Dynamic Optimistic Constrained OCO with Memory via Delay Equivalence

Mohammed Abdullah
L2S, CentraleSupélec
Université Paris-Saclay, CNRS
France
mohammed.abdullah@centralesupelec.fr

George Iosifidis
Delft University of Technology
The Netherlands
G.Iosifidis@tudelft.nl

Salah Eddine Elayoubi
L2S, CentraleSupélec
Université Paris-Saclay, CNRS
France
salaheddine.elayoubi@centralesupelec.fr

Tijani Chahed
Télécom SudParis
Institut Polytechnique de Paris
France
tijani.chahed@telecom-sudparis.eu

Abstract

Online Convex Optimization with memory and time-varying constraints (COCO-M) is a recently-introduced framework that captures the dynamics of stateful online learning and non-stochastic control with budget constraints. Despite its expressive power, COCO-M remains largely understudied. In this work, we tackle this problem in its full scale, providing an optimistic algorithm that incorporates untrusted predictions to tame universal dynamic regret and cumulative constraint violation. Our analysis leverages a potent reduction from memory to delay and treats state-dependent gradients as delayed feedback. We establish the first bounds that adapt to both environment non-stationarity (path length P_T) and prediction accuracy, while lifting previous restrictive assumptions such as functional separability. En route to these results, we derive findings of independent interest for the memory-to-delay reduction, and for delayed OCO with time-varying constraints. The proposed framework ensures sublinear worst-case guarantees that improve with prediction accuracy, reaching $\mathcal{O}(1)$ regret under perfect predictions, with cumulative constraint violation (CCV) $\mathcal{O}(\sqrt{T})$ for unknown P_T and $\mathcal{O}(\log T)$ for known or online observable P_T , while recovering the optimal COCO rates when memory disappears.

1 Introduction

Online Convex Optimization (OCO) is a powerful framework for sequential decision making: at each round $t = 1 \dots, T$, a learner selects $x_t \in \mathcal{X}$, an adversary reveals a convex loss function $f_t : \mathcal{X} \to \mathbb{R}$, and the learner incurs cost $f_t(x_t)$. Performance is measured via regret, $\mathcal{R}_T$, which compares the learner's cumulative loss against a benchmark sequence $\{u_t\}_t$. Depending on the choice of benchmark, one obtains static regret against a fixed decision $u_t = x^\star, \forall t$; worst-case dynamic regret against per-round minimizers, $u_t = x_t^\star = \arg\min_{x \in \mathcal{X}} f_t(x), \forall t$; or the more general *universal* dynamic regret, which allows comparison to any sequence $\{u_t\}_t$. Over the years, OCO has been extended to a wide range of settings with many practical applications [1, 2].

A key extension is OCO with memory (OCO-M), where the loss at time t depends on a window of past decisions $x_{t-m}, \dots, x_t$. This formulation arises naturally in applications such as recommender systems, portfolio optimization, and continual learning [3]. Importantly, OCO-M underlies the

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

framework of non-stochastic control (NSC), which optimizes stateful systems under adversarial disturbances [4]. In many such problems, the learner must also satisfy time-average constraints of adversarially-varying functions $g_t(x_t) \leq 0$. In this *constrained* OCO (COCO) setting, the goal is to control both regret and cumulative constraint violation (CCV), $Q_T \doteq \sum_t \max\{g_t(x_t), 0\}$. In practice, these constraints often exhibit memory as well, depending jointly on past decisions. Following [5], we refer to this setting as COCO-M.

Despite its many applications in stateful online decision problems and NSC with time-average constraints, COCO-M remains largely unexplored. This gap is not due to a lack of attention, but reflects inherent technical barriers. Indeed, even simplified versions of this problem challenge existing OCO methods: memory induces delay between decisions and their fully-observed feedback, time-varying constraints interact with long-term feasibility requirements, and dynamic benchmarks introduce additional non-stationarity. As a result, current techniques are insufficient to address COCO-M in its full generality, motivating the development of new methodological approaches.

Prior Work. We review briefly here the main lines of pertinent work and refer the reader to Appendix A for a comprehensive discussion. While COCO attracts growing attention, see [6] and references therein, only few works consider dynamic regret. Under this benchmark, the learner must additionally adapt to the variability of comparator, measured by its path length $P_T = \sum_t \|u_t - u_{t-1}\|$. [7] established worst-case dynamic regret $\mathcal{O}(\sqrt{(1+P_T)T})$ and $Q_T = \mathcal{O}(\sqrt{T})$ for time-invariant constraints. For time-varying adversarial constraints, [8] obtained $\mathcal{O}(\sqrt{(1+P_T)T})$ universal dynamic regret and $Q_T = \mathcal{O}(T^{3/4})$. Subsequent works [9, 10] refined these guarantees under different assumptions, and [11] provided the first bounds for universal dynamic regret and CCV. However, none of these important works considers memory. The few studies of dynamic regret in memory-dependent problems are confined to the special case of switching costs (one-round memory) [12], [13], or to NSC without time-average constraints [14], [15], [16]. Thus, dynamic regret for COCO-M, and more broadly for constrained NSC, remains essentially open.

From a methodological standpoint, this work brings together memory, delay, and optimism. Existing static optimistic memory-to-delay reductions [5, 17] require separability across memory variables, which in COCO-M yields a restrictive benchmark and prevents a direct application to constrained NSC; we drop this assumption by introducing a more general memory-to-delay reduction. Learning with delay is typically treated with multiple interleaved learners, e.g., [18], while [19] unified delay and optimism by interpreting the delayed gradients as "bad hints". This line of work focuses on static regret, with few exceptions such as [20] which does not consider constraints. Finally, optimistic learning, i.e., incorporating potentially inaccurate predictions in OCO, has been studied with dynamic regret in [21] and [22], but without delay or memory effects. In summary, current methods do not provide the tools required to address even intermediate variants of the problem considered here.

Contributions. Motivated by this gap, we study COCO-M under universal dynamic regret while leveraging (possibly inaccurate) predictions of the losses and constraints. We propose a meta-learning framework that combines an adaptive hedge-based meta-learner with a collection of optimistic delayed Online Gradient Descent (OGD) base learners, each tuned to a different value of the (unknown) path length P_T. Both the meta-learner and the base learners are bespoke. Moreover, a key idea in our work is that learning with memory can be effectively mapped to learning with delay; we formalize this connection and employ it to study the targeted COCO-M problem. To the best of our knowledge, this is the first work to address this problem, and its contributions can be summarized as follows:

- Develops an optimistic delayed OCO algorithm and derives prediction-dependent universal dynamic regret guarantees (Theorem 1). This is the algorithm engine of the framework.
- Extends this framework to constrained OCO with time-varying constraints, obtaining regret and cumulative constraint violation bounds for unknown and known P_T (Theorems 2 and 3).
- Establishes a reduction from memory-dependent OCO to delayed OCO via partial gradient aggregation, without imposing function separability (Lemma 2). Building on this, it employs delayed OCO to deliver universal dynamic regret and CCV bounds for COCO-M (Corollary 1).

In all cases, our optimistic COCO-M bounds subsume the best-known results of simpler problem versions, e.g., without memory (or delay) and/or without predictions; and, when dropping the optimism, we obtain fully-adaptive COCO-M bounds, covering an important gap in literature. The details of proofs and intermediate results (some of independent interest) are included in Appendices.

2 Background & Preliminaries

Notation. Let $\mathcal{X} \subset \mathbb{R}^d$ be a nonempty, closed, bounded and convex set, with ℓ_2-norm diameter $D \doteq \sup_{x,y\in\mathcal{X}} \|x-y\|$. We write $[T] := \{1,\dots,T\}$ and, when needed, $[T_m] := \{m,\dots,T\}$, where m is the memory window, and use the shorthand notation:

$$x_{t-m}^t \doteq (x_{t-m},\dots,x_t) \in \mathcal{X}^{m+1}, \quad x_{a:b} = \sum_{i=a}^{b} x_i, \quad \text{and} \quad z^+ \doteq \max\{z,0\},\ z\in\mathbb{R}.$$

We denote with $\tilde{f}_t$ the prediction of the function f_t and with $\nabla_{x_s} f_t(\cdot)$ the partial gradient (a subgradient) of f_t with respect to x_s. Also, Δ_N is the N-dimensional unit simplex.

Metrics & Benchmarks. At each round $t\in[T_m]$, the learner decides $x_t\in\mathcal{X}$ and incurs loss $f_t(x_{t-m}^t)$ and constraint violation $g_t(x_{t-m}^t)$, where $f_t, g_t : \mathcal{X}^{m+1} \mapsto \mathbb{R}$. We compare the learner against a comparator sequence $u_1^T \doteq (u_1,u_2,\dots,u_T)$ that belongs to the feasible benchmark set:

$$\mathcal{U}_T := \left\{u_1^T \in \mathcal{X}^T : g_t\big(u_{t-m}^t\big) \le 0,\ \forall t\in[T_m]\right\},$$

where we use the convention $x_0 = u_0 = 0$, to streamline presentation. Observe that $\mathcal{U}_T$ is confined to satisfy the constraints on a per-round basis. This is necessary so as to avoid the impossibility result of [23] and is in line with prior COCO works (see Appendix Sec. A). Still, $\mathcal{U}_T$ expands the static-regret COCO-M benchmark set in [5]; an advantage of our new memory-to-delay reduction in Sec. 5.

The learner is assessed using the metrics of dynamic regret and cumulative constraint violation[1]:

$$\mathcal{R}_T(f) \doteq \sum_{t=m}^{T} f_t(x_{t-m}^t) - \sum_{t=m}^{T} f_t(u_{t-m}^t), \qquad \mathrm{Q_T} \doteq \sum_{t=m}^{T} g_t^+(x_{t-m}^t). \tag{1}$$

Our goal is to ensure both $\mathcal{R}_T(f)$ and Q_T are sublinear in T and the comparator's path length $P_T \doteq \sum_{t=1}^{T-1} \|u_{t+1}-u_t\|$. Here we study *universal* dynamic regret, which is measured against all feasible comparator sequences [24]; this is a more general (and more challenging) metric [11].

Meta-learning with Hedge. Optimizing for dynamic regret requires knowledge of P_T or adaptation at runtime. A key idea here is to use a meta-learner that hedges against a set of $N = \left\lceil \frac{1}{2}\log_2(1+DT)\right\rceil + 1$ base learners, that are implemented with different learning rates in a way that ensures at least one of them is tuned within a constant factor of $\sqrt{1+P_T}$ [25]. Unfortunately, in our problem the hedging is compounded by the memory effects that introduce delay in discerning the best base learner.

Memory-to-Delay. In memory-less OCO, the loss gradient is available immediately; however, in OCO-M the impact of x_t is dispersed across the next m rounds; thus, memory induces feedback delay. To state this connection in a form that also applies to constrained problems, let $L_t : \mathcal{X}^{m+1} \mapsto \mathbb{R}$ be a memory-dependent loss. In the unconstrained case, $L_t = f_t$, while in the constrained case L_t is the penalized surrogate loss. We define the *credit gradient* h_s as the sum of partial gradients of all functions influenced by x_s, i.e., $h_s = \sum_{t\in I_s} \nabla_{x_s} L_t(x_{t-m}^t)$[2], where $I_s = \{t\in[T_m], t-m\le s\le t\}$. Clearly, h_s cannot be computed until round $s+m$, which reveals a feedback delay of m rounds in OCO-M. Using this observation (formalized in Sec. 5), we shift our focus from the per-round loss L_t to the per-decision credit h_s, i.e., from memory to a delayed feedback problem. A similar transformation, at the function level, was used by [5] and [17]; we discuss the differences in Sec. 5.

Optimism. Another key idea in this work is optimism, i.e., incorporating in learning (possibly inaccurate) predictions of future gradient vectors and/or actions [26]. Optimistic learning turns non-optimistic regret bounds of the form $\mathcal{O}(T^k)$, for some $k\in[0,1]$, to $\mathcal{O}(E_T^k)$, where E_T is the accumulated prediction error. These bounds hence range from $\mathcal{O}(1)$ for accurate predictions, to the non-optimistic bound $\mathcal{O}(T^k)$ in the worse case. This improvement happens, typically, through prediction errors-adaptive regularization, e.g., see [27, 28], and do not require prior information of the prediction accuracy. Finally, an important connection between optimism and feedback delay in OCO, which we also leverage here, was recently introduced in [19].

We rely on the following standard OCO assumptions throughout the paper.

[1] If there are multiple constraints $g_{t,k}$, $k=1,\dots,n$, one may define $g_t(\cdot) \doteq \max_{k\in[n]} g_{t,k}(\cdot)$.

[2] We use ∇ throughout for simplicity; if the functions are not differentiable, ∇ should be interpreted as a subgradient ∂

Assumption 1. *The decision set $\mathcal{X}$ is nonempty, closed, bounded, and convex, with $\mathbf{0} \in \mathcal{X}$ and diameter D.*

Assumption 2. *For every $t \in [T_m]$, the functions $f_t, g_t : \mathcal{X}^{m+1} \to \mathbb{R}$ are convex in their arguments and uniformly bounded in absolute value by G.*

Assumption 3. *For every $t \in [T_m]$, the gradients of f_t and g_t, as well as their predicted counterparts, are uniformly bounded in norm by G.*

Finally, we consider an oblivious adversary, in line with [3, 29, 30] and others. Hence, the loss and constraint functions are determined at $t=0$, but are not revealed before each round t is realized.

Remark 1. *Unlike prior optimistic OCO/COCO-M works [5, 17], Assumptions 1-3 do not impose separability of the memory-dependent functions. Further, we do not impose Slater-type conditions.*

Roadmap. Given the identified memory-to-delay equivalence, we start in Section 3 by studying universal dynamic regret for the optimistic delayed OCO problem. In Section 4, we extend this analysis to the case of time-varying adversarial constraints (COCO with delay). Finally, Section 5 formalizes the memory-to-delay connection and shows how these results can be transferred to the targeted optimistic COCO-M problem.

3 Dynamic Optimistic OCO with Delay

We start with the problem of optimistic OCO with feedback delay. We use a decision-time convention that makes the delay explicit and facilitates making later the connection with memory. The decision x_t is played at time t, while its associated feedback vector h_t is fully revealed only after m rounds. Equivalently, at round t, the learner receives the delayed feedback h_{t-m}. Abusing slightly the notation (in order to facilitate presentation in Section 5), we redefine $L_{t+m} : \mathcal{X} \mapsto \mathbb{R}$ to denote the convex loss at round $t+m$ which is associated (due to delay) with the earlier decision x_t. We also redefine $h_t \in \nabla L_{t+m}(x_t)$. This notation aims to emphasize the delayed feedback. By convexity:

$$L_{t+m}(x_t) - L_{t+m}(u_t) \leq \langle h_t, x_t - u_t \rangle, \quad \forall\, u_1^T \in \mathcal{X}^T$$

Thus, we control the regret $\mathcal{R}_T(\ell) := \sum_{t=1}^T \langle h_t, x_t - u_t \rangle$ of the delayed linear losses $\ell_t(x) := \langle h_t, x \rangle$.

We study the challenging case where the comparator path length P_T is unknown, and employ a meta-learning construction with N base learners, similarly in spirit to [25]. At each round t, every base learner $i \in [N]$ proposes an action $x_t^{(i)} \in \mathcal{X}$, and the meta-learner plays a *fused* action:

$$x_t = X_t w_t, \qquad \text{where } X_t := [x_t^{(1)}, \ldots, x_t^{(N)}] \qquad \text{and} \quad w_t \in \Delta_N.$$

For any fixed base learner $i \in [N]$, the delayed linearized regret decomposes as:

$$\mathcal{R}_T(\ell) = \underbrace{\sum_{t=1}^T \langle h_t, x_t - x_t^{(i)} \rangle}_{\text{Term (I)}} + \underbrace{\sum_{t=1}^T \langle h_t, x_t^{(i)} - u_t \rangle}_{\text{Term (II)}}.$$

Since $x_t = X_t w_t$ and $x_t^{(i)} = X_t e_i$, where e_i is the i-th vertex of Δ_N, Term (I) becomes:

$$\text{Term (I)} = \sum_{t=1}^T \langle M_t, w_t - e_i \rangle, \qquad \text{with} \qquad M_t := X_t^\top h_t.$$

This way, we obtain a meta-level problem with losses M_t and a base-level problem with losses h_t; and we treat both problems with delayed learning and optimism.

Unfortunately, standard dynamic-regret meta-learning schemes such as [25] do not handle delayed expert losses, while delayed/adaptive Hedge methods such as [2, 31] do not provide the optimistic prediction-dependent form needed here. Therefore, we build a bespoke meta-learning construction, leveraging and extending the delay-as-optimism framework of [19]: the missing delayed gradients are handled through predictions (or, *hints*) rather than worst-case gradient bounds; and we use a delayed-upper-bound (DUB) tuning to set the regularization from revealed quantities, thus avoiding over-regularization and preserving the benefit of predictions.

Let us clarify first the prediction structure which is essentially common for both problems. At each round t, after observing h_{t-m}, the block $h_{t-m+1},\dots,h_t$ remains unavailable to the learner. To overcome this issue, the algorithm uses the prediction block $\widetilde{h}_{t-m+1}^{t+1}$, which includes both this missing information and the next hint $\widetilde{h}_{t+1}$. That is, at each round t, the learner needs to predict the gradient of $L_{t+1},\dots,L_{t+m+1}$, and also the x_{t+1}, as it appears in the function of round $(t+m+1)$. We emphasize that this is not the only possible way to construct the hint. As in prior optimistic works, predicted gradients may be evaluated instead at current decisions (which are known); see [19, 21, 26]. More generally, the algorithm does not require these quantities to be predicted separately: it only needs a bounded vector $\widetilde{h}_s$ that serves as a prediction of h_s.

Now, focusing on Term (I), the meta-learner leverages delayed optimistic FTRL/AdaHedge to M_t, and updates its fusing weights by executing:

$$w_{t+1} = \arg\min_{w\in\Delta_N} \left\{ \left\langle w, M_{1:t-m} + \widetilde{M}_{t-m+1:t+1} \right\rangle + \eta_{t+1}^{\mathrm{M}} R_\Delta(w) \right\}. \tag{2}$$

where we defined $\widetilde{M}_t = X_t^\top \widetilde{h}_t$. Here we used an entropic regularizer $R_\Delta(w) = \sum_{i=1}^N w_i \log w_i$, and the DUB regularization $\eta_{t+1}^{\mathrm{M}} = \eta_{t+1}(\alpha_{\mathrm{M}})$ from [19], which employs the formula:

$$\eta_{t+1}(\alpha) = \frac{2}{\alpha} \max_{j\le t-m-1} a_{j-m+1:j} + \frac{1}{\alpha}\sqrt{\sum_{r=1}^{t-m}(a_r^2 + 2\alpha b_r)}. \tag{3}$$

Parameters $a_t, b_t \ge 0$ and $\alpha > 0$ are problem-specific, and for our meta-learning are defined as:

$$\begin{aligned} &a_t^{\mathrm{M}} = 2\|\bar{e}_t(M)\|_\infty, \quad b_t^{\mathrm{M}} = \mathrm{huber}(\|\bar{e}_t(M)\|_\infty, \|\bar{n}_t(M)\|_\infty), \quad \alpha_{\mathrm{M}} = \log N, \quad \text{where} \\ &\bar{e}_t(M) := M_{t-m:t} - \widetilde{M}_{t-m:t}, \quad \bar{n}_t(M) := M_{t-m} + \widetilde{M}_{t-m+1:t+1} - \widetilde{M}_{t-m:t}, \quad \text{and} \\ &\mathrm{huber}(x,y) := \frac{1}{2}x^2 - \frac{1}{2}(x-y)_+^2 \le \min\left\{\frac{1}{2}x^2, xy\right\}. \end{aligned} \tag{4}$$

There are few interesting remarks here. First, the meta-learner operates on linear losses, so w_{t+1} itself does not need to be predicted. Second, before choosing w_{t+1}, the meta-learner already knows X_{t+1} and hence the only unknown quantity is the future feedback h_{t+1}. Finally, observe that we need to predict only the gradient vectors, not matrices X_t (see Lemma 3 in Appendix).

Turning our attention to Term (II), each base learner i is tuned to a path-length guess ρ_i, using the dyadic grid of possible values, as in [25]. However, here the base learners need also to handle the feedback delay and include the predictions. To enable this, we rely again on the toolbox of [19] which we modify for dynamic regret. This extension involves a non-trivial analysis due to the interaction of the prediction errors with the comparator variation (see Theorem 4 and Corol. 2 in Appendix); yet, interestingly, its implementation requires minimal changes. Namely, each base learner runs:

$$x_{t+1}^{(i)} = \arg\min_{x\in\mathcal{X}} \left\{ \left\langle h_{t-m} + \widetilde{h}_{t-m+1:t+1} - \widetilde{h}_{t-m:t}, x \right\rangle + \frac{\eta_{t+1}^i}{2}\|x - x_t^{(i)}\|_2^2 \right\}, \tag{5}$$

$$\eta_{t+1}^i = \eta_{t+1}(\alpha_i) \text{ with } a_t^i = D\|\bar{e}_t(h)\|_2, \;\; b_t^i = \mathrm{huber}(\|\bar{e}_t(h)\|_2, \|\bar{n}_t(h)\|_2), \;\; \alpha_i = D(D+2\rho_i). \tag{6}$$

Alg. 1 summarizes the execution of this meta-learning framework, and the next theorem its regret.

Theorem 1 (Delayed optimistic OGD)**.** *Let h_t be the delayed gradient feedback and let $\tilde{h}_t$ be its prediction. For any comparator sequence $u_1^T \in \mathcal{U}_T$, Algorithm 1 ensures universal dynamic regret:*

$$\mathcal{R}_T(L) = \mathcal{O}\left((m+1)^{3/2}\left(1 + \sqrt{1+\log N} + \sqrt{1+P_T}\right)\sqrt{\sum_{t=1}^T \|h_t - \tilde{h}_t\|^2} \right).$$

Proof outline. By convexity, the delayed loss regret is upper bounded by the delayed linearized regret, $\mathcal{R}_T(L) \le \mathcal{R}_T(\ell) =$ Term (I) + Term(II). Term (I) is the meta-regret of Alg. 1, and is controlled by the delayed optimistic FTRL/AdaHedge update of [19, Theorem 12] with entropic regularizer, see Lemma 3. Term (II) is the delayed dynamic regret of the best-tuned base learner, Alg. 2, and is controlled by Theorem 4 and Corollary 2. Finally, the path-length grid, following the meta-learning idea of [25], ensures one base learner is tuned to P_T up to constants.

Algorithm 1 Universal Meta-Algorithm for Delayed Optimistic OCO

Require: Grid of path-length scales $\{\rho_i\}_{i=1}^N$, initial weights $w_m = \mathbf{1}/N$.
Require: Constraint parameters V, Φ, and λ.
Activate N base learners by running Algorithm 2 with scale ρ_i, for each $i \in [N]$.
for $t = m, \dots, T$ **do**
  Receive $x_t^{(i)}$ from each base learner and set $X_t = [x_t^{(1)}, \dots, x_t^{(N)}]$.
  Play the convex combination $x_t = X_t w_t$.
  Observe the delayed feedback h_{t-m} and the hints $\widetilde{h}_{t-m+1}^{t+1}$.
  Send h_{t-m} and $\widetilde{h}_{t-m+1:t+1}$ to all base learners.
  Receive the next predictions $x_{t+1}^{(i)}$ and set $X_{t+1} = [x_{t+1}^{(1)}, \dots, x_{t+1}^{(N)}]$.
  Define the delayed meta-loss and optimistic hints

$$M_{t-m} = X_{t-m}^\top h_{t-m}, \qquad \widetilde{M}_s = X_s^\top \widetilde{h}_s, \quad s = t-m+1, \dots, t+1.$$

  Compute η_{t+1}^{M} using (3) with (4).
  Update w_{t+1} using (2).
end for

Algorithm 2 Delayed Optimistic OGD Base Learner

Require: Path-length scale ρ, initial points $x_0, \dots, x_m$.
Require: Constraint parameters V, Φ, and λ.
for $t = m, \dots, T$ **do**
  Submit $x_t^{(i)}$ to the meta-algorithm.
  Receive the delayed feedback h_{t-m} and the hints $\widetilde{h}_{t-m+1:t+1}$.
  Compute η_{t+1}^{ρ} using (3) with (6).
  Update $x_{t+1}^{(i)}$ using (5).
end for

Discussion. The factor $(m+1)^{3/2}$ should be interpreted as the cost of maintaining a prediction-adaptive bound under delayed feedback. If predictions are ignored and a uniform gradient bound is available, one can instead use fixed worst-case tuning and obtain a delayed dynamic-regret bound with linear dependence on $m+1$. We refer to Appendix B.1.2 for a more detailed discussion of this dependence. However, such a worst-case tuning would lose the benefit of accurate predictions and is less suitable for the constrained setting, where the surrogate gradients may not admit a useful uniform Lipschitz bound. Under worst-case predictions accuracy, Theorem 1 guarantees regret $\mathcal{O}\big(m^{3/2}\sqrt{P_T}\sqrt{\sum_{t=1}^T \|h_t\|^2},\big)$, which matches the lower bound of [25] for $m = 1$, and the delay dependency (equivalently, memory length) of [3] (static regret; no predictions). Interestingly, under perfect predictions we obtain regret $\mathcal{R}_T(L) = \mathcal{O}(1)$. The bound holds for any value of m and recovers (up to constants) the respective no-memory bound $\mathcal{O}(\sqrt{(1+P_T)E_T(f)})$ of [22] in the known P_T case, where $E_T(f)$ is the accumulated prediction error.

Technically, the main difference from [19] is that, under a dynamic benchmark, prediction errors interact with the movement of comparator. We control this interaction using an auxiliary Young-type argument and show it does not change the desired prediction-adaptive order, except through the path-length dependence. We kindly refer the reader to Appendix B for the derivations and discussion. From a computational perspective, we use multiple base learners to adapt to unknown P_T. Still, unlike previous delayed OCO frameworks e.g., [18], we use only one copy of the algorithm for each P_T value, instead of $m+1$ parallel copies that would have been trained only on a fraction of data.

4 Handling Time-average Adversarial Constraints

This section extends the above analysis to handle time-average adversarially-varying constraints, i.e., to *constrained* optimistic OCO with delayed feedback. To facilitate presentation, we use the shifted notation from Section 3: the feedback received at round t corresponds to decision x_{t-m}. COCO methods perform OCO over a time-varying Lagrangian-type surrogate using either a primal–dual method, e.g., [32], or a penalty method where the accumulated constraint violation, Q_t, is penalized

directly in a scalarized surrogate loss $L_t(x)$ [9, 33–35]. We follow the latter approach and define:

$$L_t(x_{t-m}) = V f_t(x_{t-m}) + \Phi'\big(Q_{t-m-1}\big)\, g_t^+(x_{t-m}), \qquad Q_t = Q_{t-1} + g_t^+(x_{t-m}), \tag{7}$$

where $V>0$ is a trade-off parameter, and Φ an increasing convex penalty function. Equivalently, in the decision-time notation of Sec. 3, one sets $s=t-m$, so that the feedback associated with x_s is the gradient of the surrogate loss revealed at $s+m$. The penalty coefficient is evaluated at Q_{t-m-1}, which is available when x_{t-m} is decided. This avoids the need to predict the future queue evolution when forming the gradient prediction. As a result, the prediction error of the surrogate gradient separates naturally into the prediction error of the objective and that of the constraint violation.

We define the prediction errors at the level of the delayed gradients. For every $s \in [T]$, set

$$\Delta_s^f := \nabla f_{s+m}(x_s) - \nabla \tilde{f}_{s+m}(\tilde{x}_s), \qquad E_T(f) := \sum_{s=1}^{T} \|\Delta_s^f\|_2^2,$$

The term $E_T(g^+)$ is defined analogously. At t, all past decisions are known, thus suffices to predict only $\widetilde{x}_{t+1}$. We now recall the decomposition connecting the surrogate regret to $\mathcal{R}_T(f)$ and Q_T.

Lemma 1 (adapted from [5, 11])**.** *For any differentiable, convex and increasing penalty Φ, it is:*

$$\Phi(Q_T) - \Phi(Q_{m-1}) + V\mathcal{R}_T(f) \le \mathcal{R}_T(L) + G(m+1)\Phi'(Q_T), \tag{8}$$

where the dynamic regret is computed with respect to comparator sequences $u_1^T \in \mathcal{U}_T$.

Remark 2. The queue is maintained globally using the actually played trajectory. After the learner plays $x_t = X_t w_t$, the queue is updated from the realized constraint violation, and the corresponding surrogate feedback h_t is constructed and fed to all base learners and to the meta-learner. Thus, the base learners do not maintain separate counterfactual queues.

4.1 Unknown Path Length: Regret and CCV Bounds

We first consider the case where P_T is unknown. We apply the meta-base delayed optimistic algorithm developed in Section 3 to the surrogate loss defined in (7).

Theorem 2 (Unknown-path-length bounds with CCV-oriented tuning)**.** *Let $u_1^T \in \mathcal{U}_T$ be any feasible comparator sequence with path length P_T. Using a quadratic penalty $\Phi(Q) = Q^2$, and*

$$V := 4(m+1)^{3/2}\left(1+\sqrt{1+\log N}\right)^2 \sqrt{E_T(g^+)} + 2G^2(m+1) + Q_{m-1}^2.$$

Then, under the delayed optimistic meta-base algorithm 1 the following bounds hold

$$\mathcal{R}_T(f) \le \mathcal{O}\Big(A\sqrt{E_T(f)} + (m+1)^{3/2}(1+P_T)\sqrt{E_T(g^+)} + (m+1)\Big), \qquad \textit{and}$$

$$Q_T \le \mathcal{O}\Big(A\sqrt{T}\,E_T(g^+)^{1/4} + (m+1)^{\frac{5}{4}}\sqrt{T}\Big) \;\textit{ with }\; A = \mathcal{O}\Big((m+1)^{\frac{3}{2}}\left(1+\sqrt{1+\log N}+\sqrt{1+P_T}\right)\Big)$$

We note that P_T appearing in the bound on Q_T can be taken as the smallest feasible path length. In particular, it can be zero whenever there exists a fixed decision that satisfies all constraints.

Discussion. Theorem 2 gives an optimistic delayed-COCO guarantee under a universal dynamic benchmark. Under worst-case predictions, we obtain an $\mathcal{O}((1+P_T)\sqrt{T})$ regret bound and an $\mathcal{O}\Big(\max\{T^{3/4}, \sqrt{(1+P_T)T}\}\Big)$ CCV bound, recovering the best known rates of [11] when $m=0$. Under perfect predictions, our bounds reduce to $\mathcal{O}(1)$ regret and $\mathcal{O}(\sqrt{T})$ CCV. We note that [7] achieves $\mathcal{O}(\sqrt{(1+P_T)T})$ regret and $\mathcal{O}(\sqrt{T})$ CCV for time-invariant constraints. Our perfect-prediction regime recovers the same CCV order while handling time-varying constraints. In the known P_T setting, we get better bounds as shown later. As discussed in Appendix B.1.2, the extra $\sqrt{m+1}$ factor in the simplified delayed-regret bound can be absorbed by the meta-complexity or path-length terms when $\log N$ or P_T dominates m, leading to linear dependence on the delay. Finally, since V depends on $E_T(g^+)$, we use a doubling trick to learn this quantity online, introducing at most an $\mathcal{O}(\log T)$ overhead, as is standard in optimistic COCO, see, e.g., [5, 9].

Remark 3 (Penalty choice and path-length dependence)**.** In COCO, the surrogate loss must do two things: track the comparator for regret and penalize constraint violations to control the queue. This coupling between tracking and feasibility is what leads to a higher dependency on P_T. If P_T is known, the penalty parameter can be tuned to the true scale of comparator motion, allowing balancing regret and queue control. When P_T is unknown, this tuning is not available. Moreover, the usual meta-learning approach cannot be applied directly to the penalty parameter, since different penalty choices generate different surrogate loss sequences. Thus, one must use a single surrogate construction that works uniformly; this leads to a more conservative trade-off.

Also, the $T^{3/4}$-type term in the CCV bound comes from the same limitation. We use the quadratic penalty $\Phi(Q) = Q^2$, which is the smallest polynomial penalty that still yields sublinear queue growth. Stronger penalties, such as $\Phi(Q) = Q^\beta$ with $\beta > 2$, or exponential penalties, may improve the CCV exponent, but they also amplify the path-length contribution in the surrogate regret bound. Thus, improving CCV in this way comes at the cost of worse dependence on P_T. We study next the known-path-length regime, where this trade-off can be tuned more carefully.

4.2 Known Path Length

When P_T is known we can tune the surrogate loss directly to the scale of the comparator motion via the penalty Φ. Hence, we can direcly run only one base learner, Algorithm 2, with $\rho = P_T$.

Theorem 3 (Known-path-length case with exponential penalty)**.** *For any $u_1^T \in \mathcal{U}_T$ with path length P_T to be known. Choose $\rho = P_T, V = 1$, and $\Phi(Q) = e^{\lambda Q} - 1$, with:*

$$\lambda = \frac{1}{2\left(4(m+1)^{3/2}\left(D + \sqrt{D(D+P_T)}\right)\sqrt{E_T(g^+)} + G(m+1)\right)}.$$

Feeding these parameters into Algorithm 2, the following bounds hold:

$$\mathcal{R}_T(f) \le \mathcal{O}\left((m+1)^{3/2}\sqrt{(1+P_T)E_T(f)}\right), \qquad \textit{and} \tag{9}$$

$$Q_T \le \mathcal{O}\left(\left((m+1)^{3/2}\sqrt{(1+P_T)E_T(g^+)} + G(m+1)\right)\log T\right). \tag{10}$$

Discussion. Theorem 3 gives prediction-dependent bounds for the known P_T case, without using meta-learning. We first compare with dynamic optimistic COCO works with predictions. When $m = 0$, we recover the no-memory optimistic COCO bounds of [9]: $\mathcal{R}_T(f) = \mathcal{O}\big(\sqrt{P_T E_T(f)}\big)$, $Q_T = \mathcal{O}\big(\big(\sqrt{P_T E_T(g^+)} + G\big)\log T\big)$. Thus, our result extends their prediction-dependent guarantees to delayed and memory-dependent settings. The closest static optimistic memory-dependent works are [5, 17]. These works rely on separability of the memory terms, which facilitates isolating the contribution of each decision, but leads to a more restrictive benchmark in constrained problems and restricted class of loss functions. Our approach avoids this assumption and recovers the same type of prediction-dependent bounds while preserving the original joint feasibility benchmark. Finally, compared to [36], which studies the static memory-constrained problem without predictions, our bounds improve over their $\tilde{\mathcal{O}}(T^{2/3})$ regret and CCV rates.

Remark 4. If P_T is unknown but measurable online, a standard doubling trick recovers the bounds of this subsection. The same applies to tuning λ, since $E_T(g^+)$ is observable online.

5 Memory as Delayed Feedback

We now transfer the delayed optimistic dynamic regret guarantees to the memory problems. In this setting, the loss and constraint functions depend on a block of past decisions, $f_t(x_{t-m}^t)$ and $g_t(x_{t-m}^t)$. Thus, x_s affects all losses whose memory window contains it, i.e., all t such that $t - m \le s \le t$. Hence, its full first-order contribution is revealed only after these losses are observed, which induces a feedback delay of at most m. Note that we redefine here the learning functions, L_t, and the horizon (T_m instead of T) (further explanations about notation conventions, please see Appendix Sec. D).

The reduction is performed at the regret level. The memory losses are not assumed to be linear. Rather, we use the standard OCO linearization argument: applying the first-order convexity inequality

to each memory surrogate loss L_t around the played block x_{t-m}^t gives an upper bound in terms of partial gradients. More precisely, when L_t is nondifferentiable, we select a joint subgradient

$$G_t = (G_{t,t-m}, \dots, G_{t,t}) \in \partial L_t(x_{t-m}^t),$$

and interpret $\nabla_{x_s} L_t(x_{t-m}^t)$ below as the coordinate block $G_{t,s}$. Thus, all coordinate blocks associated with a given L_t are taken from the same joint subgradient, the same convention applies to predicted functions. For each decision index s, define

$$I_s := \{t \in [T_m] : \ t - m \le s \le t\}, \qquad h_s := \sum_{t \in I_s} \nabla_{x_s} L_t(x_{t-m}^t). \tag{11}$$

Here, I_s is the set of losses whose memory window contains x_s, and h_s is the corresponding *credit gradient*. It induces the auxiliary linearized loss $\ell_s(z) := \langle h_s, z \rangle$. Since $I_s \subseteq \{s, \dots, s+m\}$, the vector h_s is fully revealed no later than round $s+m$.

Lemma 2 (Memory-to-delay linearization)**.** *Let $L_t : \mathcal{X}^{m+1} \mapsto \mathbb{R}$ be jointly convex $\forall t \in [T_m]$, and define I_s, h_s as in* (11)*, using the joint-subgradient convention above whenever L_t is non-differentiable. Then, for any comparator sequence $u_1^T \in \mathcal{U}_T$, it holds that*

$$\sum_{t=m}^{T} L_t(x_{t-m}^t) - L_t(u_{t-m}^t) \le \sum_{s=1}^{T} \langle h_s, x_s - u_s \rangle.$$

Moreover, h_s is revealed with delay at most m.

The following corollary transfers the delayed guarantees to the memory setting.

Corollary 1 (Transfer from delayed OCO to memory)**.** *Any regret guarantee for delayed OCO with delay at most m, applied to the auxiliary linearized losses $\ell_s(z) = \langle h_s, z \rangle$, yields the corresponding guarantee for the memory-dependent regret. In particular, Theorems 1, 2, and 3 apply to the memory-dependent problem after replacing the delayed gradients by the credit gradients h_s. The corresponding objective and constraint prediction errors induced by the credit gradients are specified below.*

In the delayed case without memory, the hint predicts one delayed gradient, e.g., $\nabla f_{s+m}(x_s)$, and constructing the next hint only requires a one-step proxy decision. In the memory case, the hint must predict the whole credit gradient h_s. Thus, one may set

$$\widetilde{h}_s := \sum_{t \in I_s} \nabla_{x_s} \widetilde{L}_t(\widetilde{x}_{t-m}^t). \tag{12}$$

Consequently, constructing $\widetilde{h}_{s+1}$ may require predicting the future memory block up to $\widetilde{x}_{s+m+1}$, not only the next decision $\widetilde{x}_{s+1}$. Thus, the hints used by Algorithm 1 when P_T is unknown, or directly by Algorithm 2 when P_T is known, are

$$\widetilde{h}_{s-m+1:s+1} = \sum_{\tau=s-m+1}^{s+1} \sum_{t \in I_\tau} \nabla_{x_\tau} \widetilde{L}_t(\widetilde{x}_{t-m}^t).$$

Remark 5. In the constrained case, we use the surrogate loss

$$L_t(x_{t-m}^t) = V f_t(x_{t-m}^t) + \Phi'(Q_{t-m-1})\, g_t^+(x_{t-m}^t), \qquad Q_t = Q_{t-1} + g_t^+(x_{t-m}^t). \tag{13}$$

The regret decomposition Lemma of [5] applies to this surrogate too for $u_1^T \in \mathcal{U}_T$. Thus, once the dynamic regret of L_t is controlled via Corollary 1, the objective-regret and CCV bounds follow as in the delayed case. Equivalently, the prediction errors for the final COCO-M bounds are defined at the level of the objective and constraint credit gradients, $E_T^{\mathrm{mem}}(f)$ and $E_T^{\mathrm{mem}}(g^+)$, where

$$E_T^{\mathrm{mem}}(f) := \sum_{s=1}^{T} \left\| \sum_{t \in I_s} \left[\nabla_{x_s} f_t(x_{t-m}^t) - \nabla_{x_s} \widetilde{f}_t(\widetilde{x}_{t-m}^t) \right] \right\|^2, \tag{14}$$

$$E_T^{\mathrm{mem}}(g^+) := \sum_{s=1}^{T} \max_{r \in I_s} \left\| \sum_{\substack{t \in I_s \\ t \ge r}} \left[\nabla_{x_s} g_t^+(x_{t-m}^t) - \nabla_{x_s} \widetilde{g}_t^+(\widetilde{x}_{t-m}^t) \right] \right\|^2. \tag{15}$$

The latter quantity provides a penalty-independent upper envelope for the normalized constraint credit-gradient prediction error, as shown in Appendix D.

Discussion. Prior optimistic approaches to OCO with memory [5, 17], isolate the contribution of each decision at the level of loss functions, which requires separability across the memory variables. Our reduction keeps the memory-dependent loss in its original jointly convex form and linearizes only at the regret level. The price is that optimistic hints must predict credit gradients rather than single gradients. This may require predicting future memory blocks, but avoids imposing component-wise feasibility on separable parts of the constraint, and preserves the original joint feasibility benchmark. For the constraints, different memory contributions carry different queue-dependent penalty coefficients. We therefore use a penalty-independent envelope of the aggregated prediction errors, which parallels the separable setting [5] while controlling the corresponding weighted error. It vanishes under perfect predictions and preserves an $O(\sqrt{T})$ worst-case dependence, up to memory-dependent factors.

6 Conclusions and Limitations

The proposed framework contributes in closing a gap in COCO with memory, and NSC with time-average budget constraints, and can be used to tackle important problems in ML, operations research, control of cyber-physical systems, and so on. Aiming for universal dynamic regret ensures the learning is against the most demanding comparators; while the inclusion of optimism enables the bounds to improve, reaching path-length-oblivious regret $\mathcal{R}_T = \mathcal{O}(1)$, together with $Q_T = \mathcal{O}(\sqrt{T})$ when P_T is unknown, and $Q_T = \mathcal{O}(\log T)$ when known or observable at runtime.

There are few limitations which call for follow up work. First, the framework requires predicting, apart from future functions, also some future actions. This cyclic dependency is implicit in almost all optimistic learning works (unless the losses are linear); still, it is worth investigating whether it can be avoided, see discussion in [21]. Secondly, the constraint violation does not scale down to $\mathcal{O}(1)$ under perfect predictions. While this matches the best-known optimistic COCO bounds (without memory), it is certainly exciting to explore further improvements. Finally, expanding the set of eligible benchmarks in COCO is a key challenge; our work contributes in this direction but the question of approaching closer to horizon-long feasibility remains open.

Acknowledgments

This work was supported in part by the French government through the France 2030 program under the Celtic RAI-6Green project, and in part by the European Commission under Grant No. 101192462 (FLECON-6G).

References

[1] Elad Hazan. Introduction to online convex optimization. 2016.

[2] Francesco Orabona. A modern introduction to online learning, 2026. URL `https://arxiv.org/abs/1912.13213`.

[3] Oren Anava, Elad Hazan, and Shie Mannor. Online learning for adversaries with memory: price of past mistakes. *Advances in Neural Information Processing Systems*, 28, 2015.

[4] Elad Hazan and Karan Singh. *Introduction to Online Control*. Cambridge University Press, 2026.

[5] Mohammed Abdullah, George Iosifidis, Salah Eddine Elayoubi, and Tijani Chahed. Constrained online convex optimization with memory and predictions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(24):19524–19532, Mar. 2026. doi: 10.1609/aaai.v40i24.39031.

[6] Abhishek Sinha and Rahul Vaze. Beyond $\tilde{O}(\sqrt{T})$ constraint violation for online convex optimization with adversarial constraints. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

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

[8] 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.

[9] Jordan Lekeufack and Michael I Jordan. An optimistic algorithm for online convex optimization with adversarial constraints. *arXiv preprint arXiv:2412.08060*, 2024.

[10] Qingsong Liu, Wenfei Wu, Longbo Huang, and Zhixuan Fang. Simultaneously achieving sublinear regret and constraint violations for online convex optimization with time-varying constraints. *Performance Evaluation*, 152:102240, 2021.

[11] Subhamon Supantha and Abhishek Sinha. Universal dynamic regret and constraint violation bounds for constrained online convex optimization. *arXiv preprint arXiv:2510.01867*, 2025.

[12] Lijun Zhang, Wei Jiang, Shiyin Lu, and Tianbao Yang. Revisiting smoothed online learning. In *Advances in Neural Information Processing Systems 34*, pages 13599–13612, 2021.

[13] Sifan Yang, Wenhao Yang, Wei Jiang, Yuanyu Wan, and Lijun Zhang. Smoothed online convex optimization with delayed feedback. In James Kwok, editor, *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 6812–6820. International Joint Conferences on Artificial Intelligence Organization, 8 2025. doi: 10.24963/ijcai.2025/758. URL `https://doi.org/10.24963/ijcai.2025/758`. Main Track.

[14] Peng Zhao, Yu-Hu Yan, Yu-Xiang Wang, and Zhi-Hua Zhou. Non-stationary online learning with memory and non-stochastic control. *Journal of Machine Learning Research*, 24(206):1–70, 2023.

[15] Dheeraj Baby and Yu-Xiang Wang. Optimal dynamic regret in lqr control. *Advances in Neural Information Processing Systems*, 35:24879–24892, 2022.

[16] Hongyu Zhou, Zirui Xu, and Vasileios Tzoumas. Efficient online learning with memory via frank-wolfe optimization: Algorithms with bounded dynamic regret and applications to control. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 8266–8273, 2023. doi: 10.1109/CDC49753.2023.10384153.

[17] Naram Mhaisen and George Iosifidis. Optimistic online non-stochastic control via ftrl. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pages 7116–7123. IEEE, 2024.

[18] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Online learning under delayed feedback. In *International conference on machine learning*, pages 1453–1461. PMLR, 2013.

[19] Genevieve E Flaspohler, Francesco Orabona, Judah Cohen, Soukayna Mouatadid, Miruna Oprescu, Paulo Orenstein, and Lester Mackey. Online learning with optimism and delay. In *International Conference on Machine Learning*, pages 3363–3373. PMLR, 2021.

[20] Yuanyu Wan, Chang Yao, Mingli Song, and Lijun Zhang. Non-stationary online convex optimization with arbitrary delays. In *Proceedings of the 41st International Conference on Machine Learning*, pages 49991–50011, 2024.

[21] Pedro Zattoni Scroccaro, Arman Sharifi Kolarijani, and Peyman Mohajerin Esfahani. Adaptive composite online optimization: Predictions in static and dynamic environments. *IEEE Transactions on Automatic Control*, 68(5):2906–2921, 2023.

[22] Naram Mhaisen and George Iosifidis. On the dynamic regret of following the regularized leader: Optimism with history pruning. In *International Conference on Machine Learning*, pages 43990–44016. PMLR, 2025.

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

[24] Lijun Zhang, Tianbao Yang, Rong Jin, and Zhi-Hua Zhou. Dynamic regret of strongly adaptive methods. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 5882–5891, 2018.

[25] Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive online learning in dynamic environments. *Advances in neural information processing systems*, 31, 2018.

[26] Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. In *Conference on Learning Theory*, pages 993–1019, 2013.

[27] Mehryar Mohri and Scott Yang. Accelerating online convex optimization via adaptive prediction. In *Artificial Intelligence and Statistics*, pages 848–856. PMLR, 2016.

[28] Pooria Joulani, András György, and Csaba Szepesvári. A modular analysis of adaptive (non-)convex optimization: Optimism, composite objectives, variance reduction, and variational bounds. *Theor. Comput. Sci.*, 808:108–138, February 2020.

[29] Neri Merhav, Erik Ordentlich, Gadiel Seroussi, and Marcelo J Weinberger. On sequential strategies for loss functions with memory. *IEEE Transactions on Information Theory*, 48(7): 1947–1958, 2002.

[30] Andras Gyorgy and Gergely Neu. Near-optimal rates for limited-delay universal lossy source coding. *IEEE Transactions on Information Theory*, 60(5):2823–2834, 2014.

[31] Alexander Korotin, Vladimir V'yugin, and Evgeny Burnaev. Adaptive hedging under delayed feedback. *Neurocomputing*, 397:356–368, 2020.

[32] Jianjun Yuan and Andrew Lamperski. Online convex optimization for cumulative constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6140–6149, 2018.

[33] Abhishek Sinha and Rahul Vaze. Optimal algorithms for online convex optimization with adversarial constraints. *Advances in Neural Information Processing Systems*, 37:41274–41302, 2024.

[34] Abhishek Sinha and Rahul Vaze. Beyond $\tilde{o}(\sqrt{t})$ constraint violation for online convex optimization with adversarial constraints. *arXiv preprint arXiv:2505.06709*, 2025.

[35] Nikolaos Liakopoulos, Apostolos Destounis, Georgios Paschos, Thrasyvoulos Spyropoulos, and Panayotis Mertikopoulos. Cautious regret minimization: Online optimization with long-term budget constraints. Proceedings of Machine Learning Research, pages 3944–3952. PMLR, 09–15 Jun 2019.

[36] Xin Liu, Zixian Yang, and Lei Ying. Online nonstochastic control with adversarial and static constraints. In *International Conference on Machine Learning*, pages 22277–22288. PMLR, 2023.

[37] 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.

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

[39] Dheeraj Baby and Yu-Xiang Wang. Optimal dynamic regret in exp-concave online learning. In *Proceedings of the 34th Annual Conference On Learning Theory*, pages 359–409, 2021.

[40] Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Dynamic regret of convex and smooth functions. In *Advances in Neural Information Processing Systems*, volume 33, pages 12510–12520, 2020.

[41] Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Adaptivity and non-stationarity: Problem dependent dynamic regret for online convex optimization. *Journal of Machine Learning Research*, 25(98):1–52, 2024.

[42] Lijun Zhang, Shiyin Lu, and Tianbao Yang. Minimizing dynamic regret and adaptive regret simultaneously. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 309–319, 2020.

[43] Tianbao Yang, Lijun Zhang, Rong Jin, and Jinfeng Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *International Conference on Machine Learning*, pages 449–457. PMLR, 2016.

[44] Lijun Zhang, Tianbao Yang, Jinfeng Yi, Rong Jin, and Zhi-Hua Zhou. Improved dynamic regret for non-degenerate functions. In *Advances in Neural Information Processing Systems 30*, pages 732–741, 2017.

[45] Lijun Zhang, Guanghui Wang, Jinfeng Yi, and Tianbao Yang. A simple yet universal strategy for online convex optimization. In *International Conference on Machine Learning*, pages 26605–26623. PMLR, 2022.

[46] Xi Chen, Yining Wang, and Yu-Xiang Wang. Non-stationary stochastic optimization under $l_{p,q}$-variation measures. *arXiv preprint arXiv:1708.03020*, 2017.

[47] Rahul Vaze and Abhishek Sinha. $o(\sqrt{\{T\}})$ static regret and instance dependent constraint violation for constrained online convex optimization. *arXiv preprint arXiv:2502.05019*, 2025.

[48] Rahul Vaze and Abhishek Sinha. $o(\sqrt{T})$ static regret and instance dependent constraint violation for constrained online convex optimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[49] 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.

[50] Daron Anderson, George Iosifidis, and Douglas J Leith. Lazy lagrangians for optimistic learning with budget constraints. *IEEE/ACM Transactions on Networking*, 31(5):1935–1949, 2023.

[51] 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.

[52] Heyan Huang, Ping Wu, Haolin Lu, and Zhengyang Liu. On the constrained online convex optimization with feedback delay. *Expert Systems with Applications*, 287:127871, 2025. ISSN 0957-4174.

[53] Yingying Li, Subhro Das, and Na Li. Online optimal control with affine constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8527–8537, 2021.

[54] N. Merhav, E. Ordentlich, G. Seroussi, and M.J. Weinberger. On sequential strategies for loss functions with memory. *IEEE Transactions on Information Theory*, 48(7):1947–1958, 2002. doi: 10.1109/TIT.2002.1013135.

[55] Raman Arora, Ofer Dekel, and Ambuj Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. ICML'12, 2012.

[56] Chang-Heng Wang, Jaime Llorca, Antonia M. Tulino, and Tara Javidi. Dynamic cloud network control under reconfiguration delay and cost. *IEEE/ACM Transactions on Networking*, 27(2): 491–504, 2019.

[57] Samrat Mukhopadhyay and Abhishek Sinha. Online caching with optimal switching regret. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 1546–1551, 2021.

[58] Tareq Si Salem, Giovanni Neglia, and Stratis Ioannidis. No-regret caching via online mirror descent. *ACM Trans. Model. Perform. Eval. Comput. Syst.*, 8(4), 2023.

[59] Kush Bhatia and Karthik Sridharan. Online learning with dynamics: A minimax perspective. In *Advances in Neural Information Processing Systems*, volume 33, pages 15020–15030, 2020.

[60] Raunak Kumar, Sarah Dean, and Robert Kleinberg. Online convex optimization with unbounded memory. In *Advances in Neural Information Processing Systems*, volume 36, pages 26229–26270, 2023.

[61] Naman Agarwal, Brian Bullins, Elad Hazan, Sham Kakade, and Karan Singh. Online control with adversarial disturbances. In *International Conference on Machine Learning*, pages 111–119. PMLR, 2019.

[62] Nanfei Jiang, Spencer Hutchinson, and Mahnoosh Alizadeh. Online nonstochastic control with convex safety constraints. *arXiv preprint arXiv:2501.18039*, 2025.

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

[64] Yu-Hu Yan, Peng Zhao, and Zhi-Hua Zhou. Online non-stochastic control with partial feedback. *Journal of Machine Learning Research*, 24(273):1–50, 2023.

[65] Hongyu Zhou and Vasileios Tzoumas. Safe non-stochastic control of linear dynamical systems. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 5033–5038. IEEE, 2023.

[66] Marcelo J Weinberger and Erik Ordentlich. On delayed prediction of individual sequences. *IEEE Transactions on Information Theory*, 48(7):1959–1976, 2002.

[67] Kent Quanrud and Daniel Khashabi. Online learning with adversarial delays. In *Advances in Neural Information Processing Systems 28*, pages 1270–1278, 2015.

[68] 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.

[69] Yu-Guan Hsieh, Franck Iutzeler, Jérôme Malick, and Panayotis Mertikopoulos. Multi-agent online optimization with delays: Asynchronicity, adaptivity, and optimism. *Journal of Machine Learning Research*, 23(78):1–49, 2022.

[70] Yuanyu Wan. Black-box reductions for decentralized online convex optimization in changing environments. In *Proceedings of Thirty Eighth Conference on Learning Theory*, pages 5605–5631, 2025.

[71] Juncheng Wang, Min Dong, Ben Liang, Gary Boudreau, and Hatem Abou-Zeid. Delay-tolerant oco with long-term constraints: Algorithm and its application to network resource allocation. *IEEE/ACM Transactions on Networking*, 31(1):147–163, 2023. doi: 10.1109/TNET.2022.3188285.

[72] Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization: Competing with dynamic comparators. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, pages 398–406, 2015.

The structure of this appendix is as follows.

1. Section A provides a comprehensive discussion of related work.
2. Section B provides the proof of Theorem 1. This requires deriving a dynamic-regret bound for our delayed optimistic OCO algorithm; the section includes both the auxiliary results and their proofs.
3. Section C provides the proofs of Theorem 2 and Theorem 3, respectively in Sections C.1 and C.2.
4. Section D provides the proofs of the memory-to-delay equivalence results stated in the main paper, namely Lemma 2 and Corollary 1.

A Discussion of Related Work

This section provides a detailed discussion of related works. Table 1 presents a concise high-level comparison with selected prior results. The proposed method is at the nexus of different OCO flavors, namely OCO with time-average adversarially-changing constraints, OCO with memory, OCO with delay and OCO with predictions. Therefore, we review below most relevant works.

A.1 Dynamic Regret in OCO and COCO

Regarding the standard OCO model, the seminal work [37] proved the *universal* dynamic regret bound $\mathcal{O}((1+\mathcal{P}_T)\sqrt{T})$ for Online Gradient Descent (OGD), where $\mathcal{P}_T$ is the norm-based path-length of the benchmark. More recently, [25] showed this problem has a lower bound $\Omega(\sqrt{(1+\mathcal{P}_T)T})$ and designed a tight algorithm (i.e., meeting this bound) using Hedge [38]. Namely, they deploy a set of OGD *experts* (or, base learners) with learning rates that cover the entire range of path-length values, and a meta-learning algorithm that discerns at runtime which base learner is aligned with the realized functions. Refinements of this result include curvature-specific bounds [39], variation-based guarantees [40, 41], and methods which achieve both adaptive and dynamic regret [42]. On the other hand, [43] showed that OGD achieves $\mathcal{O}(\sqrt{(1+\mathcal{P}_T)T})$ *worst-case* dynamic regret, where $\mathcal{P}_T$ here is the path-length of the per-round minimizers. Under additional assumptions, tighter worst-case dynamic regret bounds were established by follow-up works, e.g., see [44–46].

There is a recent flurry of research in COCO, with important works improving the bounds of the most general problem (time-varying adversarial constraints) while dropping assumptions (such as the Slater condition); see for instance [33], [47], [48], [6] which follow a penalty-based approach like the one considered in this work. A different class of COCO techniques rely on primal-dual algorithms with adaptive regularization, e.g., see [32, 49–51]. However, dynamic regret in COCO is largely underexplored. The few exceptions include [7] that achieved $\mathcal{O}(\sqrt{(1+\mathcal{P}_T)T})$ worst-case dynamic regret and $\mathcal{O}(\sqrt{T})$ CCV for time-invariant functions, that was subsequently improved to $\mathcal{O}(\sqrt{(1+\mathcal{P}_T)T})$ regret and $\mathcal{O}(\log T)$ CCV by [8]. In case of time-varying adversarial constraints, [8] achieved $\mathcal{O}(\sqrt{(1+\mathcal{P}_T)T})$ universal dynamic regret and $\mathcal{O}(T^{3/4})$ CCV. More recently, [9] proposed an algorithm with worst-case dynamic regret $\mathcal{O}(\sqrt{(1+\mathcal{P}_T)T})$ and $\tilde{\mathcal{O}}(\sqrt{T})$ CCV. [10] introduced an algorithm with regret $\mathcal{O}(\sqrt{(1+\mathcal{P}_T)T})$ and CCV $\mathcal{O}(\max(\sqrt{T}, V_g))$, where $V_g = \sum_t \sup_{x\in\mathcal{X}} ||g_t(x) - g_{t-1}(x)||_2$. Other recent works include [52] which considers time-invariant constraints, and [11] which established universal dynamic regret and CCV guarantees under adversarial constraints using a penalty-based method. None of these works considers memory dependency; a property that changes fundamentally the required methodological approach.

A.2 Memory Effects: COCO-M and NSC

The first study on OCO-M appeared in [54], followed later by [55]. A seminal work in this area is [3] which proved that the Follow-The-Regularized-Leader algorithm can be applied to a proxy memory-less function and guarantee sublinear regret for the memory-based function. A special case of OCO-M is when the learner is penalized for decision switches, i.e., the loss function has memory of just $m = 1$. This setting captures several practical problems such as workload management in datacenters [56], caching with fetching delays, etc., [57], [58]. On the other hand, there are studies for the case of infinite (but discounted) memory effects, e.g., see [59], [60].

Reference	Constraints	Loss	Memory		Predictions	Regret	Assumptions
			Obj.	Constr.			
[36]	adversarial	convex	✓	✓	✗	Static	
[53]	static affine	convex	✓	✓	✗	Static	
[3]	none	convex	✓	✗	✗	Static	
[17]	none	linear	✓	✗	✓	Static	separability
[33]	adversarial	convex	✗	✗	✗	Static	
[50]	adversarial	convex	✗	✗	✓	Static	
[5]	adversarial	convex	✓	✓	✓	Static	separability
[11]	adversarial	convex	✗	✗	✗	Dynamic	
[9]	adversarial	convex	✗	✗	✓	Dynamic	
This work	adversarial	convex	✓	✓	✓	Dynamic	

Table 1: Summary of the most relevant prior works. The current study is the first to consider loss and constraint functions with memory dependency, under universal dynamic regret while using (imperfect) predictions.

The advent of non-stochastic control [61] renewed the interest in OCO-M. In NSC, the action at each round is a weighted sum of the past disturbances, and these weights are learned through OCO-M. Such formulations have a vast range of practical applications, see [4]. However, most NSC works impose *deterministic* constraints on the state and input [53, 62–64] or assume the constraint at $t+1$ is revealed before that slot [65]; thus, they do not focus on time-average constraints nor on unknown constraints. Notable exceptions include [36] which analyzes an adversarial COCO-M setting and proves $\mathcal{O}\big(T^{2/3}\log^2 T\big)$ static regret and CCV; and [5] that improves these bounds to $\mathcal{O}\big(m^{3/2}\sqrt{T\log T}\big)$ static regret and $\mathcal{O}\big(m^{3/2}\sqrt{T\log T}\ \vee\ T^{3/4}\big)$ CCV.

Dynamic regret is significantly less explored for this class of problems, with exceptions confined to the special case of switching costs, cf.[12], [13], or to NSC without budget constraints, e.g., [14], [15], [16]. Therefore, the problem of dynamic regret for COCO-M, and time-average constrained NSC, remains open. Our work aspires to fill this gap.

A.3 Delay and Predictions in OCO

Two key methods we employ in this work are OCO with delay, which we use to handle memory effects, unearthing an equivalence between these two OCO versions; and optimism, which expedites the learning convergence. Starting with the former, [66] was the first to study the setting where the effect of decision x_t is observed after d rounds, and extended vanilla OCO algorithms to cope with this delay. Significant follow-up works include [18] which handls arbitrary feedback delays, and [67] that improved the regret bounds to $\mathcal{O}(\sqrt{\sum_t d_t})$. These works share the main elegant idea, namely running multiple delay-free copies of a base learning algorithm over the delay window. However, this replication is computationally costly and each copy is trained only on partial observations, as it does not utilize the feedback from every round. Few other works avoid replication and incorporate delays directly into the analysis. For example, [68] derived adaptive FTRL/Mirror-Descent bounds through a single-instance reduction based on prediction drift; and [69] developed a delayed dual averaging with data- and delay-dependent guarantees. Finally, [31] studied the delayed Hedge problem without using replication. All these studies focus on static regret.

More recently, [20] proved a universal dynamic regret bound of $\mathcal{O}(\sqrt{dT}(P_T+1))$ by leveraging the delayed Hedge of [31] to meta-learn over a set of delayed-OGD base learners; and [70] extended this to a decentralized setting. Finally, [71] studied worse-case dynamic regret in COCO with fixed delay. Unlike these important works, our method is adaptive and does not require an a priori Lipschitz bound on the gradients. Besides, our approach differs conceptually as we handle delay through optimism.

Optimistic learning builds on the idea that predictions of forthcoming loss gradients can expedite the learning by reducing the regret bounds commensurately to the accuracy of these predictions, [26]. This framework has been extended in various ways, e.g., by achieving data-adaptive optimistic bounds [27, 28]. Our approach here is motivated by the insightful proposal of [19], that interpreted delayed feedback as a "bad hint" or, equivalently, used predictions to recover missing gradients in OCO with delay. Here, we extend this idea, for the first time, to the universal dynamic regret setting.

Such demanding benchmarks are indeed rare in optimistic learning, even for the simpler problem without feedback delay. The few exceptions include [21, 22, 72]. However, these works do not consider delayed feedback or memory, nor their extension to these settings is straightforward (see main paper). Hence, our optimistic delayed OCO algorithm is of independent interest.

Summary. To the best of our knowledge, this is the first work to tackle memory and optimism, both in the loss and constraint functions, under universal dynamic regret for adversarial COCO. This result extends the recently-introduced COCO-M problem of [5] to the universal dynamic setting, while being able to drop key assumptions such as the function separability and also expand the benchmark set. Furthermore, our analysis yields several byproducts of independent interest: *(i)* we enable the study of Non-Stochastic Control with adversarial time-varying constraints under dynamic regret guarantees, whereas prior works focus on static regret [5, 36]; *(ii)* we establish dynamic regret bounds for optimistic delayed OCO (and delayed COCO), while [20] study dynamic regret for delayed OCO but cannot benefit from predictions; *(iii)* we extend the "delay-as-optimism" framework of [19] to the dynamic setting by handling the interaction between prediction errors and comparator variation, while other dynamic optimistic OCO algorithms such as [22] do not handle memory nor delay.

B Proofs of Section 3

B.1 Proof of Theorem 1

We provide the proof of Theorem 1, which we restate here for the readers convenience.

Let h_t be the delayed gradient feedback and let $\tilde{h}_t$ be its prediction. For any comparator sequence $u_1^T \in \mathcal{U}_T$, Algorithm 1 ensures dynamic regret:

$$\mathcal{R}_T(L) = \mathcal{O}\left((m+1)^{3/2} \left(1 + \sqrt{1+\log N} + \sqrt{1+P_T}\right) \sqrt{\sum_{t=1}^{T} \|h_t - \tilde{h}_t\|^2} \right).$$

The proof of this result comprises two stages. First, we bound Term (II) and then Term (I). For the former, we introduce an optimistic dynamic regret variant of OGD that uses adaptive regularization and can handle delays. For Term (I), we adapt the static regret delayed optimistic algorithm of [19].

B.1.1 Bounding Term(II)

We first analyze the undelayed optimistic problem. This step is useful because the delayed update will later be recovered through the delay-as-optimism reduction. In the dynamic setting, the main challenge is the movement of the comparator. We therefore use the projected one-step geometry, which yields a clean telescoping argument and exposes the additional interaction between the prediction error and the comparator variation. We introduce the following algorithm and respective bound:

Theorem 4 (Optimistic dynamic OGD)**.** *Let $(\eta_t)_{t\geq 1}$ be a nondecreasing positive sequence. Let $L_t : \mathcal{X} \to \mathbb{R}$ be convex losses, and let $h_t = \nabla L_t(x_t)$. In this theorem, feedback is non-delayed (after playing x_t the learner observes h_t). consider the projected optimistic one-step update*

$$x_{t+1} = \Pi_X\left(x_t - \frac{1}{\eta_{t+1}}\left(h_t + \tilde{h}_{t+1} - \tilde{h}_t\right)\right), \qquad x_1 = \Pi_X\left(x_0 - \frac{1}{\eta_1}\tilde{h}_1\right),$$

we have that for any comparator sequence $u_1, \ldots, u_T \in \mathcal{U}_T$, the regret satisfies:

$$\mathcal{R}_T(L) \leq \eta_{T+1}\left(D^2 + \tfrac{3}{2}D\,P_T\right) + \sum_{t=1}^{T} \min\left(D\|e_t\|_2,\ \frac{1}{\eta_{t+1}}\,\mathrm{huber}\big(\|e_t\|_2, \|n_t\|_2\big)\right) - \sum_{t=1}^{T}\langle e_t, \delta_t\rangle, \tag{16}$$

where $e_t := h_t - \tilde{h}_t$, $n_t := h_t + \tilde{h}_{t+1} - \tilde{h}_t$, and $\delta_t := u_t - u_{t+1}$.

Theorem 4 will serve as a technical bridge with the delayed analysis. Related optimistic dynamic-regret guarantees have been studied before; see, for instance, [72] and, more recently [21] and [22]. However, these results are developed for the standard non-delayed feedback model and cannot be used

here. The purpose of Theorem 4 is to derive a non-delayed optimistic dynamic-regret bound in a form compatible with the delayed-upper-bound (DUB) argument, both in the m-step delayed setting of [19] and in the one-step delayed adaptive-tuning argument of [2]. After applying the delay-as-optimism reduction, this structure will allow us to choose the regularization adaptively from observed quantities, avoiding over-regularization and preserving the benefit of accurate predictions. The additional term compared with the static-benchmark case of [19] is $-\sum_{t=1}^{T}\langle e_t, \delta_t\rangle$, which couples the prediction error with the movement of the comparator.

We now pass from the undelayed optimistic bound to the delayed setting. By Lemma 2 of [19], delayed feedback can be viewed as optimism with a particular choice of "bad" hint. In our notation, this hint is

$$\tilde{h}_{t+1}^{\mathrm{bad}} := \tilde{h}_{t-m+1:t+1} - h_{t-m+1:t}.$$

Applying this representation to the bound in (16), we obtain the same form of guarantee after replacing e_t and n_t by

$$\bar{e}_t := h_{t-m:t} - \tilde{h}_{t-m:t}, \qquad \bar{n}_t := h_{t-m} + \tilde{h}_{t-m+1:t+1} - \tilde{h}_{t-m:t}.$$

The algebraic verification of this substitution is deferred to Appendix B.2.2, using Corollary 3 presented there.

The genuinely new term in the dynamic optimistic analysis is

$$-\sum_{t=1}^{T}\langle \bar{e}_t, \delta_t\rangle, \qquad \delta_t := u_t - u_{t+1}.$$

It appears because the comparator used here changes from u_t to u_{t+1}. In a standard non-optimistic analysis, this change only contributes through the path length. In the optimistic analysis, however, the update direction depends on predictions, so the movement of the comparator interacts with the prediction error $\bar{e}_t$. The term $-\langle \bar{e}_t, \delta_t\rangle$ therefore measures whether the prediction error helps or hurts the tracking of the moving comparator. However this term is yet controlled and upper bounded by the desired bound that is of order $\sqrt{T}$ in the worst case and shrinks with prefect prediction and/or with static benchmark to be zero.

Corollary 2 (DUB-tuned bound for Term (II))**.** *Let*

$$a_t := D\|\bar{e}_t\|_2, \qquad b_t := \mathrm{huber}(\|\bar{e}_t\|_2, \|\bar{n}_t\|_2),$$

Assume that the base learner uses the DUB-tuned regularization

$$\eta_{t+1} = \frac{2}{\alpha}\max_{j\le t-m-1} a_{j-m+1:j} + \frac{1}{\alpha}\sqrt{\sum_{i=1}^{t-m}\big(a_i^2 + 2\alpha b_i\big)}, \quad \textit{with } \alpha := D\big(D+2P_T\big). \tag{17}$$

Then the delayed dynamic regret of the best-tuned base learner satisfies

$$\text{Term (II)} \le 4(m+1)^{3/2}\left(D + \sqrt{D(D+P_T)}\right)\sqrt{\sum_{t=1}^{T}\|h_t - \tilde{h}_t\|^2}.$$

Compared with [20], our method is adaptive and does not require an a priori Lipschitz bound on the gradients. The standard black-box reduction of [18] handles delays by running multiple delay-free copies of a base algorithm, but this replication can be computationally costly and each copy is trained only on a partial subsequence of the observations. Other works avoid replication and incorporate delays directly into the analysis. For example, [68] derives adaptive FTRL/Mirror-Descent bounds through a single-instance reduction based on prediction drift. More recently, [69] develops delayed dual averaging with data- and delay-dependent guarantees, including an optimistic variant. These works, however, focus on static regret. Our contribution is complementary. Building on the optimistic delayed-OCO perspective of [19], we develop delayed optimistic guarantees for a dynamic comparator sequence. This distinction is important because static-regret techniques do not directly control the path length of a moving comparator. We then use these dynamic delayed-OCO guarantees as a bridge to constrained OCO with memory through the credit-gradient reduction.

B.1.2 Bounding Term(I)

We now turn our focus to Term(I) and bound it using an optimistic delayed AdaHedge algorithm. Using $x_t = X_t w_t$ and $x_t^i = X_t e_i$, it follows that $x_t - x_t^i = X_t(w_t - e_i)$, thus we can write:

$$\text{Term (I)} \leq \sum_{t=1}^{T} \langle M_t,\ w_t - e_i \rangle, \quad \text{where } M_t := X_t^\top h_t.$$

This is the meta-regret of the delayed expert-combination scheme. This part follows from the delayed optimistic OCO framework of [19] applied to the linear meta-losses $M_t = X_t^\top h_t$.

Lemma 3 (Bound on Term (I))**.** *Under the assumptions above, and using the static delayed optimistic FTRL meta-learner of [19, Theorem 12], for every expert $i \in [N]$, we have:*

$$\textit{Term (I)} \leq 8D(m+1)^{\frac{3}{2}} \left(1 + \sqrt{1 + \log N}\right) \sqrt{\sum_{t=1}^{T} \|h_t - \tilde{h}_t\|^2}. \tag{18}$$

• Finally, the proof of Theorem 1 is obtained by recalling that:

$$\mathcal{R}_T(L) \leq \mathcal{R}_T(\ell) = \text{Term (I)} + \text{Term (II)},$$

and hence the results follows by combining Lemma 3 with Corollary 2.

Discussion The factor $(m+1)^{3/2}$ in Theorem 1 should be understood as the price of obtaining a prediction-adaptive bound under delayed feedback. The factor $m+1$ comes from the block prediction error $\bar{e}_t = h_{t-m:t} - \tilde{h}_{t-m:t}$, whereas the additional $\sqrt{m}$ comes from the delayed-upper-bound (DUB) argument. More precisely, it appears when controlling the unrevealed window term

$$\max_{t \in [T]} a_{t-m:t-1} \leq D\sqrt{m} \sqrt{\sum_{t=1}^{T} \|\bar{e}_t\|^2}.$$

This term is the cost of choosing the regularization adaptively from delayed observations, which avoids over-regularization and preserves the benefit of accurate predictions.

If predictions are ignored and a uniform gradient bound is available, one can instead use a fixed worst-case tuning and recover a delayed dynamic-regret bound with linear dependence on $m+1$. Thus, the $(m+1)^{3/2}$ dependence is not intrinsic to delayed OCO itself, but to the adaptive optimistic tuning under delay.

We also note that the sharper bounds in (35) and (33) retain more information than the simplified $(m+1)^{3/2}$ statement. In particular, within the prediction-adaptive bound itself, the additional $\sqrt{m}$ factor in Term (I) can be dominated by the meta-complexity term when $m = O(\log N)$, while the corresponding term in Term (II) can be dominated by the path-length term when $P_T \gtrsim Dm$.

Remark 6. *One may alternatively implement the adaptive DUB tuning of [19, Theorem 12], which yields the same theoretical order, but yields for better performance in practice.*

B.2 Auxiliary Results and Proofs for Theorem 1 and Section 3

This subsection provides the intermediate results that are required for the proof of the main result of Section 3, namely of Theorem 1 that we proved in the previous subsection.

B.2.1 Proof of Theorem 4

We remind the reader that this theorem assumes no delays, and thus after committing x_t at round t, the gradient $h_t = \nabla_{x_t} L_t(x_t)$ is revealed directly and there is no delay of m steps.

Recall that:

$$e_t := h_t - \tilde{h}_t, \qquad n_t := h_t + \tilde{h}_{t+1} - \tilde{h}_t, \qquad v_t := x_t - x_{t+1}, \qquad \delta_t := u_t - u_{t+1},$$

Proof. Since each L_t is convex and $h_t = \nabla_{x_t} L_t(x_t)$, we have

$$L_t(x_t) - L_t(u_t) \le \langle h_t, x_t - u_t \rangle, \qquad \text{and thus} \qquad \mathcal{R}_T(L) \le \sum_{t=1}^{T} \langle h_t, x_t - u_t \rangle.$$

Step 1: Exact decomposition. For each $t \in \{1, \dots, T\}$, we have

$$\langle h_t, x_t - u_t \rangle = \langle e_t, v_t \rangle + \langle n_t, x_{t+1} - u_t \rangle + \langle \tilde{h}_t, x_t - u_t \rangle - \langle \tilde{h}_{t+1}, x_{t+1} - u_t \rangle. \tag{19}$$

Summing (19) over $t = 1, \dots, T$ yields

$$\begin{aligned} \sum_{t=1}^{T} \langle h_t, x_t - u_t \rangle = \sum_{t=1}^{T} \langle n_t, x_{t+1} - u_t \rangle + \sum_{t=1}^{T} \langle e_t, v_t \rangle \\ + \sum_{t=1}^{T} \Big(\langle \tilde{h}_t, x_t - u_t \rangle - \langle \tilde{h}_{t+1}, x_{t+1} - u_t \rangle \Big). \end{aligned} \tag{20}$$

Now the last sum telescopes as follows:

$$\sum_{t=1}^{T} \Big(\langle \tilde{h}_t, x_t - u_t \rangle - \langle \tilde{h}_{t+1}, x_{t+1} - u_t \rangle \Big) = \langle \tilde{h}_1, x_1 - u_1 \rangle + \sum_{t=1}^{T} \langle \tilde{h}_{t+1}, \delta_t \rangle, \tag{21}$$

where the last equality uses $\delta_T = 0$ and $\tilde{h}_{T+1} = 0$. Substituting (21) into (20), we obtain:

$$\sum_{t=1}^{T} \langle h_t, x_t - u_t \rangle = \sum_{t=1}^{T} \langle n_t, x_{t+1} - u_t \rangle + \sum_{t=1}^{T} \big(\langle e_t, v_t \rangle + \langle \tilde{h}_{t+1}, \delta_t \rangle \big) + \langle \tilde{h}_1, x_1 - u_1 \rangle. \tag{22}$$

Since $n_t = e_t + \tilde{h}_{t+1}$, we have:

$$\langle e_t, v_t \rangle + \langle \tilde{h}_{t+1}, \delta_t \rangle = \langle e_t, v_t - \delta_t \rangle + \langle n_t, \delta_t \rangle.$$

Hence, it is:

$$\sum_{t=1}^{T} \langle h_t, x_t - u_t \rangle = \sum_{t=1}^{T} \langle n_t, x_{t+1} - u_{t+1} \rangle + \sum_{t=1}^{T} \langle e_t, v_t - \delta_t \rangle + \langle \tilde{h}_1, x_1 - u_1 \rangle. \tag{23}$$

Step 2: One-step projected inequality. By the Euclidean projection optimality condition, it is:

$$\left\langle x_t - \frac{1}{\eta_{t+1}} n_t - x_{t+1},\ u_{t+1} - x_{t+1} \right\rangle \le 0.$$

Rearranging, gives:

$$\langle n_t, x_{t+1} - u_{t+1} \rangle \le \frac{\eta_{t+1}}{2} \Big(\|x_t - u_{t+1}\|^2 - \|x_{t+1} - u_{t+1}\|^2 - \|v_t\|^2 \Big). \tag{2}$$

Substituting this into (23), we obtain:

$$\sum_{t=1}^{T} \langle h_t, x_t - u_t \rangle \le A_T + \sum_{t=1}^{T} \Big(\langle e_t, v_t - \delta_t \rangle - \frac{\eta_{t+1}}{2} \|v_t\|^2 \Big) + \langle \tilde{h}_1, x_1 - u_1 \rangle, \tag{24}$$

where

$$A_T := \sum_{t=1}^{T} \frac{\eta_{t+1}}{2} \Big(\|x_t - u_{t+1}\|^2 - \|x_{t+1} - u_{t+1}\|^2 \Big).$$

Step 3: Local bound preserving the Huber structure. Rewrite the local term in (24) as:

$$\langle e_t, v_t - \delta_t \rangle - \frac{\eta_{t+1}}{2} \|v_t\|^2 = \Big(\langle e_t, v_t \rangle - \frac{\eta_{t+1}}{2} \|v_t\|^2 \Big) - \langle e_t, \delta_t \rangle. \tag{25}$$

We first bound the bracketed term as in the proof of Theorem 16 of [19]. Since:

$$x_{t+1} = \Pi_{\mathcal{X}}\left(x_t - \frac{1}{\eta_{t+1}} n_t\right) \qquad \text{and} \qquad x_t = \Pi_{\mathcal{X}}(x_t),$$

the nonexpansiveness of Euclidean projection implies that:

$$\|v_t\| = \|x_t - x_{t+1}\| \leq \frac{1}{\eta_{t+1}} \|n_t\|.$$

Also, since $x_t, x_{t+1} \in \mathcal{X}$ and $\operatorname{diam}(\mathcal{X}) = D$, it is $\|v_t\| \leq D$. Hence, we can write:

$$\|v_t\| \leq \min\left(D, \frac{\|n_t\|}{\eta_{t+1}}\right).$$

Applying Lemmas 17–18 of [19] with the Euclidean norm, we obtain:

$$\langle e_t, v_t \rangle - \frac{\eta_{t+1}}{2} \|v_t\|^2 \leq \min\left(D\|e_t\|,\ \frac{1}{\eta_{t+1}} \operatorname{huber}\big(\|e_t\|, \|n_t\|\big)\right). \tag{26}$$

Therefore, we can write:

$$\langle e_t, v_t - \delta_t \rangle - \frac{\eta_{t+1}}{2} \|v_t\|^2 \leq \min\left(D\|e_t\|,\ \frac{1}{\eta_{t+1}} \operatorname{huber}\big(\|e_t\|, \|n_t\|\big)\right) - \langle e_t, \delta_t \rangle. \tag{27}$$

Step 4: Bound the weighted telescope A_T. We rewrite A_T as

$$A_T = \frac{\eta_2}{2}\|x_1 - u_2\|^2 - \frac{\eta_{T+1}}{2}\|x_{T+1} - u_{T+1}\|^2 + \sum_{t=2}^{T}\left[\frac{\eta_{t+1} - \eta_t}{2}\|x_t - u_{t+1}\|^2 + \frac{\eta_t}{2}\left(\|x_t - u_{t+1}\|^2 - \|x_t - u_t\|^2\right)\right]$$

Since $\{\eta_t\}$ is nondecreasing and $\|x_t - u_{t+1}\| \leq D$, the first and third terms yield:

$$\frac{\eta_2}{2}\|x_1 - u_2\|^2 + \sum_{t=2}^{T} \frac{\eta_{t+1} - \eta_t}{2}\|x_t - u_{t+1}\|^2 \leq \frac{\eta_{T+1} D^2}{2}.$$

Moreover, we can write:

$$\|x_t - u_{t+1}\|^2 - \|x_t - u_t\|^2 = 2\langle x_t - u_t,\ u_t - u_{t+1} \rangle + \|\delta_t\|^2 \leq 2D\|\delta_t\| + \|\delta_t\|^2.$$

Thus, we get:

$$A_T \leq \frac{\eta_{T+1} D^2}{2} + \eta_{T+1} \sum_{t=1}^{T-1} \left(D\|\delta_t\| + \frac{1}{2}\|\delta_t\|^2\right). \tag{28}$$

Step 5: Final bound. Substituting (27), (28), and the initialization estimate

$$\langle \tilde{h}_1, x_1 - u_1 \rangle \leq \frac{\eta_{T+1} D^2}{2}, \quad \text{into} \quad (24) \quad \text{we obtain:}$$

$$\begin{aligned}\sum_{t=1}^{T} \langle h_t, x_t - u_t \rangle \leq{}& \frac{\eta_{T+1} D^2}{2} + \eta_{T+1} \sum_{t=1}^{T-1} \left(D\|\delta_t\| + \frac{1}{2}\|\delta_t\|^2\right) \\ &+ \sum_{t=1}^{T} \min\left(D\|e_t\|,\ \frac{1}{\eta_{t+1}} \operatorname{huber}\big(\|e_t\|, \|n_t\|\big)\right) - \sum_{t=1}^{T} \langle e_t, \delta_t \rangle + \frac{\eta_{T+1} D^2}{2}.\end{aligned}$$

Therefore, it holds:

$$\begin{aligned}\sum_{t=1}^{T} \langle h_t, x_t - u_t \rangle \leq{}& \eta_{T+1}\left(D^2 + \tfrac{3}{2} D\, P_T\right) + \sum_{t=1}^{T} \min\left(D\|e_t\|,\ \frac{1}{\eta_{t+1}} \operatorname{huber}\big(\|e_t\|, \|n_t\|\big)\right) \\ &- \sum_{t=1}^{T} \langle e_t, \delta_t \rangle.\end{aligned}$$

And this proves the claim. □

B.2.2 Delay as optimism

We now transfer the previous undelayed optimistic bound to the delayed-feedback setting. The key observation, following Lemma 2 of [19], is that a delayed optimistic update can be written as an undelayed optimistic one-step update with a particular choice of "bad" hint. This allows us to reuse Theorem 4 without reproving the whole dynamic regret argument. In the delayed setting, the gradient $h_t \in \nabla L_t(x_t)$ is revealed only at time $t+m$. Throughout the delayed analysis, we use the convention

$$h_r = \widetilde{h}_r = 0, \qquad r \le 0,$$

and empty sums are set to zero. Thus, the first m rounds are warm-start rounds: the corresponding decisions are included in the regret, but no genuine delayed feedback has yet been revealed. Define the bad hint

$$\tilde{h}_{t+1}^{\mathrm{bad}} := \tilde{h}_{t-m+1:t+1} - h_{t-m+1:t}.$$

Then the optimistic one-step update

$$x_{t+1} = \Pi_{\mathcal{X}}\left(x_t - \frac{1}{\eta_{t+1}}\big(h_t + \tilde{h}_{t+1}^{\mathrm{bad}} - \tilde{h}_t^{\mathrm{bad}}\big)\right)$$

is equivalent to the delayed optimistic update

$$x_{t+1} = \Pi_{\mathcal{X}}\left(x_t - \frac{1}{\eta_{t+1}}\big(h_{t-m} + \tilde{h}_{t-m+1:t+1} - \tilde{h}_{t-m:t}\big)\right).$$

Indeed, the update uses the most recent revealed gradient h_{t-m}, together with predictions for the missing gradients.

Corollary 3 (Dynamic delayed optimistic OGD)**.** *As a direct application of Lemma 2 [19], let $(\eta_t)_{t\ge1}$ be a nondecreasing positive sequence, and consider the delayed optimistic update*

$$x_{t+1} = \Pi_{\mathcal{X}}\left(x_t - \frac{1}{\eta_{t+1}}\big(h_{t-m} + \tilde{h}_{t-m+1:t+1} - \tilde{h}_{t-m:t}\big)\right), \qquad t = 1, \dots, T.$$

For any comparator sequence $u_1^T \in \mathcal{X}^T$, define:

$$P_T := \sum_{t=1}^{T-1} \|u_{t+1} - u_t\|, \qquad \delta_t := u_t - u_{t+1},$$

with the convention $u_{T+1} := u_T$. Also define:

$$\bar{e}_t := h_{t-m:t} - \tilde{h}_{t-m:t}, \quad \textit{and} \quad \bar{n}_t := h_{t-m} + \tilde{h}_{t-m+1:t+1} - \tilde{h}_{t-m:t}.$$

Then the dynamic regret

$$\mathcal{R}_T(L) := \sum_{t=1}^{T} L_t(x_t) - \sum_{t=1}^{T} L_t(u_t)$$

satisfies

$$\mathcal{R}_T(L) \le \eta_{T+1}\Big(D^2 + \frac{3}{2}D\,P_T\Big) + \sum_{t=1}^{T} \min\left\{D\|\bar{e}_t\|,\ \frac{1}{\eta_{t+1}}\,\mathrm{huber}\big(\|\bar{e}_t\|, \|\bar{n}_t\|\big)\right\} - \sum_{t=1}^{T}\langle \bar{e}_t, \delta_t\rangle. \tag{29}$$

B.2.3 Proof of Corollary 2

From the delayed dynamic bound, obtained by combining Theorem 4 with the delay to optimism reduction (with bad hints), we have

$$\text{Term (II)} \le \eta_{T+1} D\big(D + 2P_T\big) + \sum_{t=1}^{T} \min\left\{D\|\bar{e}_t\|,\ \frac{1}{\eta_{t+1}}\,\mathrm{huber}(\|\bar{e}_t\|, \|\bar{n}_t\|)\right\} - \sum_{t=1}^{T}\langle \bar{e}_t, \delta_t\rangle. \tag{30}$$

The first two terms have the form required by the delayed-upper-bound Lemma 22 of [19], with the regularization scale now depending on the path length. The last term is specific to the dynamic comparator and will be controlled separately by introducing an axillary variable β.

For any $\beta > 0$, Young's inequality gives

$$-\langle \bar{e}_t, \delta_t \rangle \leq \frac{1}{2\beta}\|\bar{e}_t\|^2 + \frac{\beta}{2}\|\delta_t\|^2.$$

Summing over t, and taking $\beta = \sqrt{\frac{\sum_{t=1}^T \|\bar{e}_t\|^2}{\sum_{t=1}^T \|\delta_t\|^2}}$, we can write:

$$-\sum_{t=1}^T \langle \bar{e}_t, \delta_t \rangle \leq \sqrt{\left(\sum_{t=1}^T \|\bar{e}_t\|^2\right)\left(\sum_{t=1}^T \|\delta_t\|^2\right)}.$$

Therefore we can update the bound of the term as follows:

$$\text{Term (II)} \leq \eta_{T+1} D\big(D + 2P_T\big) + \sum_{t=1}^T \min\left\{a_t, \frac{b_t}{\eta_{t+1}}\right\} + \sqrt{\left(\sum_{t=1}^T \|\bar{e}_t\|^2\right)\left(\sum_{t=1}^T \|\delta_t\|^2\right)}, \quad (31)$$

where

$$a_t := D\|\bar{e}_t\|, \qquad b_t := \mathrm{huber}(\|\bar{e}_t\|, \|\bar{n}_t\|).$$

We now apply the DUB tuning Lemma with $\alpha = D\big(D + 2P_T\big)$, and step:

$$\eta_{t+1} = \frac{2}{\alpha} \max_{j \leq t-m-1} a_{j-m+1:j} + \frac{1}{\alpha}\sqrt{\sum_{i=1}^{t-m} a_i^2 + 2\alpha b_i},$$

where quantities involving negative indices are interpreted according to the zero-padding convention stated above.

We allow η_t to be dependent on P_T here because this bound corresponds to the base learner tuned to the correct path-length scale. This gives:

$$\eta_{T+1} D\big(D + 2P_T\big) + \sum_{t=1}^T \min\left\{a_t, \frac{b_t}{\eta_{t+1}}\right\} \leq 2\left(2\max_{t\in[T]} a_{t-m:t-1} + \sqrt{\sum_{t=1}^T (a_t^2 + 2\alpha b_t)}\right), \quad (32)$$

because $D(D + 2P_T)/\alpha = 1$. We next bound the two terms in (32). First, using:

$$\mathrm{huber}(x, y) \leq \frac{1}{2}x^2, \text{ we obtain: } b_t \leq \frac{1}{2}\|\bar{e}_t\|^2.$$

Hence, we can write:

$$a_t^2 + 2\alpha b_t \leq D^2\|\bar{e}_t\|^2 + \alpha\|\bar{e}_t\|^2 = (D^2 + \alpha)\|\bar{e}_t\|^2,$$

and therefore,

$$\sum_{t=1}^T (a_t^2 + 2\alpha b_t) \leq (D^2 + \alpha)\sum_{t=1}^T \|\bar{e}_t\|^2.$$

Second, by Cauchy–Schwarz,

$$a_{t-m:t-1} = \sum_{j=\max\{1,t-m\}}^{t-1} a_j \leq \sqrt{m}\left(\sum_{j=\max\{1,t-m\}}^{t-1} a_j^2\right)^{1/2}.$$

Since $a_j = D\|\bar{e}_j\|$, we get

$$\max_{t\in[T]} a_{t-m:t-1} \leq D\sqrt{m}\sqrt{\sum_{t=1}^T \|\bar{e}_t\|^2}.$$

Combining the previous estimates yields:

$$\eta_{T+1}D(D+2P_T)+\sum_{t=1}^{T}\min\left\{a_t,\frac{b_t}{\eta_{t+1}}\right\}\leq\left(4D\sqrt{m}+2\sqrt{D^2+\alpha}\right)\sqrt{\sum_{t=1}^{T}\|\bar{e}_t\|^2}.$$

Since $\alpha=D(D+2P_T)$, we have $D^2+\alpha=2D(D+P_T)$. Thus:

$$2\sqrt{D^2+\alpha}=2\sqrt{2D(D+P_T)}\leq 3\sqrt{D(D+P_T)}.$$

It remains to bound the interaction term. Since it is $\|\delta_t\|\leq D$, we can write:

$$\sum_{t=1}^{T}\|\delta_t\|^2\leq D\sum_{t=1}^{T}\|\delta_t\|=D\,P_T.$$

Therefore,

$$\sqrt{\left(\sum_{t=1}^{T}\|\bar{e}_t\|^2\right)\left(\sum_{t=1}^{T}\|\delta_t\|^2\right)}\leq\sqrt{D\,P_T}\sqrt{\sum_{t=1}^{T}\|\bar{e}_t\|^2}.$$

Since it is:

$$\sqrt{D\,P_T}\leq\sqrt{D(D+P_T)},$$

we obtain the looser, but cleaner, bound:

$$\text{Term (II)}\leq 4\left(D\sqrt{m}+\sqrt{D(D+P_T)}\right)\sqrt{\sum_{t=1}^{T}\|\bar{e}_t\|^2}.$$

Finally, given that it holds:

$$\|\bar{e}_t\|^2\leq(m+1)\sum_{s=t-m}^{t}\|h_s-\tilde{h}_s\|^2,$$

we get,

$$\text{Term (II)}\leq 4(m+1)\left(D\sqrt{m}+\sqrt{D(D+P_T)}\right)\sqrt{\sum_{t=1}^{T}\|h_t-\tilde{h}_t\|^2}. \tag{33}$$

a cleaner term:

$$\text{Term (II)}\leq 4(m+1)^{3/2}\left(D+\sqrt{D(D+P_T)}\right)\sqrt{\sum_{t=1}^{T}\|h_t-\tilde{h}_t\|^2}.$$

This proves the corollary.

B.2.4 Proof of Lemma 3

Lemma 3 bounds Term (I) using the optimistic delayed AdaHedge meta-learner of [19]. At round t, the meta-loss is the linear function

$$\langle M_t,w\rangle,\qquad M_t:=X_t^\top h_t.$$

Thus, its actual and predicted gradients are

$$M_t=X_t^\top h_t,\qquad\widetilde{M}_t:=X_t^\top\widetilde{h}_t.$$

In the delayed setting, the meta-learner needs a prediction of the unavailable block of gradient meta-losses. Thus, at the time when w_{t+1} is computed, we use the block hint:

$$\tilde{M}_{t-m+1:t+1}:=\sum_{s=t-m+1}^{t+1}\tilde{M}_s=\sum_{s=t-m+1}^{t+1}X_s^\top\tilde{h}_s,$$

which predicts

$$M_{t-m+1:t+1} := \sum_{s=t-m+1}^{t+1} M_s.$$

Importantly, only the gradient-related quantities h_s need to be predicted. The matrices X_s are known when the corresponding meta-update is performed. In particular, after processing the feedback available at round t, each base learner outputs its candidate decision for round $t+1$, so:

$$X_{t+1} = [x_{t+1}^1, \ldots, x_{t+1}^N]$$

is available before the meta-learner computes w_{t+1}. Therefore, there is no need to predict X_{t+1}. We now define the *meta-loss prediction error*

$$\varepsilon_t := \left\| \sum_{s=t-m}^{t} (\tilde{M}_s - M_s) \right\|_\infty .$$

As before, we use the convention $M_r = \widetilde{M}_r = 0$ for $r \le 0$. We first bound Term (I) in terms of ε_t, and later relate ε_t to prediction errors $E_T(f)$ and $E_T(g^+)$. Specifically, we want to bound:

$$\text{Term (I)} = \sum_{t=1}^{T} \langle M_t,\ w_t - e_i \rangle.$$

In the considered delayed setting, at round $t+1$ only the losses $M_{1:t-m} := (M_1, \ldots, M_{t-m})$ are revealed, while the missing block $M_{t-m+1:t+1}$ may be predicted by a hint. For each $t \in [T]$, define

$$b_t := \mathrm{huber}\left(\left\| \sum_{s=t-m}^{t} (\tilde{M}_s - M_s) \right\|_\infty , \|M_t\|_\infty \right) \text{ and } a_t := 2\min\left\{ \left\| \sum_{s=t-m}^{t} (\tilde{M}_s - M_s) \right\|_\infty , \|M_t\|_\infty \right\}$$

where we use the ℓ_1-geometry on $\mathcal{W} = \Delta_N$, so that $\mathrm{diam}(\mathcal{W}) = 2$.

Then, by Theorem 12 of [19], we can write:

$$\text{Term (I)} \le \left(\frac{\psi(e_i)}{\alpha} + 1 \right) \left(2 \max_{t \in [T]} a_{t-m:t-1} + \sqrt{\sum_{t=1}^{T} \left(a_t^2 + 2\alpha\, b_t \right)} \right). \tag{34}$$

We use the shifted negative-entropy regularizer:

$$\psi(w) = \sum_{j=1}^{N} w_j \log(w_j) + \log N, \qquad w \in \Delta_N,$$

and choose $\alpha = \sup_{w \in \Delta_N} \psi(w) = \log N$. Since $\psi(e_i) = \log N$, we have that:

$$\frac{\psi(e_i)}{\alpha} + 1 = 2.$$

Then we can write

$$a_{t,F} = 2\min\{\varepsilon_t, \|M_t\|_\infty\} \le 2\varepsilon_t. \quad \text{and} \quad b_t = \mathrm{huber}(\varepsilon_t, \|M_t\|_\infty) \le \frac{1}{2}\varepsilon_t^2.$$

Therefore,

$$a_{t,F}^2 + 2\alpha b_t \le 4\varepsilon_t^2 + \alpha\varepsilon_t^2 = (4 + \log N)\varepsilon_t^2, \quad \text{and thus:} \quad \sum_{t=1}^{T} \left(a_{t,F}^2 + 2\alpha b_t \right) \le (4 + \log N) \sum_{t=1}^{T} \varepsilon_t^2.$$

Next,

$$a_{t-m:t-1} = \sum_{j=\max\{1,t-m\}}^{t-1} a_{j,F} \le 2 \sum_{j=\max\{1,t-m\}}^{t-1} \varepsilon_j.$$

By Cauchy–Schwarz,

$$a_{t-m:t-1} \le 2\sqrt{m}\left(\sum_{j=\max\{1,t-m\}}^{t-1}\varepsilon_j^2\right)^{1/2} \le 2\sqrt{m}\left(\sum_{j=1}^{T}\varepsilon_j^2\right)^{1/2}.$$

Thus,

$$\max_{t\in[T]} a_{t-m:t-1} \le 2\sqrt{m}\sqrt{\sum_{t=1}^{T}\varepsilon_t^2}.$$

We now relate ε_t to the prediction error of the credit gradients. Since it is $\tilde{M}_s = X_s^\top \tilde{h}_s$ and $M_s = X_s^\top h_s$, we have $\tilde{M}_s - M_s = X_s^\top(\tilde{h}_s - h_s)$. Therefore, we can write:

$$\varepsilon_t = \left\|\sum_{s=t-m}^{t} X_s^\top(\tilde{h}_s - h_s)\right\|_\infty \le \sum_{s=t-m}^{t}\|X_s^\top(\tilde{h}_s - h_s)\|_\infty.$$

Now, for any vector v it holds:

$$\|X_s^\top v\|_\infty = \max_{i\in[N]}|\langle x_s^i, v\rangle| \le \max_{i\in[N]}\|x_s^i\|\,\|v\| \le D\|v\|,$$

since $\|x_s^i\| \le D$ for all $i \in [N]$. Therefore, $\varepsilon_t \le D\sum_{s=t-m}^{t}\|h_s - \tilde{h}_s\|$, and applying Cauchy–Schwarz we can write:

$$\varepsilon_t^2 \le D^2(m+1)\sum_{s=t-m}^{t}\|h_s - \tilde{h}_s\|^2.$$

Summing over t and using that each term $\|h_s - \tilde{h}_s\|^2$ appears in at most $m+1$ windows yields

$$\sum_{t=1}^{T}\varepsilon_t^2 \le D^2(m+1)^2\sum_{t=1}^{T}\|h_t - \tilde{h}_t\|^2 \;\Rightarrow\; \sqrt{\sum_{t=1}^{T}\varepsilon_t^2} \le D(m+1)\sqrt{\sum_{t=1}^{T}\|h_t - \tilde{h}_t\|^2}.$$

Returning to (34) and using the bounds derived above for each term, we obtain

$$\text{Term (I)} \le 2D(m+1)\left(4\sqrt{m} + \sqrt{4+\log N}\right)\sqrt{\sum_{t=1}^{T}\|h_t - \tilde{h}_t\|^2}. \tag{35}$$

A cleaner upper bound is

$$\text{Term (I)} \le 8D(m+1)^{\frac{3}{2}}\left(1 + \sqrt{1+\log N}\right)\sqrt{\sum_{t=1}^{T}\|h_t - \tilde{h}_t\|^2}. \tag{36}$$

C Proofs of Section 4

C.1 Proof of Theorem 2

Proof. Define

$$A_N := 8D(m+1)^{3/2}\left(1 + \sqrt{1+\log N}\right),$$

and

$$A_P := 4(m+1)^{3/2}\left(D + \sqrt{D(D+P_T)}\right).$$

For compactness, let

$$A := A_N + A_P.$$

From the delayed optimistic OCO analysis, the surrogate dynamic regret satisfies

$$\mathcal{R}_T(L) \le A\sqrt{\sum_{t=1}^{T} \|h_t - \widetilde{h}_t\|_2^2}.$$

We now relate the surrogate-gradient prediction error to the prediction errors of the objective and constraint functions. In reveal-time notation, the surrogate loss is

$$L_t(x_{t-m}) = V f_t(x_{t-m}) + \Phi'(Q_{t-m-1}) g_t^+(x_{t-m}).$$

Equivalently, setting $s = t - m$, the same surrogate can be written in decision-time notation as

$$L_{s+m}(x_s) = V f_{s+m}(x_s) + \Phi'(Q_{s-1}) g_{s+m}^+(x_s).$$

Thus, the feedback associated with decision x_s is

$$h_s = \nabla L_{s+m}(x_s).$$

Since Φ' is nondecreasing and $Q_{s-1} \le Q_T$, we have

$$\|h_s - \widetilde{h}_s\|_2 \le V\|\Delta_s^f\|_2 + \Phi'(Q_T)\|\Delta_s^{g^+}\|_2.$$

Relabeling s as t, and using the quadratic penalty $\Phi(Q) = Q^2$, so that $\Phi'(Q_T) = 2Q_T$, Minkowski's inequality gives

$$\sqrt{\sum_{t=1}^{T} \|h_t - \widetilde{h}_t\|_2^2} \le V\sqrt{E_T(f)} + 2Q_T\sqrt{E_T(g^+)}.$$

Therefore,

$$\mathcal{R}_T(L) \le AV\sqrt{E_T(f)} + 2AQ_T\sqrt{E_T(g^+)}.$$

Combining this with the delayed regret decomposition lemma gives

$$Q_T^2 - Q_{m-1}^2 + V\mathcal{R}_T(f) \le AV\sqrt{E_T(f)} + 2AQ_T\sqrt{E_T(g^+)} + 2G(m+1)Q_T.$$

Define

$$B_T := A\sqrt{E_T(g^+)} + G(m+1).$$

Then

$$Q_T^2 - Q_{m-1}^2 + V\mathcal{R}_T(f) \le AV\sqrt{E_T(f)} + 2B_TQ_T.$$

For the regret bound, rearranging gives

$$V\mathcal{R}_T(f) \le AV\sqrt{E_T(f)} + 2B_TQ_T - Q_T^2 + Q_{m-1}^2.$$

Using $2ab - b^2 \le a^2$, with $a = B_T$ and $b = Q_T$, we obtain

$$V\mathcal{R}_T(f) \le AV\sqrt{E_T(f)} + B_T^2 + Q_{m-1}^2.$$

Thus,

$$\mathcal{R}_T(f) \le A\sqrt{E_T(f)} + \frac{B_T^2 + Q_{m-1}^2}{V}.$$

We now choose

$$V := 4(m+1)^{3/2}\left(1 + \sqrt{1 + \log N}\right)^2\sqrt{E_T(g^+)} + 2G^2(m+1) + Q_{m-1}^2.$$

Since

$$B_T^2 + Q_{m-1}^2 = \left(A\sqrt{E_T(g^+)} + G(m+1)\right)^2 + Q_{m-1}^2,$$

we have

$$B_T^2 + Q_{m-1}^2 \le 2A^2E_T(g^+) + 2G^2(m+1)^2 + Q_{m-1}^2.$$

Therefore,

$$\begin{aligned}\frac{B_T^2+Q_{m-1}^2}{V} &\le \frac{2A^2E_T(g^+)}{4(m+1)^{3/2}\left(1+\sqrt{1+\log N}\right)^2\sqrt{E_T(g^+)}} \\ &\quad + \frac{2G^2(m+1)^2}{2G^2(m+1)} + \frac{Q_{m-1}^2}{Q_{m-1}^2} \\ &\le \frac{A^2}{2(m+1)^{3/2}\left(1+\sqrt{1+\log N}\right)^2}\sqrt{E_T(g^+)} + (m+1) + 1.\end{aligned}$$

Hence,

$$\mathcal{R}_T(f) \le A\sqrt{E_T(f)} + \frac{A^2}{2(m+1)^{3/2}\left(1+\sqrt{1+\log N}\right)^2}\sqrt{E_T(g^+)} + (m+1) + 1.$$

Substituting the definitions of A_N and A_P, and hiding problem-dependent constants, gives

$$A = \mathcal{O}\Big((m+1)^{3/2}\left(1+\sqrt{1+\log N}+\sqrt{1+P_T}\right)\Big).$$

Moreover, since

$$\frac{\left(1+\sqrt{1+\log N}+\sqrt{1+P_T}\right)^2}{\left(1+\sqrt{1+\log N}\right)^2} = \mathcal{O}(1+P_T),$$

we obtain

$$\boxed{\begin{aligned}\mathcal{R}_T(f) \le \mathcal{O}\Bigg(&(m+1)^{3/2}\left(1+\sqrt{1+\log N}+\sqrt{1+P_T}\right)\sqrt{E_T(f)} \\ &+ (m+1)^{3/2}(1+P_T)\sqrt{E_T(g^+)} + (m+1)\Bigg).\end{aligned}}$$

We now prove the CCV bound. Using the lower bound

$$\mathcal{R}_T(f) \ge -GDT$$

in the coupled inequality gives

$$Q_T^2 - 2B_TQ_T - \left(AV\sqrt{E_T(f)} + VGDT + Q_{m-1}^2\right) \le 0.$$

Since $Q_T \ge 0$, the implication

$$x^2 - 2ax - b \le 0 \quad\Longrightarrow\quad x \le a + \sqrt{a^2+b}$$

yields

$$Q_T \le B_T + \sqrt{B_T^2 + AV\sqrt{E_T(f)} + VGDT + Q_{m-1}^2}.$$

Using

$$\sqrt{x+y+z+w} \le \sqrt{x}+\sqrt{y}+\sqrt{z}+\sqrt{w},$$

we get

$$Q_T \le 2B_T + \sqrt{AV\sqrt{E_T(f)}} + \sqrt{VGDT} + Q_{m-1}.$$

Using the crude bound $E_T(f) \le T$, we have

$$\sqrt{AV\sqrt{E_T(f)}} \le \sqrt{AV\sqrt{T}} = \frac{\sqrt{A}}{T^{1/4}}\sqrt{VT}.$$

Also,

$$\sqrt{VGDT} = \mathcal{O}(\sqrt{VT}),$$

where problem-dependent constants are hidden. Therefore,

$$Q_T \le 2B_T + \mathcal{O}\left(\left(1 + \frac{\sqrt{A}}{T^{1/4}}\right)\sqrt{VT}\right) + Q_{m-1}.$$

It remains to expand $\sqrt{VT}$. By the choice of V,

$$V = 4(m+1)^{3/2}\left(1+\sqrt{1+\log N}\right)^2\sqrt{E_T(g^+)} + 2G^2(m+1) + Q_{m-1}^2.$$

Hence,

$$\sqrt{VT} \le 2(m+1)^{3/4}\left(1+\sqrt{1+\log N}\right)\sqrt{T}\,E_T(g^+)^{1/4} + \sqrt{2}\,G\sqrt{m+1}\sqrt{T} + Q_{m-1}\sqrt{T}.$$

Substituting this into the previous inequality gives

$$\begin{aligned} Q_T \le \mathcal{O}\Bigg(& A\sqrt{E_T(g^+)} + G(m+1) + Q_{m-1} \\ & + \left(1 + \frac{\sqrt{A}}{T^{1/4}}\right)\Big[(m+1)^{3/4}\left(1+\sqrt{1+\log N}\right)\sqrt{T}\,E_T(g^+)^{1/4} \\ & \qquad + \sqrt{m+1}\sqrt{T} + Q_{m-1}\sqrt{T}\Big]\Bigg). \end{aligned}$$

Finally, substituting

$$A = \mathcal{O}\left((m+1)^{3/2}\left(1+\sqrt{1+\log N}+\sqrt{1+P_T}\right)\right),$$

we get

$$\sqrt{A} = \mathcal{O}\left((m+1)^{3/4}\sqrt{1+\sqrt{1+\log N}+\sqrt{1+P_T}}\right).$$

Therefore,

$$\begin{aligned} Q_T \le \mathcal{O}\Bigg(& (m+1)^{3/2}\left(1+\sqrt{1+\log N}+\sqrt{1+P_T}\right)\sqrt{E_T(g^+)} + (m+1) + Q_{m-1} \\ & + \left[1 + \frac{(m+1)^{3/4}\sqrt{1+\sqrt{1+\log N}+\sqrt{1+P_T}}}{T^{1/4}}\right] \text{this term is of order } (m+1)^{3/4} \\ & \quad \times \left[(m+1)^{3/4}\left(1+\sqrt{1+\log N}\right)\sqrt{T}\,E_T(g^+)^{1/4} + \sqrt{m+1}\sqrt{T} + Q_{m-1}\sqrt{T}\right]\Bigg). \end{aligned}$$

Thus

$$\boxed{Q_T \le \mathcal{O}\left((m+1)^{3/2}\left(1+\sqrt{1+\log N}+\sqrt{1+P_T}\right)\sqrt{T}E_T(g^+)^{\frac{1}{4}} + (m+1)^{\frac{5}{4}}\sqrt{T}\right).}$$

This proves the stated regret and CCV bounds. □

C.2 Proof of Theorem 3

Proof. Since P_T is known, we do not need the meta-learner. The base learner is tuned directly to the path length P_T. Define

$$A_P := 4(m+1)^{3/2}\left(D + \sqrt{D(D+P_T)}\right).$$

Then the delayed optimistic dynamic OCO bound for the base learner gives

$$\mathcal{R}_T(L) \le A_P \sqrt{\sum_{t=1}^{T} \|h_t - \tilde{h}_t\|_2^2}.$$

Thus

$$\Phi(Q_T) - \Phi(Q_{m-1}) + \mathcal{R}_T(f) \le A_P \sqrt{E_T(f)} + \Big(A_P \sqrt{E_T(g^+)} + G(m+1)\Big)\Phi'(Q_T).$$

Now choose

$$\Phi(Q) = e^{\lambda Q} - 1, \qquad \Phi'(Q) = \lambda e^{\lambda Q}.$$

Then

$$e^{\lambda Q_T} - e^{\lambda Q_{m-1}} + \mathcal{R}_T(f) \le A_P \sqrt{E_T(f)} + \lambda\Big(A_P \sqrt{E_T(g^+)} + G(m+1)\Big)e^{\lambda Q_T}.$$

Let

$$B_T := A_P \sqrt{E_T(g^+)} + G(m+1).$$

By the choice of λ,

$$\lambda = \frac{1}{2B_T},$$

and therefore $\lambda B_T = 1/2$. Hence

$$\frac{1}{2} e^{\lambda Q_T} + \mathcal{R}_T(f) \le e^{\lambda Q_{m-1}} + A_P \sqrt{E_T(f)}.$$

We first bound the regret. From the previous inequality,

$$\mathcal{R}_T(f) \le A_P \sqrt{E_T(f)} + e^{\lambda Q_{m-1}} - \frac{1}{2} e^{\lambda Q_T}.$$

Since the last term is non-positive, we get

$$\mathcal{R}_T(f) \le A_P \sqrt{E_T(f)} + e^{\lambda Q_{m-1}}.$$

The term Q_{m-1} only depends on the initial delayed rounds and is absorbed into the constants. Therefore,

$$\mathcal{R}_T(f) = \mathcal{O}\Big(A_P \sqrt{E_T(f)}\Big).$$

Using the definition of A_P, this yields

$$\mathcal{R}_T(f) = \mathcal{O}\Big((m+1)^{3/2} \sqrt{(1+P_T)E_T(f)}\Big).$$

We now bound the CCV. Using the lower bound

$$\mathcal{R}_T(f) \ge -GDT,$$

we obtain

$$\frac{1}{2} e^{\lambda Q_T} \le e^{\lambda Q_{m-1}} + A_P \sqrt{E_T(f)} + GDT.$$

Thus,

$$e^{\lambda Q_T} \le 2\left(e^{\lambda Q_{m-1}} + A_P \sqrt{E_T(f)} + GDT\right).$$

Taking logarithms gives

$$Q_T \le \frac{1}{\lambda} \log\Big(2\left(e^{\lambda Q_{m-1}} + A_P \sqrt{E_T(f)} + GDT\right)\Big).$$

Since $1/\lambda = 2B_T$, we get

$$Q_T \le 2\Big(A_P \sqrt{E_T(g^+)} + G(m+1)\Big) \log\Big(2\left(e^{\lambda Q_{m-1}} + A_P \sqrt{E_T(f)} + GDT\right)\Big).$$

Under the standard boundedness assumptions, the logarithmic factor is of order $\mathcal{O}(\log T)$. Therefore,

$$Q_T = \mathcal{O}\Big(\Big(A_P \sqrt{E_T(g^+)} + G(m+1)\Big) \log T\Big).$$

Substituting the definition of A_P, we obtain

$$Q_T = \mathcal{O}\Big(\Big((m+1)^{3/2} \sqrt{(1+P_T)E_T(g^+)} + G(m+1)\Big) \log T\Big).$$

This proves the theorem. □

D Proof of Section 5

D.1 Proof of Lemma 2

We start with the proof of the main result of this section, namely Lemma 2.

Proof. The proof follows from two remarkably simple observations: first, convexity allows us to linearize each memory loss around the played memory window; second, after re-indexing the resulting terms, the contribution of each decision x_s can be grouped into a single gradient quantity that is revealed only after, at most, m future rounds.

In detail, recall that the memory regret is:

$$\mathcal{R}_T(L) := \sum_{t=m}^{T} L_t(x_{t-m}, \dots, x_t) - \sum_{t=m}^{T} L_t(u_{t-m}, \dots, u_t).$$

Since L_t is convex in its joint argument $(z_0, \dots, z_m) \in \mathcal{X}^{m+1}$, the first-order convexity inequality gives, for every $t \in [T_m] := \{m, \dots, T\}$:

$$L_t(x^t_{t-m}) - L_t(u_{t-m:t}) \le \sum_{j=0}^{m} \left\langle \nabla_{x_{t-j}} L_t(x^t_{t-m}),\, x_{t-j} - u_{t-j} \right\rangle.$$

Here $\nabla_{x_{t-j}} L_t(x^t_{t-m})$ denotes the partial gradient of L_t with respect to its $(m-j)$-th coordinate, evaluated at the played decision $x_{t-m:t}$. Summing the above inequality over t, we obtain:

$$\mathcal{R}_T(L) \le \sum_{t=m}^{T} \sum_{j=0}^{m} \left\langle \nabla_{x_{t-j}} L_t(x^t_{t-m}),\, x_{t-j} - u_{t-j} \right\rangle.$$

At this stage, the right-hand side is still indexed by the loss time t. However, for the online-learning interpretation, it is more natural to regroup the terms according to the decision time. To do so, we set:

$$s := t - j, \qquad \text{or, equivalently:} \qquad t = s + k, \;\; k := j.$$

Then, for a fixed s, the index k must satisfy

$$0 \le k \le m \qquad \text{and} \qquad m \le s + k \le T.$$

Therefore the admissible range of k is

$$k \in \big[\max\{0, m-s\},\; \min\{m, T-s\}\big].$$

Using this change of variables and swapping the order of summation, we can write:

$$\mathcal{R}_T(L) \le \sum_{t=1}^{T} \left\langle \sum_{k=\max\{0,m-t\}}^{\min\{m,T-t\}} \nabla_{x_t} L_{t+k}(x^{t+k}_{t+k-m}),\, x_t - u_t \right\rangle,$$

where we just used t instead of s. This motivates the definition of the *credit gradient*

$$h_t := \sum_{k=\max\{0,m-t\}}^{\min\{m,T-t\}} \nabla_{x_t} L_{t+k}(x^{t+k}_{t+k-m}).$$

With this notation, we can write:

$$\mathcal{R}_T(L) \le \sum_{t=1}^{T} \langle h_t,\, x_t - u_t \rangle.$$

Indexing convention and delayed interpretation. The original memory-dependent losses are indexed by $t \in [T_m] := \{m, \dots, T\}$, since a full memory window is only available from round m onward. After linearization, however, the auxiliary delayed problem is naturally indexed by the decision times $s \in [T]$. Indeed, even the early decisions $x_1, \dots, x_m$ may appear in later memory windows, and the corresponding partial gradients must be assigned to the credit gradient of the decision that generated them. Algorithmically, feedback starts at round m: whenever a memory loss L_t is revealed, it contributes partial gradients to all decisions in its memory window $x_{t-m:t}$.

This also explains why the resulting auxiliary problem is a delayed linear problem. By construction, h_t aggregates the effect of the single decision x_t on the losses

$$L_t,\ L_{t+1},\ \dots,\ L_{t+m}.$$

Hence h_t is not entirely known when x_t is played; it is only fully revealed once all these losses have been observed, that is, by round $t + m$. Therefore, $\{h_t\}_{t=1}^{T}$ is a sequence of linear losses revealed with delay at most m. Consequently, the original memory-dependent regret is upper bounded by the regret of this delayed linear problem. Thus, any applicable delayed-OCO regret guarantee for the auxiliary linear losses can be transferred to the memory-dependent problem through the credit gradients h_s. The resulting guarantee has the same structure as in the delayed setting; the only difference is the effective gradients are now the credit gradients h_t, whose size introduces constants depending on the memory length m.

□

At a conceptual level, the key idea is that we postpone the construction of the auxiliary delayed loss until after linearizing the memory-dependent regret. The contribution of each decision is then captured through the corresponding coordinate block of the joint gradient and aggregated into its credit gradient. This is also why constructing optimistic hints may require predicting the future block $x_{t+1:t+m}$, rather than only x_{t+1}. In the separable case, this recovers the usual forward-gradient aggregation, while our construction does not require separability and therefore preserves the original joint feasibility benchmark. In contrast, [5] applies the penalty to an individual component of the separable constraint, $g_{t+i}^{+,i}(x_t)$, whereas here the penalty is applied to the joint constraint function.

D.2 Proof of Remark 5

Proof. To transfer the delayed-feedback guarantees of Sections 3–4 to the memory setting, it remains to relate the prediction error of the credit gradients to the memory prediction errors defined in Remark 5.

In the unconstrained case, $L_t = f_t$, and therefore

$$h_s - \widetilde{h}_s = \sum_{t \in I_s} \left[\nabla_{x_s} f_t(x_{t-m}^t) - \nabla_{x_s} \widetilde{f}_t(\widetilde{x}_{t-m}^t) \right].$$

Hence, directly from (14),

$$\left(\sum_{s=1}^{T} \|h_s - \widetilde{h}_s\|^2 \right)^{1/2} = \sqrt{E_T^{\mathrm{mem}}(f)}.$$

Therefore, Theorem 1 applies directly to the memory problem through Corollary 1.

We next consider the constrained case. To recover the prediction-error structure used in Section 4, it suffices to show that

$$\left(\sum_{s=1}^{T} \|h_s - \widetilde{h}_s\|^2 \right)^{1/2} \le V \sqrt{E_T^{\mathrm{mem}}(f)} + \Phi'(Q_T) \sqrt{E_T^{\mathrm{mem}}(g^+)}. \tag{37}$$

Recall that

$$L_t(x_{t-m}^t) = V f_t(x_{t-m}^t) + \Phi'(Q_{t-m-1}) g_t^+(x_{t-m}^t),$$

and

$$h_s = \sum_{t \in I_s} \nabla_{x_s} L_t(x_{t-m}^t).$$

For the predicted credit gradient, we use the same queue coefficient, namely

$$\widetilde{L}_t(\widetilde{x}_{t-m}^t) = V\widetilde{f}_t(\widetilde{x}_{t-m}^t) + \Phi'(Q_{t-m-1})\widetilde{g}_t^+(\widetilde{x}_{t-m}^t).$$

Indeed, $t \in I_s$ implies $t \le s+m$, and therefore $t-m-1 \le s-1$. Hence, $\Phi'(Q_{t-m-1})$ is already available when the hint associated with x_s is formed.

Using the definitions of h_s and $\widetilde{h}_s$, we obtain

$$\begin{aligned} h_s - \widetilde{h}_s = V \sum_{t\in I_s} \left[\nabla_{x_s} f_t(x_{t-m}^t) - \nabla_{x_s}\widetilde{f}_t(\widetilde{x}_{t-m}^t)\right] \\ + \sum_{t\in I_s} \Phi'(Q_{t-m-1}) \left[\nabla_{x_s} g_t^+(x_{t-m}^t) - \nabla_{x_s}\widetilde{g}_t^+(\widetilde{x}_{t-m}^t)\right]. \end{aligned} \tag{38}$$

It remains to control the second term. Fix s and order $I_s = \{t_1 < \cdots < t_{|I_s|}\}$. Since both Q_t and Φ' are nondecreasing,

$$0 \le \frac{\Phi'(Q_{t_1-m-1})}{\Phi'(Q_{s-1})} \le \cdots \le \frac{\Phi'(Q_{t_{|I_s|}-m-1})}{\Phi'(Q_{s-1})} \le 1,$$

whenever $\Phi'(Q_{s-1}) > 0$.

A discrete summation-by-parts identity then gives

$$\begin{aligned} &\sum_{j=1}^{|I_s|} \frac{\Phi'(Q_{t_j-m-1})}{\Phi'(Q_{s-1})} \left[\nabla_{x_s} g_{t_j}^+(x_{t_j-m}^{t_j}) - \nabla_{x_s}\widetilde{g}_{t_j}^+(\widetilde{x}_{t_j-m}^{t_j})\right] \\ &= \frac{\Phi'(Q_{t_1-m-1})}{\Phi'(Q_{s-1})} \sum_{j=1}^{|I_s|} \left[\nabla_{x_s} g_{t_j}^+(x_{t_j-m}^{t_j}) - \nabla_{x_s}\widetilde{g}_{t_j}^+(\widetilde{x}_{t_j-m}^{t_j})\right] \\ &\quad + \sum_{j=2}^{|I_s|} \left(\frac{\Phi'(Q_{t_j-m-1}) - \Phi'(Q_{t_{j-1}-m-1})}{\Phi'(Q_{s-1})}\right) \sum_{r=j}^{|I_s|} \left[\nabla_{x_s} g_{t_r}^+(x_{t_r-m}^{t_r}) - \nabla_{x_s}\widetilde{g}_{t_r}^+(\widetilde{x}_{t_r-m}^{t_r})\right]. \end{aligned}$$

All coefficients on the right-hand side are nonnegative, and their sum is

$$\frac{\Phi'(Q_{t_{|I_s|}-m-1})}{\Phi'(Q_{s-1})} \le 1.$$

Therefore, by the triangle inequality,

$$\begin{aligned} &\left\| \sum_{t\in I_s} \frac{\Phi'(Q_{t-m-1})}{\Phi'(Q_{s-1})} \left[\nabla_{x_s} g_t^+(x_{t-m}^t) - \nabla_{x_s}\widetilde{g}_t^+(\widetilde{x}_{t-m}^t)\right] \right\| \\ &\qquad \le \max_{r\in I_s} \left\| \sum_{\substack{t\in I_s \\ t\ge r}} \left[\nabla_{x_s} g_t^+(x_{t-m}^t) - \nabla_{x_s}\widetilde{g}_t^+(\widetilde{x}_{t-m}^t)\right] \right\|. \end{aligned} \tag{39}$$

Multiplying (39) by $\Phi'(Q_{s-1})$ and using $\Phi'(Q_{s-1}) \le \Phi'(Q_T)$ yields

$$\begin{aligned} &\left\| \sum_{t\in I_s} \Phi'(Q_{t-m-1}) \left[\nabla_{x_s} g_t^+(x_{t-m}^t) - \nabla_{x_s}\widetilde{g}_t^+(\widetilde{x}_{t-m}^t)\right] \right\| \\ &\qquad \le \Phi'(Q_T) \max_{r\in I_s} \left\| \sum_{\substack{t\in I_s \\ t\ge r}} \left[\nabla_{x_s} g_t^+(x_{t-m}^t) - \nabla_{x_s}\widetilde{g}_t^+(\widetilde{x}_{t-m}^t)\right] \right\|. \end{aligned} \tag{40}$$

If $\Phi'(Q_{s-1}) = 0$, monotonicity implies $\Phi'(Q_{t-m-1}) = 0$ for every $t \in I_s$, so (40) holds trivially.

Combining (38) and (40), and then applying Minkowski's inequality across s, gives

$$\left(\sum_{s=1}^{T}\|h_s-\widetilde{h}_s\|^2\right)^{1/2} \le V\left(\sum_{s=1}^{T}\left\|\sum_{t\in I_s}\left[\nabla_{x_s}f_t(x_{t-m}^t)-\nabla_{x_s}\widetilde{f}_t(\widetilde{x}_{t-m}^t)\right]\right\|^2\right)^{1/2} + \Phi'(Q_T)\left(\sum_{s=1}^{T}\max_{r\in I_s}\left\|\sum_{\substack{t\in I_s\\ t\ge r}}\left[\nabla_{x_s}g_t^+(x_{t-m}^t)-\nabla_{x_s}\widetilde{g}_t^+(\widetilde{x}_{t-m}^t)\right]\right\|^2\right)^{1/2}.$$

By (14)–(15), this is exactly

$$\left(\sum_{s=1}^{T}\|h_s-\widetilde{h}_s\|^2\right)^{1/2} \le V\sqrt{E_T^{\mathrm{mem}}(f)}+\Phi'(Q_T)\sqrt{E_T^{\mathrm{mem}}(g^+)}.$$

Thus, in the unconstrained case, the delayed prediction-error term reduces exactly to $\sqrt{E_T^{\mathrm{mem}}(f)}$, so Theorem 1 transfers directly. In the constrained case, (37) has the same form as the prediction-error decomposition used in Section 4. Hence, after replacing $E_T(f)$ and $E_T(g^+)$ by $E_T^{\mathrm{mem}}(f)$ and $E_T^{\mathrm{mem}}(g^+)$, respectively, the proofs of Theorems 2 and 3 carry over through Corollary 1. □

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: The abstract and introduction state the main theoretical contributions: the dynamic delayed optimistic OCO result, the extension to constrained OCO, and the memory-to-delay transfer. The claims are supported by the theorems in the main paper and the proofs in the appendix, together with an extended comparison with related work.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: The main limitation discussed in the paper is the prediction requirement. In the general nonlinear setting, constructing gradient hints may require predicting future decisions or future decision blocks, especially in the memory setting. This is standard in optimistic OCO, and additional structure, such as linearity, can reduce this requirement. However, we focus on the general case where such structure may not be available. This and other limitations are discussed in Section 6.

   Guidelines:

   - The answer [N/A] 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?

   Answer: [Yes].

   Justification: The assumptions are stated in Section 2, and the main theoretical results are stated in the main paper, with complete proofs provided in the appendix. The main paper also specifies the parameters required to run each algorithm.

   Guidelines:

   - The answer [N/A] 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: [N/A].

   Justification: The paper is theoretical and does not include experimental results. The main claims are supported by mathematical analysis and complete proofs; moreover, the algorithmic templates are explicitly provided and can be readily implemented.

   Guidelines:

   - The answer [N/A] 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: [N/A].

   Justification: The paper does not use datasets and does not include experiments requiring code. The proposed algorithms are described explicitly in the paper.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [N/A].

   Justification: The paper does not include experiments, training procedures, or test settings.

   Guidelines:

   - The answer [N/A] 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: [N/A].

Justification: The paper does not include experimental results, so statistical significance and error bars are not applicable.

Guidelines:

- The answer [N/A] 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.
- 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: [N/A].

   Justification: The paper does not include experiments requiring computational resources.

   Guidelines:

   - The answer [N/A] 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: The work is theoretical and does not involve human subjects, private data, deployed systems, or high-risk models. The research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] 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: [Yes].

    Justification: The introduction motivates the work through applications in online decision-making, resource allocation, constrained control systems, and online non-stochastic control. Since the paper is theoretical and does not propose a deployed system, possible negative impacts are indirect and depend on how the framework is instantiated in applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper does not release datasets, models, or systems with high misuse risk.

    Guidelines:

    - The answer [N/A] 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: [N/A].

Justification: The paper does not use existing datasets, codebases, pretrained models, or other external assets. Prior work is credited through citations.

Guidelines:

- The answer [N/A] 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.
- 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: [N/A].

Justification: The paper does not introduce or release new datasets, code assets, or models.

Guidelines:

- The answer [N/A] 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: [N/A].

Justification: The paper does not involve crowdsourcing or human-subject research.

Guidelines:

- The answer [N/A] 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: [N/A].

Justification: The paper does not involve human-subject research, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] 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.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A].

Justification: The core methodology and theoretical results do not involve LLMs as an important, original, or non-standard component. Use only for editing and grammar.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.