Constrained Online Convex Optimization with Memory and Predictions

Mohammed Abdullah[1,2], **George Iosifidis**[3], **Salah Eddine Elayoubi**[1], **Tijani Chahed**[2]
[1]Université Paris-Saclay, CentraleSupélec, CNRS, L2S, Gif-sur-Yvette, France
[2]Institut Polytechnique de Paris, Télécom SudParis, Palaiseau, France
[3]Delft University of Technology, The Netherlands
{mohammed.abdullah, salaheddine.elayoubi}@centralesupelec.fr,
G.Iosifidis@tudelft.nl, tijani.chahed@telecom-sudparis.eu

Abstract

We study Constrained Online Convex Optimization with Memory (COCO-M), where both the loss and the constraints depend on a finite window of past decisions made by the learner. This setting extends the previously studied unconstrained online optimization with memory framework and captures practical problems such as the control of constrained dynamical systems and scheduling with reconfiguration budgets. For this problem, we propose the first algorithms that achieve sublinear regret and sublinear cumulative constraint violation under time-varying constraints, both with and without predictions of future loss and constraint functions. Without predictions, we introduce an adaptive penalty approach that guarantees sublinear regret and constraint violation. When short-horizon and potentially unreliable predictions are available, we reinterpret the problem as online learning with delayed feedback and design an optimistic algorithm whose performance improves as prediction accuracy improves, while remaining robust when predictions are inaccurate. Our results bridge the gap between classical constrained online convex optimization and memory-dependent settings, and provide a versatile learning toolbox with diverse applications.

1 Introduction

Online Convex Optimization (OCO) is the workhorse model for sequential decisions under adversarial uncertainty. In its basic version, a learner picks an decision x_t from a convex set $\mathcal{X}$ at the start of each round t; an adversary then reveals a convex loss function $f_t : \mathcal{X} \mapsto \mathbb{R}$ and the learner suffers $f_t(x_t)$. The learning is assessed by the metric of regret $\mathcal{R}_T$, i.e., the distance of the accumulated loss from that of the best-in-hindsight decision $x^\star = \arg\min_{x\in\mathcal{X}} \sum_{t=1}^{T} f_t(x)$, and the goal is to ensure sublinear regret. Since its conception (Zinkevich 2003), OCO has been extended to an impressive range of problems (Orabona 2025).

One of these extensions is OCO with memory (OCO-M), where the loss at each round t depends on the previous m decisions of the learner $x_{t-m}, \ldots, x_t$. Data caching with fetching costs, communication systems with reconfiguration delays, user-engagement in recommender systems, investment portfolio selection, and model training in continual learning, are only some of the problems that can be tackled with OCO-M (Anava, Hazan, and Mannor 2015). Further, the recent online non-stochastic control (NSC) framework owes its success to OCO-M, as such stateful systems can be controlled with memory-based functions (Hazan and Singh 2025).

Beyond minimizing losses, real-world systems must often satisfy average constraints of time-varying functions, $g_t(x_t) \leq 0$. This constrained OCO (COCO) extension is attracting growing interest and has several flavors. In some cases the goal is to ensure sublinear long-term violation (LTV), $\sum_t g_t(x_t)$, and in others to bound the cumulative constraint violation (CCV), $\sum_t \max\{g_t(x_t), 0\}$; also, the functions may be known when x_t is decided or not, and they might be static $g_t = g$, stochastically-perturbed or selected by an adversary. Importantly, similarly to losses, the constraints may exhibit memory and depend jointly on the m recent decisions. Examples include energy budget constraints over m-slot windows in smart grid, thermal envelopes that integrate recent power inputs in processors, QoE user metrics capturing service volatility, and battery-health limits tied to cumulative depth-of-discharge, to mention only few. Apart from such operational or resource constraints, LTV and CCV are also relevant for multi-criteria optimization.

It is clear from the above that COCO-M is an important and practical extension of OCO which, nevertheless, remains largely unexplored. This work contributes in addressing this gap by studying several instances of this problem.

Contributions. We study the most compounded version of COCO-M where the constraints are unknown and adversarially varying, and we are interested in CCV, which we denote $\mathcal{V}_T$. We consider two problems, one with memory effects on losses and constraints (COCO-M^2), and another with memory-less constraints (COCO-M). Following a penalty-based relaxation analysis (Zangwill 1967) we design an algorithm that achieves regret $\mathcal{R}_T = \mathcal{O}\big(m^{3/2}\sqrt{T\log T}\big)$, and CCV $\mathcal{V}_T = \mathcal{O}\big(\max\{T^{3/4}, m^{3/2}\sqrt{T\log T}\}\big)$ for COCO-M^2, improving upon the $\mathcal{R}_T, \mathcal{V}_T = \mathcal{O}(T^{2/3}\log^2 T)$ bounds of (Liu, Yang, and Ying 2023) for $T \in [3, 10^{49}]$, the only prior work related to COCO-M^2. For COCO-M, we achieve $\mathcal{V}_T = \mathcal{O}(T^{3/4})$ which improves to $\mathcal{V}_T = \mathcal{O}(m^{3/2}\sqrt{T}\log T)$ for short memory.

We take the next step and study, for the first time, the problem through the lens of *optimistic learning* (OL),

(Rakhlin and Sridharan 2013b). That is, we assume the availability of untrusted predictions about the gradients of forthcoming losses and constraints, and design an algorithm that, under a more restrictive benchmark than in the no-prediction setting, ensures

$$\mathcal{R}_T = \mathcal{O}\left(\sqrt{\mathcal{E}_T(f)}\right),\ \mathcal{V}_T = \mathcal{O}\left(\left(\sqrt{\mathcal{E}_T(g^+)}+m\right)\log T\right),$$

where $\mathcal{E}_T(f)$ and $\mathcal{E}_T(g^+)$ denote the total prediction errors for the loss and constraint functions over the T rounds. These bounds diminish with the predictions' accuracy, becoming $\mathcal{R}_T = \mathcal{O}(\log T)$, $\mathcal{V}_T = \mathcal{O}(m \log T)$ for perfect predictions, and $\mathcal{R}_T = \mathcal{O}(m^2\sqrt{T})$, $\mathcal{V}_T = \mathcal{O}(m^2\sqrt{T}\log(T))$ when the prediction fail maximally. These rates subsume the optimistic COCO bounds *without memory* (Lekeufack and Jordan 2024); and the optimistic *unconstrained* OCO-M bounds (Mhaisen and Iosifidis 2024).

To streamline the presentation of the material, we defer all proofs to the Appendix where, the interested reader, can also find extensive discussion of related work, analysis of special problem cases and numerical experiments.

2 Related work

COCO Bounds. The COCO literature falls in two strands. First, works that assume constraints are *static or known*. The earliest work here (Mahdavi, Jin, and Yang 2012), considers fixed affine constraints; (Chaudhary and Kalathil 2022) study fixed but unknown constraints observed via stochastic feedback; and (Qiu, Wei, and Kolar 2023), (Yu and Neely 2020) address static unknown constraints to get LTV $\mathcal{O}(1)$. This regime is less relevant to our setting but provides useful insights. The second strand considers constraints that are both *time-varying and unknown*. Here, (Guo et al. 2022) match the $\mathcal{O}(\sqrt{T})$ regret and obtain $\mathcal{O}(T^{3/4})$ CCV. Other papers, e.g., (Wang, Wan, and Zhang 2025) focused on projection free algorithms for this problem. (Sinha and Vaze 2024) achieve $\mathcal{O}(\sqrt{T})$ regret with $\mathcal{O}(\sqrt{T}\log T)$ CCV. For a dynamic benchmark, (Wang, Yan, and Liu 2025) provide a bound of $\mathcal{O}(T^{(1+V_x)/2})$ and $\mathcal{O}(T^{V_g})$ violation, where V_x and $V_g \in [0,1]$ quantify the functions variability. However, there are no OCO-M papers with either CCV or LTV.

COCO-M and Non-stochastic Control. Since their introduction by (Agarwal, Hazan, and Singh 2019), NSC methods have used disturbance–action (DAC) policies: the control at round t is a weighted sum of the last m disturbances, and these weights are learned through OCO-M. In that sense NSC is relevant to our study. However, most NSC papers impose *deterministic* constraints on the state and input (Jiang, Hutchinson, and Alizadeh 2025; Li, Das, and Li 2021; Nonhoff and Müller 2021; Yan, Zhao, and Zhou 2023) or assume the constraint at $t+1$ is revealed one round before (Zhou and Tzoumas 2023); in both cases the goal is per–round feasibility rather than an average–sense guarantee. The sole exception is (Liu, Yang, and Ying 2023), who analyze a fully adversarial setting and obtain regret and CCV bounds of $\mathcal{O}(T^{2/3}\log^2 T)$ when the memory length is fixed to $m = \log T$. Our work tightens these bounds to $\mathcal{O}(m^{3/2}\sqrt{T\log T})$ regret and $\mathcal{O}(m^{3/2}\sqrt{T\log T} \vee T^{3/4})$ CCV. Besides, our bounds hold for any memory length m.

Optimism. Look-ahead gradients can compress regret in proportion to their prediction error. In OCO, this is well studied (Rakhlin and Sridharan 2013a; Mohri and Yang 2016; Joulani, Gyorgy, and Szepesvari 2020; Flaspohler et al. 2021). Extending optimism to OCO-M is harder because one must predict farther than the next round; (Mhaisen and Iosifidis 2024) is the only work that tackles this challenge for linear losses in NSC under *imperfect* predictions. In online control, perfect look-ahead predictions can yield exponential improvements in dynamic regret. For example, Yu et al. (Yu et al. 2020) analyze quadratic, time-invariant losses with adversarial disturbances under model-predictive control, and Li et al. (Li, Chen, and Li 2019) study time-varying convex losses without disturbances. In both cases, the dynamic-regret bound decreases exponentially with the length of the prediction window. Predictions may also be viewed through the lens of "context" (Li et al. 2022) in stochastic MDPs with presume finite states and action sets. Lastly, prior works (Yu et al. 2022; Zhang, Li, and Li 2021) assume full-horizon predictions, blocking the learner from using updates; we instead let it incorporate the latest forecasts each round.

Finally, OL in COCO remains surprisingly sparse. (Anderson, Iosifidis, and Leith 2023) proposed a primal-dual algorithm to achieve $\mathcal{R}_T = \mathcal{O}(1)$ and LTV $\mathcal{O}(\sqrt{T})$ under perfect predictions; (Zhang, Guo, and Liu 2025) achieves $\mathcal{R}_T = \mathcal{O}(\sqrt{V_T})$, where V_T captures the aggregate variation of successive gradients, and grantees LTV $= \mathcal{O}(1)$ under the Slater condition, while (Lekeufack and Jordan 2024) used instead a penalty method, attaining $\mathcal{R}_T = \mathcal{O}(\sqrt{\bar{\mathcal{E}}_T(f)})$ and CCV $\mathcal{O}(\log T(\sqrt{\bar{\mathcal{E}}_T(g^+)} + 1))$, where $\bar{\mathcal{E}}_T(f)$ and $\bar{\mathcal{E}}_T(g^+)$ denote the prediction errors. None of these works consider memory in the objective or constraints.

Summary. OCO-M with adversarial time-varying cumulative constraints has only been studied in NSC by (Liu, Yang, and Ying 2023), which we strictly outperform for $T \in [1, 10^{49}]$. The optimistic version of the problem is introduced by this work, and we obtain bounds which for $m = 0$ match the OL COCO results (Lekeufack and Jordan 2024). A summary of the most relevant COCO results is provided in the appendix (Table **??**).

3 Preliminaries

Notation. The diameter of a non-empty, closed and convex decision set $\mathcal{X} \subset \mathbb{R}^d$, is defined as $\|\mathcal{X}\| \doteq \sup_{x,y\in\mathcal{X}} \|x-y\|$, where $\|\cdot\|$ is the ℓ_2 norm. We write $\mathcal{T} := \{m, \dots, T\}$ and at each round $t \in \mathcal{T}$ the learner selects an decision $x_t \in \mathcal{X}$. The *memory length* is denoted with m, and we use:

$$x_{t-m}^t \ \doteq\ (x_{t-m}, \dots, x_t) \in \mathcal{X}^{m+1}, \quad x_{a:b} \doteq \sum_{i=a}^{b} x_i.$$

For a function $f_t(x_{t-m}, \dots, x_t)$ we define its memory-less version $\hat{f}_t(x_t) \doteq f_t(x_t, \dots, x_t)$ with $\hat{f}_t : \mathcal{X} \mapsto \mathbb{R}$, and its prediction is denoted by $\tilde{f}_t$. We use $f_t^+(x) \doteq \max\{f_t(x), 0\}$,

and abuse notation to denote $\nabla f_t(x_t)$ the gradient of f_t at x_t or its subgradient if it is non-differentiable.

Background. In OCO the learner selects an decision $x_t \in \mathcal{X}$ before the convex loss $f_t : \mathcal{X} \mapsto \mathbb{R}$ is revealed. The performance of the learner is measured with the regret:

$$\texttt{OCO}: \quad \mathcal{R}_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x\in\mathcal{X}} \sum_{t=1}^{T} f_t(x), \tag{1}$$

and the learner wishes to achieve $\lim_{T\to\infty} \mathcal{R}_T/T = 0$, for any possible sequence of loss functions $\{f_t\}_t$.

In a more recent extension of this framework, the learner's decisions need additionally to satisfy a time-average (budget) of constraints $g_t : \mathcal{X} \mapsto \mathbb{R}\ \forall t$. In this Constrained OCO (COCO) formulation, the regret is defined as:

$$\mathcal{R}_T^c = \sum_{t=1}^{T} f_t(x_t) - \min_{x\in\mathcal{X}_T} \sum_{t=1}^{T} f_t(x), \tag{2}$$

where the set of eligible decisions is modified to:

$$\mathcal{X}_T = \Big\{ x \in \mathcal{X} \;\Big|\; g_t(x) \le 0, \forall t \in \mathcal{T} \Big\}. \tag{3}$$

Observe that we restrict the benchmark to satisfy the constraints at every round and not on average; a necessary compromise to avoid the impossibility result of COCO, cf. (Mannor, Tsitsiklis et al. 2009). The learner here aims to achieve sublinear regret *and* constraint violation:

$$\texttt{COCO}: \quad \mathcal{R}_T^c = o(T), \quad \mathcal{V}_T^c \triangleq \sum_{t=1}^{T} g_t^+(x_t) = o(T). \tag{4}$$

In this work we are interested in functions with m-length memory where the loss $f_t : \mathcal{X}^{m+1} \mapsto \mathbb{R}$ at each round t, depends on the previous $m>0$ decisions of the learner. Following (Anava, Hazan, and Mannor 2015), the regret for this OCO-M problem is defined as:

$$\texttt{OCO-M}: \mathcal{R}_T^m = \sum_{t=m}^{T} f_t(x_{t-m}^t) - \min_{x\in\mathcal{X}} \sum_{t=m}^{T} f_t(x,\dots,x), \tag{5}$$

where the benchmark is defined using the respective *memory-less* functions $\hat{f}_t(x) \doteq f_t(x,\dots,x)$ that are assumed convex. In this work, we make a further step and introduce the $\texttt{COCO-M}^2$ framework, where the constraints also exhibit memory effects, $g_t : \mathcal{X}^{m+1} \mapsto \mathbb{R}$. We thus define:

$$\mathcal{R}_T^{mc} = \sum_{t=m}^{T} f_t(x_{t-m}^t) - \min_{x\in\mathcal{X}_T^m} \sum_{t=m}^{T} f_t(x,\dots,x), \tag{6}$$

where the set of eligible decisions is:

$$\mathcal{X}_T^m = \Big\{ x \in \mathcal{X} \;\Big|\; g_t(x,\dots,x) \le 0, \forall t \in \mathcal{T} \Big\} \tag{7}$$

and, as in the typical COCO, the learner aims to achieve:

$$\texttt{COCO-M}^2: \ \mathcal{R}_T^{mc},\ \mathcal{V}_T^{mc} \doteq \sum_{t=1}^{T} g_t^+(x_{t-m}^t) = o(T). \tag{8}$$

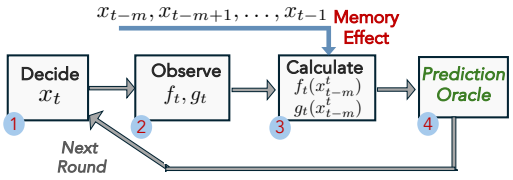

Figure 1: decision stages of $\texttt{COCO-M}$ (with predictions).

Finally, the COCO setting where only the loss functions have memory while the constraints, $g_t : \mathcal{X} \mapsto \mathbb{R}$, are memoryless, is of independent interest. In this case, the definition of regret $\mathcal{R}_T^{mc}$ remains the same, but the benchmark is $x^\star \in \mathcal{X}_T$, and the constraint violation changes to:

$$\texttt{COCO-M}: \mathcal{R}_T^{mc},\ \mathcal{V}_T^{c} \doteq \sum_{t=1}^{T} g_t^+(x_t) = o(T). \tag{9}$$

We denote this problem as $\texttt{COCO-M}$ to distinguish it from the above problem with *double* memory ($\texttt{M}^2$).

Regarding the solution of COCO problems, the main techniques use a simple idea: apply an OCO algorithm on some type of (time-varying) Lagrange function, $\mathcal{L}_t(\cdot)$, that scalarizes the objective and constraints. These techniques can be classified in two broad categories. Those that introduce explicit dual variables μ and perform primal-dual iterations on $\mathcal{L}_t(x,\mu)$, i.e., employ coordinated learning in the primal and dual space, e.g., see (Yuan and Lamperski 2018; Valls et al. 2020). And the second category that draws from penalty methods (Zangwill 1967) and creates again a Lagrange-type function, $\mathcal{L}_t(x)$, where the constraint violation is penalized with some parameter (Liakopoulos et al. 2019; Leith and Iosifidis 2023; Lekeufack and Jordan 2024; Sinha and Vaze 2024). We follow this latter approach.

Learning Model & Assumptions. We consider the most general COCO model where both the loss and the constraint functions may change over time, and in each round they are revealed *after* the learner commits its decision. Regarding the adversary model, we follow (Anava, Hazan, and Mannor 2015; Merhav et al. 2002; Gyorgy and Neu 2014) and consider an oblivious adversary which implies that these functions are determined in advance (i.e., at $t = 0$) but, of course, are not revealed. Finally, we consider full-information feedback for all the arguments of the memory functions; see Fig. 1. We also use the following standard OCO assumptions.

Assumption 1. *$\mathcal{X}$ is closed and convex, with $\|\mathcal{X}\| < \infty$.*

Assumption 2. *For every t, functions $f_t(\cdot)$ and $g_t(\cdot)$ are convex and F-, G-bounded, respectively.*

Assumption 3 (Lipschitz continuous)**.** *For every time $t \in \mathcal{T}$, let $f_t, g_t : \mathcal{X}^{m+1} \mapsto \mathbb{R}$ and $\tilde{f}_t, \tilde{g}_t : \mathcal{X}^{m+1} \mapsto \mathbb{R}$.*

All functions are Lipschitz continuous, i.e., there exist finite constants $L_{t,f}, L_{t,g} \ge 0$ such that, $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}^{m+1}$,

$$\big|f_t(\mathbf{x}) - f_t(\mathbf{y})\big|,\ \big|\tilde{f}_t(\mathbf{x}) - \tilde{f}_t(\mathbf{y})\big| \le L_f \,\|\mathbf{x}-\mathbf{y}\|,$$

$$\big|g_t(\mathbf{x}) - g_t(\mathbf{y})\big|,\ \big|\tilde{g}_t(\mathbf{x}) - \tilde{g}_t(\mathbf{y})\big| \le L_g \,\|\mathbf{x}-\mathbf{y}\|.$$

Finally, the following remark is without loss of generality.

Remark 1. *If we have n constraints $g_{t,k}$ with $k = 1, \ldots, n$, we can follow the standard approach and define $g_t := \max_k g_{t,k}$. Hence, to streamline the presentation, we consider constraints that map to $\mathbb{R}$, not to $\mathbb{R}^n$.*

4 Algorithm Design & Analysis

Our method is closer to penalty COCO techniques, while the memory effect is handled as in (Anava, Hazan, and Mannor 2015), i.e., we perform the analysis using the memory-less functions and lift the results to the original problem. Namely, we use the memory-less versions of $f_t, g_t : \mathcal{X}^{m+1} \mapsto \mathbb{R}$, which are defined on $\mathcal{X}$ as:

$$\hat{f}_t(x) := f_t(x, \ldots, x), \qquad \hat{g}_t(x) := g_t(x, \ldots, x),$$

and following (Lekeufack and Jordan 2024; Sinha and Vaze 2024), we define the (memory-less) surrogate function:

$$\hat{\mathcal{L}}_t(x) \;=\; \hat{f}_t(x) \;+\; \Phi'\big(\hat{V}_t\big)\, \hat{g}_t^+(x), \tag{10}$$

where $\Phi : \mathbb{R}_+ \mapsto \mathbb{R}$ is a non-negative, convex monotonically increasing *penalty* function with $\Phi(0) = 0$, and Φ' its derivative. Observe that Φ is applied to the cumulative memory-less constraint violation which is denoted $\hat{V}_t$ and defined:

$$\hat{V}_t = \begin{cases} \hat{\mathcal{V}}_t^{\mathrm{mc}}, & \text{in } \texttt{COCO-M}^2, \\ \mathcal{V}_t^{\mathrm{c}}, & \text{in } \texttt{COCO-M}. \end{cases}$$

This quantity evolves with time as:

$$\hat{V}_t = \begin{cases} \hat{V}_{t-1} + \hat{g}_t^+(x_t), & \text{in } \texttt{COCO-M}^2, \\ \hat{V}_{t-1} + g_t^+(x_t), & \text{in } \texttt{COCO-M}. \end{cases}$$

The intuition of $\hat{\mathcal{L}}_t$ is that we penalize the violation of constraints at round t, with a penalty commensurate to the accumulated constraint violation. Different from (Lekeufack and Jordan 2024), (Sinha and Vaze 2024), we use:

$$\Phi(\mathrm{V}) = \lambda V^2,$$

with parameter $\lambda > 0$ determined below. This penalty leads to tighter bounds compared to the exponential function in those prior works; we revisit this discussion later.

Now, focusing on the surrogate function, we observe that it operates on decisions drawn from the convex set $\mathcal{X}$, and under Assumptions 1-3 it is convex with bounded gradients for fixed T. Namely, by the triangle inequality, we get:

$$\sup_{t,x} \big\| \nabla \hat{\mathcal{L}}_t(x) \big\| \;\le\; L_f + \Phi'(V_T) L_g = L_f + 2\lambda V_T L_g. \tag{11}$$

Therefore, performing Online Gradient Descent (OGD), i.e.,

$$x_{t+1} = \mathcal{P}_{\mathcal{X}}\Big(x_t - \eta_t \nabla \hat{\mathcal{L}}_t(x_t) \Big), \tag{12}$$

where $\mathcal{P}_{\mathcal{X}}$ is the ℓ_2 projection on $\mathcal{X}$, ensures sublinear regret for these surrogate functions, which we denote $\mathcal{R}_T(\hat{\mathcal{L}})$.

Lemma 1. *Under Assumptions 1-3, performing OGD on* (10) *with step* $\eta_t = \frac{\sqrt{2}\|\mathcal{X}\|}{2\sqrt{\sum_{\tau=1}^{t} \|\nabla \hat{\mathcal{L}}_t(x_t)\|^2}}$, *we obtain:*

$$\mathcal{R}_T(\hat{\mathcal{L}}) \triangleq \sum_{t=1}^{T} \hat{\mathcal{L}}_t(x_t) - \min_{x \in \mathcal{X}_T^{\mathrm{b}}} \sum_{t=1}^{T} \hat{\mathcal{L}}_t(x)$$

$$\le \sqrt{2}\|\mathcal{X}\| \sqrt{\sum_{t=1}^{T} \|\nabla \hat{\mathcal{L}}_t(x_t)\|^2}$$

Algorithm 1: Learning for COCO-M and COCO-M^2

Require: initial history $x_0^{m-1} \in X^m$; dual seed $\hat{\mathrm{V}}_0^{m-1} \leftarrow \mathbf{0}$
for $t = m$ **to** $T-1$ **do**
  Play x_t and observe $f_t(\cdot)$, $g_t(\cdot)$
  Calculate $f_t(x_{t-m}^t)$ and $g_t(x_t)$. *// for COCO-M*
  Calculate $f_t(x_{t-m}^t)$, $g_t(x_{t-m}^t)$. *// for COCO-M^2*
  $\mathrm{V}_t \leftarrow \mathrm{V}_{t-1} + g_t^+(x_t)$ *// dual update for COCO-M*
  $\hat{\mathrm{V}}_t \leftarrow \hat{\mathrm{V}}_{t-1} + g_t^+(x_t, \ldots, x_t)$ *// for COCO-M^2*
  Compute surrogate gradient $\nabla \hat{\mathcal{L}}_t(x_t)$ via (10)
  $x_{t+1} \leftarrow$ solution of (12) *// devise next decision*
end for

where the benchmark set depends on the problem: $\mathcal{X}_T^{\mathrm{b}} = \mathcal{X}_T^m$ for COCO-M^2 and $\mathcal{X}_T^{\mathrm{b}} = \mathcal{X}_T$ for COCO-M.

In the sequel, we explain how this result sets the basis for tackling the COCO-M and COCO-M^2 problems.

Problem COCO-M^2: Double Memory Effect

We start with the more compounded problem COCO-M^2 in (8), having memory in losses and constraints. Without loss of generality, we assume these effects are of the same length m. The solution is summarized in Algorithm 1. In brief, in each round t, the learner commits its decision $x_t \in \mathcal{X}$, observes the current loss and constraint functions and calculates the loss and constraint violation. Then, it updates the constraint violation, calculates the t-round gradient $\nabla \hat{\mathcal{L}}_t(x_t)$ and uses OGD to devise the next decision.

To characterize the performance of this algorithm, we take two steps. First, we utilize the following *regret decomposition* result, which links the regret of the memory-less Lagrangian $\mathcal{R}_T(\hat{\mathcal{L}})$ to the regret of the memory-less loss (denoted $\hat{\mathcal{R}}_T^c$) and the constraint violation $\hat{\mathcal{V}}_T^{mc}$ over the memory-less functions.

Lemma 2 (Regret decomposition (Sinha and Vaze 2024))**.** *For any OCO algorithm, if Φ is a convex increasing function, we have for any $t \ge m$ and $x^\star \in \mathcal{X}_T^m$*

$$\hat{\mathcal{R}}_T^c + \Phi(\hat{V}_T) - \Phi(\hat{V}_m) \le \mathcal{R}_T(\hat{\mathcal{L}}). \tag{13}$$

And, secondly, we transfer these results to the original problem with memory, by observing that we can write:

$$\mathcal{R}_T^c = \underbrace{\sum_{t=m}^{T} f_t(x_{t-m}^t) - f_t(x_t, \ldots, x_t)}_{\text{memory deviation}} + \hat{\mathcal{R}}_T^c \tag{14}$$

$$\mathcal{V}_T^{mc} = \sum_{t=m}^{T} g_t^+(x_{t-m}^t) - g_t^+(x_t, \ldots, x_t) + \hat{\mathcal{V}}_T^{mc}. \tag{15}$$

We bound this memory deviation by exploiting the functions Lipschitz continuity together with the one-step OGD bound. The following theorem summarizes the achieved bounds.

Theorem 1 (Regret and CCV for COCO-M^2)**.** *Assume* (i) *Assumptions 1, 2, 3 hold;* (ii) *the constraint is $g_t(x_{t-m}^t)$, and* (iii) *we use the step* $\eta_t = \frac{\sqrt{2}\|\mathcal{X}\|}{2\sqrt{\sum_{\tau=1}^{t} \|\nabla \hat{\mathcal{L}}_t(x_t)\|^2}}$ *and the penalty*

$\Phi(V)=\lambda V^2$ *with* $\lambda=\frac{1}{\sqrt{T}}$. *Then,* $\forall\, T\geq m$ *it is:*

$$\mathcal{R}_T^{mc}=\mathcal{O}\Big(m^{\frac{3}{2}}\sqrt{T\log(T)}\Big) \tag{16}$$

$$\mathcal{V}_T^{mc}=\mathcal{O}\left(\max\{T^{3/4},m^{\frac{3}{2}}\sqrt{T\log(T)}\right). \tag{17}$$

Discussion. In Theorem 1 we see the effect of m: longer memory amplifies both the regret and CCV, thus the performance degrades as the window grows. The only directly comparable study is (Liu, Yang, and Ying 2023), which assumes $m=\log T$ and achieves strictly inferior bounds (Table **??**) for any $T\in[3,10^{49}]$. When memory vanishes ($m=0$), our bounds collapse to $\mathcal{O}(\sqrt{T})$ for the regret and $\mathcal{O}(T^{3/4})$ for the CCV; which are looser than the bounds of (Sinha and Vaze 2024). This gap arises because we employ a *quadratic* penalty that yields sharper results for the memory problem, whereas (Sinha and Vaze 2024) use an exponential penalty in the memory–free setting. We will show later that, for COCO−M and sufficiently short memory, this gap vanishes as we also use there a different penalty function.

Problem COCO−M

We continue with the case where memory affects only the losses, while the constraints are time-varying memory-less functions. The goal is to ensure sublinear regret $\mathcal{R}_T^{mc}$ and constraint violation $\mathcal{V}_T^c$, see (9). The algorithm is identical to the one above, with only changes in the calculation of the constraint violation and the dual update. In particular, following the same steps, we first invoke the regret–decomposition lemma and translate the bound back to the original problem using solely (14), because, here, the term $\hat{V}_T$ appearing in the lemma coincides with the cumulative violation $\mathcal{V}_T^c$. We thus get the next result.

Theorem 2 (Regret and CCV with memory-free constraint)**.** *Given that* (i) *Assumptions 1, 2, and 3 hold,* (ii) *the constraint is memory-less* ($g_t=(x_t)$)*, and* (iii) *we use the adaptive step* $\eta_t=\frac{\sqrt{2}\|\mathcal{X}\|}{2\sqrt{\sum_{\tau=1}^t\|\nabla\hat{\mathcal{L}}_t(x_t)\|^2}}$ *and the penalty* $\Phi(V)=\lambda V^2$ *with* $\lambda=\frac{1}{\sqrt{T}}$. *Then, for any* $T\geq m$ *we have:*

$$\mathcal{R}_T^{mc}=\mathcal{O}\Big(m^{\frac{3}{2}}\sqrt{T\log(T)}\Big),\ \ \mathcal{V}_T^c=\mathcal{O}\left(T^{3/4}\right). \tag{18}$$

Discussion Relatively to COCO−M^2 (Theorem 1), two points stand out: (i) The CCV no longer depends on the window m; it remains $\mathcal{O}(T^{3/4})$ regardless of the loss functions' memory. (ii) The regret retains the $m^{3/2}\sqrt{T\log T}$ factor. Compared to the regret bound $\mathcal{O}\big(m^{3/2}\sqrt{T}\big)$ of (Anava, Hazan, and Mannor 2015), this carries an extra $\sqrt{\log T}$ factor; this is the price of enforcing time–varying constraints.

Furthermore, it is important to stress that when the memory length satisfies $m\leq\big(T^{1/6}/\log T\big)^{1/3}$, thus $m^{3/2}\sqrt{T\log T}\leq T^{3/4}$, we can replace the quadratic penalty with an exponential one with a tuned λ, to achieve tighter guarantees for CCV: $\mathcal{R}_T^{mc}=\mathcal{O}\big(\sqrt{T}+m^{3/2}\sqrt{T\log T}\big)$ and $\mathcal{V}_T^c=\mathcal{O}\big(\sqrt{T\log T}+m^{3/2}\sqrt{T}\log T\big)$. While $\mathcal{V}_T^c$ depends now on m, this bound is smaller than $\mathcal{O}(T^{3/4})$. In several practical applications indeed m is a constant and much smaller than any expression of the growing T. Hence, for all these problems we can enable these improved rates. Finally, we note that for $m=0$ the bounds reduce to those in (Sinha and Vaze 2024). We provide the details for these cases in the Appendix.

5 Benefiting from Predictions

We next study COCO−M^2 when predictions about forthcoming loss and constraint functions are available see Figure 1 for the decision stages. We study problem COCO−M in the Appendix. This form of learning, known as *Optimistic Learning* (OL), achieves bounds that shrink with the initially-unknown predictions' accuracy. Predictions are widely studied in OCO (Rakhlin and Sridharan 2013a), less so in COCO (Lekeufack and Jordan 2024), and only recently in OCO-M (Mhaisen and Iosifidis 2024) – still, without constraints. In classical OCO, it suffices to predict the next gradient; alas in the presence of memory the forecasting should involve the gradients of the next m slots.

This compounded problem requires modifying the approach in Sec. 4. First, we use the *memory-based* surrogate:

$$\mathcal{L}_t\big(x_{t-m}^t\big)\ =\ f_t\big(x_{t-m}^t\big)\ +\ \Phi'\big(V_{t-m-1}\big)\,g_t^+\big(x_{t-m}^t\big), \tag{19}$$

with $\Phi(V)=\exp(\lambda V)-1$ that has delayed argument V_{t-m-1}. We study the COCO−M^2 problem, where:

$$V_t=V_{t-1}+g_t^+\big(x_{t-m}^t\big),\quad \text{i.e., } V_T=\mathcal{V}_T^{mc}.$$

Secondly, instead of performing OL on (19), we turn the problem on its head and interpret the losses and constraints as having delayed gradients instead of depending on past decisions: at round t the learner selects x_t but is only able to observe the gradient of this decision at the end of $t+m$, i.e., after all functions influenced by x_t are revealed. This change of vantage point allow us to replace the memory effect with a delay effect, which then is handled through a particular version of OL. Before we proceed, we need the following.

Assumption 4 (Separability)**.** *Every memory-based function can be decomposed into components, each depending on a decision from a different round:*

$$f_t\big(x_{t-m}^t\big)=\sum_{i=0}^m f_t^i(x_{t-i}),\qquad g_t\big(x_{t-m}^t\big)=\sum_{i=0}^m g_t^i(x_{t-i}).$$

where f_t^i, g_t^i are defined on $\mathcal{X}$, and i marks that their argument was decided in round $t-i$.

Assumption 5 (Linearity)**.** *Every function* $f_t^i(\cdot)$ *and* $g_t^i(\cdot)$ *is linear, for all* $t\in\mathcal{T}$, $i\in[0,m]$.

Now, the key idea for pivoting to delayed learning is introducing a *forward* loss function to capture the influence of each x_t on the system operation, i.e., $Z_t(x_t)\doteq$

$$\sum_{i=0}^m\mathcal{L}_{t+i}^i(x_t)\triangleq\sum_{i=0}^m f_{t+i}^i(x_t)+\Phi'(V_{t-m-1+i})\,g_{t+i}^{i,+}(x_t).$$

Despite the similarities with $\mathcal{L}_t\big(x_{t-m}^t\big)$, Z_t depends only on round t decision (is memory-less) and includes function components from the next m rounds (has delayed gradient),

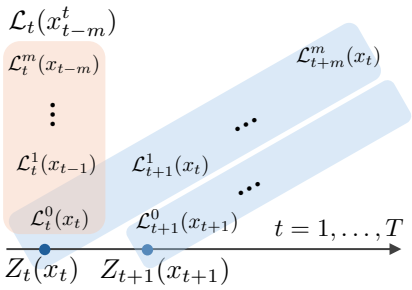

Figure 2: **Diagonal**: $Z_t(x_t)$ depends only on x_t, but includes loss/constraint components from next m rounds. **Vertical**: $\mathcal{L}_t(x_{t-m}^t)$ includes function components only from round t, but depends on all past m decisions.

see Figure 2). Our strategy is to bound the regret of Z_t and translate that bound to the initial problem. A similar construction was used for OCO in (Mhaisen and Iosifidis 2024).

The next new decomposition lemma ties the regret and CCV to memory-based surrogate loss and forward function.

Lemma 3. *For any valid penalty function Φ and under Assumption*[1] *4 it holds:*

$$\begin{aligned}
&\Phi\big(V_T\big) - \Phi\big(V_{m-1}\big) + \mathrm{R}_T^{mc} \\
&\quad\le\ \mathcal{R}_T(\mathcal{L}) + G(m+1)\,\Phi'\big(V_T\big) \qquad (20)\\
&\quad\le\ \mathcal{R}_T(Z) + G(m+1)\,\Phi'\big(V_T\big). \qquad (21)
\end{aligned}$$

where the regret is defined for benchmarks in the set:

$$\mathcal{X}_T^{\mathrm{mp}} \;=\; \big\{\, x \in \mathcal{X} : g_t^i(x) \le 0,\ \forall t \in \mathcal{T},\ i \le m \big\}.$$

Equipped with this result, it suffices to bound $\mathcal{R}_T(Z)$, which we achieve by utilizing predictions to cope with its delayed gradients. In particular, we leverage (Flaspohler et al. 2021) that proposed a suite of delayed–optimistic algorithms (without memory or constraints), and adapt their *Optimistic Delayed AdaF* (ODAF) algorithm, which has the tightest guarantees, for our problem. ODAF relies on FTRL (McMahan 2011) which decides the next decision using all the previous gradients. In the standard delayed-feedback OCO, the entire gradient $\nabla f_t(x_t)$ is revealed in one shot, exactly m rounds after x_t is decided. With a memory window of length m the situation is subtler: each decision x_t affects *some of the components* of the loss functions in rounds $t, \ldots, t+m$; hence, the gradient information arrives piecemeal and may be delayed up to m rounds before fully revealed. Specifically, at round t the learner possesses:

- ✓ *revealed* gradients $\nabla Z_\tau(x_\tau)$, for $\tau = 1, \ldots, t-m-1$ (coming from $x_1, \ldots, x_{t-m-1}$);
- ○ *delayed* gradients that still depend on $x_{t-m}, \ldots, x_{t-1}$;
- × *unseen* gradients that will depend on decision x_t.

With this in mind, we perform OL using an oracle that provides the next forward function gradient:

$$\nabla \tilde{Z}_t(\tilde{x}_t) = \sum_{i=0}^{m} \Big[\nabla \tilde{f}_{t+i}^{\,i}(\tilde{x}_t) + \Phi'\big(V_{t-m-1+i}\big)\,\nabla \tilde{g}_{t+i}^{\,i,+}(\tilde{x}_t)\Big],$$

[1] We trivially set $f_t(\cdot) = g_t(\cdot) = 0$ for $t \le m$ or $t > T$.

Algorithm 2: Optimistic learning for COCO-M^2

Require: initial history $x_0^m \in \mathcal{X}^{m+1}$; dual seed $\hat{V}_0^{m-1} \leftarrow \mathbf{0}$

1: **for** $t = m+1$ **to** $T-1$ **do**
2: Play x_t and observe $f_t(\cdot)$, $g_t(\cdot)$
3: Calculate $f_t(x_{t-m}^t)$, $g_t(x_{t-m}^t)$.
4: $V_t \leftarrow V_{t-1} + g_t^+(x_{t-m}^t)$
5: Compute the prediction error ϵ_{t-m} as in (22).
6: Compute the predictions h_{t+1}
7: Decide decision: $x_{t+1} \leftarrow \mathtt{ODAF}\,(\nabla Z_{1:t-m}, h_{t+1}, \epsilon_{t-m})$.
8: **end for**

as well as the missing past gradients, which we combine into a single *hint* vector: $h_t \doteq$

$$\underbrace{\sum_{i=0}^{m-1}\Bigg(\sum_{j=0}^{m-i-1}\Big[\nabla f_{t-m+i+j}^{\,j} + \Phi'\big(V_{t+i+j-2m-1}\big)\nabla g_{t-m+i+j}^{\,j,+}\Big]}_{\text{available at } t} +$$

$$\underbrace{\sum_{j=m-i}^{m}\Big[\nabla \tilde{f}_{t-m+i+j}^{\,j} + \Phi'\big(V_{t+i+j-2m-1}\big)\nabla \tilde{g}_{t-m+i+j}^{\,j,+}\Big]\Bigg)}_{\text{future predictions}} + \nabla \tilde{Z}_t$$

where we denote $\nabla Z_t(x_t)$ with ∇Z_t, and $\nabla \tilde{Z}_t(\tilde{x}_t)$ with $\nabla \tilde{Z}_t$.

Similarly to other OL algorithms (Rakhlin and Sridharan 2013b), ODAF performs an update (here, using FTRL) whose regularization is scaled by the prediction error. After the decision is committed, the losses f_t and g_t are revealed, rendering available the forward function $Z_{t-m}(x_{t-m})$. At the end of t, the learner therefore knows the gradients ∇Z_τ for all $\tau = 1, \ldots, t-m$ and can evaluate the error of the hint h_{t-m}, which covered the window $\tau = (t-2m)(:t-m)$,

$$\epsilon_{t-m}(Z) = \Bigg\| \sum_{\tau=t-2m}^{t-m} \nabla Z_\tau - h_{t-m} \Bigg\|^2, \qquad (22)$$

with $\mathcal{E}_T(Z) = \sum_{t=m}^{T} \epsilon_t(Z)$ denoting the cumulative prediction error. The loss-function prediction error is then:

$$\epsilon_{t-m}(f) = \Bigg\| \sum_{s=t-2m}^{t-m} \sum_{i=t-s}^{m} \Big(\nabla \tilde{f}_{s+i}^{\,i}(\tilde{x}_s) - \nabla f_{s+i}^{\,i}(x_s)\Big) \Bigg\|^2,$$

where the outer sum runs over the last m decisions $x_s, s = t-m, \ldots, m$, whose delayed contributions have not been fully revealed at $t-m$, and the inner sum selects only those slices that arrive *after* $t-m$ (i.e., delays $i \ge t-s$), and measures the difference between their predicted and true gradients. Similarly, we can define the constraint-function error as $\epsilon_{t-m}(g^+)$, and we denote $\mathcal{E}_T(f) = \sum_{t=m}^{T} \epsilon_t(f)$ and $\mathcal{E}_T(g^+) = \sum_{t=m}^{T} \epsilon_t(g^+)$ the cumulative errors.

Having clarified the prediction and error calculations, we proceed to present the learning mechanism, which is summarized in Algorithm 2. At each round t the learner chooses an decision x_t, observes the realized loss and constraint functions (line 3), and updates the multiplier $\Phi'\big(V_{t-m-1}\big)$ for use at the next step. It then evaluates the prediction–error

(line 5) and *predicts* the forward loss h_{t+1} for the next round; finally, it feeds the ODAF routine (line 7) with the cumulative *revealed* gradients $\nabla Z_{1:t-m}$, the hinth_{t+1}, and the prediction error ϵ_{t-m}, to find the next decision. Due to lack of space, the details for ODAF are deferred to Appendix. Essentially, using the hints and prediction errors designed specifically for our problem, one can readily call the algorithm from (Flaspohler et al. 2021). The next theorem establishes regret and CCV guarantees for this optimistic setting.

Theorem 3. *Under the following conditions:*

- *Assumptions 1, 2, 3, 4 and 5 hold;*
- *The update rule is ODAF;*
- $\Phi(V)=\exp(\lambda V)-1$, *with* $\lambda = \frac{1}{2\left(C\sqrt{\mathcal{E}_T(g^+)}+G(m+1)\right)}$,

the following bounds hold:

$$\mathcal{R}_T^{mc} = \mathcal{O}\left(\sqrt{\mathcal{E}_T(f)}\right), \tag{23}$$

$$\mathcal{V}_T^{mc} = \mathcal{O}\left(\left(\sqrt{\mathcal{E}_T(g^+)}+m\right)\log T\right). \tag{24}$$

Discussion. Let us start by noting that the value of λ depends on the (unknown) prediction error $\mathcal{E}_T(g)$, but we can adjust it online via a *doubling trick* that adds only an extra $\log T$ to the bounds, similar in spirit to (Lekeufack and Jordan 2024); the full analysis is in Appendix. As prediction accuracy improves, the bounds tighten, where under perfect prediction, they reach $\mathcal{O}(1)$ regret and $\mathcal{O}(m\log T)$ CCV. On the other hand, even if the predictors fail completely, the algorithm guarantees regret $\mathcal{O}(m^2\sqrt{T})$ and CCV $\mathcal{O}(m^2\sqrt{T}\log T)$. These bounds are tighter than those of Section 4, but they refer to a more restrictive benchmark $\mathcal{X}_T^{mp} \subset \mathcal{X}_T^m$. What is more, the predictions accuracy does not need to be known in advance, and, further, our solution allows the oracle to update its forecast at every round and benefit from more accurate information whenever available. This flexibility is crucial as predictions can indeed improve with time. Finally, we note that these bounds include as special cases important prior works. When $m = 0$, the rates coincide with those of (Lekeufack and Jordan 2024), who study COCO with predictions but *no* memory. For OCO-M, (Mhaisen and Iosifidis 2024) obtain $\mathcal{O}(1)$ regret under perfect predictions, relying on the delayed-feedback framework of (Flaspohler et al. 2021); our analysis yields the same bound when constraints are omitted. Our work extends these ideas to time-varying *and* memory-dependent constraints.

There are also some important notes in place regarding the Assumptions. First, observe that the surrogate function uses the delayed penalty V_{t-m-1} because at t the freshest *known* value is V_{t-1}. Relying on V_{t-m-1} let us form $\nabla\tilde{Z}_t$ without forecasting the entire future constraint $g_t(x_{t-m}^t)$, and yet it does not affect the bound. Secondly, due to Assumption 5, the gradient of f_{t+i}^i is the constant coefficient vector (independent of x_t), and for the constraint $g_{t+i}^{i,+}(x) = \max\{0,\, a_{t+i}^i x + b_{t+i}^i\}$, we only need to predict the sign of $a_{t+i}^i x_t + b_{t+i}^i$. Thus the predictor only guesses the half-space of x_t, a far weaker requirement than guessing the exact $\tilde{x}_t$. However, this assumption can be lifted if a predictor is available that directly provides an estimate $\tilde{x}_t$ of x_t.

Finally, there is an interesting trade-off between the assumptions about the predictions and the problem structure. In general, satisfying the memory-based constraints $g_t(x_{t-m}^t)$ would require not only forecasting the future loss and constraints, but also their dependence on the yet-unknown decisions $x_{t+1}, \ldots, x_{t+m}$, which in turn demands perfect predictions for a look-ahead horizon $H = \Theta(\log T)$ as in (Yu et al. 2020). For shorter or imperfect forecasts, the sublinear bounds are not guaranteed. To sidestep this, we invoke Assumption 4 and replace the comparator set $\mathcal{X}_T^m$ with $\mathcal{X}_T^{\text{mp}}$. This relaxation allows us recasting it as a memoryless problem with delayed gradients. Consequently, we recover sublinear bounds on both regret an CCV, even under untrusted predictions. The reader might recall that similar concessions about the benchmark set are made in traditional COCO, where $\mathcal{X}$ is reduced to $\mathcal{X}_T$ so as to avoid the impossibility result of (Mannor, Tsitsiklis et al. 2009). Making bolder assumptions for the availability of more informative predictions to learn against an expanded benchmark set is certainly a direction where our framework can be extended.

6 Conclusions

As discussed, COCO-M^2 appears in many real systems, e.g., smart-grid energy budgets, battery-health limits, etc., and directly captures the handling of constraints in NSC. Our penalty method tightens the only prior COCO-M^2 rates of (Liu, Yang, and Ying 2023) ($\mathcal{O}(T^{2/3}\log^2 T)$) to $\mathcal{R}_T = \mathcal{O}\big(m^{3/2}\sqrt{T\log T}\big)$ and $\mathcal{V}_T = \mathcal{O}\big(\max\{T^{3/4},\, m^{3/2}\sqrt{T\log T}\}\big)$, and extends the analysis to the COCO-M case. Moreover, this is the first work to study *untrusted* gradient forecasts for time-varying COCO problems with memory. The proposed optimistic algorithm achieves $\mathcal{R}_T = \mathcal{O}\big(\sqrt{\mathcal{E}_{Tf}}\big)$ and $\mathcal{V}_T = \mathcal{O}\big(\max\{\sqrt{\mathcal{E}_{Tg}}\log T,\, m\log T\}\big)$, matching the results of (Lekeufack and Jordan 2024) without memory and reducing to them when $m = 0$. Indeed, previous COCO, OCO-M and optimistic OCO bounds emerge as special cases of our framework. Finally, as future work, these techniques can be extended to dynamic (adaptive) regret metrics via static-to-dynamic reductions, and with the design of penalties that react to time-varying windows and constraint hardness.

Acknowledgments

The work was supported in part by the Dutch National Growth Fund through the 6G flagship project "Future Network Services" and by the European Commission under Grants 101139270 (ORIGAMI) and 101192462 (FLECON-6G), and in part by the French government through the France 2030 program within the Celtic RAI-6green project.

References

Agarwal, N.; Hazan, E.; and Singh, K. 2019. Logarithmic regret for online control. *Advances in Neural Information Processing Systems (NeurIPS)*, 32.

Anava, O.; Hazan, E.; and Mannor, S. 2015. Online learning for adversaries with memory: price of past mistakes. *Advances in Neural Information Processing Systems (NeurIPS)*, 28.

Anderson, D.; Iosifidis, G.; and Leith, D. J. 2023. Lazy Lagrangians for Optimistic Learning with Budget Constraints. *IEEE/ACM Transactions on Networking*, 31(5): 1935–1949.

Chaudhary, S.; and Kalathil, D. 2022. Safe Online Convex Optimization with Unknown Linear Safety Constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 6175–6182.

Flaspohler, G. E.; Orabona, F.; Cohen, J.; Mouatadid, S.; Oprescu, M.; Orenstein, P.; and Mackey, L. 2021. Online learning with optimism and delay. In *International Conference on Machine Learning (ICML)*, 3363–3373. PMLR.

Guo, H.; Liu, X.; Wei, H.; and Ying, L. 2022. Online Convex Optimization with Hard Constraints: Towards the Best of Two Worlds and Beyond. *Advances in Neural Information Processing Systems (NeurIPS)*, 35: 36426–36439.

Gyorgy, A.; and Neu, G. 2014. Near-Optimal Rates for Limited-Delay Universal Lossy Source Coding. *IEEE Transactions on Information Theory*, 60(5): 2823–2834.

Hazan, E.; and Singh, K. 2025. Introduction to Online Control. arXiv:2211.09619.

Jiang, N.; Hutchinson, S.; and Alizadeh, M. 2025. Online Nonstochastic Control with Convex Safety Constraints. *arXiv preprint arXiv:2501.18039*.

Joulani, P.; Gyorgy, A.; and Szepesvari, C. 2020. A modular analysis of adaptive (non-)convex optimization: Optimism, composite objectives, variance reduction, and variational bounds. *Theoretical Computer Science*, 808: 108–138.

Leith, D. J.; and Iosifidis, G. 2023. Penalized FTRL with Time-Varying Constraints. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, 311–326.

Lekeufack, J.; and Jordan, M. I. 2024. An Optimistic Algorithm for Online Convex Optimization with Adversarial Constraints. *arXiv preprint arXiv:2412.08060*.

Li, T.; Yang, R.; Qu, G.; Shi, G.; Yu, C.; Wierman, A.; and Low, S. 2022. Robustness and Consistency in Linear Quadratic Control with Untrusted Predictions. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(1): 1–35.

Li, Y.; Chen, X.; and Li, N. 2019. Online Optimal Control with Linear Dynamics and Predictions: Algorithms and Regret Analysis. *Advances in Neural Information Processing Systems (NeurIPS)*, 32.

Li, Y.; Das, S.; and Li, N. 2021. Online optimal control with affine constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 8527–8537.

Liakopoulos, N.; Destounis, A.; Paschos, G.; Spyropoulos, T.; and Mertikopoulos, P. 2019. Cautious Regret Minimization: Online Optimization with Long-Term Budget Constraints. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, 3944–3952. PMLR.

Liu, X.; Yang, Z.; and Ying, L. 2023. Online Nonstochastic Control with Adversarial and Static Constraints. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Mahdavi, M.; Jin, R.; and Yang, T. 2012. Trading Regret for Efficiency: Online Convex Optimization with Long Term Constraints. *Journal of Machine Learning Research*, 13(1): 2503–2528.

Mannor, S.; Tsitsiklis, J. N.; et al. 2009. Online Learning with Sample Path Constraints. *Journal of Machine Learning Research*, 10(3).

McMahan, B. 2011. Follow-the-Regularized-Leader and Mirror Descent: Equivalence Theorems and ℓ_1 Regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 525–533. JMLR Workshop and Conference Proceedings.

Merhav, N.; Ordentlich, E.; Seroussi, G.; and Weinberger, M. J. 2002. On Sequential Strategies for Loss Functions with Memory. *IEEE Transactions on Information Theory*, 48(7): 1947–1958.

Mhaisen, N.; and Iosifidis, G. 2024. Optimistic Online Non-Stochastic Control via FTRL. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*.

Mohri, M.; and Yang, S. 2016. Accelerating Online Convex Optimization via Adaptive Prediction. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Nonhoff, M.; and Müller, M. A. 2021. An online convex optimization algorithm for controlling linear systems with state and input constraints. In *2021 American Control Conference (ACC)*, 2523–2528. IEEE.

Orabona, F. 2025. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*.

Qiu, S.; Wei, X.; and Kolar, M. 2023. Gradient-Variation Bound for Online Convex Optimization with Constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 9534–9542.

Rakhlin, A.; and Sridharan, K. 2013a. Optimization, Learning, and Games with Predictable Sequences. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Rakhlin, S.; and Sridharan, K. 2013b. Optimization, Learning, and Games with Predictable Sequences. *Advances in Neural Information Processing Systems (NeurIPS)*, 26.

Sinha, A.; and Vaze, R. 2024. Optimal Algorithms for Online Convex Optimization with Adversarial Constraints. *Advances in Neural Information Processing Systems (NeurIPS)*, 37: 41274–41302.

Valls, V.; Iosifidis, G.; Leith, D.; and Tassiulas, L. 2020. Online Convex Optimization with Perturbed Constraints: Optimal Rates against Stronger Benchmarks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Wang, J.; Yan, B.; and Liu, Y. 2025. Doubly-Bounded Queue for Constrained Online Learning: Keeping Pace with Dynamics of Both Loss and Constraint. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 21135–21143.

Wang, Y.; Wan, Y.; and Zhang, L. 2025. Revisiting Projection-Free Online Learning with Time-Varying Constraints. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(20): 21339–21347.

Yan, Y.-H.; Zhao, P.; and Zhou, Z.-H. 2023. Online Non-Stochastic Control with Partial Feedback. *Journal of Machine Learning Research*, 24(273): 1–50.

Yu, C.; Shi, G.; Chung, S.-J.; Yue, Y.; and Wierman, A. 2020. The power of predictions in online control. *Advances in Neural Information Processing Systems (NeurIPS)*, 33: 1994–2004.

Yu, C.; Shi, G.; Chung, S.-J.; Yue, Y.; and Wierman, A. 2022. Competitive Control with Delayed Imperfect Information. In *2022 American Control Conference (ACC)*, 2604–2610. IEEE.

Yu, H.; and Neely, M. J. 2020. 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.

Yuan, J.; and Lamperski, A. 2018. Online Convex Optimization for Cumulative Constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, 6140–6149.

Zangwill, W. I. 1967. Non-Linear Programming via Penalty Functions. *Management Science*, 13(5): 344–358.

Zhang, H.; Guo, H.; and Liu, X. 2025. On the Power of Optimism in Constrained Online Convex Optimization. In Kwok, J., ed., *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-25)*, 6976–6983. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Zhang, R.; Li, Y.; and Li, N. 2021. On the Regret Analysis of Online LQR Control with Predictions. In *2021 American Control Conference (ACC)*, 699–703. IEEE.

Zhou, H.; and Tzoumas, V. 2023. Safe non-stochastic control of linear dynamical systems. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*.

Zinkevich, M. 2003. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *Proceedings of the International Conference on Machine Learning (ICML)*.