Optimal Algorithms for Online Convex Optimization with Adversarial Constraints

Abhishek Sinha, Rahul Vaze
School of Technology and Computer Science
Tata Institute of Fundamental Research
Mumbai 400005, India
abhishek.sinha@tifr.res.in, rahul.vaze@gmail.com

Abstract

A well-studied generalization of the standard online convex optimization (OCO) framework is constrained online convex optimization (COCO). In COCO, on every round, a convex cost function and a convex constraint function are revealed to the learner after it chooses the action for that round. The objective is to design an online learning policy that simultaneously achieves a small regret while ensuring a small cumulative constraint violation (CCV) against an adaptive adversary interacting over a horizon of length T. A long-standing open question in COCO is whether an online policy can simultaneously achieve $O(\sqrt{T})$ regret and $\tilde{O}(\sqrt{T})$ CCV without any restrictive assumptions. For the first time, we answer this in the affirmative and show that a simple first-order policy can simultaneously achieve these bounds. Furthermore, in the case of strongly convex cost and convex constraint functions, the regret guarantee can be improved to $O(\log T)$ while keeping the CCV bound the same as above. We establish these results by effectively combining adaptive OCO policies as a blackbox with Lyapunov optimization - a classic tool from control theory. Surprisingly, the analysis is short and elegant.

1 Introduction

Online convex optimization (OCO) is a standard framework for modelling and analyzing a broad family of online decision problems under uncertainty. In the OCO problem, on every round t, an online policy first selects an action x_t from a closed and convex admissible set (*a.k.a.* decision set) $\mathcal{X}$. Then the adversary reveals a convex cost function f_t, resulting in a cost of $f_t(x_t)$. The goal of an online policy is to choose an admissible action sequence $\{x_t\}_{t=1}^T$ so that its cumulative cost is not much larger than that of any fixed admissible action chosen in hindsight. In particular, the objective is to minimize the static regret defined below

$$\text{Regret}_T \equiv \sup_{\{f_t\}_{t=1}^T} \sup_{x^\star \in \mathcal{X}} \text{Regret}_T(x^\star), \text{ where } \text{Regret}_T(x^\star) \equiv \sum_{t=1}^T f_t(x_t) - \sum_{t=1}^T f_t(x^\star). \tag{1}$$

The term *static* refers to using a fixed benchmark, specifically only one action $x^\star$ throughout the horizon of length T.

In this paper, we consider a generalization of the standard OCO framework. In this problem, on every round t, the online policy first chooses an admissible action $x_t \in \mathcal{X}$, and then the adversary chooses a convex cost function $f_t : \mathcal{X} \to \mathbb{R}$ and k constraints of the form $g_{t,i}(x) \leq 0,\ i \in [k]$, where $g_{t,i} : \mathcal{X} \to \mathbb{R}$ is a convex function for each $i \in [k]$[1]. Since $g_{t,i}$'s are revealed after the action x_t is

[1]Notations: For any natural number n, we define $[n] \equiv \{1, 2, \ldots, n\}$. For any real number z, we define $(z)^+ \equiv \max(0, z)$.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-1306

chosen, an online policy need not necessarily take feasible actions on each round, and the obvious metric of interest in addition to (1) is the total cumulative constraint violation (CCV) $\mathbb{V}(T)$ defined as

$$\text{CCV}_T \equiv \mathbb{V}(T) = \max_{i=1}^{k} \mathbb{V}_i(T) \quad \text{where} \quad \mathbb{V}_i(T) = \sum_{t=1}^{T} (g_{t,i}(x_t))^+. \tag{2}$$

Let $\mathcal{X}^\star$ be the feasible set consisting of all admissible actions that satisfy all constraints $g_{t,i}(x) \leq 0,\ i \in [k], t \in [T]$. Under the standard assumption that $\mathcal{X}^\star$ is not empty, the goal is to design an online policy to simultaneously achieve a small regret (1) with respect to any admissible benchmark $x^\star \in \mathcal{X}^\star$ and a small CCV (2). We refer to this problem as the constrained OCO (COCO). The assumption $\mathcal{X}^\star \neq \varnothing$ will be relaxed in Section 3 for the Online Constraint Satisfaction (OCS) problem where the cost functions are set to zero, and the objective is to minimize just the CCV.

COCO arises in many applications, including online portfolio optimization with risk constraints, resource allocation in cloud computing with time-varying demands, pay-per-click online ad markets with budget constraints [Liakopoulos et al., 2019], online recommendation systems, dynamic pricing, revenue management, robotics and path planning problems, and multi-armed bandits with fairness constraints [Sinha, 2024a]. The necessity for revealing the constraints sequentially may also arise, *e.g.,* in communication-limited settings, where it might be infeasible to reveal all constraints defining the feasible set at a time (*e.g.,* combinatorial auctions). See Section 4 for an application of the COCO framework in fraud detection which involves binary classification with a highly-imbalanced dataset.

1.1 Related Work

Unconstrained OCO: In a seminal paper, Zinkevich [2003] showed that for solving (1), the ubiquitous projected online gradient descent (OGD) policy achieves an $O(\sqrt{T})$ regret for convex cost functions with uniformly bounded sub-gradients. A number of follow-up papers proposed adaptive and parameter-free versions of OGD [Hazan et al., 2007, Orabona and Pál, 2018]. See Orabona [2019], Hazan [2022] for textbook treatments of the OCO framework and associated algorithms.

Constrained OCO (COCO): (A) Time-invariant constraints: A number of papers considered COCO with time-invariant constraints, *i.e.,* $g_{t,i} = g_i, \forall\ t$ [Yuan and Lamperski, 2018, Jenatton et al., 2016, Mahdavi et al., 2012, Yi et al., 2021]. These works assume that the functions g_i's are known to the policy *a priori*. However, they allowed the policy to remain infeasible on any round to avoid the costly projection step of the vanilla projected OGD policy. Their main objective was to design an *efficient* policy (avoiding the explicit projection step) with a small regret and CCV.

(B) Time-varying constraints: Solving the COCO problem when the constraint functions, *i.e.*, $g_{t,i}$'s, change arbitrarily with time t is more challenging. In this case, except for Neely and Yu [2017] and Liakopoulos et al. [2019], most of the prior works construct some Lagrangian function and then update the primal and dual variables [Yu et al., 2017, Sun et al., 2017, Yi et al., 2023]. However, the performance bounds obtained with this approach remain suboptimal. Both Neely and Yu [2017] and Liakopoulos et al. [2019] use the drift-plus-penalty (DPP) framework introduced by Neely [2010] to solve the constrained problem under various assumptions. In particular, Neely and Yu [2017] proposed a DPP-based policy for COCO upon assuming the Slater's condition, *i.e.,* $g_{t,i}(x^\star) < -\eta$, for some $\eta > 0\ \forall i, t$. Clearly, this condition precludes the important case of non-negative constraint functions (*e.g.,* constraint functions of the form $\max(0, g_t(x))$). Furthermore, the bounds obtained upon assuming Slater's condition depend inversely with the Slater's constant η (usually hidden under the big-Oh notation). Since η could be arbitrarily small, these bounds could be arbitrarily loose. Liakopoulos et al. [2019] extended Neely and Yu [2017]'s result by considering a weaker form of the feasibility assumption without assuming Slater's condition. Furthermore, although these DPP-based results are interesting, they have not been able to provide improved regret or CVV bounds when the cost functions f_t's are strongly convex because of the linearization step inherent in this approach.

In a recent paper, Guo et al. [2022] considered COCO and obtained the best-known prior results without assuming Slater's condition. However, in addition to yielding sub-optimal bounds, their policy is quite computationally intensive since it requires solving a convex optimization problem on each round. Compared to this, all policies proposed in this paper take only a single gradient-descent step and perform only one Euclidean projection on each round. Please refer to Table 1 for a brief summary of the results and Section A.5 in the Appendix for a qualitative comparison. The COCO problem has been considered in the *dynamic* setting as well [Chen and Giannakis, 2018, Cao and Liu, 2018, Vaze, 2022, Liu et al., 2022] where the benchmark $x^\star$ in (1) is replaced by $x_t^\star$ that is also

Reference	Regret	CCV	Complexity per round	Assumptions
Mahdavi et al. [2012]	$O(\sqrt{T})$	$O(T^{3/4})$	Projection	Time-invariant constraints
Jenatton et al. [2016]	$O(T^{\max(\beta,1-\beta)})$	$O(T^{1-\beta/2})$	Projection	Time-invariant constraints
Sun et al. [2017]	$O(\sqrt{T})$	$O(T^{3/4})$	Bregman Projection	-
Neely and Yu [2017]	$O(\sqrt{T})$	$O(\sqrt{T})$	Conv-OPT	Slater condition
Yuan and Lamperski [2018]	$O(T^{\max(\beta,1-\beta)})$	$O(T^{1-\beta/2})$	Projection	Time-invariant constraints
Yu and Neely [2020]	$O(\sqrt{T})$	$O(1)$	Conv-OPT	Slater & Time-invariant constraints
Yi et al. [2021]	$O(T^{\max(\beta,1-\beta)})$	$O(T^{(1-\beta)/2})$	Conv-OPT	Time-invariant constraints
Yi et al. [2022]	$O(T^{\beta})$	$O(T^{1-\beta/2})$	Projection	Strongly convex cost
Guo et al. [2022]	$O(\sqrt{T})$	$O(T^{3/4})$	Conv-OPT	-
Guo et al. [2022]	$O(\log T)$	$O(\sqrt{T\log T})$	Conv-OPT	Strongly convex cost
Yi et al. [2023]	$O(T^{\max(\beta,1-\beta)})$	$O(T^{1-\beta/2})$	Conv-OPT	-
Yi et al. [2023]	$O(\log(T))$	$O(\sqrt{T\log T})$	Conv-OPT	Strongly convex cost
This paper	$O(\sqrt{T})$	$O(\sqrt{T}\log T)$	Projection	-
This paper	$O(\log T)$	$O(\sqrt{T\log T})$	Projection	Strongly convex cost
This paper	$O(\log T)$	$O(\frac{\log T}{\alpha})$	Projection	Strongly convex cost, $\text{Regret}_T \geq 0$,

Table 1: Summary of the results on COCO. Unless stated otherwise, we assume arbitrary time-varying convex constraints and convex cost functions. In the above table, $0 \leq \beta \leq 1$ is an adjustable parameter, α is the strong convexity parameter of the strongly convex cost functions. Conv-OPT refers to solving a constrained convex optimization problem on each round. Projection refers to the Euclidean projection operation on the convex set $\mathcal{X}$. For typical convex sets (*e.g.,* Euclidean box, probability simplex), projection operations are substantially more efficient than solving a constrained convex optimization problem.

allowed to change its actions over time. However, we focus our attention on achieving the optimal performance bounds for the static version. A special case of COCO is the ONLINE CONSTRAINT SATISFACTION (OCS) problem that does not involve any cost function, *i.e.,* $f_t = 0, \ \forall t$, and the only object of interest is the CCV. The OCS problem becomes especially interesting in the setting where the feasible set may be empty.

1.2 Our Contributions

In this paper, we consider both COCO and OCS problems and make the following contributions.

1. We propose an efficient first-order policy that simultaneously achieves $O(\sqrt{T})$ regret and $O(\sqrt{T}\log T)$ CCV for the COCO problem. Our result breaks the long-standing $O(T^{3/4})$ barrier for the CCV and matches the lower bound (derived in Theorem 3, previously missing from the literature) up to a logarithmic term. For strongly convex cost functions, the regret guarantee is improved to $O(\log T)$ while keeping the CCV bound the same as above. Under an additional assumption that the regret is non-negative, we obtain a further improved logarithmic CCV bound in the strongly convex setting (see Table 1).
2. We additionally consider a special case of the COCO problem, called Online Constraint Satisfaction (OCS), under relaxed feasibility assumptions and obtain sub-linear CCV bounds.
3. On the algorithmic side, our policy simply runs an adaptive first-order OCO algorithm as a blackbox on a specially constructed convex surrogate cost function sequence. On every round, the policy needs to compute only two gradients and an Euclidean projection. This is way more efficient compared to the policies proposed in the previous works [Guo et al., 2022, Neely and Yu, 2017], which need to solve expensive convex optimization problems on each round while yielding sub-optimal bounds. Furthermore, in the special case of time-invariant constraints, our results yield an efficient first-order OCO policy with competitive regret and CCV bounds [Mahdavi et al., 2012, Jenatton et al., 2016, Yi et al., 2021].
4. Our results are obtained by introducing a crisp and elegant potential function-based algorithmic technique for simultaneously controlling the regret and the CCV. In brief, the regret and CCV bounds are derived from a single inequality that arises from plugging in off-the-shelf adaptive regret bounds in a new regret decomposition result (Eqn. (6)). This new analytical technique might also be of independent interest.

https://doi.org/10.52202/079017-1306

5. Finally, in Section 4, we evaluate the practical performance of our algorithm in the online credit card fraud detection problem with a highly imbalanced dataset.

2 The Constrained OCO (COCO) Problem

2.1 Assumptions

We now state the assumptions considered in this paper. These assumptions are standard in literature on the COCO problem [Guo et al., 2022, Yi et al., 2021, Neely and Yu, 2017].

Assumption 1 (Convexity)**.** *The cost function $f_t : \mathcal{X} \mapsto \mathbb{R}$ and the constraint function $g_{t,i} : \mathcal{X} \mapsto \mathbb{R}$ are convex for all $t \geq 1, i \in [k]$. The admissible set (*a.k.a. *the decision set or the action set) $\mathcal{X} \subseteq \mathbb{R}^d$ is closed and convex and has a finite Euclidean diameter D.*

Assumption 2 (Lipschitzness)**.** *All cost functions $\{f_t\}_{t\geq 1}$ and the constraint functions $\{g_{t,i}\}_{i\in[k],t\geq 1}$'s are G-Lipschitz. In other words, for any $x, y \in \mathcal{X}$, we have*

$$|f_t(x) - f_t(y)| \leq G\|x - y\|,\ |g_{t,i}(x) - g_{t,i}(y)| \leq G\|x - y\|,\ \forall t \geq 1, i \in [k].$$

Unless specified otherwise, the norm $\|\cdot\|$ will refer to the standard Euclidean norm and ∇f will refer to an arbitrary subgradient of a convex function f. Assumption 2 implies that the ℓ_2-norm of the (sub)gradients of the cost and constraint functions are uniformly upper-bounded by G over the admissible set $\mathcal{X}$. Finally, we make the following feasibility assumption about the constraint functions.

Assumption 3 (Feasibility)**.** *There exists a feasible action $x^\star \in \mathcal{X}$ s.t. $g_{t,i}(x^\star) \leq 0, \forall t, i$. The feasible set $\mathcal{X}^\star$ is defined to be the set of all feasible actions. The feasibility assumption implies that $\mathcal{X}^\star \neq \varnothing$.*

The feasibility assumption distinguishes the cost functions from the constraint functions and is commonly assumed in the literature [Guo et al., 2022, Neely and Yu, 2017, Yu and Neely, 2016, Yuan and Lamperski, 2018, Yi et al., 2023, Liakopoulos et al., 2019]. In Section 3, we will consider a constraint-only variant of the problem where the feasibility assumption (Assumption 3) will be relaxed. See Appendix A.1 for a brief discussion on the assumptions.

Remarks: On each round, multiple constraints of the form $g_{t,i}(x) \leq 0, i \in [k]$ can be replaced by a single new constraint $g_t(x) \leq 0$ where the constraint function g_t is defined to be the pointwise maximum of the given constraints, *i.e.,* $g_t(x) \equiv \max_{i=1}^k g_{t,i}(x), x \in \mathcal{X}$. It is easy to verify that if each of the constraint functions $\{g_{t,i}\}_{i=1}^k$ satisfies the above assumptions, then the constraint function g_t defined above also satisfies the assumptions. Hence, throughout this section and without loss of generality, we will assume that only one constraint function is revealed on each round. That being said, under the relaxed feasibility assumption in Section 3, this trick does not work and there we will need to consider the full set of k constraint functions.

2.2 Online Policy for COCO

Recall that compared to the standard OCO problem where the only objective is to minimize the Regret [Hazan, 2022], in COCO, our objective is twofold: to *simultaneously* control the Regret *and* the CCV. See Section A.2 in the Appendix for preliminaries on the OCO problem and some standard results which will be useful in our analysis. In the following, we propose a Lyapunov function-based policy that yields the optimal Regret and CCV bounds for the COCO problem. Although for simplicity, we assume that the horizon length T is known, we can use the standard doubling trick for an unknown T.

2.3 Design and Analysis of the Algorithm

To simplify the analysis, we pre-process the cost and constraint functions on each round as follows.

Pre-processing: On every round, we first clip the negative part of the constraint function to zero by passing it through the standard ReLU unit. Then, we scale both the cost and constraint functions by a positive factor β, which will be determined later. In other words, we work with the pre-processed inputs $\tilde{f}_t \leftarrow \beta f_t, \tilde{g}_t \leftarrow \beta(g_t)^+$. Hence, the pre-processed functions are βG-Lipschitz and $\tilde{g}_t \geq 0, \forall t$.

In the following, we derive the Regret and CCV bounds for the pre-processed functions. The bounds for the original problem are obtained upon scaling the results back by β^{-1} in the final step.

Algorithm 1 Online Policy for COCO

Input: Sequence of convex cost functions $\{f_t\}_{t=1}^T$ and constraint functions $\{g_t\}_{t=1}^T$, G = a common Lipschitz constant, T = Horizon length, D = Euclidean diameter of the admissible set $\mathcal{X}$, $\mathcal{P}_{\mathcal{X}}(\cdot)$ = Euclidean projection operator on the set $\mathcal{X}$

Parameter settings:

1. **Convex cost functions:** $\beta = (2GD)^{-1}, V = 1, \lambda = \frac{1}{2\sqrt{T}}, \Phi(x) = \exp(\lambda x) - 1.$
2. **α-strongly convex cost functions:** $\beta = 1, V = \frac{8G^2 \ln(Te)}{\alpha}, \Phi(x) = x^2.$

Initialization: Set $x_1 \in \mathcal{X}$ arbitrarily, $Q(0) = 0$.

for each $t = 1 : T$ **do**

  Play x_t, observe f_t, g_t, incur a cost of $f_t(x_t)$ and constraint violation of $(g_t(x_t))^+$

  $\tilde{f}_t \leftarrow \beta f_t, \tilde{g}_t \leftarrow \beta \max(0, g_t)$.

  $Q(t) = Q(t-1) + \tilde{g}_t(x_t)$.

  Compute (sub)gradient $\nabla_t = \nabla \hat{f}_t(x_t)$, where the surrogate function $\hat{f}_t$ is defined in Eqn. (5)

  $x_{t+1} = \mathcal{P}_{\mathcal{X}}(x_t - \eta_t \nabla_t)$, where

$$\eta_t = \begin{cases} \frac{\sqrt{2}D}{2\sqrt{\sum_{\tau=1}^t \|\nabla_\tau\|_2^2}}, & \text{for convex costs (AdaGrad stepsizes)} \\ \frac{1}{\sum_{s=1}^t H_s}, & \text{for strongly convex costs } (H_s = \text{strong convexity parameter of } f_s, s \geq 1) \end{cases}$$

end for each

2.3.1 Defining the Surrogate Cost Functions

Let $Q(t)$ denote the CCV for the pre-processed constraints up to round t. Clearly, $Q(t)$ satisfies the simple recursion $Q(t) = Q(t-1) + \tilde{g}_t(x_t), t \geq 1$, with $Q(0) = 0$. Recall that one of our objectives is to make $Q(t)$ small. Towards this, let $\Phi : \mathbb{R}_+ \mapsto \mathbb{R}_+$ be any non-decreasing differentiable convex potential (Lyapunov) function such that $\Phi(0) = 0$. Using the convexity of $\Phi(\cdot)$, we have

$$\begin{aligned} \Phi(Q(t)) &\leq \Phi(Q(t-1)) + \Phi'(Q(t))(Q(t) - Q(t-1)) \\ &= \Phi(Q(t-1)) + \Phi'(Q(t))\tilde{g}_t(x_t). \end{aligned} \tag{3}$$

Hence, the change (*drift*) of the potential function $\Phi(Q(t))$ on round t can be upper bounded as

$$\Phi(Q(t)) - \Phi(Q(t-1)) \leq \Phi'(Q(t))\tilde{g}_t(x_t). \tag{4}$$

Recall that, in addition to controlling the CCV, we also want to minimize the cumulative cost $\sum_{t=1}^T f_t(x_t)$ (which is equivalent to the regret minimization). Inspired by the stochastic *drift-plus-penalty* framework of Neely [2010], we combine these two objectives to a single objective of minimizing a sequence of surrogate cost functions $\{\hat{f}_t\}_{t=1}^T$ which are obtained by taking a positive linear combination of the drift upper bound (4) and the cost function. More precisely, we define

$$\hat{f}_t(x) := V\tilde{f}_t(x) + \Phi'(Q(t))\tilde{g}_t(x), \quad t \geq 1. \tag{5}$$

In the above, V is a suitably chosen non-negative parameter to be determined later. In brief, the proposed policy for COCO, described in Algorithm 1, simply runs an adaptive OCO policy on the surrogate cost function sequence $\{\hat{f}_t\}_{t\geq 1}$, with a specific choice of the potential function $\Phi(\cdot)$, the parameter V, and step-size sequence $\{\eta_t\}_{t\geq 1}$, as dictated by the following analysis.

2.3.2 The Regret Decomposition Inequality

Let $x^\star \in \mathcal{X}^\star$ be any feasible action guaranteed by Assumption (3). Plugging in the definition of surrogate costs (5) into the drift inequality (4), and using the fact that $g_\tau(x^\star) \leq 0, \forall \tau \geq 1$, we have

$$\Phi(Q(\tau)) - \Phi(Q(\tau-1)) + V(\tilde{f}_\tau(x_\tau) - \tilde{f}_\tau(x^\star)) \leq \hat{f}_\tau(x_\tau) - \hat{f}_\tau(x^\star), \ \forall \tau \geq 1.$$

Summing the above inequalities for rounds $1 \leq \tau \leq t$, and using the fact that $\Phi(0) = 0$, we obtain

$$\Phi(Q(t)) + V\text{Regret}_t(x^\star) \leq \text{Regret}'_t(x^\star), \ \forall x^\star \in \mathcal{X}^\star, \tag{6}$$

where Regret_t on the LHS and Regret'_t on the RHS of (6) refer to the regret for learning the pre-processed cost functions $\{\tilde{f}_t\}_{t\geq 1}$ and the surrogate cost functions $\{\hat{f}_t\}_{t\geq 1}$ respectively. We will use

https://doi.org/10.52202/079017-1306

the following upper bound on the ℓ_2-norm of the (sub)gradients G_t of the surrogate cost function $\hat{f}_t$ defined in Eqn. (5):

$$G_t \equiv \|\nabla \hat{f}_t(x_t)\| \overset{(a)}{\leq} V\|\nabla \tilde{f}_t(x_t)\| + \Phi'(Q(t))\|\nabla \tilde{g}_t(x_t)\| \overset{(b)}{\leq} \beta G\big(V + \Phi'(Q(t))\big), \tag{7}$$

where in (a), we have used the triangle inequality for ℓ_2 norms and in (b), we have used the fact that all pre-processed functions are βG-Lipschitz.

2.3.3 Convex Cost and Convex Constraint Functions

We now apply the regret decomposition inequality (6) to the case of convex cost and convex constraint functions. Let us choose the regret-minimizing OCO subroutine for the surrogate cost functions to be the OGD policy with adaptive step sizes (a.k.a. *AdaGrad*) described in part 1 of Theorem 6 in the Appendix (see Algorithm 1). Plugging in the adaptive regret bound (24) on the RHS of (6), setting $\beta = (2GD)^{-1}$, and using Eqn. (7), we arrive at the following inequality valid for any $t \geq 1$:

$$\Phi(Q(t)) + V\text{Regret}_t(x^\star) \leq \sqrt{\sum_{\tau=1}^{t}\big(\Phi'(Q(\tau))\big)^2} + V\sqrt{t}. \tag{8}$$

In deriving the above result, we have utilized simple algebraic inequalities $(x+y)^2 \leq 2(x^2+y^2)$ and $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}, a, b \geq 0$. Now recall that the sequence $\{Q(t)\}_{t\geq 1}$ is non-negative and non-decreasing as $\tilde{g}_t \geq 0$. Furthermore, the derivative $\Phi'(\cdot)$ is non-decreasing as the function $\Phi(\cdot)$ is assumed to be convex. Hence, bounding all terms in the summation on the RHS of (8) from above by the last term, we arrive at the following inequality for any feasible $x^\star \in \mathcal{X}^\star$:

$$\Phi(Q(t)) + V\text{Regret}_t(x^\star) \leq \Phi'\big(Q(t)\big)\sqrt{t} + V\sqrt{t}. \tag{9}$$

The simplified regret decomposition inequality (9) constitutes the key step for the subsequent analysis.

■ Performance Analysis

An exponential Lyapunov function: We now derive the Regret and CCV bounds for the proposed policy (Algorithm 1) by choosing $\Phi(\cdot)$ to be the exponential Lyapunov function: $\Phi(x) \equiv \exp(\lambda x) - 1$, where the parameter $\lambda \geq 0$ will be fixed later. Clearly, the function $\Phi(\cdot)$ satisfies the required conditions for a Lyapunov function - it is a non-decreasing and convex function with $\Phi(0) = 0$.

Bounding the Regret: With the above choice for the Lyapunov function $\Phi(\cdot)$, Eqn. (9) implies that for any feasible $x^\star \in \mathcal{X}^\star$ and for any $t \in [T]$, we have

$$\exp(\lambda Q(t)) - 1 + V\text{Regret}_t(x^\star) \leq \lambda \exp(\lambda Q(t))\sqrt{t} + V\sqrt{t}.$$

Transposing the first term on the above inequality to the RHS and dividing throughout by V, we have:

$$\text{Regret}_t(x^\star) \leq \sqrt{t} + \frac{1}{V} + \frac{\exp(\lambda Q(t))}{V}(\lambda\sqrt{t} - 1). \tag{10}$$

Choosing any $\lambda \leq \frac{1}{\sqrt{T}}$, the last term in the above inequality becomes non-positive for any $t \in [T]$. Hence, for any $x^\star \in \mathcal{X}^\star$, we have the following regret bound

$$\text{Regret}_t(x^\star) \leq \sqrt{t} + \frac{1}{V}. \quad \forall t \in [T]. \tag{11}$$

Bounding the CCV: Since all pre-processed cost functions are $\beta G = (2D)^{-1}$-Lipschitz, we trivially have $\text{Regret}_t(x^\star) = \sum_{s=1}^{t}(\tilde{f}_s(x_s) - \tilde{f}_s(x^\star)) \geq -\frac{Dt}{2D} \geq -\frac{t}{2}$. Hence, from Eqn. (10), we have that for any $\lambda < \frac{1}{\sqrt{T}}$ and any $t \in [T]$:

$$\frac{\exp(\lambda Q(t))}{V}(1 - \lambda\sqrt{t}) \leq 2t + \frac{1}{V} \implies Q(t) \leq \frac{1}{\lambda}\ln\frac{1 + 2Vt}{1 - \lambda\sqrt{t}}. \tag{12}$$

Choosing $\lambda = \frac{1}{2\sqrt{T}}, V = 1$, and scaling the bounds back by $\beta^{-1} \equiv 2GD$, we arrive at our main result.

Theorem 1. *For the COCO problem with adversarially chosen G-Lipschitz cost and constraint functions, Algorithm 1, with* $\beta = (2GD)^{-1}, V = 1, \Phi(x) = \exp(\frac{x}{2\sqrt{T}}) - 1$, *yields the following Regret and CCV bounds for any horizon length* $T \geq 1$:

$$\textit{Regret}_t \leq 2GD(\sqrt{t} + 1), \quad \forall t \in [T], \quad \textit{CCV}_T \leq 4GD\ln\big(2\big(1 + 2T\big)\big)\sqrt{T}.$$

In the above, D denotes the Euclidean diameter of the closed and convex admissible set $\mathcal{X}$.

2.3.4 Strongly Convex Cost and Convex Constraint Functions

We now consider the setting where each of the cost functions $f_t, t \geq 1$, is α-strongly convex for some $\alpha > 0$. The constraint functions g_t's are assumed to be convex as before and not necessarily strongly convex. In this case, we choose the regret-minimizing OCO subroutine for the surrogate cost functions to be the OGD algorithm with the step-size sequence as given in part 2 of Theorem 6 in the Appendix (see Algorithm 1). Since the cost functions are known to be α-strongly convex, each of the surrogate cost functions (5) is $V\alpha$-strongly convex. Hence, using the bound from Eqn. (7), choosing the scaling parameter to be $\beta = 1$, and simplifying the generic regret bound given by Eqn. (25), we obtain the following regret bound for learning the surrogate cost functions $\{\hat{f}_s\}_{s\geq 1}$:

$$\text{Regret}'_t(x^\star) \leq \frac{VG^2}{\alpha}(1+\ln(t)) + \frac{G^2}{\alpha V}\sum_{\tau=1}^{t}\frac{(\Phi'(Q(\tau)))^2}{\tau},\ x^\star \in \mathcal{X}. \tag{13}$$

In the above, we have used the standard bound for the Harmonic sum: $\sum_{\tau=1}^{t}\frac{1}{\tau} \leq 1+\ln(t)$, as well as the fact that $(a+b)^2 \leq 2(a^2+b^2)$. Substituting the bound (13) into the regret decomposition inequality (6), and using the non-decreasing property of the sequence $\{Q(\tau)\}_{\tau\geq 1}$ and the derivative $\Phi'(\cdot)$, we obtain

$$\Phi(Q(t)) + V\text{Regret}_t(x^\star) \leq \frac{VG^2}{\alpha}(1+\ln(t)) + \frac{G^2}{\alpha V}(1+\ln(t))\big(\Phi'(Q(t))\big)^2,\ \forall x^\star \in \mathcal{X}^\star, \forall t. \tag{14}$$

Finally, choosing $\Phi(\cdot)$ as the quadratic Lyapunov function, *i.e.,* $\Phi(x) \equiv x^2$, we arrive at the following result for strongly convex cost and convex constraint functions.

Theorem 2. *For the COCO problem with adversarially chosen α-strongly convex, G-Lipschitz cost functions and G-Lipschitz convex constraint functions, Algorithm 1, with $\beta = 1, V = \frac{8G^2\ln(Te)}{\alpha}, \Phi(x) = x^2$, yields the following Regret and CCV bounds for any horizon length $T \geq 1$:*

$$\textit{Regret}_t(x^\star) \leq \frac{G^2}{\alpha}\big(1+\ln(t)\big),\ \textit{CCV}_t = O\Big(\sqrt{\frac{t\log T}{\alpha}}\Big), \forall x^\star \in \mathcal{X}^\star,\ \forall t \in [T].$$

Furthermore, if the worst-case regret is non-negative in some round t (i.e., $\sup_{x^\star\in\mathcal{X}^\star}\textit{Regret}_t(x^\star) \geq 0$*), then the CCV can be further improved to $\textit{CCV}_T = O(\frac{\log T}{\alpha})$ while keeping the regret bound the same.*

Please refer to Appendix A.6 for the proof of Theorem 2.

Remarks: The second part of the theorem is surprising because it says that when the regret is non-negative, a stronger logarithmic CCV bound holds for not necessarily strongly convex constraints. In Appendix A.7, we give example of an interesting class of adversaries, called *convex adversary*, for which the non-negative regret assumption holds true in the OCO setting.

2.4 Lower Bounds

We now show that under Assumptions 1, 2, and 3, the regret and the CCV of any online policy for the COCO problem for T rounds are both lower bounded by $\Omega(\sqrt{T})$ provided the problem is high-dimensional. Recall that if the constraint function $g_t = 0, \forall t$, then the COCO problem reduces to the standard OCO problem, and $\Omega(\sqrt{T})$ is a well-known regret lower bound for OCO [Hazan, 2022, Theorem 10]. In this case, we trivially have CCV = 0. The main challenge in proving a lower bound for COCO is *simultaneously* bounding both the regret and CCV. Prior work does not give any simultaneous lower bounds since the standard adversarial inputs used to derive the lower bound of Hazan [2022] do not satisfy the feasibility assumption (Assumption 3). We derive the lower bound by constructing a sequence of cost and constraint functions that satisfy Assumption 3 in a d-dimensional Euclidean box of unit diameter.

Theorem 3. *Under Assumptions 1, 2, and 3, for any choice of the horizon length T and online policy, there exists a problem instance with dimension $d \geq T$ where $\min(\textit{Regret}_T, \textit{CCV}_T) = \Omega(\sqrt{T})$.*

In high-dimensional problems where $d \gg T$, the above lower bound matches with the upper bound given in Theorem 1. The proof of Theorem 3 can be found in Appendix A.4.

https://doi.org/10.52202/079017-1306

3 The Online Constraint Satisfaction Problem (OCS)

In this section, we study a special case of the COCO problem, which involves only constraint functions and no cost functions. The OCS problem arises in many practical settings, including the multi-task learning problem (see Section A.3 in the Appendix for a brief discussion). In Section A.8 in the Appendix, we also establish a connection between the OCS problem and the well-studied Convex Body Chasing problem [Argue et al., 2019]. The setup is similar to the COCO setting – on every round $t \geq 1$, an online policy selects an action x_t from a closed, bounded, and convex admissible set $\mathcal{X} \subseteq \mathbb{R}^d$. After observing the current action x_t, the adversary chooses k constraints of the form $g_{t,i}(x) \leq 0, i \in [k]$, where each $g_{t,i} : \mathcal{X} \mapsto \mathbb{R}$ is a convex function. Let $\mathcal{I}$ be any sub-interval of the horizon $[1, T]$. The cumulative constraint violation (CCV) $\mathbb{V}(T)$ for the OCS problem is defined as the maximum *signed* cumulative constraint violation in any sub-interval:

$$\mathbb{V}(T) = \max_{i=1}^{k} \mathbb{V}_i(T), \text{ where } \mathbb{V}_i(T) = \max_{\mathcal{I} \subseteq [1,T]} \sum_{t \in \mathcal{I}} g_{t,i}(x_t), \ 1 \leq i \leq k. \tag{15}$$

The objective is to design an online learning policy so that $\mathbb{V}(T)$ is as small as possible. It is worth noting that in the OCS problem, we consider a soft constraint violation metric $\max_{\mathcal{I}} \sum_{t \in \mathcal{I}} g_{t,i}(x_t)$ instead of the hard violation metric $\sum_{t=1}^{T} (g_{t,i}(x_t))^+$ as in COCO. This allows for compensating the infeasibility on one round with strict feasibility on other rounds. In contrast with the COCO setting, without Assumption 3, running a no-regret policy on the pointwise maximum of the constraint functions no longer works as the CCV of any fixed benchmark could grow linearly with T. In the OCS problem, we relax the feasibility assumption (Assumption 3), and consider the following two distinct alternatives instead.

1. S-feasibility: Here, we assume that there is an admissible action $x^\star \in \mathcal{X}$ that satisfies the aggregate constraints over any interval of S rounds. However, unlike Liakopoulos et al. [2019], which also considers the same assumption, the value of the parameter S is not necessarily known to the policy *a priori*. Towards this end, we define the set of all S-feasible actions $\mathcal{X}_S$ as below:

$$\mathcal{X}_S = \{x^\star \in \mathcal{X} : \sum_{\tau \in \mathcal{I}} g_{\tau,i}(x^\star) \leq 0, \forall \text{sub-intervals } \mathcal{I} \subseteq [1, T], \ |\mathcal{I}| = S, \forall i \in [k]\}. \tag{16}$$

We now replace Assumption 3 with the following weaker version:

Assumption 4 (S-feasibility)**.** *$\mathcal{X}_S \neq \varnothing$ for some $1 \leq S \leq T$.*

Clearly, Assumption 4 is weaker than Assumption 3 as $\mathcal{X}^\star \subseteq \mathcal{X}_S, \forall S \geq 1$. Note that even when the individual constraint functions satisfy S-feasibility, their pointwise maximum need not satisfy S-feasibility. Hence, unlike COCO under Assumption 3, this problem cannot be solved by simply running a no-regret policy on the pointwise maximum of the constraints.

2. P_T-constrained adversary In this case, we drop any feasibility assumption altogether. As a consequence, any static admissible benchmark $x^\star \in \mathcal{X}$ also incurs a CCV.

Definition 1. *An adversary is called P_T-constrained if its minimum static CCV is $P_T F$,* i.e., *$\frac{1}{F} \min_{x^\star \in \mathcal{X}} \max_{\mathcal{I} \subseteq [T], i} \sum_{t \in \mathcal{I}} g_{t,i}(x^\star) = P_T$, where F is a normalizing factor denoting the maximum absolute value of the constraint functions within the compact admissible set $\mathcal{X}$.*

As before, the value of P_T is not necessarily known to the policy *a priori*.

3.1 Designing an OCS Policy with a Quadratic Lyapunov function

We define a process $\boldsymbol{Q}(t) = \big(Q_i(t), i \in [k]\big), t \geq 1$, which tracks the CCV:

$$Q_i(t) = \big(Q_i(t-1) + g_{t,i}(x_t)\big)^+, \ Q_i(0) = 0, \ t \geq 1, \ \forall i \in [k]. \tag{17}$$

Notably, in contrast to COCO, we *do not* clip the constraint functions in the above recursion. Expanding Eqn. (17), which is also known as the queueing recursion or the *Lindley process* [Asmussen, 2003, pp. 92], and using the definition in Eqn. (15), we have the following relation for all $i \in [k]$:

$$\mathbb{V}_i(T) \equiv \max_{t=1}^{T} \max(0, \max_{\tau=0}^{t-1} \sum_{s=t-\tau}^{t} g_{s,i}(x_l)) = \max_{t=1}^{T} Q_i(t). \tag{18}$$

Equation (18) indicates that to control the CCV (15), it is sufficient to control the $\boldsymbol{Q}(t)$ process. Similar to the COCO problem, we combine the classic Lyapunov method with adaptive no-regret OCO policies to control the $\boldsymbol{Q}(t)$ process.

A Quadratic Lyapunov function: We consider the quadratic potential function $\Phi(\boldsymbol{Q}(t)) \equiv \sum_{i=1}^{k} Q_i^2(t), t \geq 1$. Since $((x)^+)^2 = xx^+, \forall x \in \mathbb{R}$, from Eqn. (17), we have

$$\begin{aligned} Q_i^2(t) &= \big(Q_i(t-1) + g_{t,i}(x_t)\big)Q_i(t) = Q_i(t-1)Q_i(t) + Q_i(t)g_{t,i}(x_t), \\ &\overset{(a)}{\leq} \frac{1}{2}Q_i^2(t) + \frac{1}{2}Q_i^2(t-1) + Q_i(t)g_{t,i}(x_t), \ \forall i \in [k]. \end{aligned} \tag{19}$$

where in (a), we have used the AM-GM inequality. Rearranging Eqn. (19), the change of the potential function $\Phi(\boldsymbol{Q}(t))$ on round t can be upper bounded as follows

$$\Phi(\boldsymbol{Q}(t)) - \Phi(\boldsymbol{Q}(t-1)) = \sum_{i=1}^{k} \big(Q_i^2(t) - Q_i^2(t-1)\big) \leq 2\sum_{i=1}^{k} Q_i(t)g_{t,i}(x_t). \tag{20}$$

Similar to (5), we now define a surrogate cost function $\hat{f}_t : \mathcal{X} \mapsto \mathbb{R}$ as a linear combination of the constraint functions with the coefficients given by the vector $\boldsymbol{Q}(t)$, *i.e.,*

$$\hat{f}_t(x) \equiv 2\sum_{i=1}^{k} Q_i(t)g_{t,i}(x). \tag{21}$$

Clearly, the surrogate cost function $\hat{f}_t(\cdot)$ is convex since the coefficients $Q_i(t)$'s are non-negative and the constraint functions are convex. Our OCS policy, described below, simply runs a regret-minimizing adaptive OCO subroutine on the surrogate cost function sequence (21).

The OCS policy (Algorithm 2): Pass the surrogate cost functions $\{\hat{f}_t\}_{t\geq 1}$ to the AdaGrad algorithm which enjoys a data-dependent regret as given in part 1 of Theorem 6 in the Appendix (Eqn. (24)).

Algorithm 2 Online Policy for OCS

1: **Input:** Sequence of convex constraint functions $\{g_{t,i}\}_{i\in[k],t\geq 1}$, a closed and convex admissible set $\mathcal{X}$ with a finite Euclidean diameter D, $\mathcal{P}_{\mathcal{X}}(\cdot)$ = Euclidean projection operator on the set $\mathcal{X}$
2: **Output:** Sequence of admissible actions $\{x_t\}_{t\geq 1}$
3: **Initialization:** Set $x_1 \in \mathcal{X}$ arbitrarily, $Q_i(0) = 0, \ \forall i \in [k]$.
4: **for each** each round $t \geq 1$ **do**
5: Play x_t, observe the constraint functions $\{g_{t,i}\}_{i\in[k]}$ revealed by the adversary.
6: **[Update $\boldsymbol{Q}(t)$]** $Q_i(t) = (Q_i(t-1) + g_{t,i}(x_t))^+, i \in [k]$.
7: **[Compute a subgradient]** $\nabla_t \equiv \nabla \hat{f}_t(x_t) = 2\sum_{i=1}^{k} Q_i(t)\nabla g_{t,i}(x_t)$.
8: **[AdaGrad step]** Compute the next action $x_{t+1} = \mathcal{P}_{\mathcal{X}}(x_t - \eta_t \nabla_t)$, where $\eta_t = \frac{\sqrt{2}D}{2\sqrt{\sum_{\tau=1}^{t} \|\nabla_\tau\|_2^2}}$.
9: **end for each**

3.2 Performance Bounds

Theorem 4. *Under Assumptions 1, 2, and 4, Algorithm 2 achieves the following CCV bound for the OCS problem:* $\mathbb{V}(T) = O(\max(\sqrt{ST}, S))$.

Theorem 5. *Under Assumptions 1 and 2, Algorithm 2 achieves the following CCV bound for the OCS problem for any P_T-constrained adversary as given in Definition 1:*

$$\mathbb{V}(T) = O(P_T^{1/3}T^{2/3}) + O(\sqrt{T}).$$

Trivially, we have $S \leq T$ and $P_T \leq T$. In the non-trivial case where either S or P_T increases *sub-linearly* with the horizon length T, the above theorems yield sublinear CCV bounds.

4 Experiments: Credit Card Fraud Detection

Classification with a highly imbalanced dataset: We first formulate the credit card fraud detection problem in the COCO framework. Assume that we receive a sequence of d-dimensional feature vectors $\{z_t\}_{t\geq 1}$ and the corresponding binary labels $\{y_t\}_{t\geq 1}$ for a sequence of credit card transactions, where each transaction can either be legitimate (`label` = 0) or fraudulent (`label` = 1). The problem

https://doi.org/10.52202/079017-1306

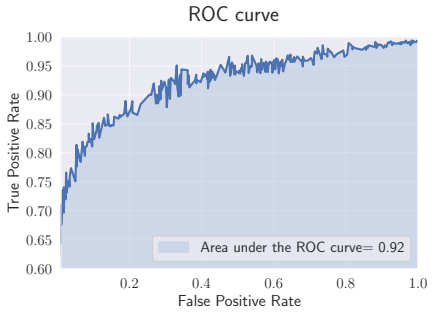

Figure 1: ROC curve obtained by varying λ

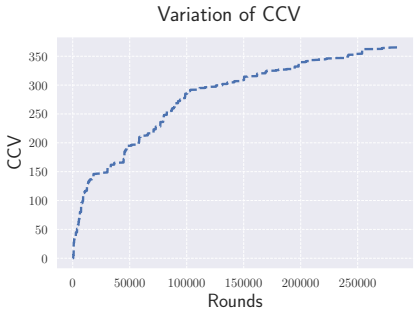

Figure 2: Typical variation of the CCV with time

is to predict the label $\hat{y}_t$ for each transaction z_t before its true label $y_t \in \{0, 1\}$ is revealed. Typically, legitimate transactions outnumber fraudulent transactions by orders of magnitude. Since the goal is to detect any fraudulent transactions (even at the cost of a few false alarms), maximizing the classification accuracy alone is insufficient due to the significant class imbalance. We propose the following reformulation for this problem within the COCO framework.

Formulation: Let $\hat{y}_t(z_t, x)$ be the likelihood of class 1 for the feature z_t, given by a parameterized model with parameter x. Hence, the log-likelihood $\mathcal{L}(t)$ of the data on round t can be expressed as:

$$\mathcal{L}(t) = y_t \log(\hat{y}_t(z_t, x)) + (1 - y_t) \log(1 - \hat{y}_t(z_t, x)).$$

We train the model by maximizing the sum of log-likelihoods for legitimate transactions, subject to the constraint that all fraudulent transactions have a likelihood value close to 1 (*i.e.,* the sum of the log-likelihoods of the fraudulent transactions remains close to zero):

$$\max_x \sum_{t=1}^{T} (1 - y_t) \log(1 - \hat{y}_t(z_t, x)), \quad \text{s.t.} \quad \sum_{t=1}^{T} y_t \log(\hat{y}_t(z_t, x)) \geq 0. \tag{22}$$

The above problem (22) can be immediately recognized to be an instance of COCO with the following cost and constraint functions:

$$f_t(x) \equiv -(1 - y_t) \log(1 - \hat{y}_t(z_t, x)), \quad g_t(x) \equiv -y_t \log(\hat{y}_t(z_t, x)), \ t \geq 1.$$

In our experiments, we consider the common scenario in which the likelihoods are modeled by the output of a feedforward neural network. Note that the feasibility assumption (Assumption 3) is naturally satisfied as the overparameterized neural network models are known to perfectly fit the data [Belkin et al., 2019]. However, in this case, the functions f_t and g_t are generally non-convex.

Experiments: We experiment with a publicly available credit card transaction dataset [Dal Pozzolo et al., 2014]. This highly imbalanced dataset contains only 492 frauds ($\sim 0.17\%$) out of $284,807$ reported transactions. Each data point has $D_{\text{in}} = 30$ features and binary labels. We choose a simple network architecture with a single hidden layer containing $H = 10$ hidden nodes and sigmoid non-linearities. Unlike previous algorithms, our algorithm is especially suitable for training neural network models as it only needs to compute the gradients (via backward pass) and evaluate the functions (via forward pass). Initially, all weights are independently sampled from a standard normal distribution. The network is then trained using Algorithm 1 on a quad-core CPU with 8 GB RAM. The projection operation corresponds to L_2-normalization. The code has been publicly released [Sinha, 2024b].

Results: Given the severe class imbalance, the area under the ROC curve, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR), is an appropriate metric to evaluate any prediction algorithm for this problem. By varying the hyperparameter λ, we obtain the ROC curve shown in Figure 1. The area under the ROC curve is computed to be ≈ 0.92, which is an excellent score (cf. ideal score = 1.0), notwithstanding the fact that, unlike the standard resampling-based techniques, the algorithm learns in an entirely online fashion starting from random initialization. Figure 2 illustrates the expected sublinear variation of CCV during one of the algorithm runs.

5 Conclusion

In this paper, we proposed efficient online policies for the COCO problem with optimal performance bounds. We also derived sublinear CCV bounds for the OCS problem under a set of relaxed assumptions. Our analysis is streamlined, leveraging Lyapunov theory and adaptive regret bounds for the standard OCO problem. In the future, exploring dynamic regret bounds and a bandit extension of the COCO problem would be interesting.

6 Acknowledgement

This work was supported by the Department of Atomic Energy, Government of India, under project no. RTI4001 and by a Google India faculty research award. The first author was also partially supported by a US-India NSF-DST collaborative grant coordinated by IDEAS-Technology Innovation Hub (TIH) at the Indian Statistical Institute, Kolkata. The authors gratefully acknowledge comments from the anonymous reviewers, which substantially improved the quality of the presentation.

References

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.

Abhishek Sinha. BanditQ - Fair Bandits with Guaranteed Rewards. In *Uncertainty in Artificial Intelligence*. PMLR, 2024a.

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.

Elad Hazan, Alexander Rakhlin, and Peter Bartlett. Adaptive online gradient descent. *Advances in neural information processing systems*, 20, 2007.

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

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

Elad Hazan. *Introduction to online convex optimization*. MIT Press, 2022.

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

Rodolphe Jenatton, Jim Huang, and Cédric Archambeau. Adaptive algorithms for online convex optimization with long-term constraints. In *International Conference on Machine Learning*, pages 402–411. PMLR, 2016.

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.

Xinlei Yi, Xiuxian Li, Tao Yang, Lihua Xie, Tianyou Chai, and Karl Johansson. Regret and cumulative constraint violation analysis for online convex optimization with long term constraints. In *International Conference on Machine Learning*, pages 11998–12008. PMLR, 2021.

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

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

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.

Xinlei Yi, Xiuxian Li, Tao Yang, Lihua Xie, Yiguang Hong, Tianyou Chai, and Karl H Johansson. Distributed online convex optimization with adversarial constraints: Reduced cumulative constraint violation bounds under slater's condition. *arXiv preprint arXiv:2306.00149*, 2023.

Michael J Neely. Stochastic network optimization with application to communication and queueing systems. *Synthesis Lectures on Communication Networks*, 3(1):1–211, 2010.

 https://doi.org/10.52202/079017-1306

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.

Tianyi Chen and Georgios B Giannakis. Bandit convex optimization for scalable and dynamic iot management. *IEEE Internet of Things Journal*, 6(1):1276–1286, 2018.

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.

Rahul Vaze. On dynamic regret and constraint violations in constrained online convex optimization. In *2022 20th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*, pages 9–16, 2022. doi: 10.23919/WiOpt56218.2022.9930613.

Qingsong Liu, Wenfei Wu, Longbo Huang, and Zhixuan Fang. Simultaneously achieving sublinear regret and constraint violations for online convex optimization with time-varying constraints. *ACM SIGMETRICS Performance Evaluation Review*, 49(3):4–5, 2022.

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.

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. A low complexity algorithm with $o(\sqrt{T})$ regret and $o(1)$ constraint violations for online convex optimization with long term constraints. *arXiv preprint arXiv:1604.02218*, 2016.

CJ Argue, Sébastien Bubeck, Michael B Cohen, Anupam Gupta, and Yin Tat Lee. A nearly-linear bound for chasing nested convex bodies. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 117–122. SIAM, 2019.

Søren Asmussen. *Applied probability and queues*, volume 2. Springer, 2003.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Andrea Dal Pozzolo, Olivier Caelen, Yann-Ael Le Borgne, Serge Waterschoot, and Gianluca Bontempi. Learned lessons in credit card fraud detection from a practitioner perspective. *Expert systems with applications*, 41(10):4915–4928, 2014. The dataset is available for download at https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/.

Abhishek Sinha. Source code for "Optimal Algorithms for Online Convex Optimization with Adversarial Constraints"; A. Sinha, R. Vaze. https://github.com/abhishek-sinha-tifr/COCO, 2024b.

Max Hopkins, Daniel M. Kane, Shachar Lovett, and Gaurav Mahajan. Realizable learning is all you need. In *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 3015–3069. PMLR, 02–05 Jul 2022.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Jacob Abernethy, Chansoo Lee, Abhinav Sinha, and Ambuj Tewari. Online linear optimization via smoothing. In *Conference on Learning Theory*, pages 807–823, 2014.

Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Delay-tolerant online convex optimization: Unified analysis and adaptive-gradient algorithms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Ofer Dekel, Philip M Long, and Yoram Singer. Online multitask learning. In *International Conference on Computational Learning Theory*, pages 453–467. Springer, 2006.

Keerthiram Murugesan, Hanxiao Liu, Jaime Carbonell, and Yiming Yang. Adaptive smoothed online multi-task learning. *Advances in Neural Information Processing Systems*, 29, 2016.

Nikhil Bansal, Martin Böhm, Marek Eliáš, Grigorios Koumoutsos, and Seeun William Umboh. Nested convex bodies are chaseable. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1253–1260. SIAM, 2018.

Sébastien Bubeck, Bo'az Klartag, Yin Tat Lee, Yuanzhi Li, and Mark Sellke. Chasing nested convex bodies nearly optimally. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1496–1508. SIAM, 2020.

 https://doi.org/10.52202/079017-1306

A Appendix

A.1 Discussion on Assumptions 1, 2 and 3

Assumptions 1 and 2 are standard in the online learning literature. The feasibility assumption (Assumption 3) is analogous to the *realizability* assumption in learning theory [Hopkins et al., 2022] and is commonly used in the COCO literature [Neely and Yu, 2017, Yu and Neely, 2016, Yuan and Lamperski, 2018, Yi et al., 2023, Liakopoulos et al., 2019]. Assumption 3 requires the existence of a single admissible action $x^\star \in \mathcal{X}$ that satisfies the constraints in *every* round. Consequently, all constraint functions are required to be non-positive over a non-empty common subset. This assumption is weakened in Section 3, Assumption 4, which only requires the existence of a fixed admissible action $x^\star$ that satisfies the constraints *on average*. Specifically, Assumption 4 requires that the sum of the constraint functions evaluated at some admissible $x^\star$ over any interval of length S is non-positive. Notably, throughout the paper, we *do not* assume Slater's condition as it does not hold in many problems of interest [Yu and Neely, 2016]. As a result, unlike many previous works [Yu et al., 2017], our bounds are *independent* of Slater's constant, which can be problem-dependent. Furthermore, we do not restrict the sign of either cost or constraint functions, allowing them to take both positive and negative values.

A.2 Preliminaries on Online Convex Optimization (OCO)

The standard OCO problem can be described as a repeated game between an online policy and an adversary [Hazan, 2022]. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a convex decision set, which we refer to as the *admissible* set. In each round $t \geq 1$, an online policy selects an action $x_t \in \mathcal{X}$. After the action x_t is chosen, the adversary reveals a convex cost function $f_t : \mathcal{X} \mapsto \mathbb{R}$. The goal of the online policy is to choose an admissible action sequence $\{x_t\}_{t\geq 1}$ so that its total cost over a horizon of length T is not significantly larger than the total cost incurred by any fixed admissible action $x^\star \in \mathcal{X}$. More precisely, the objective is to minimize the static regret, defined as:

$$\text{Regret}_T \equiv \sup_{x^\star \in \mathcal{X}} \text{Regret}_T(x^\star), \text{ where } \text{Regret}_T(x^\star) \equiv \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^\star). \tag{23}$$

Algorithm 3 Online Gradient Descent (OGD)

1: **Input:** Non-empty closed convex set $\mathcal{X} \subseteq \mathbb{R}^d$, sequence of convex cost functions $\{f_t\}_{t\geq 1}$, step sizes $\eta_1, \eta_2, \ldots, \eta_T > 0$, Euclidean projection operator $\mathcal{P}_{\mathcal{X}}(\cdot)$ onto the set $\mathcal{X}$
2: **Initialization**: Set $x_1 \in \mathcal{X}$ arbitrarily
3: **for each** round $t \geq 1$ **do**
4: Play x_t, observe f_t, incur a cost of $f_t(x_t)$.
5: Compute a (sub)gradient $\nabla_t \equiv \nabla f_t(x_t)$.
6: Update $x_{t+1} = \mathcal{P}_{\mathcal{X}}(x_t - \eta_t \nabla_t)$.
7: **end for each**

In a seminal paper, Zinkevich [2003] showed that the online gradient descent policy, outlined in Algorithm 3, run with an appropriately chosen constant step size sequence, achieves a sublinear regret bound $\text{Regret}_T = O(\sqrt{T})$ for Lipschitz-continuous convex cost functions. In Theorem 6, we recall two standard results on further refined data-dependent adaptive regret bounds achieved by the OGD policy with appropriately chosen adaptive step size sequences.

Theorem 6. *Consider the generic OGD policy outlined in Algorithm 3.*

1. *[Duchi et al., 2011], [Orabona, 2019, Theorem 4.14] Let the cost functions $\{f_t\}_{t\geq 1}$ be convex and the step size sequence be adaptively chosen as $\eta_t = \frac{\sqrt{2}D}{2\sqrt{\sum_{\tau=1}^{t} G_\tau^2}}, t \geq 1$, where D is the Euclidean diameter of the admissible set $\mathcal{X}$ and $G_t = \|\nabla f_t(x_t)\|_2, t \geq 1$. Then Algorithm 3 achieves the following regret bound:*

$$\textit{Regret}_T \leq \sqrt{2}D\sqrt{\sum_{t=1}^{T} G_t^2}. \tag{24}$$

The OGD policy with the above adaptive step-size sequence is known as (a variant of) the AdaGrad policy in the literature [Duchi et al., 2011].

2. *[Hazan et al., 2007, Theorem 2.1] Let the cost functions $\{f_t\}_{t\geq 1}$ be strongly convex and let $H_t > 0$ be the strong convexity parameter[2] for the cost function f_t. Let the step size sequence be adaptively chosen as $\eta_t = \frac{1}{\sum_{s=1}^{t} H_s}, t \geq 1$. Then Algorithm 3 achieves the following regret bound:*

$$Regret_T \leq \frac{1}{2} \sum_{t=1}^{T} \frac{G_t^2}{\sum_{s=1}^{t} H_s}. \tag{25}$$

Similar adaptive regret bounds are known for various other online learning policies as well. For structured domains, one can use other algorithms such as AdaFTRL [Orabona and Pál, 2018] which gives better regret bounds for high-dimensional problems. Furthermore, for problems with combinatorial structures, adaptive oracle-efficient algorithms, *e.g.,* Follow-the-Perturbed-Leader (FTPL)-based policies, can be employed [Abernethy et al., 2014, Theorem 11]. Our proposed policies are agnostic to the specific online learning subroutine used for the surrogate OCO problem - what matters is that the subroutine provides adaptive regret bounds similar to (24) and (25). This flexibility allows for an immediate extension of our algorithm to a wide range of settings, such as delayed feedback [Joulani et al., 2016] or combinatorial actions.

A.3 Online Multi-task Learning as an Instance of the OCS Problem

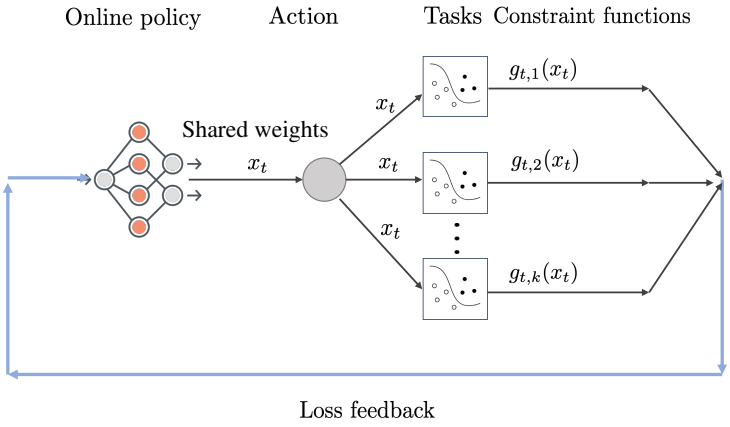

Figure 3: A schematic for the online multi-task learning problem

Consider the problem of online multi-task learning where a single model is trained to perform a number of related tasks [Ruder, 2017, Dekel et al., 2006, Murugesan et al., 2016]. See Figure 3 for a simplified schematic of the multi-task learning pipeline. In this setup, the action x_t naturally corresponds to the shared weight vector that specifies the common model for all tasks. The loss function for the j^{th} task on round t is given by the function $g_{t,j}(\cdot), j \in [k]$. A task is assumed to be satisfactorily completed (*e.g.,* correct prediction in the case of classification problems) on any round if the corresponding loss is non-positive. As an example, using linear predictors for the binary classification problem, the requirement for the j^{th} task on round t can be taken to be $g_{t,j}(x_t) \equiv \langle z_{t,j}, x_t \rangle \leq 0$, where $z_{t,j}$ is the feature vector for the j^{th} task. The goal in multi-task learning is to sequentially update the shared weight vectors $\{x_t\}_{t=1}^{T}$ so that all tasks are successfully completed. Formally, we require that the maximum cumulative loss of each task over any sub-interval grows sub-linearly. Since the weight vector is shared across the tasks, the above goal would be impossible to achieve had the tasks not been related to each other [Ruder, 2017]. Theorem 4 and Theorem 5 give performance bounds for Algorithm 2 under different task-relatedness assumptions.

[2]The strong convexity of f_t implies that $f_t(y) \geq f_t(x) + \langle \nabla f_t(x), y - x \rangle + \frac{H_t}{2}\|x - y\|^2, \forall x, y \in \mathcal{X}, \forall t$.

https://doi.org/10.52202/079017-1306

A.4 Proof of Theorem 3

We prove the Theorem via constructing an explicit input sequence for which no online policy can have better than $\Omega(\sqrt{T})$ regret and CCV.

Action space $\mathcal{X}$: Let $d = T$. Let $\mathcal{X}$ be the d-dimensional cuboid $0 \le x_i \le \frac{1}{\sqrt{d}},\ 1 \le i \le d$. Clearly, the Euclidean diameter of $\mathcal{X}$ is 1.

Input: For each round we will only consider the case when only one constraint is revealed, *i.e.*, $k = 1$. On round $t = 1, \ldots, d$, choose the constraint g_t to be $x_t \le \frac{1}{4\sqrt{d}}$ or $x_t \ge \frac{3}{4\sqrt{d}}$ with equal probability of $\frac{1}{2}$ for $\mathbf{x} = (x_1, \ldots, x_d) \in \mathcal{X}$. Thus, at round t, only the t^{th} dimension has an effective constraint. If the chosen g_t is $x_t \le \frac{1}{4\sqrt{d}}$ then pick $f_t = |x - \frac{1}{4\sqrt{d}}|$, otherwise pick $f_t = |x - \frac{3}{4\sqrt{d}}|$.

For any online policy $\mathcal{A}$, the expected constraint violation at round t is at least $\frac{1}{8\sqrt{d}}$. Thus, the overall expected constraint violation over rounds $t = 1, \ldots, d$ is at least $\frac{\sqrt{d}}{8}$. Moreover, the expected cost $\mathbb{E}[f_t(x_t)]$ of $\mathcal{A}$ is at least $\frac{1}{8\sqrt{d}}$ for each $t = 1, \ldots, d$, and the overall cost $\mathbb{E}\big[\sum_{t=1}^{T} f_t(x_t)\big]$ is at least $\frac{\sqrt{d}}{8}$.

Recall that the choice of input has to satisfy Assumption 3, i.e., $\mathcal{X}^\star \neq \varnothing$. We next demonstrate that for the prescribed input $\exists\ \mathbf{x}^\star \in \mathcal{X}^\star$.

Choosing a feasible $\mathbf{x}^\star$: When g_t is such that the constraint is $x_t \le \frac{1}{4\sqrt{d}}$ choose $\mathbf{x}^\star \in \mathcal{X}$ such that $x_t^\star = \frac{1}{4\sqrt{d}}$ for $t = 1, 2, \ldots, d$, while if g_t is such that $x_t \ge \frac{3}{4\sqrt{d}}$, then choose $x_t^\star = \frac{3}{4\sqrt{d}}$ for $t = 1, 2, \ldots, d$. Thus, a single vector $\mathbf{x}^\star$ satisfies all the revealed constraints. Moreover, with this choice of $\mathbf{x}^\star$, the overall cost of $\mathbf{x}$, $\sum_t f_t(\mathbf{x}^\star)$, is 0.

Since $d = T$, we get that for any online policy $\mathcal{A}$ its regret is at least $\Omega(\sqrt{T})$ and the cumulative constraint violation is $\Omega(\sqrt{T})$. ■

A.5 Comparison with Previous Works

A.5.1 Neely and Yu [2017], Yu et al. [2017] and Liakopoulos et al. [2019]

Policies proposed by Yu et al. [2017] and Liakopoulos et al. [2019] are almost identical to Neely and Yu [2017]. The policy proposed in Neely and Yu [2017], however, is highly customized, does not fully exploit the best guarantees available for the standard OCO problem, and obtains sub-optimal performance bounds that depend inversely on Slater constant, which is assumed to be strictly positive. In a nutshell, Neely and Yu [2017] choose the next action x_{t+1} using the algorithm described below. For all rounds $t \ge 1$, define the following evolution for $Q(t)$:

$$Q(t) = (Q(t-1) + g_t(x_t) + \nabla^T g_t(x_t)(x_t - x_{t-1}))^+,\ Q(0) = 0. \tag{26}$$

The next action is chosen by solving the following quadratic optimization problem:

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \big[\langle V \nabla^T f_t(x_t) + Q(t) \nabla g_t(x_t), x\rangle + \alpha \|x - x_{t-1}\|^2\big],$$

where V and α are suitably chosen parameters.

In comparison, we have a different and simpler update rule:

$$Q(t) = Q(t-1) + (2GD)^{-1}(g_t(x_t))^+,\ Q(0) = 0. \tag{27}$$

We then construct a convex surrogate function $\hat{f}_t(x) \equiv f_t(x) + \frac{1}{4GD\sqrt{T}} e^{\frac{Q(t)}{2\sqrt{T}}} (g_t(x))^+$, whose gradient is then passed directly to the AdaGrad subroutine.

Remarks: We emphasize that Theorem 1, which shows that it is possible to simultaneously achieve $O(\sqrt{T})$ regret and $\tilde{O}(\sqrt{T})$ CCV in the convex setting without assuming Slater's condition, is highly surprising and unexpected. In fact, Liakopoulos et al. [2019, Section 4] had previously commented that:

"... *On the other hand, the point* $O(\sqrt{T})$, $O(\sqrt{T})$ *achieved by Neely and Yu [2017] for* $K = 1$ *is not part of our achievable guarantees; we attribute this gap to the stricter Slater assumption studied by Neely and Yu [2017].*"

Theorem 1 squarely falsifies the last conjecture.

A.5.2 Guo et al. [2022]

The policy in Guo et al. [2022] is a slightly modified form of the policy proposed in [Neely and Yu, 2017]. In particular, it chooses the action x_t by solving the following quadratic optimization problem over $\mathcal{X}$:

$$x_t = \arg\min_{x\in\mathcal{X}} \left[\langle \nabla f_{t-1}(x_{t-1}), x - x_{t-1}\rangle + Q(t-1)\gamma_{t-1} g_{t-1}^{+}(x) + \alpha_{t-1}\|x - x_{t-1}\|^2\right],$$

where the Q variables are updated as follows:

$$Q(t) = \max(Q(t-1) + \gamma_{t-1} g_{t-1}^{+}(x_t), \eta_t).$$

Here $\alpha_t, \eta_t, \gamma_t$ are suitably chosen learning rate parameters. Essentially, this policy is trying to find the local optimum of an augmented Lagrangian under the online information model (f_t and g_t are revealed after action x_t is chosen). Since their augmented Lagrangian involves the constraint function g_{t-1}, their policy needs to solve a full-fledged constrained convex optimization problem over the set $\mathcal{X}$ after having full access to the constraint function. In comparison, our policy, rather than using approximations to Lagrangian and adding regularizers, makes full use of the well-developed theory for OCO and uses first-order methods that need to compute only a gradient and perform one Euclidean projection on each round.

A.5.3 Jenatton et al. [2016]

The policy proposed by Jenatton et al. [2016] is based on the idea of primal-dual algorithm for optimizing the augmented Lagrangian

$$L_t(\lambda, x) = f_t(x) + \lambda g_t(x) - \frac{\theta_t}{2}\lambda^2,$$

where $\frac{\theta_t}{2}\lambda^2$ is the augmentation term. The primal variable x_t and the dual variable λ_t are updated by executing projected gradient descent and gradient ascent on the Lagrangian as follows:

$$x_{t+1} = \mathcal{P}_{\mathcal{X}}(x_t - \eta_t \nabla_x L_t(x_t, \lambda_t))$$

and

$$\lambda_{t+1} = (\lambda_t + \mu_t \nabla_\lambda L_t(x_t, \lambda_t))^{+},$$

where θ_t, η_t, and μ_t are parameters to be chosen.

A.6 Proof of Theorem 2

Bounding the CCV: Choosing $\Phi(x) = x^2$ in Eqn. (14), we have for any feasible $x^\star \in \mathcal{X}^\star$:

$$Q^2(t) + V\text{Regret}_t(x^\star) \le \frac{VG^2}{\alpha}(1 + \ln(t)) + \frac{4G^2Q^2(t)\ln(Te)}{\alpha V}, \tag{28}$$

where, on the last term in the RHS, we have used the fact that $t \le T$. Setting $V = \frac{8G^2\ln(Te)}{\alpha}$, and transposing the last term on the RHS to the left, the above inequality yields

$$Q^2(t) + 2V\text{Regret}_t(x^\star) \le \frac{2VG^2}{\alpha}(1 + \ln(t)). \tag{29}$$

Since the cost functions are assumed to be G-Lipschitz (Assumption 2), we trivially have $\text{Regret}_t(x^\star) = \sum_{t=1}^{T}(f_t(x_t) - f_t(x^\star)) \ge -GDt$. Hence, from Eqn. (29), we obtain

$$Q^2(t) \le 2VGDt + \frac{2VG^2}{\alpha}(1 + \ln(t)) \implies Q(t) \overset{(a)}{\le} 4G\sqrt{\frac{GD}{\alpha}t\ln(Te)} + \frac{4G^2\ln(Te)}{\alpha}.$$

where step (a), we have substituted $V = \frac{8G^2\ln(Te)}{\alpha}$. Hence, we have the following bound $\text{CCV}_t = O\left(\sqrt{\frac{t\log T}{\alpha}}\right)$.

https://doi.org/10.52202/079017-1306

Bounding the regret: Using the above choice for the parameter V and the fact that $Q^2(t) \geq 0$, from Eqn. (29), we have

$$2V\text{Regret}_t(x^\star) \leq \frac{2VG^2}{\alpha}(1+\ln(t)).$$

This leads to the following logarithmic bound for regret for any feasible $x^\star \in \mathcal{X}^\star$:

$$\text{Regret}_t(x^\star) \leq \frac{G^2}{\alpha}(1+\ln(t)). \quad \blacksquare$$

A sharper CCV bound under the non-negative regret assumption: We now establish an improved CCV bound when the worst-case regret is non-negative on some round $t \geq 1$. Let $\sup_{x^\star \in \mathcal{X}^\star} \text{Regret}_t(x^\star) \geq 0$ for some round $t \geq 1$. Letting $V = \frac{8G^2 \ln(Te)}{\alpha}$ as above, from Eq. (29) we have

$$Q^2(t) \leq \frac{2VG^2}{\alpha}(1+\ln(t)) \implies Q(t) = O\big(\frac{\ln T}{\alpha}\big), t \in [T]. \blacksquare$$

Comment: From the above proof, it immediately follows that the same conclusion holds even under the weaker assumption of $-\text{Regret}_T = O(\frac{\log T}{\alpha})$.

A.7 Adversaries Ensuring Non-negative Regret

Convex adversary: An adversary is called *convex* if for any sequence of action sequence $\{x_t\}_{t=1}^T$, the adversary chooses the cost function sequence $\{f_t\}_{t=1}^T$ such that for any $T \geq 1$, we have

$$\sum_{t=1}^{T} f_t(x_t) \geq \sum_{t=1}^{T} f_t(\bar{x}_T), \tag{30}$$

where $\bar{x}_T \equiv \frac{1}{T}\sum_{t=1}^T x_t$. Hence, by definition, a convex adversary guarantees a non-negative regret with respect to the average action $\bar{x}_T$ for all rounds. In the following, we give two examples of convex adversaries.

1. Fixed adversary: An adversary which always selects a fixed convex function f on all rounds is a convex adversary. In this case, Eqn. (30) holds due to the Jensen's inequality.

2. Minimax adversary: Let $\mathcal{F}$ denote an arbitrary non-empty set of convex functions defined on the admissible set $\mathcal{X}$. Consider an adversary $\mathcal{M}$, which, upon seeing the selected action x_t, chooses the worst cost function f_t from the set $\mathcal{F}$ on round t :

$$f_t \in \arg\max_{f \in \mathcal{F}} f(x_t).$$

We now show that $\mathcal{M}$ is a convex adversary. By definition, for any round $\tau \in [T]$, we have

$$f_\tau(x_\tau) \geq f_t(x_\tau) \implies f_\tau(x_\tau) \geq \frac{1}{T}\sum_{t=1}^{T} f_t(x_\tau).$$

Summing up the above inequalities for each $\tau \in [T]$, we have

$$\sum_{\tau=1}^{T} f_\tau(x_\tau) \geq \sum_{t=1}^{T}\frac{1}{T}\sum_{\tau=1}^{T} f_t(x_\tau) \overset{(a)}{\geq} \sum_{t=1}^{T} f_t(\bar{x}_T), \tag{31}$$

where inequality (a) follows upon applying Jensen's inequality to each cost function. Eqn. (31) shows that $\mathcal{M}$ is a convex adversary.

P.S. It can be easily seen that Fixed adversary is a special case of Minimax adversary where $\mathcal{F} = \{f\}$.

A.8 Connection Between OCS and the Convex Body Chasing Problem

A well-studied problem related to the OCS problem is the *nested convex body chasing (NCBC)* problem [Bansal et al., 2018, Argue et al., 2019, Bubeck et al., 2020], where at each round t, a convex set $\chi_t \subseteq \chi$ is revealed such that $\chi_t \subseteq \chi_{t-1}$, where $\chi_0 = \chi \subseteq \mathbb{R}^d$ is a convex, compact, and bounded set. The objective is to choose $x_t \in \chi_t$ so as to minimize the total movement cost across rounds $C = \sum_{t=1}^{T} \|x_t - x_{t-1}\|_2$, where $x_0 \in \chi$ is some fixed action. In NCBC, action x_t is chosen *after* the set χ_t is revealed. This is in contrast to the OCS problem, where x_t must be chosen *before* the constraints $g_{t,i}$'s are revealed at round t. Moreover, note that the nested condition $\chi_t \subseteq \chi_{t-1}$ is stricter than Assumption 3, which is applicable to the OCS problem. However, as we show next, a feasible algorithm for NCBC also provides an upper bound on the CCV of the OCS problem under Assumption 3.

In this reduction, we define χ_t as the intersection of the first kt convex constraints $g_{\tau,i} \leq 0, 1 \leq \tau \leq t, i \in [k]$, revealed up to round t for the OCS problem. It is easy to see that χ_t is convex and $\chi_t \subseteq \chi_{t-1}, \forall t$. Let x_t be the action chosen by an algorithm $\mathcal{A}$ for the NCBC problem after the set χ_t is revealed. Note that $\chi_t \neq \varnothing$, thanks to Assumption 3. We now choose $y_t := x_{t-1}$ as the action for the OCS problem on round t, ensuring that action y_t is chosen before the set χ_t is revealed. The resulting i^{th} constraint violation for the OCS problem at round t is given by

$$g_{t,i}(y_t) \overset{(a)}{\leq} g_{t,i}(y_t) - g_{t,i}(y_{t+1}) \leq G\|y_t - y_{t+1}\|,$$

where (a) follows from the feasibility of $\mathcal{A}$ for NCBC, $y_{t+1} = x_t \in \chi_t$ and hence $g_{t,i}(y_{t+1}) \leq 0$. Summing across rounds $t = 1, \ldots, T$, and taking the max over all the k constraints, we get that the CCV using $\mathcal{A}$ for the OCS is upper bounded by $\sum_{t=2}^{T} G\|y_t - y_{t+1}\| \leq \sum_{t=2}^{T} G\|x_{t-1} - x_t\| \leq G \cdot C_{\mathcal{A}}$, where $C_{\mathcal{A}}$ is the movement cost of $\mathcal{A}$ for the NCBC problem.

From prior work Bansal et al. [2018], Argue et al. [2019], Bubeck et al. [2020], it is known that for NCBC, a Steiner point-based algorithm that chooses x_t as the Steiner point of χ_t can achieve $C_{\mathcal{A}} = O(\sqrt{d \log d})$, where $\chi \subset \mathbb{R}^d$. Thus, the Steiner point-based algorithm (even though computationally intensive) provides an $O(\sqrt{d \log d})$ constraint violation for the OCS as well. However, this result is effective for problems where $\sqrt{d \log d} = o(T)$. Our result efficiently overcomes this hurdle and provides a bound under weaker feasibility assumptions even beyond $\sqrt{d \log d} = o(T)$ – a setting that is better motivated in practice for modern deep learning applications which are characteristically high-dimensional.

A.9 Proof of Theorem 4

Generalized regret decomposition: Fix any S-feasible benchmark $x^\star \in \mathcal{X}_S$, as given by Eqn. (16). Then, from Eqn. (20), we have

$$\begin{aligned}
\Phi(\tau) - \Phi(\tau-1) &\leq 2\sum_{i=1}^{k} Q_i(\tau) g_{\tau,i}(x_\tau) \\
&= 2\sum_{i=1}^{k} Q_i(\tau)\big(g_{\tau,i}(x_\tau) - g_{\tau,i}(x^\star)\big) + 2\sum_{i=1}^{k} Q_i(\tau) g_{\tau,i}(x^\star) \\
&= \hat{f}_\tau(x_\tau) - \hat{f}_\tau(x^\star) + 2\sum_{i=1}^{k} Q_i(\tau) g_{\tau,i}(x^\star).
\end{aligned}$$

Summing up the above inequalities from $\tau = 1$ to $\tau = t$, we have

$$\sum_{i=1}^{k} Q_i^2(t) = \Phi(t) \leq \text{Regret}'_t(x^\star) + 2\sum_{i=1}^{k}\sum_{\tau=1}^{t} Q_i(\tau) g_{\tau,i}(x^\star), \tag{32}$$

where $\text{Regret}'(\cdot)$ refers to the regret of the surrogate costs as before. We now bound the last term by making use of the S-feasibility of the action $x^\star$ as given by Eqn. (16). Let us now divide the entire interval $[1, t]$ into disjoint and consecutive sub-intervals $\{\mathcal{I}_j\}_{j=1}^{\lceil t/S \rceil}$, each of length S (except the last interval which could be of a smaller length). Let $Q_i^\star(j)$ be the value of the variable $Q_i(\cdot)$ at the beginning of the j^{th} interval. We have

$$\sum_{\tau=1}^{t} Q_i(\tau) g_{\tau,i}(x^\star) = \sum_{j=1}^{\lceil t/S \rceil} \sum_{\tau \in \mathcal{I}_j} \big(Q_i(\tau) - Q_i^\star(j)\big) g_{\tau,i}(x^\star) + \sum_{j=1}^{\lceil t/S \rceil} Q_i^\star(j) \sum_{\tau \in \mathcal{I}_j} g_{\tau,i}(x^\star). \tag{33}$$

https://doi.org/10.52202/079017-1306

Using the boundedness assumption, let $g_{t,i}(x) \le F, \forall x \in \mathcal{X}, t, i$. Using the Lipschitzness property of the queueing dynamics (17) with respect to time, we have

$$\max_{\tau \in \mathcal{I}_j} |Q_i(\tau) - Q_i^\star(j)| \le F(S-1).$$

Substituting the above bound into Eqn. (33), we obtain

$$\sum_{\tau=1}^{t} Q_i(\tau) g_{\tau,i}(x^\star) \le \big(1 + \frac{t}{S}\big) F^2 S(S-1) + F(S-1)(Q_i(t) + F(S-1)), \tag{34}$$

where in the last term, we have used the S-feasibility of the action $x^\star$ in all intervals, except possibly the last interval. Substituting the bound (34) into Eqn. (32), we arrive at the following extended regret decomposition inequality:

$$\sum_{i=1}^{k} Q_i^2(t) \quad \le \quad \text{Regret}_t'(x^\star) + 2kF^2St + 2FS\sum_{i=1}^{k} Q_i(t) + 4F^2S^2k. \tag{35}$$

Eqn. (35) leads to the following bound on the cumulative constraint violation.

A.9.1 CCV Bound

We now apply the generalized regret decomposition bound given in (35) to the case of convex constraint functions. Substituting the regret bound (24) of the AdaGrad policy into Eqn. (35), we have

$$\sum_{i=1}^{k} Q_i^2(t) \le c_1 \sqrt{\sum_{\tau=1}^{t} \Big(\sum_{i=1}^{k} Q_i^2(\tau) \Big)} + c_2 St + c_3 S \sum_{i=1}^{k} Q_i(t) + c_4 S^2$$

where the constants $c_1 \equiv O(GD\sqrt{k}), c_2 = O(kF^2), c_3 = O(F), c_4 = O(kF^2)$ are problem-specific parameters that depend on the bounds on the gradients and the maximum value of the constraint functions, the number of constraints, and the diameter of the admissible set. Defining $Q^2(t) \equiv \sum_i Q_i^2(t)$, we obtain:

$$Q^2(t) \le c_1 \sqrt{\sum_{\tau=1}^{t} Q^2(\tau)} + c_2 St + c_3 S \sum_{i=1}^{k} Q_i(t) + c_4 S^2.$$

Since $Q_i(t) \le Ft, \forall i$, the above inequality can be simplified to

$$Q^2(t) \le c_1 \sqrt{\sum_{\tau=1}^{t} Q^2(\tau)} + c_2' St + c_4 S^2, \ \forall t \ge 1, \tag{36}$$

where we have defined $c_2' \equiv c_3 kF + c_2$. To solve the above system of inequalities, note that for each $1 \le \tau \le t$, we have

$$Q^2(\tau) \le c_1 \sqrt{\sum_{\tau=1}^{t} Q^2(\tau)} + c_2' St + c_4 S^2.$$

Summing up the above inequalities for $1 \le \tau \le t$ and defining $Z_t \equiv \sqrt{\sum_{\tau=1}^{t} Q^2(\tau)}$, we obtain

$$\begin{aligned} Z_t^2 &\le c_1 t Z_t + c_2' St^2 + c_4 S^2 t \\ \textit{i.e., } Z_t^2 &\le 3\max(c_1 t Z_t, c_2' St^2, c_4 S^2 t) \\ \textit{i.e., } Z_t &= O(\max(t, t\sqrt{S}, S\sqrt{t})). \end{aligned}$$

Substituting the above bound for $Z(t)$ in Eqn. (36), we have for any $t \ge 1$:

$$\begin{aligned} Q^2(t) &= O(\max(Z_t, St, S^2)) \\ \textit{i.e., } Q(t) &= O(\max(\sqrt{Z_t}, \sqrt{St}, S)) \\ \text{Hence, } Q_i(t) \le Q(t) &= O(\max(\sqrt{t}, \sqrt{t}S^{1/4}, \sqrt{S}t^{1/4}, \sqrt{St}, S)) \\ &= O(\max(\sqrt{St}, S)), \ \forall i \in [k]. \end{aligned}$$

The final result follows upon appealing to Eqn. (18). ■

A.10 Proof of Theorem 5

We will use a similar line of arguments used in the analysis of an S-constrained adversary for a suitable value of S to be determined later. We start from Eqn. (33), which holds for any value of the sub-interval length $S \geq 1$ and any arbitrary adversary. Furthermore, from the definition of a P_T-constrained adversary, we know that there exists a benchmark $x^\star \in \mathcal{X}$ such that for any interval $\mathcal{I}_j$ and any $i \in [k]$, we have:

$$\sum_{\tau \in \mathcal{I}_j} g_{\tau,i}(x^\star) \leq P_T F,$$

where F is the maximum absolute value of the constraint functions as given in Definition 1. Hence,

$$\begin{aligned}\sum_{j=1}^{\lceil t/S \rceil} Q_i^\star(j) \sum_{\tau \in \mathcal{I}_j} g_{\tau,i}(x^\star) &\leq P_T F \sum_{j=1}^{\lceil t/S \rceil} Q_i^\star(j) \\ &\leq \frac{P_T F}{S} \sum_{j=1}^{\lceil t/S \rceil} \sum_{\tau \in \mathcal{I}_j} \big(Q_i^\star(j) - Q_i(\tau)\big) + \frac{P_T F}{S} \sum_{\tau=1}^{t} Q_i(\tau).\end{aligned}$$

Hence, from Eqn. (33), we have that

$$\begin{aligned}\sum_{\tau=1}^{t} Q_i(\tau) g_{\tau,i}(x^\star) &\leq \big(1 + \frac{t}{S}\big) F^2 S(S-1) + (1 + \frac{t}{S}) P_T F^2 (S-1) + \frac{P_T F}{S} \sum_{\tau=1}^{t} Q_i(\tau) \\ &\leq F^2 (S + P_T)(S + t) + \frac{P_T F}{S} \sum_{\tau=1}^{t} Q_i(\tau).\end{aligned}$$

Substituting the above bound into Eqn. (32), we have that

$$\sum_{i=1}^{k} Q_i^2(t) \leq \text{Regret}_t'(x^\star) + 2kF^2(S + P_T)(S + t) + \frac{2P_T F}{S} \sum_{\tau=1}^{t} \sum_{i=1}^{k} Q_i(\tau).$$

Plugging in the regret bound of the AdaGrad policy for the surrogate cost functions, the above equation yields

$$\sum_{i=1}^{k} Q_i^2(t) \leq GD\sqrt{2k} \sqrt{\sum_{\tau=1}^{t} \big(\sum_{i=1}^{k} Q_i^2(\tau)\big)} + 2kF^2(S + P_T)(S + t) + \frac{2P_T F}{S} \sum_{\tau=1}^{t} \sum_{i=1}^{k} Q_i(\tau). \tag{37}$$

Using Cauchy-Schwarz inequality, the last term of the above inequality can be upper bounded by

$$\frac{2P_T F \sqrt{kt}}{S} \sqrt{\sum_{\tau=1}^{t} \big(\sum_{i=1}^{k} Q_i^2(\tau)\big)}.$$

Hence, we have the following inequality which holds for any $1 \leq S \leq t$ and $1 \leq \tau \leq t$:

$$\sum_{i=1}^{k} Q_i^2(\tau) \leq \left(GD\sqrt{2k} + \frac{2P_T F \sqrt{kt}}{S}\right) \sqrt{\sum_{\tau=1}^{t} \big(\sum_{i=1}^{k} Q_i^2(\tau)\big)} + 2kF^2(S + P_T)(S + t). \tag{38}$$

Summing up the above inequalities for $1 \leq \tau \leq t$ and defining $Z_t^2 \equiv \sum_{\tau=1}^{t} \sum_{i=1}^{k} Q_i^2(\tau)$, we have:

$$\begin{aligned}Z_t^2 &\leq \left(GD\sqrt{2k} + \frac{2P_T F \sqrt{kt}}{S}\right) t Z_t + 2kF^2 t (S + P_T)(S + t) \\ &\leq 2 \max\left(\big(GD\sqrt{2k} + \frac{2P_T F \sqrt{kt}}{S}\big) t Z_t, 2kF^2 t (S + P_T)(S + t)\right).\end{aligned}$$

The above inequality implies that

$$\begin{aligned}Z_t &\leq 2 \max\left(\big(GD\sqrt{2k} + \frac{2P_T F \sqrt{kt}}{S}\big) t, F\sqrt{kt(S + P_T)(S + t)}\right) \\ &\leq 2 \max\left(\big(GD\sqrt{2k} + \frac{2P_T F \sqrt{kT}}{S}\big) T, FT\sqrt{2k(S + P_T)}\right),\end{aligned} \tag{39}$$

https://doi.org/10.52202/079017-1306

where in the last step, we have used the fact that $t \le T$ and $S \le T$. Now, let us choose $S \equiv P_T^{2/3} T^{1/3}$. With the above choice of S, from the above inequality, we have the following bound for Z_t :

$$Z_t \le 2 \max\left(\left(GD\sqrt{2k} + 2F\sqrt{k} P_T^{1/3} T^{1/6} \right) T, 2F\sqrt{k} P_T^{1/3} T^{7/6} \right) = O(P_T^{1/3} T^{7/6}) + O(T),$$

where we have used the fact that $P_T \le T$. Substituting the above bound in (38), we have for any $1 \le i \le k$ and any $t \le T$:

$$\sum_{i=1}^{k} Q_i^2(t) \overset{(a)}{=} O(P_T^{2/3} T^{4/3}) + O(T) + O(P_T^{2/3} T^{4/3}) = O(P_T^{2/3} T^{4/3}) + O(T),$$

where in (a), we have used the fact that $T \ge S \ge P_T$ in bounding the last term. Hence, we have the following upper bound on the queue lengths for any $1 \le t \le T$

$$\|\boldsymbol{Q}(t)\|_\infty \le \|\boldsymbol{Q}(t)\|_2 = O(P_T^{1/3} T^{2/3}) + O(\sqrt{T}).$$

The final result follows upon appealing to the relation (18). ■

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We give complete proofs of all claims made in the paper (either in the main paper or in the Appendix).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We clearly state the assumptions under which our results hold.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

https://doi.org/10.52202/079017-1306

Answer: [Yes]

Justification: We clearly state the assumptions in the main paper and give complete proofs of all claims (either in the main paper or in the Appendix)

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have publicly released the code so that it can be used to reproduce the main experimental results reported in this paper. Link to the codebase, which includes detailed instructions on how to run the code, has been included in the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have publicly released the code with clear instructions on how to replicate the experimental results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Sufficient details have been provided in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Since the paper deals with worst-case guarantees against adversarial inputs, statistical guarantees are superfluous. Choice of random seeds used in the experiments have been clearly specified in the code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

https://doi.org/10.52202/079017-1306

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute details have been mentioned in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: This work conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This is a foundational work on online learning and does not have a direct societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This theoretical paper does not pose any such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The public dataset used in the experiments has been appropriately cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

https://doi.org/10.52202/079017-1306

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code we have released is well documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-1306