\leq G_g D + 2F.$$

Therefore, it follows from the Hoeffding Lemma that,

$$\mathbb{E}[\exp(\lambda\epsilon_{t,i})|\mathcal{F}_t] \leq \exp\left(\frac{\lambda^2(G_g D + 2F)^2}{2}\right),$$

and therefore we can take $\rho^2 = (G_g D + 2F)^2$. Therefore, the constraint $g_t(x_t) \leq 0$ is subsumed by the model in Section 2.

Finally, note that,

$$\mathbb{E}[g_t(x_t)] = \mathbb{E}[Ax_t] - b + \mathbb{E}[\epsilon_t] = \mathbb{E}[Ax_t] - b + \mathbb{E}[\mathbb{E}[\epsilon_t|\mathcal{F}_t]] = \mathbb{E}[Ax_t] - b.$$

Therefore, the in-expectation guarantees in Proposition 1, $\mathbb{E}[Ax_t] \leq b$ imply that $\mathbb{E}[g_t(x_t)] \leq 0$. At the same time, it holds that $\bar{g}(x) = \mathbb{E}[A_t]x - \mathbb{E}[b_t] = Ax - b$ and therefore the regret notions are identical, i.e. $R_T = R_T^{\text{stoch}}$. $\square$

F   Example OCO Algorithms

In this section, we give examples of algorithms that can be used for the OCO algorithm $\mathcal{A}$ in OSOCO, i.e. those that admit the type of regret bound specified in Definition 1.

F.1   Problem setting

We consider a setting where at each round t, the player chooses an action $x_t \in \mathcal{X} \subseteq \mathbb{R}^d$ and an adversary simultaneously chooses a cost function f_t given the actions played in previous rounds. The player then observes f_t and suffers the cost $f_t(x_t)$. The game can be terminated at any time and therefore the final round T is unknown to the player. In the algorithms below, we use the flag alive to indicate that the game has not been terminated. Also, $\mathcal{X}$ is not necessarily convex. We assume that all f_t are differentiable and $\|\nabla f_t(x)\| \leq G$ for all $x \in \mathcal{X}$. We also assume that $\|x\| \leq D/2$ for all $x \in \mathcal{X}$.

F.2   Example Algorithm 1: Anytime Hedge over a Finite Cover

The simplest algorithm to handle this problem is Hedge played over a finite cover with the doubling trick. We give pseudocode for this algorithm in Algorithm 3. This is a well-known extension of the Hedge algorithm Freund and Schapire (1997) that has been noted by works such as Dani et al. (2007), Maillard and Munos (2010) and Krichene et al. (2015).

Algorithm 3 Anytime Hedge over a Finite Cover

input $\mathcal{X}$.
1: Initialize: $t = 1$, $j = 0$, alive = True.
2: **while** alive **do**
3:    Construct Δ_j-net over $\mathcal{X}$, i.e. choose points $\{z_1, ..., z_{M_j}\}$ such that $\min_{m\in[M_j]} \|x - z_m\| \leq \Delta_j$ for all $x \in \mathcal{X}$.
4:    Set distribution p_t over $[M_j]$ to be uniform.
5:    **while** $t \leq \tau_{j+1} - 1$ **and** alive **do**
6:       Sample $m_t \sim p_t$.
7:       Play $x_t = z_{m_t}$ and observe f_t.
8:       Update $p_{t+1}(m) \propto \exp(-\eta_j f_t(z_m))$.
9:       $t = t + 1$.
10:    **end while**
11:    $j = j + 1$
12: **end while**

Then, we give the regret guarantees of Algorithm 3.

Proposition 6. *Consider Algorithm 3 with $\Delta_j = 1/(G\sqrt{\tau_j})$ and $\eta_j = \sqrt{2\log(M_j)/\tau_j}/GD$. Then, it holds that*

$$\sum_{t=1}^{T} \mathbb{E}_{x_t \sim p_t}[f_t(x_t)] - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} f_t(x) \leq 5\left(GD\sqrt{4d\log(2DG\sqrt{dT})} + 2\right)\sqrt{T}$$

Proof. Let $x^\star$ be the optimal action over the entire time horizon, $x_j^\star$ be the optimal action within phase j and $m_j^\star$ be the index of the nearest net point. Also, let τ_j be the duration of phase j and $j = N$ be the index of the last phase. To put the update into the standard form for Hedge, we shift and scale the losses by defining the surrogate losses $\ell_t(m) = \frac{f_t(z_m) - \bar{f}_t}{GD}$ where $\bar{f}_t = \min_{y \in \mathcal{X}} f_t(y)$. This ensures that the $\ell_t(m) \in [0, 1]$. Then, the Hedge update becomes

$$\begin{aligned}
p_{t+1}(m) &\propto p_t(m)\exp(-\eta_j f_t(z_m)) \\
&= p_t(m)\exp(-\eta_j(GD\ell_t(m) + \bar{f}_t)) \\
&= p_t(m)\exp(-\eta_j \bar{f}_t)\exp(-\eta_j GD\ell_t(m)) \\
&\propto p_t(m)\exp(-\eta_j GD\ell_t(m)) \\
&= p_t(m)\exp(-(\sqrt{2\log(M_j)/\tau_j})\ell_t(m)).
\end{aligned}$$

This is equivalent to the classical analysis of Hedge Arora et al. (2012) with step size $\eta_j' = \sqrt{2\log(M_j)/\tau_j}$ and therefore, for any phase $j \in [0, N]$,

$$\begin{aligned}
&\sum_{t=\tau_j}^{\tau_{j+1}-1} \mathbb{E}_{x_t \sim p_t}[f_t(x_t)] - \min_{m \in M_j} \sum_{t=\tau_j}^{\tau_{j+1}-1} f_t(z_m) \\
&= GD\sum_{t=\tau_j}^{\tau_{j+1}-1} \mathbb{E}_{x_t \sim p_t}[\ell_t(x_t)] - GD\min_{m \in M_j} \sum_{t=\tau_j}^{\tau_{j+1}-1} \ell_t(z_m) \\
&\leq GD\sqrt{4\tau_j \log(M_j)},
\end{aligned} \tag{8}$$

Note that if the game is terminated in the midst of the final round, there will not be cost functions for the remaining rounds. However, we can define the "fictitious" cost functions $f_{T+1}, ..., f_{\tau_{N+1}-1}$ that extend from the end of the game to the end of the final phase. This is why we have the regret guarantee for all phases as defined in (8).

Also, for any $x \in \mathcal{X}$, there exists $z \in \{z_1, ..., z_{M_j}\}$ such that

$$f_t(x) - f_t(z) \leq G\|x - z\| \leq G\Delta_j$$

Then, we get that within a given phase j, it holds that

$$\begin{aligned}
&\sum_{t=\tau_j}^{\tau_{j+1}-1} \mathbb{E}_{x_t \sim p_t}[f_t(x_t)] - \sum_{t=\tau_j}^{\tau_{j+1}-1} f_t(x_j^\star) \\
&\leq \sum_{t=\tau_j}^{\tau_{j+1}-1} \mathbb{E}_{x_t \sim p_t}[f_t(x_t)] - \sum_{t=\tau_j}^{\tau_{j+1}-1} f_t(z_{m_j^\star}) + \sum_{t=\tau_j}^{\tau_{j+1}-1} f_t(z_{m_j^\star}) - \sum_{t=\tau_j}^{\tau_{j+1}-1} f_t(x_j^\star) \\
&\leq GD\sqrt{4\tau_j \log(M_j)} + G\Delta_j \tau_j
\end{aligned}$$

Furthermore, it is known (e.g. Example 27.1 in Shalev-Shwartz and Ben-David (2014)) that we can construct a grid for the net to get $M_j \leq (2D\sqrt{d}/\Delta_j)^d$ for desired Δ_j. Therefore, choosing $\Delta_j = 1/(G\sqrt{\tau_j})$ will need $M_j \leq (2DG\sqrt{d\tau_j})^d$ points. Also, note that N is the smallest integer such that $\sum_{j=0}^{N} 2^j = 2^{N+1} - 1 \geq T$ and therefore $N = \lceil \log_2(T+1) - 1\rceil \leq \log_2(T+1)$. Therefore, it holds that

$$\sum_{j=0}^{N} \sqrt{\tau_j} = \sum_{j=0}^{N} \sqrt{2^j} = \sum_{j=0}^{N} \sqrt{2}^j = \frac{1 - \sqrt{2}^{N+1}}{1 - \sqrt{2}} = \frac{\sqrt{2^{N+1}} - 1}{\sqrt{2} - 1} \leq 5\sqrt{T} \tag{9}$$

Putting everything together, we get that

$$
\begin{aligned}
\sum_{t=1}^{T} \mathbb{E}_{x_t \sim p_t}[f_t(x_t)] - \sum_{t=1}^{T} f_t(x^\star) &= \sum_{j=0}^{N} \sum_{t=\tau_j}^{\tau_{j+1}-1} \mathbb{E}_{x_t \sim p_t}[f_t(x_t)] - \sum_{j=0}^{N} \sum_{t=\tau_j}^{\tau_{j+1}-1} f_t(x^\star) \\
&\leq \sum_{j=0}^{N} \sum_{t=\tau_j}^{\tau_{j+1}-1} \mathbb{E}_{x_t \sim p_t}[f_t(x_t)] - \sum_{j=0}^{N} \sum_{t=\tau_j}^{\tau_{j+1}-1} f_t(x_j^\star) \\
&\leq \sum_{j=0}^{N} \left(GD\sqrt{4\tau_j \log(M_j)} + 2G\Delta_j \tau_j\right) \\
&= \sum_{j=0}^{N} \left(GD\sqrt{4d\tau_j \log(2DG\sqrt{d\tau_j})} + 2\sqrt{\tau_j}\right) \\
&\leq \left(GD\sqrt{4d\log(2DG\sqrt{dT})} + 2\right) \sum_{j=0}^{N} \sqrt{\tau_j} \\
&\leq 5\left(GD\sqrt{4d\log(2DG\sqrt{dT})} + 2\right) \sqrt{T}
\end{aligned}
$$

completing the proof. □

F.3   Example Algorithm 2: Anytime Follow the Perturbed Leader

Another algorithm that can be used for this setting is Follow the Perturbed Leader (FTPL) Kalai and Vempala (2005) which has been popular for a variety of OCO settings. We use the version from Suggala and Netrapalli (2020) with an exact optimization oracle. We use the version and analysis from Suggala and Netrapalli (2020) because they study it in the nonconvex setting, which is directly applicable to our setting because the action set is nonconvex. We give the FTPL algorithm with the doubling trick in Algorithm 4.

Algorithm 4 Anytime Follow the Perturbed Leader

---

input $\mathcal{X}$.
 1: Initialize: $t = 1$, $j = 0$, alive = True, $\tau_j := 2^j$.
 2: **while** alive **do**
 3:     **while** $t \leq \tau_{j+1} - 1$ **and** alive **do**
 4:         Let $\sigma_t = [\sigma_{t,1} \, ... \, \sigma_{t,d}]^\top$ where $\sigma_{t,i} \sim \mathrm{Exp}(\eta_j)$.
 5:         Choose $x_t \in \arg\min_{x \in \mathcal{X}} \left(\sum_{s=1}^{t-1} f_t(x) - \sigma_t^\top x\right)$
 6:         Play x_t and observe f_t.
 7:         $t = t + 1$.
 8:     **end while**
 9:     $j = j + 1$
10: **end while**

We then give the regret guarantees of this algorithm in the following.

Proposition 7. *Consider Algorithm 4 with $\eta_j = 1/\sqrt{2dG^2\tau_j}$. Then, it holds that*

$$
\sum_{t=1}^{T} \mathbb{E}_{x_t \sim p_t}[f_t(x_t)] - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} f_t(x) \leq 630 DGd^{3/2}\sqrt{2T},
$$

where p_t is the distribution over the action round t.

Proof. We will use the analysis from Suggala and Netrapalli (2020) to bound the regret within each phase. First note that Suggala and Netrapalli (2020) assumes that $\|x - y\|_\infty \leq D'$ for all $x, y \in \mathcal{X}$ and that f_t are L'-Lipschitz w.r.t. the 1-norm. In our setting, it holds that for all $x, y \in \mathcal{X}$ that

$$
\|x - y\|_\infty \leq \|x - y\| = \|x - \mathbf{0} + \mathbf{0} - y\| \leq \|x - \mathbf{0}\| + \|\mathbf{0} - y\| \leq D
$$

and that

$$|f_t(x) - f_t(y)| \leq G\|x - y\| \leq G\|x - y\|_1.$$

Therefore, we can take $D' = D$ and $L' = G$. Also, Suggala and Netrapalli (2020) assume access to a (α, β) inexact oracle, but we assume that this oracle is exact and therefore take $\alpha, \beta = 0$. Then, since the regret guarantees of FTPL hold against a non-oblivious adversary (see Theorem 8 in Suggala and Netrapalli (2020)), we can apply the regret guarantees of FTPL (Theorem 1 in Suggala and Netrapalli (2020)) to get that,

$$\sum_{t=\tau_j}^{\tau_{j+1}-1} \mathbb{E}_{x_t \sim p_t}[f_t(x_t)] - \min_{x \in \mathcal{X}} \sum_{t=\tau_j}^{\tau_{j+1}-1} f_t(x) \leq 125\eta_j (L')^2 d^2 D' 2\tau_j + \frac{dD'}{\eta_j}$$

$$= 125\eta_j G^2 d^2 D 2\tau_j + \frac{dD}{\eta_j}$$

$$= 126 d^{3/2} DG \sqrt{2\tau_j}$$

where we use $\eta_j = 1/\sqrt{2dG^2\tau_j}$. Then, putting everything together and letting $x^\star = \arg\min_{x \in \mathcal{X}} \sum_{t=1}^T f_t(x)$, we get that

$$\sum_{t=1}^T \mathbb{E}_{x_t \sim p_t}[f_t(x_t)] - \sum_{t=1}^T f_t(x^\star) = \sum_{j=0}^N \sum_{t=\tau_j}^{\tau_{j+1}-1} \mathbb{E}_{x_t \sim p_t}[f_t(x_t)] - \sum_{j=0}^N \sum_{t=\tau_j}^{\tau_{j+1}-1} f_t(x^\star)$$

$$\leq \sum_{j=0}^{N-1} \sum_{t=\tau_j}^{\tau_{j+1}-1} \mathbb{E}_{x_t \sim p_t}[f_t(x_t)] - \sum_{j=0}^N \sum_{t=\tau_j}^{\tau_{j+1}-1} f_t(x_j^\star)$$

$$\leq \sum_{j=0}^N 126 DGd^{3/2} \sqrt{2\tau_j}$$

$$\leq 630 DGd^{3/2} \sqrt{2T},$$

where we use (9) in the last inequality. □

G Details on Numerical Experiments

In this appendix, we provide details on the numerical experiments from Section 7 and additional results.

G.1 Setting Description

The setting is defined by the following.

- $f_t(x) = 3\|x - v_t\|^2$

- $g(x) = Ax - b$ (Static Constraint)

- $g_t(x) = Ax - b_t$ (Stochastic Constraint)

- $d = 2$

- $\mathcal{X} = \mathbb{B}$

- $b = 0.8$

- $A = [-1 \ -1]$

- $v_t \sim [-1, 0]^d$

- $\epsilon_t \sim \mathcal{N}(0, \rho I)$

- $b_t = b + \epsilon_t$

- $\rho = 0.01$
- $D = 2$
- $S = \sqrt{2}$
- $G = 6\sqrt{2} + 6$
- $b_{\min} = 0.8$

G.2 Algorithms Description

We consider OSOCO configured for high-probability (OSOCO-H) and OSOCO configured for expectation (OSOCO-E) alongside the algorithm from Yu et al. (2017) (which we call DPP for "drift-plus-penalty") and SO-PGD from Chaudhary and Kalathil (2022). For DPP, we consider both the nominal choice of algorithm parameters as well as a conservative choice of algorithm parameters that are chosen to keep the violation very small (referred to as DPP(cons.)). The choice of algorithm parameters for all algorithms is given in Table 2. Both OSOCO and SO-PGD require a projection in each round, which we implement with CVXPY. DPP only requires a projection on to the action set $\mathcal{X}$, which is the 2-norm ball in this setting and therefore the projection can be computed by scaling according to the norm of the point.

Table 2: Choice of algorithm parameters for simulations.

Algorithm	Version	Parameters
OSOCO	E	$\mathcal{A} = \text{HedgeDescent}$ $\eta_t = D/(G\sqrt{t})$ (in HedgeDescent) $\zeta_t = \sqrt{4\log(2d)}/(GD\sqrt{t})$ (in HedgeDescent) $\lambda = \max(1, D^2)$ $\delta = \min(1/2, b_{\min}/(2SDT))$ $\beta_t = \rho\sqrt{d\log\left(\frac{1+(t-1)D^2/\lambda}{\delta/n}\right)} + \sqrt{\lambda}S$ $\kappa = b_{\min}/T$
	H	$\mathcal{A} = \text{HedgeDescent}$ $\eta_t = D/(G\sqrt{t})$ (in HedgeDescent) $\zeta_t = \sqrt{4\log(2d)}/(GD\sqrt{t})$ (in HedgeDescent) $\lambda = \max(1, D^2)$ $\delta = 0.01$ $\beta_t = \rho\sqrt{d\log\left(\frac{1+(t-1)D^2/\lambda}{\delta/n}\right)} + \sqrt{\lambda}S$ $\kappa = 0$
DPP	-	$\alpha = T$ $V = \sqrt{T}$
	(cons.)	$\alpha = T$ $V = 0.01\sqrt{T}$
SO-PGD	-	$\lambda = 1$ $\delta = 0.01$ $\eta = D/(G\sqrt{T})$ $T_0 = T^{2/3}$

G.3 Results Description

We study the regret, the stochastic constraint violation and the static constraint violation as reported in Figure 2a, Figure 2b and Figure 2c, respectively. All figures show the results for 30 trials, where the average is indicated

by a point and the standard deviation is indicated by an errorbar or shading. These results indicate that the nominal DPP algorithm has lower regret than any other algorithm, although it also incurs significant violation of both static and stochastic constraints. When the algorithm parameters for DPP are chosen to ensure very little violation, the regret becomes substantial. The algorithms with the next lowest regret are OSOCO-T and OSOCO-S, which do not incur any stochastic constraint violation (on average) or static constraint violation.

G.4 Additional Results

In addition to the main results in the body of the paper, we also provide additional results studying the performance for larger dimensions (beyond just 2) as well as the computation time. These additional results are only given for SO-PGD and OSOCO-H because they are applicable to the exact same setting (no violation in high probability) and therefore allow for direct comparison.

G.4.1 Higher Dimensions

In order to study higher dimensions, we extend the setting described in Section G.1 by taking $v_t \sim [-\sqrt{\frac{2}{d}}, 0]^d$ and $A = -\sqrt{\frac{2}{d}}\mathbf{1}^\top$. The reason that we scale the parameters as such is that it ensures that the parameters $S = \sqrt{2}$ and $G = 6\sqrt{2} + 6$ are valid irrespective of dimension, i.e. $\|A\| \leq S$ and $\|\nabla f(x)\| \leq G$ for any d. This is necesary to ensure that the regret bound is $\tilde{\mathcal{O}}(dT^{2/3})$ for SO-PGD and $\tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$ for OSOCO. In this setting, the regret of OSOCO and SO-PGD is shown in Figure 3 for $T = 1000$ with varying dimension d. This figure shows the aggregate over 30 trials where the markers indicate the mean and the errorbars indicate the standard deviation.
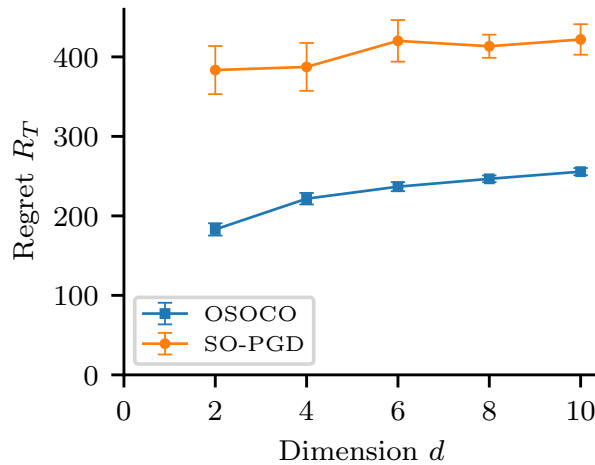

Figure 3: Regret of OSOCO and SO-PGD for $T = 1000$ with varying problem dimension. The markers indicate the average of 30 trials and the errorbars indicate the standard deviation.

G.4.2 Computation Time

We study the computation time for the original 2-dimensional setting described in Section G.1. Specifically, Figure 4 shows the average and standard deviation of the computation time for $T = 1000$. Note that we use CVXPY with the default settings to compute the projection for the algorithms. Therefore, our results do not rule out the possibility for faster implementations of either algorithm.
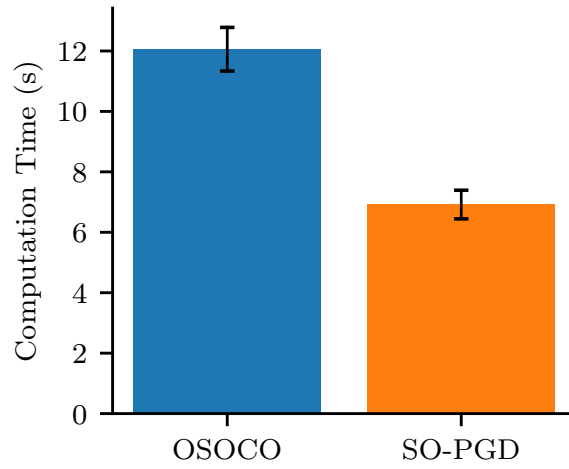
Figure 4: Computation time of OSOCO and SO-PGD for $T = 1000$. The bars indicate the average over 30 trials and the errorbars indicate the standard deviation.