Adaptive Composite Online Optimization
Predictions in Static and Dynamic Environments

Scroccaro, Pedro Zattoni; Kolarijani, Arman Sharifi; Esfahani, Peyman Mohajerin

DOI
10.1109/TAC.2023.3237486

Publication date
2023

Document Version
Final published version

Published in
IEEE Transactions on Automatic Control

Adaptive Composite Online Optimization: Predictions in Static and Dynamic Environments

Pedro Zattoni Scroccaro, Arman Sharifi Kolarijani, and Peyman Mohajerin Esfahani

***Abstract*—In the past few years, online convex optimization (OCO) has received notable attention in the control literature thanks to its flexible real-time nature and powerful performance guarantees. In this article, we propose new step-size rules and OCO algorithms that simultaneously exploit *gradient predictions*, *function predictions* and *dynamics*, features particularly pertinent to control applications. The proposed algorithms enjoy static and dynamic regret bounds in terms of the dynamics of the reference action sequence, gradient prediction error, and function prediction error, which are generalizations of known regularity measures from the literature. We present results for both convex and strongly convex costs. We validate the performance of the proposed algorithms in a trajectory tracking case study, as well as portfolio optimization using real-world datasets.**

***Index Terms*—Composite costs, dynamic environments, online convex optimization (OCO), predictions, real-time control.**

I. Introduction

THE standard framework of online convex optimization (OCO) can be described as a game between a Player and Nature, played over T rounds. Let $\mathcal{A}$ be the Player's action space. Suppose that $\mathcal{X} \subseteq \mathcal{A}$ is a convex set representing the set of possible actions of the Player. Moreover, let $\mathcal{F}$ denote a set of convex functions available to Nature. At each round t, the Player chooses an action $x_t \in \mathcal{X}$. After the Player commits with an action, Nature reveals a convex cost $f_t : \mathcal{X} \to \mathbb{R}$, where $f_t \in \mathcal{F}$. The Player suffers the loss $f_t(x_t)$. The goal of the Player is to perform as well as possible against the costs chosen by Nature. (See [1], [2], and [3] for in-depth studies of fundamental theories of OCO and its many applications.)

Manuscript received 29 April 2022; revised 22 August 2022; accepted 30 December 2022. Date of publication 16 January 2023; date of current version 26 April 2023. This work was supported by the ERC under Grant TRUST-949796. Recommended by Senior Editor Tetsuya Iwasaki and Guest Editors George J. Pappas, Anuradha M. Annaswamy, Manfred Morari, Claire J. Tomlin, Rene Vidal, and Melanie N. Zeilinger. *(Corresponding author: Pedro Zattoni Scroccaro.)*

The authors are with the Delft Center for Systems and Control, TU Delft, 2628 CD Delft, The Netherlands (e-mail: p.zattoniscroccaro@tudelft.nl; A.SharifiKolarijani@tudelft.nl; P.MohajerinEsfahani@tudelft.nl).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAC.2023.3237486.

Digital Object Identifier 10.1109/TAC.2023.3237486

A common metric to evaluate the performance of the Player is the so-called *static regret* defined as

$$\mathbf{Reg}_T^s := \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} f_t(x). \tag{1}$$

Intuitively, this metric quantifies how well the Player performs against the *best fixed* action computed in hindsight. Based on this notion of regret, OCO algorithms are designed such that the resulting action sequence $\{x_t\}_{t=1}^T$ guarantees a *sub-linear* regret w.r.t. T, i.e., $\lim_{T\to\infty}(\mathbf{Reg}_T^s/T) = 0$. In other words, such OCO strategies perform (on average) as well as the best fixed action in hindsight. A standard algorithm to choose x_t is called online mirror descent (OMD) algorithm

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \{\eta_t \langle \nabla f_t(x_t), x\rangle + \mathcal{B}_h(x, x_t)\} \tag{OMD}$$

where η_t denotes the step-size and $\mathcal{B}_h$ is the Bregman divergence functional [2]. By choosing η_t appropriately, Algorithm OMD guarantees $\mathbf{Reg}_T^s \leq O(\sqrt{T})$ [4] or $\mathbf{Reg}_T^s \leq O(\log(T))$ [5], based on the regularity of the cost set $\mathcal{F}$. Moreover, Abernethy et al. [6] showed that these regret rates are in fact optimal by the *minimax* formulation of OCO problems.

However, there are many OCO problems in which the Player and Nature *do not exactly follow* the rules of the sequential game mentioned above. In this article, we focus on the case of *OCO with predictions*. In these scenarios, we assume access to *predictions* about the costs of the problem being studied, and we use OCO algorithms combined with these predictions in order to achieve improved regret guarantees. For instance, if our OCO problem is related to estimating the evolution of dynamical parameters of a system, predictions could come from a dynamical model we have of the system (see Section IV). This approach is inspired by the classical control theory literature, in which dynamical and/or predictive models of the system being controlled are almost always assumed to exist. Moreover, there has been recent interest from both the online learning and controls communities in combining OCO techniques to control problems, e.g., [7], [8], [9], [10]. Also, most of the results presented in this work apply to problems with composite costs with nonsmooth components (e.g., $\|\cdot\|_1$). These results open up even more possibilities of connections with control applications, for instance, ℓ_1 optimization for sparse networked feedback control [11], [12]. Therefore, we hope that this work lays a theoretical foundation and also inspires new works in the intersection of OCO and control theory.

Next, we formally define important notions that will be used throughout the article.

A. Gradient Predictions

The minimax regret bounds for OCO algorithms are derived assuming a worst-case (i.e., fully adversarial) cost sequence $\{f_t\}_{t=1}^T$. The cost sequence is however *not* completely adversarial in many practical OCO problems [13]. In such problems, the Player can (partially) *predict* the *unseen* cost f_t at round t, before deciding its action x_t.[1] It is hence natural to expect that one can possibly exploit the predictability of an OCO problem to achieve tighter regret bounds.

A generic notion of the predictability of Nature's moves can be stated as follows [13]. At the outset of each round $t \in [T]$, the Player has access to the value of a function

$$M_t : \mathcal{X}^{t-1} \times \mathcal{F}^{t-1} \times \mathcal{I}^{t-1} \to \mathcal{P},$$

where $\mathcal{I}$ denotes some information space provided to the Player via an exogenous source and $\mathcal{P}$ is the space to which each predictable entity belongs. In particular, a certain class of OCO problems with predictability is the class of OCO problems with *gradient predictions*. Observe that here $\mathcal{P} \subseteq \mathcal{A}^*$, where $\mathcal{A}^*$ is the dual space of the action space $\mathcal{A}$. To exploit gradient predictions in OCO problems, Rakhlin and Sridharan [13] proposed the optimistic mirror descent (OptMD) algorithm

$$\begin{aligned} x_t &= \arg\min_{x\in\mathcal{X}} \{\eta_t \langle M_t, x\rangle + \mathcal{B}_h(x, y_{t-1})\} \\ y_t &= \arg\min_{y\in\mathcal{X}} \{\eta_t \langle \nabla f_t(x_t), y\rangle + \mathcal{B}_h(y, y_{t-1})\} \end{aligned} \tag{OptMD}$$

where $\{M_t\}_{t=1}^T$ is a generic gradient prediction sequence.[2] Rakhlin and Sridharan [14] further provided an adaptive step-size rule for Algorithm OptMD such that $\mathbf{Reg}_T^s \le O(1+\sqrt{D_T})$, where

$$D_T := \sum_{t=1}^T \|\nabla f_t(x_t) - M_t\|_*^2. \tag{2}$$

When the Player has access to $\nabla f_t(\cdot)$ *before* choosing x_t, we say that the Player has access to *perfect gradient predictions*. In this scenario, Ho-Nguyen and Kılınç-Karzan [15] showed that by setting $M_t := \nabla f_t(y_{t-1})$, $\eta_t \le 1/\beta$ and when $\mathcal{F}$ represents β-*smooth* functions, Algorithm OptMD guarantees $\mathbf{Reg}_T^s \le O(1)$.

B. Problem With D_T

In the following, we argue that regret bounds given in terms of D_T are not suitable for exploiting gradient predictions, mainly because x_t *is not available* at the beginning of round t (see Algorithm OptMD). In some works that prove regret bounds in terms of D_T (e.g., [14], [16]), it is argued that for "predictable sequences," external knowledge of the gradient sequence can be used to achieve tighter regret bounds. For example, Jadbabaie et al. [16] stated that: "...one can get a tighter bound for regret once the learner advances a sequence of conjectures $\{M_t\}_{t=1}^T$ well-aligned with the gradients." However, consider the following scenario: at the beginning of round t, the Player has access to a prediction of $\nabla f_t(\cdot)$, namely $\nabla \hat{f}_t(\cdot)$. Now, based on those regret bounds given in terms of D_T, how one would choose M_t when using Algorithm OptMD? Naturally, we want to choose M_t so that D_T is as small as possible (recall that $D_T = \sum_{t=1}^T \|\nabla f_t(x_t) - M_t\|_*^2$). However, since x_t is not available at the beginning of round t, we cannot set $M_t = \nabla \hat{f}_t(x_t)$. Thus, from these regret bounds, it is not clear how one should choose M_t in order to exploit this type of gradient prediction. Moreover, Ho-Nguyen and Kılınç-Karzan [15] showed that when perfect gradient predictions are available (that is, $\nabla \hat{f}_t(\cdot) = \nabla f_t(\cdot)$), constant static regret is achievable. Still, this constant regret result is not recovered by the regret bound $\mathbf{Reg}_T^s \le O(1+\sqrt{D_T})$ given in [14], even when perfect gradient predictions are available. In fact, since smoothness of the cost is not assumed in [14], if it was possible to choose M_t such that $D_T = 0$ (i.e., such that $\mathbf{Reg}_T^s \le O(1)$), this would contradict the lower bound for first-order optimization methods [17], [18, Remark 1]. Therefore, we conclude that in order to effectively exploit gradient predictions, a different approach must be used.

C. Dynamic Environments and Regularity Measures

In the regret notion (1), the Player's cumulative loss competes against the loss of the best fixed action in hindsight. There are, on the other hand, many OCO problems where the best fixed action is not accessible or does not exist [19]. Thus, in those cases, the use of the regret (1) is not convenient anymore. The term OCO problems in *dynamic environments* is used in the literature for such problems [20].

To generalize the standard regret notion in order to tackle these scenarios, Zinkevich [21] proposed to compare the Player's performance against a general dynamical *reference sequence* $\{u_t\}_{t=1}^T \in \mathcal{X}^T$. The resulting metric is called the *dynamic regret*, defined as

$$\mathbf{Reg}_T^d := \sum_{t=1}^T f_t(x_t) - \sum_{t=1}^T f_t(u_t). \tag{3}$$

Unfortunately, it is impossible to achieve a sublinear dynamic regret for an arbitrarily chosen $\{u_t\}_{t=1}^T$ [22]. Thus, in order to achieve *meaningful* dynamic regret bounds, it is common to place extra *regularity assumptions* on the costs and/or the reference sequence. For example, Hall and Willett [23] considered the bounded variability of the reference sequence in terms of $C_T := \sum_{t=1}^T \|u_{t+1} - u_t\|$. For convex costs, the authors showed that Algorithm OMD guarantees $\mathbf{Reg}_T^d \le O(\sqrt{T}(1+C_T))$. The authors further considered that the Player has access to dynamical models $\Phi_t : \mathcal{X} \to \mathcal{X}$ of the reference sequence, that is, models that approximate the true dynamical models $\Phi_t^\star$, i.e., $u_{t+1} = \Phi_t^\star(u_t)$. They employ $\Phi_t(x_t)$ instead of x_t in Algorithm OMD and prove $\mathbf{Reg}_T^d \le O(\sqrt{T}(1+C_T'))$, where

$$C_T' := \sum_{t=1}^T \|u_{t+1} - \Phi_t(u_t)\|. \tag{4}$$

When Φ_t approximates the true dynamics well enough, we may have $C_T' \le C_T$, which in turn implies tighter dynamic regret bounds. Subsequently, Jadbabaie et al. [16] studied dynamical environments to account for the cases with gradient predictions. The authors showed that Algorithm OptMD guarantees $\mathbf{Reg}_T^d \le O(\sqrt{1+D_T}(1+C_T))$ in such cases. Finally, using an expert-based algorithm called *Ader*, Zhang et al. [24] showed that it guarantees the optimal bound $\mathbf{Reg}_T^d \le O(\sqrt{T(1+C_T')})$.

[1] This assumption deviates from the standard OCO protocol, where Nature reveals f_t only *after* the Player chooses x_t.

[2] Notice that Algorithm OptMD reduces to Algorithm OMD when $M_t = 0$.

Another important regularity measure popular in the literature is the *temporal variability* of the cost sequence

$$V_T := \sum_{t=2}^{T} \max_{x \in \mathcal{X}} |f_t(x) - f_{t-1}(x)|. \tag{5}$$

In the setting of stochastic optimization with noisy gradients, Besbes et al. [19] showed that a restarted gradient descent algorithm incurs dynamic regret bounded by $O(T^{2/3}(1+v)^{1/3})$, where v is an upper bound of V_T known in advance, and Jadbabaie et al. [16] provided an algorithm, which guarantees a dynamic regret bound of $\tilde{O}(\sqrt{D_T+1} + \min\{\sqrt{(D_T+1)C_T}, (D_T+1)^{1/3}T^{1/3}V_T^{1/3}\})$,[3] for the specific case when the regret is defined w.r.t. the reference sequence $u_t = \arg\min_{x \in \mathcal{X}} f_t(x)$, also known as *restricted dynamic regret* [25].

D. Composite Cost, Implicit Updates, and Function Predictions

A cost function f_t is called composite if it can be decomposed as $f_t(\cdot) = s_t(\cdot) + r_t(\cdot)$. For example, Duchi et al. [26] considered the case when $r_t(\cdot) = r(\cdot)$ for all t, and proposes the composite objective mirror descent (COMID) algorithm

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \{\eta_t \langle \nabla s_t(x_t), x\rangle + \eta_t r(x) + \mathcal{B}_h(x, x_t)\} \tag{COMID}$$

where differently from OMD, the fixed part $r(\cdot)$ is not linearized. This can be advantageous when, for example, $r(\cdot) = \|\cdot\|_1$. In this case, using COMID would lead to sparse updates, whereas OMD would not [26]. In the offline optimization literature (i.e., $f_t(\cdot) = f(\cdot)$ for all t), algorithms that partially linearize the cost function are called *proximal gradient methods* [27], [28]. These algorithms are usually used when s is smooth, but r is not. Then, by linearizing only the smooth component of f, a proximal gradient method can lead to convergence rates that match the one of OMD for smooth costs (e.g., $O(1/T)$ rate instead of $O(1/\sqrt{T})$). Intuitively, when smoothness is necessary to prove a convergence rate for some first-order algorithms, one can usually deal with nonsmooth components by not linearizing them in the proximal updates.

Somewhat related to proximal gradient methods are the so-called implicit updates, also known as implicit online mirror descent (IOMD) [29], [30], [31]

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \{\eta_t f_t(x) + \mathcal{B}_h(x, x_t)\}. \tag{IOMD}$$

Kulis and Bartlett [30] proved regret bounds for IOMD that match the ones from OMD. McMahan [32] and Song et al. [33] quantified the advantage of implicit updates through nonnegative, data-dependent quantities. Recently, Campolongo and Orabona [31] showed that an adaptive version of IOMD guarantees $O(\min\{V_T, \sqrt{T}\})$. Moreover, in dynamic environments, Campolongo and Orabona [25] showed that a version of IOMD guarantees $O(\min\{V_T, \sqrt{T(1+\tau)}\})$, where τ is a known upper bound of C_T. When τ is not known, a similar bound $\tilde{O}(\min\{V_T, \sqrt{T(1+C_T)}\})$ can be achieved by combining implicit updates with experts and strongly adaptive algorithms [25].

When a linearized version of the cost f_t is used in our OCO strategy, e.g., OMD algorithm, it is natural to expect that we only need *gradient predictions* to exploit information of unseen costs, as it is done in the OptMD algorithm. However, when using strategies that partially linearize the cost f_t (or do not linearize it at all), one should not hope that gradient predictions of the cost can be effectively used. Therefore, in order to exploit predictive information about cost functions, we will require *gradient predictions* of its linearized component and *function predictions* of its nonlinearized component. For example, for the composite cost $f_t(x) = s_t(x) + r_t(x)$, if we decide to linearize $s_t(x)$ and not linearize $r_t(x)$, we will require gradient predictions of s_t and function predictions of r_t, denoted as $\hat{r}_t$.

E. Problem Description and Related Works

In this article, we consider OCO problems with composite costs of the form $f_t(\cdot) = s_t(\cdot) + r_t(\cdot)$, in both static and dynamic environments. Recall that Ho-Nguyen and Kılınç-Karzan [15] observed that perfect gradient predictability in the form of $M_t = \nabla f_t(y_{t-1})$ implies that Algorithm OptMD guarantees constant static regret. Motivated by this observation and the discussion presented in Section I-B, we extend this idea to the case of an "imperfect" gradient predictability. To do so, we introduce the *gradient prediction error measure*

$$D_t' := \sum_{\tau=1}^{t} \|\nabla s_\tau(y_{\tau-1}) - \nabla \hat{s}_\tau(y_{\tau-1})\|_*^2 \tag{6}$$

where $\{y_{\tau-1}\}_{\tau=1}^t$ are points generated by an online algorithm. Notice that we changed the notation from M_t to $\nabla \hat{s}_\tau(y_{\tau-1})$. We do it so that the connection between the gradient of s_t and the gradient predictions is clearer. Also, notice that this measure refers to gradient predictions only for the s_t component of f_t. Thus, we also introduce the *function prediction error*

$$V_t' := \sum_{\tau=1}^{t} |r_\tau(x_\tau) - \hat{r}_\tau(x_\tau) + \hat{r}_\tau(y_\tau) - r_\tau(y_\tau)| \tag{7}$$

where $\{x_\tau\}_{\tau=1}^t$ and $\{y_\tau\}_{\tau=1}^t$ are points generated by an online algorithm. When $s_t = 0$, V_T' can be interpreted as a generalization of V_T for the case when function predictions are available. Namely, when function predictions are not available, by setting $\hat{r}_t = r_{t-1}$, we get $V_T' \le 2V_T$. Moreover, in dynamic environments, we further suppose that the Player has access to a (possibly approximate) dynamical model Φ_t of the reference sequence $\{u_t\}_{t=1}^T$. This is a useful assumption, which has been used in practical applications of OCO algorithms [10], [34]. We are now set to state the problem considered in this article.

Problem: Design and analyze OCO algorithms such that the corresponding regret bounds exploit

1) (possibly imperfect) gradient and/or function predictions of the components of the cost sequence $\{f_t\}_{t=1}^T$
2) (possibly approximate) dynamical models Φ_t of the reference sequence $\{u_t\}_{t=1}^T$.

Other than the works already mentioned in Section I, several studies in the literature propose algorithms that take advantage of the predictability of the cost sequence. Several works exploit predictions in OCO problems with switching costs. In this scenario, at round t, the Player suffers the loss $f_t(x_t, x_{t-1}) = c_t(x_t) + \gamma\|x_t - x_{t-1}\|$, where c_t is a convex function and $\gamma\|x_t - x_{t-1}\|$ is the switching cost. In order to exploit predictions in these problems, it is usually necessary to have a *window of future cost predictions* [35], [36], [37], [38], [39]. Another application where predictions have been used is

[3] The $\tilde{O}$ notation hides poly-logarithmic terms.

the so-called *online control* problem. For this class of problems, due to the dynamics of the system, the cost f_t may depend on the whole history of previous actions, and a window of predictions is again necessary [40], [41], [42]. Thus, since in this work, the cost at time t only depends on x_t and we only use predictions about the very next cost, results on OCO with switching costs and online control are not directly comparable to this paper's results. Dekel et al. [43] studied online linear optimization. The authors supposed that at the outset of each round, the Player has access to a vector (or hint) that is correlated with the cost to be incurred to the Player. If *all* hints are sufficiently good and the action set possesses certain geometrical properties, the authors showed $\mathbf{Reg}_T^s \leq O(\log(T))$. Recently, Bhaskara et al. [44] extended this result to the case when not all hints are correlated with the true cost vector. In dynamic environments, Lesage–Landry et al. [45] showed that tighter dynamic regret bounds can be achieved by only using predictions that meet certain conditions. Ravier et al. [46] employed gradient predictions in order to obtain possibly tighter dynamic regret bounds. However, the proposed approach yields regret bounds that lack worst-case guarantees. Chang and Shahrampour [47] proposed an online optimistic Newton method that exploits gradient and hessian predictions and prove dynamic regret bounds for this algorithm.

F. Contributions and Organization

A summary of the main results is now given as follows:

1) *Static regret for convex costs*: In static environments, we propose a novel algorithm that uses step-sizes that adapt to the quality of gradient and function predictions, guaranteeing $\mathbf{Reg}_T^s \leq O(1+\sqrt{D_T'}+\min\{V_T,'\sqrt{T}\})$ for convex costs (see Theorem 2.5). This result generalizes the best-case $\mathbf{Reg}_T^s \leq O(1)$ [15] and worst-case $\mathbf{Reg}_T^s \leq O(\sqrt{T})$ [21] regret rates.
2) *Static regret for strongly convex costs*: When the costs are strongly convex and we have access to the r_t components, we propose an adaptive step-size η_t which improves the regret to $\mathbf{Reg}_T^s \leq O(1+\log(1+D_T'))$ (see Theorem 2.9). This result generalizes the best-case $\mathbf{Reg}_T^s \leq O(1)$ [15] and worst-case $\mathbf{Reg}_T^s \leq O(\log(T)$ [48] regret rates.
3) *Dynamic regret for convex costs*: For dynamic environments, we introduce a new variant of Algorithm OptMD that simultaneously exploits gradient predictions, function predictions, and the dynamics of the reference sequence. We show that it guarantees regret $\mathbf{Reg}_T^d \leq O\left((1+C_T')(1+\sqrt{D_T'}+\min\{V_T,'\sqrt{(1+C_T')T}\})\right)$ for convex costs (see Theorem 2.15).
4) *Dynamic regret for implicit updates*: Using fully implicit updates (i.e., when $s_t = 0$), we show that the proposed algorithm guarantees the dynamic regret bound $\mathbf{Reg}_T^d \leq O(\min\{V_T,'\sqrt{(1+\tau)T}\})$, where τ is a known upper bound to C_T' (see Theorem 2.17). This result generalizes the dynamic regret bounds of [25], for the case of function predictions.
5) *Dynamic regret for fully adaptive step-size*: Finally, when we have access to the r_t component of the costs, we propose a step-size η_t, which adapts to gradient predictions and C_t' on the fly. The resulting algorithm guarantees $\mathbf{Reg}_T^d \leq O(\sqrt{(\theta_T+D_T')(1+C_T')})$, where θ_T is a parameter used to control the size of the step-size (see Theorem 2.18).

For the ease of the readers, in an arXiv version of this article, we also present tables positioning the above contributions within the existing OCO literature reviewed earlier, with a particular focus on the predictions and composite features in the context of static regret bounds [49, Appendix A]. The rest of this article is organized as follows. The main results are provided in Section II. To improve the flow of this article, we moved the proofs of our main results to Section III. Numerical experiments are presented in Section IV.

II. Main Results

A. Mathematical Preliminaries

Let the action set $\mathcal{X} \subset \mathbb{R}^n$. We denote by $\|\cdot\|_*$ the dual norm of $\|\cdot\|$. Also, we define $[T] := \{1, 2, \ldots, T\}$.

Definition 2.1 (Bregman Divergence): Let $h : \mathcal{X} \to \mathbb{R}$ be a differentiable convex function. The Bregman divergence of $x, y \in \mathcal{X}$, w.r.t. the function h is $\mathcal{B}_h(x, y) := h(x) - h(y) - \langle \nabla h(y), x - y\rangle$.

Definition 2.2 (α-Strong convexity): A function $f : \mathcal{X} \to \mathbb{R}$ is *α-strongly convex* w.r.t. a norm $\|\cdot\|$ if $f(x) - f(y) \leq \langle \nabla f(x), x - y\rangle - \frac{\alpha}{2}\|x - y\|^2$, for all $x, y \in \mathcal{X}$.

Definition 2.3 (β-Smoothness): A function $f : \mathcal{X} \to \mathbb{R}$ is *β-smooth* w.r.t. a norm $\|\cdot\|$ if it is differentiable and $\|\nabla f(x) - \nabla f(y)\|_* \leq \beta\|x - y\|$, for all $x, y \in \mathcal{X}$.

Next, we collect several assumptions, which we will employ in the results to follow.

Assumption 2.4 (Regularity Assumptions): Let $\mathcal{A}$ be a Banach space equipped with the norm $\|\cdot\|$. Suppose that

1) The set $\mathcal{X}$ is a convex subset of $\mathcal{A}$.
2) The map $h : \mathcal{A} \to \mathbb{R}$ is differentiable and 1-strongly convex on $\mathcal{X}$.
3) Each member of the cost sequence $\{s_t\}_{t=1}^T$ is convex and β-smooth. Each member of the cost sequence $\{r_t\}_{t=1}^T$ is convex.
4) $\mathcal{B}_h(x, y) \leq R^2$ for all $x, y \in \mathcal{X}$, where $R > 0$.
5) For all $t \in [T]$, the gradient prediction $\nabla\hat{s}_t$ satisfies $\|\nabla s_t(x) - \nabla\hat{s}_t(x)\|_* \leq \sigma < \infty$ for any $x \in \mathcal{X}$.
6) For all $t \in [T]$, the function prediction $\hat{r}_t$ is convex and $|r_t(x) - \hat{r}_t(x)| < \infty$ for any $x \in \mathcal{X}$.

In particular, the last two points of Assumption 2.4 simply state that the gradient and function predictions cannot be arbitrarily bad, which would naturally prevent the use of such predictive information. Next, we provide static and dynamic regret bounds that exploit gradient/function predictability and/or dynamical models of the reference sequence.

B. Static Environments

Our first result concerns convex costs in static environments. In order to exploit predictive information of composite costs of the form

$$f_t(\cdot) = s_t(\cdot) + r_t(\cdot)$$

we propose the optimistic composite mirror descent (OptCMD) algorithm

$$x_t = \arg\min_{x\in\mathcal{X}} \{\eta_t\langle\nabla\hat{s}_t(y_{t-1}), x\rangle + \eta_t\hat{r}_t(x) + \mathcal{B}_h(x, y_{t-1})\}$$

$$y_t = \arg\min_{y\in\mathcal{X}} \{\eta_t\langle\nabla s_t(x_t), y\rangle + \eta_t r_t(y) + \mathcal{B}_h(y, y_{t-1})\} \quad \text{(OptCMD)}$$

where $\nabla \hat{s}_t$ is a gradient prediction of ∇s_t and $\hat{r}_t$ is the function prediction of r_t. Notice that unlike algorithms COMID and IOMD, Algorithm OptCMD makes use of an auxiliary variable y_t. However, x_t is still the decision variable of all OCO algorithms discussed in this article. Algorithm OptCMD can be interpreted as an extension of OptMD for composite costs with smooth and nonsmooth components. As hinted in Section I-B, one needs smooth functions to properly exploit gradient predictions of costs. Therefore, the intuition behind Algorithm OptCMD is similar to the one from proximal gradient algorithms: we handle nonsmooth components by not linearizing them in the proximal updates, while linearizing the smooth ones. This leads to using *function predictions* of the nonsmooth component r_t, instead of gradient predictions.

Theorem 2.5 (Static regret: convex costs): Suppose that Assumption 2.4 holds. Using the adaptive step-size

$$\eta_1 = \frac{1}{2\beta}, \quad \eta_t = \left(4\beta^2 + \left(V'_{t-1}\right)^2 + D'_{t-1}\right)^{-\frac{1}{2}}$$

for all $t > 1$, Algorithm OptCMD guarantees

$$\mathbf{Reg}_T^s \leq O\left(1 + \sqrt{D'_T} + \min\left\{V_T, \sqrt{T}\right\}\right). \tag{8}$$

Remark 2.6 (Intuition on adaptive step-size η_t): In Theorem 2.5, for simplicity, consider the case when $r_t(x) = 0\ \forall x \in \mathcal{X}$, i.e., $V'_T = 0$. For this scenario, we want to guarantee $O(\sqrt{T})$ regret in the worst-case, and in order to do so, it is known we need $\eta_t = O(1/\sqrt{t})$. On the other hand, with perfect gradient predictions (i.e., $D'_T = 0$), we want to guarantee $O(1)$ regret, and in order to do so, we need $\eta_t \leq O(1/\beta)$ [15]. Now, if we want to guarantee a regret bound that generalizes these two extreme cases, it is natural that our step size should also generalize $\eta_t = O(1/\sqrt{t})$ and $\eta_t \leq O(1/\beta)$, which is precisely the behavior of the η_t we designed. Similar intuitions can be derived from the other scenarios and results presented in this article.

The result of Theorem 2.5 is also related to [50, Th. 3], where the authors prove regret bounds for the so-called *Composite Adaptive Optimistic Follow-the-Regularized-Leader* (CAO-FTRL) algorithm. The key differences between these results are: the CAO-FTRL algorithm uses FTRL update steps, which can be computationally more expensive than the mirror descent steps of Algorithm OptCMD; the CAO-FTRL algorithm assumes knowledge of r_t at the beginning of round t, thus, is less general than Algorithm OptCMD; and finally, the regret bound of [50, Th. 3] is presented in terms of D_T. Here, we re-emphasize that our regret bounds depend on D'_T instead of D_T, which solves the issues raised in Section I-B. Key points to achieve this result are our proposed adaptive step-size (see remark above), and the extra assumption that the costs are β-smooth. In particular, since smooth costs have Lipschitz continuous gradients, we are able to control the difference between, possibly approximate, gradient predictions.

Next, we discuss how the bound of Theorem 2.5 generalizes several regret bounds from the literature.

Remark 2.7 (Generality of regret bound): First, let us consider the case when r_t is known at the beginning of round t, i.e., $V'_T = 0$. In this case, f_t is β-smooth convex, and Algorithm OptCMD reduces to Algorithm OptMD. In this scenario, when perfect predictions are available, setting $\nabla \hat{s}_t = \nabla s_t$ implies that $D'_T = 0$, and the regret inequality (8) reduces to $\mathbf{Reg}_T^s \leq O(1)$, recovering the result of Ho-Nguyen and Kılınç-Karzan [15]. On the other hand, in view of Assumption 2.4, the regret inequality (8) also recovers the minimax static regret $\mathbf{Reg}_T^s \leq O(\sqrt{T})$ in the worst case, that is, even if the gradient predictions are completely uncorrelated with the true gradients and we end up with $D'_T = O(T)$. Next, consider the case when ∇s_t is known at the beginning of round t, i.e., $D'_T = 0$. In this case, f_t is a general convex function and (8) reduces to $O(1 + \min\{V_T, \sqrt{T}\})$. Again, when perfect predictions are available we recover the optimal constant regret bound, by simply setting $\hat{r}_t = r_t$. This bound generalizes the $O(1 + \min\{V_T, \sqrt{T}\})$ bound of Campolongo and Orabona [31], which is known to be optimal [31, Th. 6.3]. In this case, if our function predictions are good, V'_T may be small and we guarantee small regret. On the other hand, we still guarantee the standard $O(\sqrt{T})$ regret in the worst-case.

Next, we state a static regret result for strongly convex costs. This stronger assumption on the costs allows us to achieve tighter bounds. For this result, we need the following assumption.

Assumption 2.8 (Extra regularity assumptions): Suppose that the action space $\mathcal{A}$ is an Euclidean space equipped with the two-norm $\|\cdot\|_2$ and $h(x) = \frac{1}{2}\|x\|_2^2$. Moreover, suppose we have access to perfect function prediction of r_t. That is, we are able to set $\hat{r}_t = r_t$ for all t.

Notice that under Assumption 2.8, the Bregman divergence $\mathcal{B}_h(x, y) = \frac{1}{2}\|x - y\|_2^2$. Concerning the perfect prediction of r_t, this is the case, for example, when this term corresponds to a fixed known regularizer, e.g., $r_t(x) = \|x\|$, or naturally when $r_t(x) = 0$ for all t.

Theorem 2.9 (Static regret: strongly convex costs): Suppose that assumptions 2.4 and 2.8 hold and that the costs $\{s_t\}_{t=1}^T$ are α-strongly convex. Using the adaptive step-size

$$\eta_1 = \frac{1}{2\beta}, \quad \eta_t = \left(2\beta + \frac{\alpha}{2\sigma^2} D'_{t-1}\right)^{-1}$$

for all $t > 1$, Algorithm OptCMD with $\hat{r}_t = r_t$ guarantees

$$\mathbf{Reg}_T^s \leq O\left(1 + \log(1 + D'_T)\right). \tag{9}$$

Remark 2.10 (Generality of bound for strongly convex costs): Employing a similar line of argument as in Remark 2.7, we state two observations. With perfect gradient predictions, inequality (9) becomes $\mathbf{Reg}_T^s \leq O(1)$, again recovering the result of Ho-Nguyen and Kılınç-Karzan [15]. Moreover, the optimal regret bound $\mathbf{Reg}_T^s \leq O(\log(T))$ is also recovered in the worst-case.

C. Dynamic Environments

As previously mentioned, when working in dynamic environments, we would like to exploit gradient predictions, function predictions, and *knowledge of reference sequence dynamics*. Thus, in this scenario, we propose the *Optimistic Dynamic Composite Mirror Descent* (OptDCMD) algorithm

$$\begin{aligned} x_t &= \arg\min_{x \in \mathcal{X}} \left\{\eta_t \langle \nabla \hat{s}_t(y_{t-1}), x\rangle + \eta_t \hat{r}_t(x) + \mathcal{B}_h(x, y_{t-1})\right\} \\ \tilde{y}_t &= \arg\min_{y \in \mathcal{X}} \left\{\eta_t \langle \nabla s_t(x_t), y\rangle + \eta_t r_t(y) + \mathcal{B}_h(y, y_{t-1})\right\} \\ y_t &= \Phi_t(\tilde{y}_t). \end{aligned} \tag{OptDCMD}$$

This algorithm can be viewed as a combination of Algorithm OptCMD and the DMD algorithm of Hall and Willett [23]. To the best of our knowledge, no result in the literature has presented

a regret analysis of an algorithm that combines gradient predictions, function predictions, and knowledge about the dynamics of the reference sequence. In what follows, we assume that the Player has access to dynamical models Φ_t of $\{u_t\}_{t=1}^T$. Let us further make the following assumptions.

Assumption 2.11 (Lipschitz-likeness of $\mathcal{B}_h$): For all $x, y, z \in \mathcal{X}$, there exist a scalar $\gamma > 0$ such that the Bregman divergence satisfies the Lipschitz-like condition $\mathcal{B}_h(x, z) - \mathcal{B}_h(y, z) \leq \gamma\|x - y\|$.

Remark 2.12 (Mildness of Assumption 2.11): It follows that Assumption 2.11 holds when the mapping h is Lipschitz on $\mathcal{X}$ [16], which is a mild assumption once $\mathcal{X}$ is usually a compact set. Some examples are $h(x) = \frac{1}{2}\|x\|_2^2$ (i.e., the euclidean case), or the "KL divergence case" [18].

Assumption 2.13 (Non-expansiveness of Φ_t): For all $x, y \in \mathcal{X}$ and $\mathcal{B}_h$, the mapping Φ_t is nonexpansive, that is, $\mathcal{B}_h(\Phi_t(x), \Phi_t(y)) - \mathcal{B}_h(x, y) \leq 0$.

Remark 2.14 (Necessity of Assumption 2.13): Observe that Assumption 2.13 is a restriction on the class of dynamical models Φ_t. The reason behind this assumption is to control the impact of a possibly unreliable prediction (made by the use of Φ_t), as the online game progresses [20], [34].

In the results that follow, by abuse of notation, we use $V_t' := \sum_{\tau=1}^t |r_\tau(x_\tau) - \hat{r}_\tau(x_\tau) + \hat{r}_\tau(\tilde{y}_\tau) - r_\tau(\tilde{y}_\tau)|$.

Theorem 2.15 (Dynamic regret: convex costs): Suppose that assumptions 2.4, 2.11, and 2.13 hold. Define the adaptive step-size

$$\eta_1 = \frac{1}{2\beta}, \quad \eta_t = \left(4\beta^2 + \left(V_{t-1}'\right)^2 + D_{t-1}'\right)^{-\frac{1}{2}}$$

for all $t > 1$. Then, Algorithm (OptDCMD) guarantees

$$\mathbf{Reg}_T^d \leq O\left((1 + C_T')\left(1 + \sqrt{D_T'} + \min\left\{V_T,' \sqrt{(1 + C_T')T}\right\}\right)\right). \tag{10}$$

Remark 2.16 (Comparison with literature): Let us consider the case when r_t is known at the beginning of round t, i.e., $V_T' = 0$. Observe that when Φ_t approximates the true dynamics of the comparator sequence $\{u_t\}_{t=1}^T$, we may have $C_T' \leq C_T$. Moreover, we also recover $C_T' = C_T$ if we choose Φ_t as the identity map. Therefore, compared to the bound $\mathbf{Reg}_T^d \leq O((C_T + 1)\sqrt{D_T + 1})$ of Jadbabaie et al. [16], our result improves it in the sense that it is given in terms of C_T' and D_T', instead of C_T and D_T (recall the discussion of Section I-B). Moreover, recall that we have $\|\nabla s_t(y_{t-1}) - \nabla \hat{s}_t(y_{t-1})\|_* \leq \sigma$ by Assumption 2.4. Hence, it follows that $O(\sqrt{1 + D_T'}(1 + C_T')) = O(\sqrt{T}(1 + C_T'))$ in the worst-case, and we recover the bound of Hall and Willett [23]. However, Zhang et al. [24] proposed an algorithm called *Ader*, which achieves the *optimal* bound $\mathbf{Reg}_T^d \leq O(\sqrt{T(1 + C_T')})$. Thus, in the worst-case, our regret bound does not recover the optimal one. Comparing (10) with the $O(\min\{V_T, \sqrt{T(1 + \tau)}\})$ dynamic regret bound of Campolongo and Orabona [25], where τ is a known upper bound of C_T, we see that (10) has worst dependence of C_T'. This is mainly due to the fact that, in order to exploit gradient prediction, we need to have $\eta_t \leq 1/(2\beta)$. Since in [25] a fully implicit algorithm is used (see Algorithm IOMD), the step size can depend linearly on τ, in other words, it can be as large as necessary.

In the next theorem, we show that if the component $s_t = 0$, that is, $f_t = r_t$, we achieve a bound that generalizes [25, Th. 5.1] using function predictions, i.e., using V_T' instead of V_T. Notice that in this case, the updates of Algorithm OptDCMD are fully implicit updates, just like in Algorithm IOMD.

Theorem 2.17 (Dynamic regret: implicit updates): Suppose that assumptions 2.4, 2.11, and 2.13 hold. Furthermore, let $s_t = 0$ for all t, and τ be an upper bound of C_T'. Define the adaptive step-size

$$\eta_t = \frac{\tau}{V_{t-1}'}.$$

for all $t > 1$. Then, Algorithm (OptDCMD) guarantees

$$\mathbf{Reg}_T^d \leq O\left(\min\left\{V_T,' \sqrt{(1 + \tau)T}\right\}\right). \tag{11}$$

As mentioned in Remark 2.16, the Ader algorithm of Zhang et al. [24] guarantees the optimal worst-case dynamic regret bound of $\mathbf{Reg}_T^d \leq O(\sqrt{T(1 + C_T')})$, without prior knowledge of C_T' or an upper bound on it. In order to achieve this bound, an expert-tracking algorithm based on online gradient descent (OGD) updates is used. In our final result, we show that by using a step-size that adapts to C_t', a similar regret bound can be achieved while also exploiting gradient predictions.

Theorem 2.18 (Dynamic regret: fully adaptive step-size): Suppose that assumptions 2.4, 2.11, and 2.13 hold. Furthermore, assume have access to r_t, thus, we can choose $\hat{r}_t = r_t$. Set the adaptive step-size to $\eta_1 = \eta_2 = 1/(2\beta)$ and

$$\eta_t = \sqrt{\frac{C_{t-2}' + 1}{D_{t-1}' + \theta_t}}$$

for $t > 2$, where θ_t is chosen such that $\eta_t \leq \eta_{t-1} \leq \frac{1}{2\beta}$ and $\theta_t \geq \theta_{t-1}$ for all t. In this scenario, Algorithm OptDCMD guarantees

$$\mathbf{Reg}_T^d \leq O\left(\sqrt{(\theta_T + D_T')(1 + C_T')}\right). \tag{12}$$

Remark 2.19 (Comments on θ_t): From the definition of the step-size used in Theorem 2.18, we notice that the more the reference sequence $\{u_t\}_{t=1}^T$ varies (i.e., the bigger C_t' is), the larger η_t should be. Intuitively, we need larger step-sizes to "track" a reference sequence that changes a lot. On the other hand, in order to exploit gradient prediction, we also need $\eta_t \leq 1/(2\beta)$. Thus, θ_t can be interpreted as a trade-off parameter, which must be big enough so that $\eta_t \leq 1/(2\beta)$, but also not too big so that the algorithm is not able to "track" $\{u_t\}_{t=1}^T$. Also notice that, in the case of perfect gradient predictions (i.e. $D_T' = 0$), the regret bound of Theorem 2.18 becomes $\mathbf{Reg}_T^d \leq O(1 + C_T')$, since in this scenario we need $\theta_t = O(1 + C_t')$ in order to guarantee that $\eta_t \leq 1/(2\beta)$. This dynamic regret bound is similar to the one presented in [22], where the authors do not use any kind of gradient predictions, but assume *strongly convex* costs and a specific reference sequence defined as $u_t = \arg\min_{x \in \mathcal{X}} f_t(x)$.

In Theorem 2.18, notice that feedback about $\|u_t - \Phi_{t-1}(u_{t-1})\|$ after round t is necessary to implement the proposed step-size η_t. Although this information may not be available in the most general case of arbitrary costs f_t and reference sequence u_t, it is reasonable to assume this type of feedback in many applications. For example, in the case, where the reference sequence is a fixed point (and the dynamic regret reduces to static regret), the feedback assumption trivially holds since in

this case $\|u_t - u_{t-1}\| = 0$. Another example is the case of quadratic costs (see experimental results in [22] and [34]), which is ubiquitous in control applications. In this case, the gradient feedback ∇f_t constrains the information about u_t, thus, our step-sizes can be implemented. Finally, another common case is when $u_t = \arg\min_{x\in\mathcal{X}} f_t(x)$. When the cost f_t is revealed after round t, its optimizer can be computed, and again our step-sizes η_t can be implemented, although it may be computationally expensive to do so.

Differently from the approach proposed in Theorem 2.18, algorithms based on the doubling-trick or experts have been proposed as a way to adapt to C_T' without knowing it in advance [16], [24], [25]. We leave it as an open question whether or not these tools can be used to prove tighter dynamic regret bounds when using gradient and/or function predictions. Moreover, our regret bounds can serve as the basis for the design and analyses of algorithms that learn gradient/function predictors and minimize regret simultaneously. For instance, in order to learn good predictors, it may be necessary to explore the action space by playing actions perturbed by some noise. This strategy may lead to regret bounds that depend on the prediction error (i.e, D_T' and/or V_T') and terms that depend on the perturbation noise. Studying the trade-off between exploration (playing perturbed action to learn good predictors and minimize D_T' and/or V_T') and exploitation (playing actions with low noise) is an interesting future work direction.

III. Technical Proofs

A. Auxiliary Lemmas

The following lemma is a straightforward generalization of the standard mirror descent inequality and is stated without proof.

Lemma 3.1: Suppose that $\mathcal{X}$ is a closed convex set. Let $\varphi : \mathcal{X} \to \mathbb{R}$ be a convex function and $\eta > 0$. Define

$$u := \arg\min_{x\in\mathcal{X}} \{\eta\varphi(x) + \mathcal{B}_h(x, v)\}.$$

It follows that, for all $z \in \mathcal{X}$ and $g(u) \in \partial\varphi(u)$

$$\eta\langle g(u), u - z\rangle \le \mathcal{B}_h(z, v) - \mathcal{B}_h(z, u) - \mathcal{B}_h(u, v).$$

The next lemma relates the proximal gradient updates (e.g., as in Algorithm OptCMD) with the gradients of the linearized components.

Lemma 3.2: Suppose that $\mathcal{X}$ is a closed convex set in a Banach space $\mathbb{S}$ equipped with a norm $\|\cdot\|$. Let h be 1-strongly convex w.r.t. $\|\cdot\|$. Let $w_1, w_2 \in \mathbb{S}^*$, $v \in \mathcal{X}$, $r : \mathcal{X} \to \mathbb{R}$ is a convex function and $\eta > 0$. Define

$$u_1 := \arg\min_{x_1\in\mathcal{X}} \left\{\langle w_1, x_1\rangle + r(x_1) + \frac{1}{\eta}\mathcal{B}_h(x_1, v)\right\}$$

$$u_2 := \arg\min_{x_2\in\mathcal{X}} \left\{\langle w_2, x_2\rangle + r(x_2) + \frac{1}{\eta}\mathcal{B}_h(x_2, v)\right\}.$$

Then, it holds that $\|u_1 - u_2\| \le \eta\|w_1 - w_2\|_*$.

Proof: From the optimality of u_1 and u_2 [51, Th. 3.1.24], we have

$$\eta\langle w_1 + g(u_1), u_2 - u_1\rangle \ge \langle\nabla h(u_1) - \nabla h(v), u_1 - u_2\rangle \text{ and}$$

$$\eta\langle -w_2 - g(u_2), u_2 - u_1\rangle \ge \langle\nabla h(v) - \nabla h(u_2), u_1 - u_2\rangle$$

where $g(u) \in \partial r(u)\ \forall u \in \mathcal{X}$. Adding these two inequalities up, we get

$$\eta\langle w_1 - w_2 + g(u_1) - g(u_2), u_2 - u_1\rangle \ge \langle\nabla h(u_1) - \nabla h(u_2), u_1 - u_2\rangle. \tag{13}$$

Since h is 1-strongly convex, it follows that:

$$\langle\nabla h(u_1) - \nabla h(u_2), u_1 - u_2\rangle \ge \|u_1 - u_2\|^2. \tag{14}$$

Combining (13) and (14), using the Cauchy–Schwarz inequality and the monotonicity of the subgradient $\langle g(u_1) - g(u_2), u_2 - u_1\rangle \le 0$, we have $\|u_1 - u_2\|^2 \le \eta\|w_1 - w_2\|_*\|u_2 - u_1\|$. As a result, the claim follows. ■

The next lemma is useful for upper bounding quantities arising from the use of adaptive step-sizes in OCO algorithms.

Lemma 3.3: Let $\{a_k\}_{k=1}^T, \{b_k\}_{k=1}^{T+1}, \{c_k\}_{k=1}^T$, be nonnegative sequences, with $b_{t+1} \ge b_t$ and $c_{t+1} \ge c_t$. Then, for $T \ge 1$

$$\sum_{t=1}^T a_t\sqrt{\frac{b_t}{c_t + \sum_{k=1}^t a_k}} \le 2\sqrt{b_T\left(c_T + \sum\nolimits_{t=1}^T a_t\right)}.$$

Proof: The proof is by induction. For $T = 1$, one can show analytically that the inequality holds. Suppose that the inequality holds for some $T - 1 > 2$. Thus, it follows that:

$$\begin{aligned}
&\sum_{t=1}^T a_t\sqrt{\frac{b_t}{c_t + \sum_{k=1}^t a_k}} \\
&= \sum_{t=1}^{T-1} a_t\sqrt{\frac{b_t}{c_t + \sum_{k=1}^t a_k}} + a_T\sqrt{\frac{b_T}{c_T + \sum_{k=1}^T a_k}} \\
&\le 2\sqrt{b_{T-1}\left(c_{T-1} + \sum\nolimits_{t=1}^{T-1} a_t\right)} + a_T\sqrt{\frac{b_T}{c_T + \sum_{k=1}^T a_k}} \\
&\le 2\sqrt{b_T\left(c_T - a_T + \sum\nolimits_{t=1}^T a_t\right)} + a_T\sqrt{\frac{b_T}{c_T + \sum_{k=1}^T a_k}} \\
&= \sqrt{b_T}\left(2\sqrt{A - a_T} + \frac{a_T}{\sqrt{A}}\right)
\end{aligned}$$

where $A := c_T + \sum_{t=1}^T a_t$. As a function of $a_T \ge 0$, one can show that the R.H.S of the previous inequality is maximized when $a_T = 0$. Thus, $\sqrt{b_T}(2\sqrt{A - a_T} + \frac{a_T}{\sqrt{A}}) \le 2\sqrt{b_T A}$. This concludes the proof. ■

The next lemma is useful for upper bounding quantities arising from the use of adaptive step-sizes in OCO algorithms, especially when the costs are strongly convex.

Lemma 3.4: Given two positive reals a and b, it holds that

$$b\left(\frac{1}{b} - \frac{1}{a}\right) \le \log\left(\frac{b^{-1}}{a^{-1}}\right).$$

Proof: Let us first recall the identity $\log(\xi) \le \xi - 1$, for any $\xi > 0$. Set $\xi = a^{-1}/b^{-1}$. Notice that

$$-\log\left(\frac{b^{-1}}{a^{-1}}\right) = \log\left(\frac{a^{-1}}{b^{-1}}\right) \le \frac{a^{-1}}{b^{-1}} - 1 = b\left(\frac{1}{a} - \frac{1}{b}\right).$$

Thus, the claim is an immediate consequence of the above relation. ■

B. Main Proofs

Next, we continue with the proofs of our main results.

1) Proof of Theorem 2.5: Define $x^* := \arg\min_{x\in\mathcal{X}} \sum_{t=1}^T f_t(x)$. From the definition of f_t

$$\begin{aligned} f_t(x_t) - f_t(x^*) &= s_t(x_t) + r_t(x_t) - s_t(x^*) - r_t(x^*) \\ &= s_t(x_t) - s_t(x^*) + r_t(x_t) - \hat{r}_t(x_t) \\ &\quad + \hat{r}_t(x_t) - r_t(y_t) + r_t(y_t) - r_t(x^*) \\ &\le \Delta_t + \langle \nabla s_t(x_t), x_t - x^*\rangle \\ &\quad + \langle \hat{g}_t(x_t), x_t - y_t\rangle + \langle g_t(y_t), y_t - x^*\rangle \\ &= \Delta_t + \langle \nabla s_t(x_t) - \nabla \hat{s}_t(y_{t-1}), x_t - y_t\rangle \\ &\quad + \langle \nabla \hat{s}_t(y_{t-1}) + \hat{g}_t(x_t), x_t - y_t\rangle \\ &\quad + \langle \nabla s_t(x_t) + g_t(y_t), y_t - x^*\rangle \end{aligned}$$

where $g_t(y_t) \in \partial r_t(y_t)$, $\hat{g}_t(x_t) \in \partial \hat{r}_t(x_t)$, $\Delta_t := r_t(x_t) - \hat{r}_t(x_t) + \hat{r}_t(y_t) - r_t(y_t)$ and the inequality follows from the convexity of s_t, r_t and $\hat{r}_t$. Using Lemma 3.1, we get

$$\begin{aligned} &f_t(x_t) - f_t(x^*) \\ &\le \Delta_t + \langle \nabla s_t(x_t) - \nabla \hat{s}_t(y_{t-1}), x_t - y_t\rangle \\ &\quad + \frac{1}{\eta_t}\left(\mathcal{B}_h(x^*, y_{t-1}) - \mathcal{B}_h(x^*, y_t)\right. \\ &\quad \left. - \mathcal{B}_h(x_t, y_{t-1}) - \mathcal{B}_h(y_t, x_t)\right) \\ &\le \Delta_t + \|\nabla s_t(x_t) - \nabla \hat{s}_t(y_{t-1})\|_* \|x_t - y_t\| \\ &\quad + \frac{1}{\eta_t}\left(\mathcal{B}_h(x^*, y_{t-1}) - \mathcal{B}_h(x^*, y_t)\right. \\ &\quad \left. - \mathcal{B}_h(x_t, y_{t-1}) - \mathcal{B}_h(y_t, x_t)\right) \\ &= A_t + B_t + C_t \end{aligned}$$

where we define

$$\begin{aligned} A_t &:= \|\nabla s_t(x_t) - \nabla \hat{s}_t(y_{t-1})\|_* \|x_t - y_t\| \\ &\quad - \frac{1}{2\eta_t}\mathcal{B}_h(y_t, x_t) - \frac{1}{\eta_t}\mathcal{B}_h(x_t, y_{t-1}) \\ B_t &:= \frac{1}{\eta_t}\left(\mathcal{B}_h(x^*, y_{t-1}) - \mathcal{B}_h(x^*, y_t)\right), \\ C_t &:= \Delta_t - \frac{1}{2\eta_t}\mathcal{B}_h(y_t, x_t). \end{aligned}$$

We will proceed by upper bounding $\sum_{t=1}^T A_t$ and $\sum_{t=1}^T B_t$ separately.

(Upper bounding $\sum_{t=1}^T A_t$): Starting from the fact that $-\mathcal{B}_h(x, y) \le -\frac{1}{2}\|x - y\|^2$ and that $ab \le \rho a^2 + \frac{b^2}{4\rho}$ for any $\rho > 0$, we have

$$\begin{aligned} A_t &= \|\nabla s_t(x_t) - \nabla \hat{s}_t(y_{t-1})\|_* \|x_t - y_t\| \\ &\quad - \frac{1}{2\eta_t}\mathcal{B}_h(y_t, x_t) - \frac{1}{\eta_t}\mathcal{B}_h(x_t, y_{t-1}) \\ &\le \|\nabla s_t(x_t) - \nabla \hat{s}_t(y_{t-1})\|_* \|x_t - y_t\| \\ &\quad - \frac{1}{4\eta_t}\|y_t - x_t\|^2 - \frac{1}{2\eta_t}\|x_t - y_{t-1}\|^2 \\ &\le \eta_{t+1}\|\nabla s_t(x_t) - \nabla \hat{s}_t(y_{t-1})\|_*^2 \\ &\quad + \left(\frac{1}{4\eta_{t+1}} - \frac{1}{4\eta_t}\right)\|x_t - y_t\|^2 - \frac{1}{2\eta_t}\|x_t - y_{t-1}\|^2 \\ &\le 2\eta_{t+1}\|\nabla s_t(x_t) - \nabla s_t(y_{t-1})\|_*^2 - \frac{1}{2\eta_t}\|x_t - y_{t-1}\|^2 \\ &\quad + 2\eta_{t+1}\|\nabla s_t(y_{t-1}) - \nabla \hat{s}_t(y_{t-1})\|_*^2 + \frac{R^2}{2\eta_{t+1}} - \frac{R^2}{2\eta_t} \\ &\le \left(2\beta^2\eta_{t+1} - \frac{1}{2\eta_t}\right)\|x_t - y_{t-1}\|^2 + \frac{R^2}{2\eta_{t+1}} - \frac{R^2}{2\eta_t} \\ &\quad + 2\eta_{t+1}\|\nabla s_t(y_{t-1}) - \nabla \hat{s}_t(y_{t-1})\|_*^2 \\ &\le 2\eta_{t+1}\|\nabla s_t(y_{t-1}) - \nabla \hat{s}_t(y_{t-1})\|_*^2 + \frac{R^2}{2\eta_{t+1}} - \frac{R^2}{2\eta_t} \end{aligned} \tag{15}$$

where used the facts that η_t is nonincreasing, Assumption 2.4 and the fact that $\eta_t \le \frac{1}{2\beta}$, which implies $2\beta^2\eta_{t+1} - \frac{1}{2\eta_t} \le 0$. Next, we will bound the two terms of (15) separately. Summing the first term over $t = 1, \ldots, T$, we get

$$\begin{aligned} &2\sum_{t=1}^T \eta_{t+1}\|\nabla s_t(y_{t-1}) - \nabla \hat{s}_t(y_{t-1})\|_*^2 \\ &= 2\sum_{t=1}^T \frac{\|\nabla s_t(y_{t-1}) - \nabla \hat{s}_t(y_{t-1})\|_*^2}{\sqrt{4\beta^2 + (V_t')^2 + D_t'}} \\ &\le 4\sqrt{4\beta^2 + (V_T')^2 + D_T'} \le 4V_T' + 4\sqrt{4\beta^2 + D_T'} \end{aligned}$$

where the inequalities follow from the definition of D_t', Lemma 3.3 and $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$. Summing the second and third terms of (15) over $t = 1, \ldots, T$ and telescoping the sum, we get

$$\frac{R^2}{2}\sum_{t=1}^T \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right) \le \frac{R^2}{2\eta_{T+1}}.$$

Putting these bounds together, we arrive at

$$\begin{aligned} \sum_{t=1}^T A_t &\le 4V_T' + \frac{R^2}{2\eta_{T+1}} + 4\sqrt{4\beta^2 + D_T'} \\ &= \left(4 + \frac{R^2}{2}\right)\left(V_T' + \sqrt{4\beta^2 + D_T'}\right). \end{aligned} \tag{16}$$

(Upper bounding $\sum_{t=1}^T B_t$): Rearranging and telescoping the sum, we have

$$\begin{aligned} \sum_{t=1}^T B_t &\le \sum_{t=1}^T \frac{1}{\eta_t}\left(\mathcal{B}_h(x^*, y_{t-1}) - \mathcal{B}_h(x^*, y_t)\right) \\ &\le \frac{1}{\eta_1}\mathcal{B}_h(x^*, y_0) + \sum_{t=1}^{T-1}\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)\mathcal{B}_h(x^*, y_t) \\ &\le \frac{R^2}{\eta_T} \end{aligned} \tag{17}$$

where we used Assumption 2.4. Putting (16) and (17) together, we arrive at

$$\begin{aligned}\mathbf{Reg}_T^s &\le \sum_{t=1}^{T}(A_t+B_t+C_t)\\ &\le \left(4+\frac{R^2}{2}\right)\left(V_T'+\sqrt{4\beta^2+D_T'}\right)+\frac{R^2}{\eta_T}\\ &\quad+\sum_{t=1}^{T}\left(\Delta_t-\frac{1}{2\eta_t}\mathcal{B}_h(y_t,x_t)\right)\\ &\le \left(4+\frac{3R^2}{2}\right)\left(V_T'+\sqrt{4\beta^2+D_T'}\right)\\ &\quad+\sum_{t=1}^{T}\left(\Delta_t-\frac{1}{2\eta_t}\mathcal{B}_h(y_t,x_t)\right) \qquad (18)\\ &\le \left(5+\frac{3R^2}{2}\right)\left(V_T'+\sqrt{4\beta^2+D_T'}\right) \qquad (19)\end{aligned}$$

where we used the definition of η_{T+1} and the fact that $V_T'=\sum_{t=1}^{T}|\Delta_t|$ by definition. Similar to [31, Th. 6.2], we will proceed to bound the regret in a second way, which in turn will imply the regret is upper bounded by the minimum of (19) and the second bound. In particular, we will focus on the following part of (18)

$$\begin{aligned}&\left(5+\frac{3R^2}{4}\right)V_T'-\sum_{t=1}^{T}\frac{1}{2\eta_t}\mathcal{B}_h(y_t,x_t)\\ &=\left(5+\frac{3R^2}{4}\right)\sum_{t=1}^{T}|\Delta_t|-\sum_{t=1}^{T}\frac{1}{4\eta_t}\|x_t-y_t\|^2=c\lambda_T\end{aligned}$$

where $\lambda_T:=\sum_{t=1}^{T}(|\Delta_t|-\frac{1}{4c\eta_t}\|x_t-y_t\|^2)$ and $c:=5+\frac{3R^2}{4}$. Thus

$$\mathbf{Reg}_T^s\le c\lambda_T+\left(4+\frac{3R^2}{4}\right)\sqrt{4\beta^2+D_T'}. \qquad (20)$$

Next, we will proceed to prove an upper bound to λ_T^2, which will naturally imply an upper bound to $c\lambda_T$. To do so, we will first prove an upper bound to the term $|\Delta_t|-\frac{1}{4c\eta_t}\|x_t-y_t\|^2$.

(Upper bounding $|\Delta_t|-\frac{1}{4c\eta_t}\|x_t-y_t\|^2$): From the definition of Δ_t and convexity of r_t and $\hat{r}_t$, we have that

$$\begin{aligned}\Delta_t &\le \langle g_t(x_t)-\hat{g}_t(y_t),x_t-y_t\rangle\\ &\le \sqrt{2}R\|g_t(x_t)-\hat{g}_t(y_t)\|_*+\frac{\|x_t-y_t\|^2}{4c\eta_t} \qquad (21)\end{aligned}$$

and

$$\begin{aligned}\Delta_t &\le \|g_t(x_t)-\hat{g}_t(y_t)\|_*\|x_t-y_t\|\\ &\le c\eta_t\|g_t(x_t)-\hat{g}_t(y_t)\|_*^2+\frac{\|x_t-y_t\|^2}{4c\eta_t} \qquad (22)\end{aligned}$$

for any $g_t(x_t)\in\partial r_t(x_t)$ and $\hat{g}_t(y_t)\in\partial\hat{r}_t(y_t)$, where we used the facts that $\frac{\|x_t-y_t\|^2}{4c\eta_t}\ge 0$ and $ab\le\rho a^2+\frac{b^2}{4\rho}$ for any $\rho>0$.

Similarly, we also have that

$$-\Delta_t\le\sqrt{2}R\|g_t(y_t)-\hat{g}_t(x_t)\|_*+\frac{\|x_t-y_t\|^2}{4c\eta_t} \text{ and} \qquad (23)$$

$$-\Delta_t\le c\eta_t\|g_t(y_t)-\hat{g}_t(x_t)\|_*^2+\frac{\|x_t-y_t\|^2}{4c\eta_t} \qquad (24)$$

Combining (21), (22), (23), and (24), we get that

$$|\Delta_t|-\frac{\|x_t-y_t\|^2}{4c\eta_t}\le\min\left\{\sqrt{2}RG_t,c\eta_tG_t^2\right\} \qquad (25)$$

where $G_t:=\max\{\|g_t(x_t)-\hat{g}_t(y_t)\|_*,\|g_t(y_t)-\hat{g}_t(x_t)\|_*\}$.

(Upper bounding λ_T^2): Notice that

$$\lambda_t-\lambda_{t-1}=|\Delta_t|-\frac{\|x_t-y_t\|^2}{4c\eta_t}\le\min\left\{\sqrt{2}RG_t,c\eta_tG_t^2\right\}. \qquad (26)$$

Defining $\lambda_0^2:=0$, we have that

$$\begin{aligned}\lambda_T^2&=\sum_{t=1}^{T}\left(\lambda_t^2-\lambda_{t-1}^2\right)\\ &=\sum_{t=1}^{T}\left(\left(\lambda_t-\lambda_{t-1}\right)^2+2\left(\lambda_t-\lambda_{t-1}\right)\lambda_{t-1}\right)\\ &\le\sum_{t=1}^{T}\left(2R^2G_t^2+2c\eta_t\lambda_{t-1}G_t^2\right)\end{aligned}$$

where the last inequality follows from (26). Next, notice that by definition

$$\eta_t\lambda_{t-1}=\frac{\sum_{k=1}^{t-1}\left(|\Delta_k|-\frac{\|x_k-y_k\|^2}{2c\eta_k}\right)}{\sqrt{4\beta^2+\left(\sum_{k=1}^{t-1}|\Delta_k|\right)^2+D_{t-1}'}}\le\frac{\sum_{k=1}^{t-1}|\Delta_k|}{\sum_{k=1}^{t-1}|\Delta_k|}=1.$$

Thus, we have that

$$\lambda_T^2\le\sum_{t=1}^{T}\left(2R^2G_t^2+2cG_t^2\right)=(2R^2+2c)\sum_{t=1}^{T}G_t^2.$$

Taking the square root and substituting it into (20), we get the second regret bound

$$\begin{aligned}\mathbf{Reg}_T^s&\le c\sqrt{2R^2+2c}\sqrt{\sum_{t=1}^{T}G_t^2}\\ &\quad+\left(5+\frac{3R^2}{4}\right)\sqrt{4\beta^2+D_T'}. \qquad (27)\end{aligned}$$

Finally, combining (19) and (27), arrive at

$$\begin{aligned}\mathbf{Reg}_T^s&\le\min\left\{\left(5+\frac{3R^2}{4}\right)V_T',c\sqrt{2R^2+2c}\sqrt{\sum_{t=1}^{T}G_t^2}\right\}\\ &\quad+\left(5+\frac{3R^2}{4}\right)\sqrt{4\beta^2+D_T'}\\ &=O\left(1+\sqrt{D_T'}+\min\left\{V_T',\sqrt{T}\right\}\right).\end{aligned}$$

■

2) Proof of Theorem 2.9: In order to prove Theorem 2.9, first we will prove a version of this theorem for general Bregman divergences and a general notion of strong convexity (see Lemma 3.8). This result is achieved by exploiting a certain technical assumption (see Assumption 3.6). Then, we will show that for the euclidean case (i.e., $\mathcal{B}_h(x,y)=\frac{1}{2}\|x-y\|_2^2$), this technical assumption always holds and Theorem 2.9 follows.

Definition 3.5 (α-Strong convexity w.r.t. $\mathcal{B}_h$): A function $f:\mathcal{X}\to\mathbb{R}$ is *α-strongly convex* w.r.t. $\mathcal{B}_h$ if $f(x)-f(y)\leq\langle\nabla f(x),x-y\rangle-\alpha\mathcal{B}_h(y,x)$, for all $x,y\in\mathcal{X}$.

Assumption 3.6 (Technical assumption): For $1/\eta>\alpha>0$, there exists a constant $\lambda>0$ such that $\lambda\mathcal{B}_h(x,y)-\frac{1}{\eta}\mathcal{B}_h(y,z)-\alpha\mathcal{B}_h(x,z)\leq 0$, for all $x,y,z\in\mathcal{X}$.

Before stating the general version of Theorem 2.9, we make a short remark on Assumption 3.6.

Remark 3.7 (Mildness of Assumption 3.6): Notice that η, α, and $\mathcal{B}_h(x,y)$ are all non-negative. Thus, for a general choice of h, one should expect to be able to choose a small enough λ to ensure that the inequality in Assumption 3.6 holds. In particular, when $\mathcal{B}_h(x,y)=\frac{1}{2}\|x-y\|_2^2$, we will show that Assumption 3.6 holds for $\lambda=\alpha/2$.

Lemma 3.8 (Strongly convex case with general divergence): Suppose that Assumptions 2.4 and 3.6 hold and that the costs $\{f_t\}_{t=1}^T$ are α-strongly convex w.r.t. B_h. Using the adaptive step-size

$$\eta_1=\frac{1}{2\beta},\quad \eta_t=\left(\frac{\lambda}{\sigma^2}D'_{t-1}+2\beta\right)^{-1}$$

for all $t>1$, Algorithm OptCMD with $\hat{r}_t=r_t$ guarantees

$$\mathbf{Reg}_T^s\leq O\left(1+\log(1+D'_T)\right).$$

Proof: Let $x^*:=\arg\min_{x\in\mathcal{X}}\sum_{t=1}^T f_t(x)$. Since s_t is α-strongly convex w.r.t. $\mathcal{B}_h$, we have

$$s_t(x_t)-s_t(x^*)\leq\langle\nabla s_t(x_t),x_t-x^*\rangle-\alpha\mathcal{B}_h(x^*,x_t).$$

Thus

$$\begin{aligned}
&f_t(x_t)-f_t(x^*)=s_t(x_t)+r_t(x_t)-s_t(x^*)-r_t(x^*)\\
&=s_t(x_t)-s_t(x^*)+r_t(x_t)-r_t(x_t)\\
&\quad+r_t(x_t)-r_t(y_t)+r_t(y_t)-r_t(x^*)\\
&\leq\langle\nabla s_t(x_t),x_t-x^*\rangle-\alpha\mathcal{B}_h(x^*,x_t)+\langle g_t(x_t),x_t-y_t\rangle\\
&\quad+\langle g_t(y_t),y_t-x^*\rangle\\
&=\langle\nabla s_t(x_t)-\nabla\hat{s}_t(y_{t-1}),x_t-y_t\rangle\\
&\quad+\langle\nabla s_t(x_t)+g_t(y_t),y_t-x^*\rangle\\
&\quad+\langle\nabla\hat{s}_t(y_{t-1})+g_t(x_t),x_t-y_t\rangle-\alpha\mathcal{B}_h(x^*,x_t)
\end{aligned}$$

where $g_t(y_t)\in\partial r_t(y_t)$, $g_t(x_t)\in\partial r_t(x_t)$ and the inequality follows from the convexity of s_t and r_t. Using Lemma 3.1, we get

$$\begin{aligned}
&f_t(x_t)-f_t(x^*)\\
&\leq\langle\nabla s_t(x_t)-\nabla\hat{s}_t(y_{t-1}),x_t-y_t\rangle-\alpha\mathcal{B}_h(x^*,x_t)\\
&\quad+\frac{1}{\eta_t}\left(\mathcal{B}_h(x^*,y_{t-1})-\mathcal{B}_h(x^*,y_t)\right.\\
&\quad\left.-\mathcal{B}_h(x_t,y_{t-1})-\mathcal{B}_h(y_t,x_t)\right)\\
&\leq A_t+B_t
\end{aligned}$$

where we define

$$\begin{aligned}
A_t&:=\frac{1}{\eta_t}\left(\mathcal{B}_h(x^*,y_{t-1})-\mathcal{B}_h(x^*,y_t)-\mathcal{B}_h(y_t,x_t)\right)\\
&\quad-\alpha\mathcal{B}_h(x^*,x_t)\\
B_t&:=\|\nabla s_t(x_t)-\nabla\hat{s}_t(y_{t-1})\|_*\|x_t-y_t\|-\frac{1}{\eta_t}\mathcal{B}_h(x_t,y_{t-1}).
\end{aligned}$$

With the above notations at hand, it follows that:

$$\mathbf{Reg}_T^s=\sum_{t=1}^T\left(f_t(x_t)-f_t(x^*)\right)\leq\sum_{t=1}^T A_t+\sum_{t=1}^T B_t. \tag{28}$$

We proceed by bounding $\sum_{t=1}^T A_t$ and $\sum_{t=1}^T B_t$ separately.

(Upper bounding $\sum_{t=1}^T A_t$): Observe that

$$\begin{aligned}
\sum_{t=1}^T A_t&=\sum_{t=1}^T\frac{1}{\eta_t}\left(\mathcal{B}_h(x^*,y_{t-1})-\mathcal{B}_h(x^*,y_t)\right)\\
&\quad-\sum_{t=1}^T\left(\frac{1}{\eta_t}\mathcal{B}_h(y_t,x_t)+\alpha\mathcal{B}_h(x^*,x_t)\right)\\
&\leq\frac{\mathcal{B}_h(x^*,y_0)}{\eta_1}+\sum_{t=1}^T\left(\frac{1}{\eta_{t+1}}-\frac{1}{\eta_t}\right)\mathcal{B}_h(x^*,y_t)\\
&\quad-\sum_{t=1}^T\left(\frac{1}{\eta_t}\mathcal{B}_h(y_t,x_t)+\alpha\mathcal{B}_h(x^*,x_t)\right).
\end{aligned}$$

Assumption 2.4 and $\eta_1=\frac{1}{2\beta}$ imply that $\frac{\mathcal{B}_h(x^*,y_0)}{\eta_1}\leq 2\beta R^2$.

From the definition of η_t, we have that

$$\frac{1}{\eta_{t+1}}-\frac{1}{\eta_t}=\frac{\lambda}{\sigma^2}\|\nabla s_t(y_{t-1})-\nabla\hat{s}_t(y_{t-1})\|_*^2. \tag{29a}$$

Hence, we obtain

$$\begin{aligned}
&\sum_{t=1}^T\left(\frac{1}{\eta_{t+1}}-\frac{1}{\eta_t}\right)\mathcal{B}_h(x^*,y_t)\\
&\quad=\lambda\sum_{t=1}^T\frac{\|\nabla s_t(y_{t-1})-\nabla\hat{s}_t(y_{t-1})\|_*^2}{\sigma^2}\mathcal{B}_h(x^*,y_t)\\
&\quad\leq\lambda\sum_{t=1}^T\mathcal{B}_h(x^*,y_t)
\end{aligned}\tag{29b}$$

where the inequality follows from the fifth item in Assumption 2.4. In light of the upper bounds derived in (29), we then infer that

$$\begin{aligned}
\sum_{t=1}^T A_t&\leq 2\beta R^2+\sum_{t=1}^T\left(\lambda\mathcal{B}_h(x^*,y_t)\right.\\
&\quad\left.-\frac{1}{\eta_t}\mathcal{B}_h(y_t,x_t)-\alpha\mathcal{B}_h(x^*,x_t)\right)\leq 2\beta R^2
\end{aligned}\tag{30}$$

where the second inequality follows from Assumption 3.6.

(Upper bounding $\sum_{t=1}^T B_t$): Invoking Lemma 3.2, we conclude that

$$\|y_t-x_t\|\leq\eta_t\|\nabla s_t(x_t)-\nabla\hat{s}_t(y_{t-1})\|_*$$

and as a result

$$B_t \le \eta_t \|\nabla s_t(x_t) - \nabla \hat{s}_t(y_{t-1})\|_*^2 - \frac{1}{\eta_t}\mathcal{B}_h(x_t, y_{t-1}).$$

Notice that

$$\begin{aligned} B_t &\le 2\eta_t \|\nabla s_t(x_t) - \nabla s_t(y_{t-1})\|_*^2 \\ &\quad + 2\eta_t \|\nabla s_t(y_{t-1}) - \nabla \hat{s}_t(y_{t-1})\|_*^2 - \frac{1}{\eta_t}\mathcal{B}_h(x_t, y_{t-1}) \\ &\le 2\eta_t \beta^2 \|x_t - y_{t-1}\|^2 + 2\eta_t \|\nabla s_t(y_{t-1}) - \nabla \hat{s}_t(y_{t-1})\|_*^2 \\ &\quad - \frac{1}{\eta_t}\mathcal{B}_h(x_t, y_{t-1}). \end{aligned}$$

where we made use of the identity $\|a-b\|^2 \le 2\|a-c\|^2 + 2\|c-b\|^2$ and the β-smoothness of s_t. Using Lemma $-\mathcal{B}_h(x,y) \le -\frac{1}{2}\|x-y\|^2$, we arrive at

$$\begin{aligned} B_t &\le \left(2\eta_t\beta^2 - \frac{1}{2\eta_t}\right)\|x_t - y_{t-1}\|^2 \\ &\quad + 2\eta_t \|\nabla s_t(y_{t-1}) - \nabla \hat{s}_t(y_{t-1})\|_*^2. \end{aligned}$$

From the definition of η_t, we have that $2\eta_t\beta^2 - \frac{1}{2\eta_t} \le 0 \,\forall t \ge 1$, and summing B_t over $t = 1, \dots, T$ yields

$$\begin{aligned} \sum_{t=1}^T B_t &\le 2\sum_{t=1}^T \eta_t \|\nabla s_t(y_{t-1}) - \nabla \hat{s}_t(y_{t-1})\|_*^2 \\ &\le 2\sum_{t=1}^T \eta_{t+1} \|\nabla s_t(y_{t-1}) - \nabla \hat{s}_t(y_{t-1})\|_*^2 \\ &\quad + 2\sum_{t=1}^T (\eta_t - \eta_{t+1}) \|\nabla s_t(y_{t-1}) - \nabla \hat{s}_t(y_{t-1})\|_*^2. \end{aligned}$$

By virtue of the fifth item in Assumption 2.4, it follows that:

$$\begin{aligned} &\sum_{t=1}^T (\eta_t - \eta_{t+1}) \|\nabla s_t(y_{t-1}) - \nabla \hat{s}_t(y_{t-1})\|_*^2 \\ &\quad \le \sigma^2 \sum_{t=1}^T (\eta_t - \eta_{t+1}) = \sigma^2 (\eta_1 - \eta_{T+1}) \le \sigma^2 \eta_1 = \frac{\sigma^2}{2\beta}. \end{aligned}$$

Based on the above analyses, it is straightforward to see that

$$\sum_{t=1}^T B_t \le \frac{\sigma^2}{\beta} + 2\sum_{t=1}^T \eta_{t+1} \|\nabla s_t(y_{t-1}) - \nabla \hat{s}_t(y_{t-1})\|_*^2. \quad (31)$$

Notice that by the definition η_t, we have

$$\|\nabla s_t(y_{t-1}) - \nabla \hat{s}_t(y_{t-1})\|_*^2 = \frac{\sigma^2}{\lambda}\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)$$

and as a result

$$\sum_{t=1}^T B_t \le \frac{\sigma^2}{\beta} + \frac{2\sigma^2}{\lambda}\sum_{t=1}^T \eta_{t+1}\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right).$$

Using Lemma 3.4 to upper bound the RHS of the inequality above, we have that

$$\begin{aligned} \sum_{t=1}^T B_t &\le \frac{\sigma^2}{\beta} + \frac{2\sigma^2}{\lambda}\sum_{t=1}^T \log\left(\frac{\eta_{t+1}^{-1}}{\eta_t^{-1}}\right) \\ &= \frac{\sigma^2}{\beta} + \frac{2\sigma^2}{\lambda}\left(\log\left(\frac{1}{\eta_{T+1}}\right) - \log\left(\frac{1}{\eta_1}\right)\right) \\ &= \frac{\sigma^2}{\beta} + \frac{2\sigma^2}{\lambda}\log\left(\frac{\eta_1}{\eta_{T+1}}\right) \end{aligned}$$

which immediately yields

$$\sum_{t=1}^T B_t \le \frac{\sigma^2}{\beta} + \frac{2\sigma^2}{\lambda}\log\left(1 + \frac{\lambda}{2\beta\sigma^2}D_T'\right). \quad (32)$$

(Regret upper bound): In light of (30) and (32), we get

$$\mathbf{Reg}_T^s \le 2\beta R^2 + \frac{\sigma^2}{\beta} + \frac{2\sigma^2}{\lambda}\log\left(1 + \frac{\lambda}{2\beta\sigma^2}D_T'\right).$$

The lemma immediately follows. ■

Finally, for the euclidean case (i.e., $\mathcal{B}_h(x,y) = \frac{1}{2}\|x-y\|_2^2$) and choosing $\lambda = \alpha/2$, we have

$$\begin{aligned} &\lambda\mathcal{B}_h(x^*, y_t) - \frac{1}{\eta_t}\mathcal{B}_h(y_t, x_t) - \alpha\mathcal{B}_h(x^*, x_t) \\ &\quad = \frac{\alpha}{4}\|x^* - y_t\|_2^2 - \frac{1}{2\eta_t}\|y_t - x_t\|_2^2 - \frac{\alpha}{2}\|x^* - x_t\| \\ &\quad \le \frac{\alpha}{2}\|x_t - y_t\|_2^2 - \frac{1}{2\eta_t}\|y_t - x_t\|_2^2 \le 0, \end{aligned}$$

where the second inequality follows from $\|a-b\|^2 \le 2\|a-c\|^2 + 2\|c-b\|^2$ and the third inequality follows from $\eta_t^{-1} \ge \beta \ge \alpha$. Thus, we have shown that Assumption 3.6 holds for all t, and Theorem 2.9 follows from Lemma 3.8. ■

3) Proof of Theorem 2.15: Let $u_t \in \mathcal{X}$. Following similar steps to the ones from the proof of Theorem 2.5, one can show that $f_t(x_t) - f_t(u_t) \le A_t + B_t + C_t$, where we define

$$\begin{aligned} A_t &:= \|\nabla s_t(x_t) - \nabla \hat{s}_t(y_{t-1})\|_* \|x_t - \tilde{y}_t\| \\ &\quad - \frac{1}{2\eta_t}\mathcal{B}_h(\tilde{y}_t, x_t) - \frac{1}{\eta_t}\mathcal{B}_h(x_t, y_{t-1}) \\ B_t &:= \frac{1}{\eta_t}\left(\mathcal{B}_h(u_t, y_{t-1}) - \mathcal{B}_h(u_t, \tilde{y}_t)\right), \; C_t := \Delta_t \\ &\quad - \frac{1}{2\eta_t}\mathcal{B}_h(\tilde{y}_t, x_t). \end{aligned}$$

Moreover, still following steps similar to the proof of Theorem 2.5, we can show that

$$\sum_{t=1}^T A_t \le \left(4 + \frac{R^2}{2}\right)\left(V_T' + \sqrt{4\beta^2 + D_T'}\right). \quad (33)$$

(Upper bounding $\sum_{t=1}^T B_t$): Adding $\pm\frac{1}{\eta_t}\mathcal{B}_h(u_{t+1}, y_t)$ and $\pm\frac{1}{\eta_t}\mathcal{B}_h(\Phi_t(u_t), y_t)$ to B_t and summing the result over $t = 1, \dots, T$, we get

$$\begin{aligned} \sum_{t=1}^T B_t = \sum_{t=1}^T \frac{1}{\eta_t}&\left(\mathcal{B}_h(u_t, y_{t-1}) - \mathcal{B}_h(u_{t+1}, y_t) + \mathcal{B}_h(u_{t+1}, y_t)\right. \\ &\left. - \mathcal{B}_h(\Phi_t(u_t), y_t) + \mathcal{B}_h(\Phi_t(u_t), \Phi_t(\tilde{y}_t)) - \mathcal{B}_h(u_t, \tilde{y}_t)\right) \end{aligned}$$

where we made use of $y_t = \Phi_t(\tilde{y}_t)$. By Assumption 2.11, it holds that for some positive real γ

$$\mathcal{B}_h(u_{t+1}, y_t) - \mathcal{B}_h(\Phi_t(u_t), y_t) \le \gamma\|u_{t+1} - \Phi_t(u_t)\|.$$

By Assumption 2.13, it further holds that

$$\mathcal{B}_h(\Phi_t(u_t), \Phi_t(\tilde{y}_t)) - \mathcal{B}_h(u_t, \tilde{y}_t) \leq 0.$$

By virtue of the last two inequalities, we arrive at

$$\sum_{t=1}^{T} B_t \leq \sum_{t=1}^{T} \frac{1}{\eta_t} \left(\mathcal{B}_h(u_t, y_{t-1}) - \mathcal{B}_h(u_{t+1}, y_t) + \gamma \|u_{t+1} - \Phi_t(u_t)\|\right). \tag{34}$$

Next, observe that

$$\begin{aligned}
&\sum_{t=1}^{T} \frac{1}{\eta_t} \left(\mathcal{B}_h(u_t, y_{t-1}) - \mathcal{B}_h(u_{t+1}, y_t)\right) \\
&\leq \frac{1}{\eta_1}\mathcal{B}_h(u_1, y_0) + \sum_{t=2}^{T} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) \mathcal{B}_h(u_t, y_{t-1}) \\
&\leq \frac{R^2}{\eta_1} + R^2 \sum_{t=2}^{T} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) \leq \frac{R^2}{\eta_T}
\end{aligned}$$

where we made use Assumption 2.4. Considering inequality (34), one can conclude based on the above arguments that

$$\begin{aligned}
\sum_{t=1}^{T} B_t &\leq \frac{R^2}{\eta_T} + \sum_{t=1}^{T} \left(\frac{\gamma}{\eta_t}\|u_{t+1} - \Phi_t(u_t)\|\right) \\
&\leq \frac{R^2}{\eta_T} + \frac{\gamma}{\eta_T} \sum_{t=1}^{T} \|u_{t+1} - \Phi_t(u_t)\| = \frac{1}{\eta_T}(R^2 + \gamma C_T')
\end{aligned} \tag{35}$$

where the second inequality follows from $\eta_t \geq \eta_{t+1}$. Putting (33) and (35) together, we arrive at:

$$\begin{aligned}
\mathbf{Reg}_T^s &\leq \sum_{t=1}^{T} (A_t + B_t + C_t) \\
&\leq \left(4 + \frac{R^2}{2}\right) \left(V_T' + \sqrt{4\beta^2 + D_T'}\right) \\
&\quad + (R^2 + \gamma C_T')\frac{1}{\eta_T} + \sum_{t=1}^{T} \left(\Delta_t - \frac{1}{2\eta_t}\mathcal{B}_h(y_t, x_t)\right) \\
&\leq \left(4 + \frac{3R^2}{2} + \gamma C_T'\right) \left(V_T' + \sqrt{4\beta^2 + D_T'}\right) \\
&\quad + \sum_{t=1}^{T} \left(\Delta_t - \frac{1}{2\eta_t}\mathcal{B}_h(y_t, x_t)\right) \qquad (36) \\
&\leq \left(5 + \frac{3R^2}{2} + \gamma C_T'\right) \left(V_T' + \sqrt{4\beta^2 + D_T'}\right) \qquad (37)
\end{aligned}$$

where we used the definition of η_{T+1} and the fact that $V_T' = \sum_{t=1}^{T} |\Delta_t|$. Define $c := 5 + \frac{3R^2}{4} + \gamma C_T'$ and $\lambda_T := \sum_{t=1}^{T}(|\Delta_t| - \frac{1}{4c\eta_t}\|x_t - y_t\|^2)$. By following the same steps of the last part of the proof of Theorem 2.5, one can show

$$\mathbf{Reg}_T^s \leq c\sqrt{2R^2 + 2c}\sqrt{\sum_{t=1}^{T} G_t^2} + \left(4 + \frac{3R^2}{4} + \gamma C_T'\right)\sqrt{4\beta^2 + D_T'}. \tag{38}$$

Finally, combining (37) and (38), arrive at

$$\begin{aligned}
\mathbf{Reg}_T^s &\leq \min \\
&\times \left\{\left(5 + \frac{3R^2}{4} + \gamma C_T'\right) V_T',\ c\sqrt{2R^2 + 2c}\sqrt{\sum_{t=1}^{T} G_t^2}\right\} \\
&\quad + \left(4 + \frac{3R^2}{4} + \gamma C_T'\right)\sqrt{4\beta^2 + D_T'} \\
&= O\left((1 + C_T')\left(1 + \sqrt{D_T'} + \min\left\{V_T', \sqrt{(1 + C_T')T}\right\}\right)\right).
\end{aligned}$$

This concludes the proof. ■

C. Proof of Theorem 2.17

Similarly to the beginning of the proof of Theorem 2.15, one can show that $f_t(x_t) - f_t(u_t) \leq B_t + C_t$, where we define

$$B_t := \frac{1}{\eta_t}\left(\mathcal{B}_h(u_t, y_{t-1}) - \mathcal{B}_h(u_t, \tilde{y}_t)\right) \text{ and}$$

$$C_t := \Delta_t - \frac{1}{\eta_t}\mathcal{B}_h(\tilde{y}_t, x_t).$$

Next, continuing following the proof of Theorem 2.15, we have that

$$\sum_{t=1}^{T} B_t \leq \frac{R^2}{\eta_T} + \sum_{t=1}^{T} \frac{\gamma}{\eta_t}\|u_{t+1} - \Phi_t(u_t)\| \leq \frac{1}{\eta_T}\left(R^2 + \gamma\tau\right).$$

Thus, we have that

$$\begin{aligned}
\mathbf{Reg}_T^s &\leq \sum_{t=1}^{T} (B_t + C_t) \\
&\leq \frac{1}{\eta_T}\left(R^2 + \gamma\tau\right) + \sum_{t=1}^{T} \left(\Delta_t - \frac{1}{\eta_t}\mathcal{B}_h(y_t, x_t)\right) \\
&\leq \left(\frac{R^2}{\tau} + \gamma + 1\right) V_T' - \sum_{t=1}^{T} \frac{1}{\eta_t}\mathcal{B}_h(y_t, x_t).
\end{aligned}$$

Again following the steps of the proof of Theorem 2.15, we can alternatively bound the regret by

$$\mathbf{Reg}_T^s \leq c\sqrt{2R^2 + 2\tau c}\sqrt{\sum_{t=1}^{T} G_t^2} \leq O(\sqrt{(1+\tau)T})$$

where $c = \frac{R^2}{\tau} + \gamma + 1$. Combining the two regret bounds, we have

$$\mathbf{Reg}_T^s \leq O\left(\min\left\{V_T',\ \sqrt{(1+\tau)T}\right\}\right).$$

This concludes the proof. ■

1) Proof of Theorem 2.18: We start the proof by following similar steps to the ones taken in the proof of Theorem 2.9. By doing so, we arrive at

$$\mathbf{Reg}_T^d = \sum_{t=1}^{T}(f_t(x_t) - f_t(u_t)) \leq \sum_{t=1}^{T} A_t + \sum_{t=1}^{T} B_t \tag{39}$$

where

$$A_t := \|\nabla s_t(x_t) - \nabla \hat{s}_t(y_{t-1})\|_* \|x_t - \tilde{y}_t\| - \frac{1}{\eta_t}\mathcal{B}_h(\tilde{y}_t, x_t) - \frac{1}{\eta_t}\mathcal{B}_h(x_t, y_{t-1})$$

$$B_t := \frac{1}{\eta_t}(\mathcal{B}_h(u_t, y_{t-1}) - \mathcal{B}_h(u_t, \tilde{y}_t)).$$

(Upper bounding $\sum_{t=1}^T A_t$): We proceed by bounding $\sum_{t=1}^T A_t$ in the sequel. Recall that by definition, $\eta_t \leq 1/(2\beta)$ for all t. Thus, by following similar steps as taken in the proof of Theorem 2.5, we get

$$\sum_{t=1}^T A_t \leq \frac{R^2}{2\eta_{T+1}} + 2\sum_{t=1}^T \eta_{t+1}\|\nabla s_t(y_{t-1}) - \nabla \hat{s}_t(y_{t-1})\|_*^2.$$

Recall the definition of η_t. Invoking lemma Lemma 3.3, we arrive at

$$\sum_{t=1}^T A_t \leq \frac{R^2}{2\eta_{T+1}} + 4\sqrt{(\theta_T + D'_T)(1 + C'_T)}. \tag{40}$$

(Upper bounding $\sum_{t=1}^T B_t$): Following similar steps as taken in the proof of Theorem 2.15, we can bound:

$$\sum_{t=1}^T B_t \leq \frac{R^2}{\eta_T} + \sum_{t=1}^T \frac{\gamma}{\eta_t}\|u_{t+1} - \Phi_t(u_t)\|.$$

Define $\Delta u_t := \|u_{t+1} - \Phi_t(u_t)\|$ and notice that

$$\begin{aligned}\sum_{t=1}^T \frac{\gamma}{\eta_t}\Delta u_t &= \gamma\sum_{t=1}^T \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t+1}} + \frac{1}{\eta_{t+1}}\right)\Delta u_t \\ &= \gamma\sum_{t=1}^T \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t+1}}\right)\Delta u_t + \gamma\sum_{t=1}^T \frac{1}{\eta_{t+1}}\Delta u_t \\ &\leq \frac{\gamma R^2}{\eta_1} + \gamma\sum_{t=1}^T \frac{1}{\eta_{t+1}}\Delta u_t\end{aligned}$$

where for the last inequality, we assumed without loss of generality that $\Delta u_t \leq R^2$. Following these same steps again, and using the fact that $\eta_1 = \eta_2$, we get

$$\sum_{t=1}^T \frac{\gamma}{\eta_t}\|u_{t+1} - \Phi_t(u_t)\| \leq \frac{2\gamma R^2}{\eta_1} + \gamma\sum_{t=1}^T \frac{1}{\eta_{t+2}}\Delta u_t.$$

Recall the definition of η_t in Theorem 2.18. Invoking Lemma 3.3, we get

$$\gamma\sum_{t=1}^T \frac{1}{\eta_{t+2}}\|u_{t+1} - \Phi_t(u_t)\| \leq 2\gamma\sqrt{(\theta_{T+2} + D'_{T+1})(1 + C'_T)}.$$

Back to our upper bound on A_t, we now have

$$\sum_{t=1}^T B_t \leq \frac{2\gamma R^2}{\eta_1} + \frac{R^2}{\eta_T} + 2\gamma\sqrt{(\theta_{T+2} + D'_{T+1})(1 + C'_T)}. \tag{41}$$

(Regret upper bound): Considering equations (39), (40), and (41), it holds that

$$\mathbf{Reg}_T^d \leq \frac{2\sigma^2 R^2}{\eta_1} + \frac{3R^2}{2\eta_T} + (4 + 2\gamma)\sqrt{(\theta_{T+2} + D'_{T+1})(1 + C'_T)}.$$

This concludes the proof. ■

IV. Numerical Experiments

A. Tracking Dynamical Parameters

In this section, we employ a strategy based on Algorithm OptDCMD in a parameter tracking problem. The scenario presented in this section is based on the numerical experiment of [34]. Denote the parameters to be tracked by $u_t \in \mathbb{R}^4$. These parameters have dynamics described by the linear model $u_{t+1} = Au_t + v_t$. Similarly to [34], we emphasize that our online learning results hold even when the noise is adversarial with an unknown structure. For this experiment, we use

$$A = \begin{bmatrix} 1 & 0.1 & 0 & 0 \\ 0 & 1 & 0.1 & 0 \\ 0 & 0 & 1 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad v_t = \begin{cases} 2 & \text{if } \tilde{v}_t > 0 \\ -1 & \text{if } \tilde{v}_t \leq 0 \end{cases}$$

where $\tilde{v}_t$ is Gaussian noise with a random covariance matrix, and the inequalities in the definition of v_t are component-wise. The cost at time t is defined as $f_t(x_t) = \frac{1}{2}\|x_t - u_t\|_2^2 + \|x_t\|_1$, where $s_t(x_t) := \frac{1}{2}\|x_t - u_t\|_2^2$, $r_t(x_t) := \|x_t\|_1$ and x_t is the output of our tracking algorithm. We assume the Player has access to $\Phi_t(x) = Ax$, which is an approximate model of the dynamics of u_t.

To choose its action sequence $\{x_t\}_{t=1}^T$, the Player employs a variation of Algorithm OptDCMD with $h(x) = \frac{1}{2}\|x\|_2^2$ (i.e., the euclidean setup), with the difference that in the update rule of $\tilde{y}_t$, we use a constant step-size $\eta_t = 1$. This change was inspired by [22], and the fact that $\frac{1}{2}\|x_t - u_t\|_2^2$ is smooth and strongly convex. For the update rule of x_t, we use the step-size defined in Theorem 2.15 (notice that since the nonsmooth component of the cost f_t is fixed, $V'_t = 0$ for all t). We consider the following gradient prediction models:

1) *Perfect*: a perfect model $\nabla \hat{s}_t(y_{t-1}) := \nabla s_t(y_{t-1})$.
2) *Noisy*: a noisy model $\nabla \hat{s}_t(y_{t-1}) := \nabla s_t(y_{t-1}) + w_t$.
3) *Noisy+Bias*: a noisy prediction model plus a bias term $\nabla \hat{s}_t(y_{t-1}) := \nabla s_t(y_{t-1}) + w_t - 1$.
4) *Previous*: a prediction model that uses the previous cost gradient $\nabla \hat{s}_t(y_{t-1}) := \nabla s_{t-1}(y_{t-1})$.
5) *Random*: a random prediction model $\nabla \hat{s}_t(y_{t-1}) := w_t$.

where $w_t \sim \mathcal{N}(0, 0.5I)$. As a benchmark, we use the dynamic mirror descent (DMD) algorithm of Hall and Willett [23] with a constant step-size $\eta = 1$ and a dynamic version of Algorithm OptMD, which also uses the dynamical model Φ_t to update the y_t variable. We refer to this algorithm as dynamic OptMD.

Denote the regrets of Algorithm OptDCMD, the DMD algorithm and the dynamic OptMD by $\mathbf{Reg}_t^d$(OptDCMD), $\mathbf{Reg}_t^d$(DMD) and $\mathbf{Reg}_t^d$(d-OptMD), respectively. The experiments are repeated 100 times, and for each experiment, a new trajectory $\{u_t\}_{t=1}^T$ was generated. The shaded areas correspond

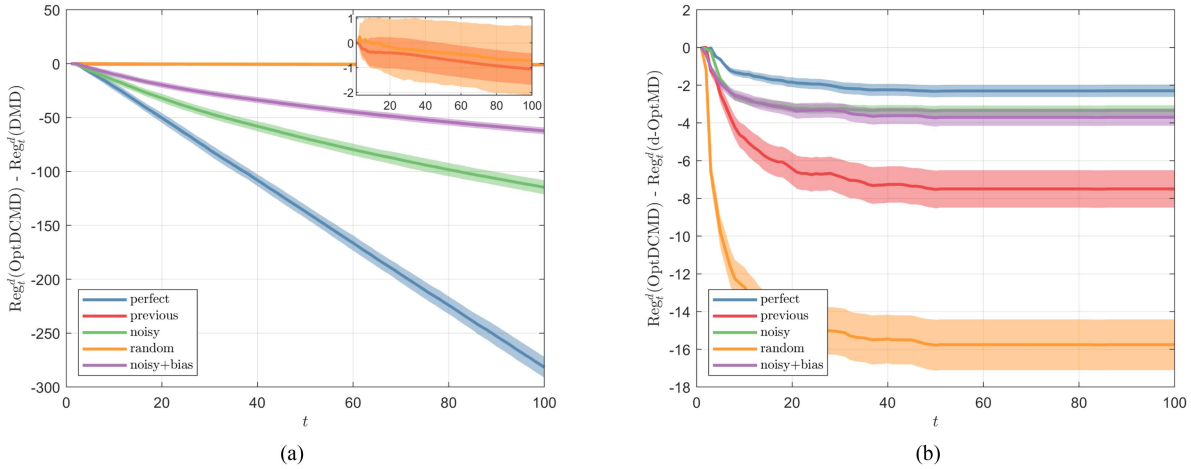

Fig. 1. Regret difference between the DMD algorithm, the dynamic version of Algorithm OptMD and Algorithm OptDCMD, for different gradient prediction models. (a) OptDCMD versus DMD. (b) OptDCMD versus dynamic OptMD.

to one standard deviation for Fig. 1(a) and 0.1 times one standard deviation for Fig. 1(b). Fig. 1(a) depicts the difference $\mathbf{Reg}_t^d$(OptDCMD)-$\mathbf{Reg}_t^d$(DMD). One can observe that all the models that use some kind of information about future gradients (perfect, noisy, noisy+bias) were able to perform better than the benchmark. This shows that indeed Algorithm OptDCMD was able to exploit predictive information about the problem. Moreover, model previous and random also perform better than the benchmark on average, showing the robustness of our algorithm against inaccurate gradient predictions. Fig. 1(a) depicts the difference $\mathbf{Reg}_t^d$(OptDCMD)-$\mathbf{Reg}_t^d$(d-OptMD). As can be seen, Algorithm OptDCMD performs better than the benchmark for all predictions models, illustrating the advantage of the composite updates Algorithm OptDCMD compared with Algorithm OptMD.

B. Portfolio Selection

In this section, we apply the result of Theorem 2.5 in a portfolio selection problem. Suppose that an investor (or the Player) has n assets in a Market (or Nature). Let the Player's action x be a probability distribution over n assets. The action set $\mathcal{X}$ is thus $\Delta_n := \{x \in \mathbb{R}^n : x(i) \geq 0, \sum_{i=1}^n x(i) = 1\}$. Let the return of an asset at round t be the ratio of the value of the asset between rounds t and $t+1$. At round t, Nature chooses a strictly positive return vector $r_t \in \mathbb{R}^n_{>0}$ such that each entry of r_t corresponds to the return of an asset. The Player's wealth ratio between rounds t and $t+1$ is $\langle r_t, x_t \rangle$. Let the Player's gain at round t be $\log(\langle r_t, x_t \rangle)$. In a game of T rounds, the goal of the Player is to maximize $\sum_{t=1}^T \log(\langle r_t, x_t \rangle)$ or, equivalently, to minimize $\sum_{t=1}^T -\log(\langle r_t, x_t \rangle)$. Hence, we have $f_t(x) = -\log(\langle r_t, x \rangle)$ and $\nabla f_t(x) = -r_t/\langle r_t, x \rangle$, for all $x \in \mathcal{X}$.[4] Notice that in this scenario, there is no nonsmooth component in the cost f_t, and Algorithm OptCMD reduces to Algorithm OptMD.

We assume that the Player has prediction models of the return vector r_t, denoted by $\hat{r}_t$. Thus, in light of the approaches proposed in this article, we define

$$\nabla \hat{f}_t(y_{t-1}) := -\frac{\hat{r}_t}{\langle \hat{r}_t, y_{t-1} \rangle}. \tag{42}$$

[4]See [1] for a more detailed description of this problem.

In what follows, we show how the Player can employ Algorithm OptMD to decide its action sequence $\{x_t\}_{t=1}^T$ considering the static regret (1). Since the costs are convex, the Player uses the step-size rule of Theorem 2.5 in Algorithm OptMD (with $V_t = 0$). We assume the return of each asset at each time t is bounded as $r_{\min} \leq r_t \leq r_{\max}$ (component-wise). By assuming $r_{\min} = 0.5$ and $r_{\max} = 1.5$, we can set the smoothness parameter $\beta = 9$.

Since Δ_n is the n-dimensional simplex, we let $h(x)$ be the *negative entropy* function $\sum_{i=1}^n x(i)\log(x(i))$. Observe that h is 1-strongly convex w.r.t. $\|\cdot\|_1$ [52]. We consider the following prediction models for the returns vector:

1) *MA(k):* a Moving Average prediction model model $\tilde{r}_t := \frac{1}{k}\sum_{i=1}^k r_{t-i}$.
2) *Previous:* a model that uses the previous return vector as its prediction $\tilde{r}_t := r_{t-1}$.
3) *Noisy:* a noisy, unbiased predictor model of the true returns vector $\tilde{r}_t := r_t + v_t$, where $v_t \sim \mathcal{N}(0, 0.3)$.
4) *Random:* a random predictor, where the entries of $\tilde{r}_t$ are chosen uniformly between $r_{\min}$ and $r_{\max}$.
5) *Recursivels(k):* for each stock, we have a prediction model of the form $\tilde{r}_t = w_1 r_{t-1} + w_2 r_{t-2} + \cdots + w_k r_{t-k} + w_{k+1}$, where the weights $w_1, \ldots, w_{k+1}$ are updated online, using a recursive least squares algorithm.

However, instead of using the output of these models directly in (42), we will use $\hat{r}_t = g(\tilde{r}_t)$. The function $g(r)$ is defined as

$$g(r) := \begin{cases} r_{\max} & \text{if } r > 1 \\ 1 & \text{if } r = 1 \\ r_{\min} & \text{if } r < 1 \end{cases}$$

and is applied component-wise for vector inputs. The interpretation behind passing the predictions $\tilde{r}_t$ through g is that, instead of using the exact predictions given by our models, we use $\tilde{r}_t$ only as an indication if a given stock is predicted to increase or decrease its value in the next round.

To simulate a stock market, we use six real-world datasets: NYSE(O), NYSE(N), DJIA, TSE, SP500, and MSCI. A detailed description of these datasets can be found in [53]. Let the number of assets of each dataset be N. As a benchmark of each experiment, we employ the *Constant Uniform Portfolio* (CUP) strategy, that is, a Player that chooses $x_t = [1/N, \ldots, 1/N]$, for

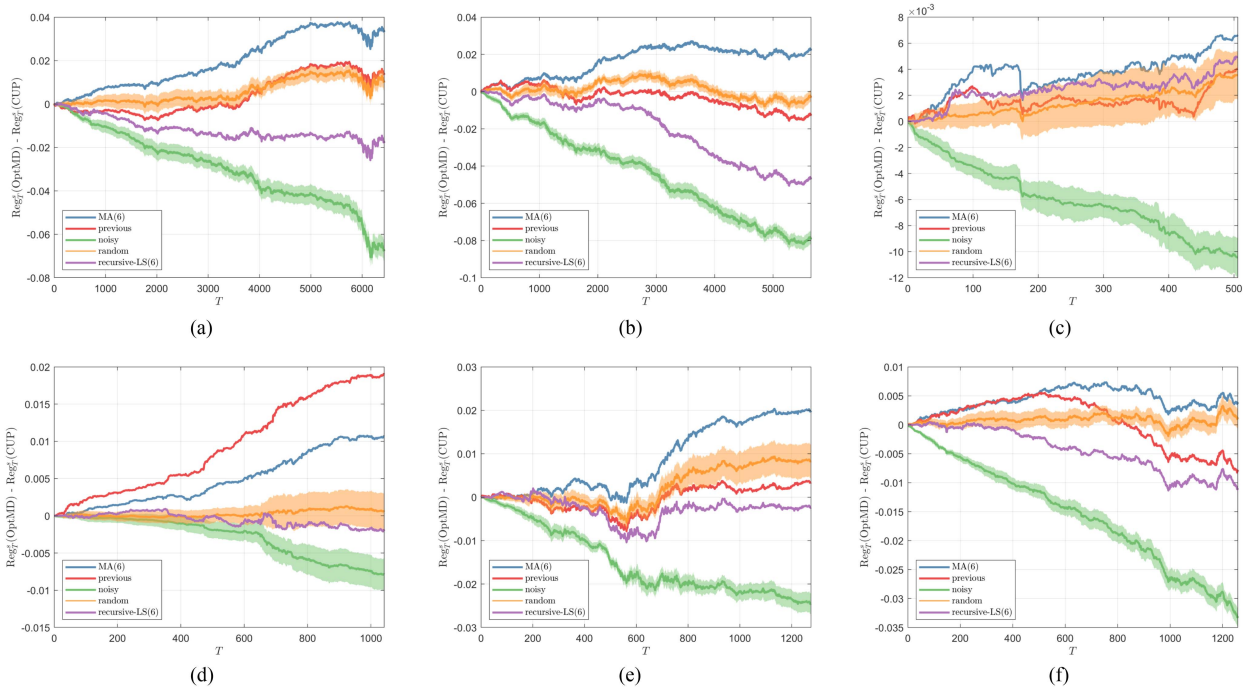

Fig. 2. Algorithm OptMD applied to the portfolio selection problem. (a) NYSE(N). (b) NYSE(O). (c) DJIA. (d) MSCI. (e) SP500. (f) TSE.

all $t \in [T]$. For the datasets considered in this experiment, the CUP strategy performed better than the Algorithm OMD, for any $\eta_t > 0$ and $x_0 = [1/N, \ldots, 1/N]$.

Denote the regrets of Algorithm OptMD and CUP strategies by $\mathbf{Reg}_T^s(\text{OptMD})$ and $\mathbf{Reg}_T^s(\text{CUP})$, respectively. Fig. 2 depicts the difference $\mathbf{Reg}_t^s(\text{OptMD}) - \mathbf{Reg}_t^s(\text{CUP})$ for each considered dataset. The experiment was repeated 10 times and the shaded areas correspond to one standard deviation. As expected, for all datasets, the *noisy* model achieved the best performance, since it uses information of r_t in the prediction $\hat{r}_t$. More interestingly, we notice that for all datasets except DJIA, the *recursiveLS(6)* prediction model performed better than all other models. Moreover, this model also performed better than the CUP benchmark strategy. In other words, at time t, we were able to generate and exploit the predictive information about the return of each stock, using only information available up to time $t-1$. Another interesting conclusion we can draw from Fig. 2 is that, in general, using either the previous return or a simple moving average as predictions lead to poor performance for the algorithm. Finally, when using the *random* models (i.e., gradient predictions uncorrelated with the true gradients), Algorithm OptMD performed generally similarly to the CUP benchmark strategy. This indicates that our approach can also be robust to bad gradient predictions (see Remark 2.7).

REFERENCES

[1] E. Hazan, "Introduction to online convex optimization," *Foundations Trends Optim.*, vol. 2, no. 3–4, pp. 157–325, 2016.

[2] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[3] S. Shalev-Shwartz et al., "Online learning and online convex optimization," *Foundations Trends® in Mach. Learn.*, vol. 4, no. 2, pp. 107–194, 2012.

[4] S. Bubeck, "Lecture Notes: Introduction to online optimization," Princeton, NJ, USA: Princeton University - Department of Operations Research and Financial Engineering, 2011. [Online]. Available: http://sbubeck.com/BubeckLectureNotes.pdf

[5] S. Shalev-Shwartz and Y. Singer, "Logarithmic regret algorithms for strongly convex repeated games," The Hebrew University, 2007. [Online]. Available: http://luthuli.cs.uiuc.edu/~daf/courses/optimization/Papers/10.1.1.116.9046.pdf}. Last visited on 2023/01/20

[6] J. Abernethy, P. Bartlett, A. Rakhlin, and A. Tewari, "Optimal strategies and minimax lower bounds for online convex games," in *Proc. Conf. Learn. Theory*, 2008, pp. 415–423.

[7] N. Agarwal, B. Bullins, E. Hazan, S. Kakade, and K. Singh, "Online control with adversarial disturbances," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 111–119.

[8] N. Agarwal, E. Hazan, and K. Singh, "Logarithmic regret for online control," *Adv. Neural Inf. Process. Syst.*, 2019.

[9] E. Hazan, S. Kakade, and K. Singh, "The nonstochastic control problem," in *Proc. 31st Int. Conf. Algorithmic Learn. Theory*, 2020, pp. 408–421.

[10] N. Wagener, C.-A. Cheng, J. Sacks, and B. Boots, "An online learning approach to model predictive control," *Proc. Robotics, Sci. Syst.*, 2019.

[11] M. Nagahara, D. E. Quevedo, and J. /Ostergaard, "Sparse packetized predictive control for networked control over erasure channels," *IEEE Trans. Autom. Control*, vol. 59, no. 7, pp. 1899–1905, Jul. 2014.

[12] M. Nagahara, D. E. Quevedo, and D. Nežić, "Maximum hands-off control: A paradigm of control effort minimization," *IEEE Trans. Autom. Control*, vol. 61, no. 3, pp. 735–747, Mar. 2016.

[13] A. Rakhlin and K. Sridharan, "Online learning with predictable sequences," in *Proc. Conf. Learn. Theory*, 2013, pp. 993–1019.

[14] S. Rakhlin and K. Sridharan, "Optimization, learning, and games with predictable sequences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3066–3074.

[15] N. Ho-Nguyen and F. Kılınç-Karzan, "Exploiting problem structure in optimization under uncertainty via online convex optimization," *Math. Program.*, vol. 177, no. 1–2, pp. 113–147, 2019.

[16] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan, "Online optimization: Competing with dynamic comparators," in *Proc. 18th Int. Conf. Artif. Intell. Statist.*, 2015, pp. 398–406.

[17] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Berlin, Germany: Springer, 2004.

[18] T. Yang, M. Mahdavi, R. Jin, and S. Zhu, "Regret bounded by gradual variation for online convex optimization," *Mach. Learn.*, vol. 95, no. 2, pp. 183–223, 2014.

[19] O. Besbes, Y. Gur, and A. Zeevi, "Non-stationary stochastic optimization," *Operations Res.*, vol. 63, no. 5, pp. 1227–1244, 2015.

[20] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 647–662, Jun. 2015.

[21] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 928–936.
[22] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro, "Online optimization in dynamic environments: Improved regret rates for strongly convex problems," in *Proc. IEEE 55th Conf. Decis. Control*, 2016, pp. 7195–7201.
[23] E. Hall and R. Willett, "Dynamical models and tracking regret in online convex programming," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 579–587.
[24] L. Zhang, S. Lu, and Z.-H. Zhou, "Adaptive online learning in dynamic environments," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1323–1333.
[25] N. Campolongo and F. Orabona, "A closer look at temporal variability in dynamic online learning," 2021, *arXiv:2102.07666*.
[26] J. C. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari, "Composite Objective Mirror Descent.," in *Colt*, vol. 10. Citeseer, 2010, pp. 14–26.
[27] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.
[28] A. Beck, *First-Order Methods in Optimization*. Philadelphia, PA, USA: SIAM, 2017.
[29] J. Kivinen and M. K. Warmuth, "Exponentiated gradient versus gradient descent for linear predictors," *Inf. Comput.*, vol. 132, no. 1, pp. 1–63, 1997.
[30] B. Kulis and P. L. Bartlett, "Implicit online learning," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 575–582.
[31] N. Campolongo and F. Orabona, "Temporal variability in implicit online learning," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 12377–12387, 2020.
[32] H. B. McMahan, "A unified view of regularized dual averaging and mirror descent with implicit updates," 2010, *arXiv:1009.3240*.
[33] C. Song, J. Liu, H. Liu, Y. Jiang, and T. Zhang, "Fully implicit online learning," 2018, *arXiv:1809.09350*.
[34] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," *IEEE Trans. Autom. Control*, vol. 63, no. 3, pp. 714–725, Mar. 2018.
[35] N. Chen, A. Agarwal, A. Wierman, S. Barman, and L. L. Andrew, "Online convex optimization using predictions," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, 2015, pp. 191–204.
[36] N. Chen, J. Comden, Z. Liu, A. Gandhi, and A. Wierman, "Using predictions in online optimization: Looking forward with an eye on the past," *ACM SIGMETRICS Perform. Eval. Rev.*, New York, NY, USA: ACM, vol. 44, no. 1, pp. 193–206, 2016.
[37] Y. Lin, G. Goel, and A. Wierman, "Online optimization with predictions and non-convex losses," in *Proc. ACM Meas. Anal. Comput. Syst.*, New York, NY, USA: ACM, vol. 4, no. 1, pp. 1–32, 2020.
[38] Y. Li and N. Li, "Leveraging predictions in smoothed online convex optimization via gradient-based algorithms," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 14520–14531, 2020.
[39] Y. Li, G. Qu, and N. Li, "Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit," *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4761–4768, Oct. 2021.
[40] Y. Li, X. Chen, and N. Li, "Online optimal control with linear dynamics and predictions: Algorithms and regret analysis," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/6d1e481bdcf159961818823e652a7725-Abstract.html
[41] C. Yu, G. Shi, S.-J. Chung, Y. Yue, and A. Wierman, "The power of predictions in online control," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 1994–2004, 2020.
[42] T. Li et al., "Robustness and consistency in linear quadratic control with untrusted predictions," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 6, no. 1, pp. 1–35, 2022.
[43] O. Dekel, A. Flajolet, N. Haghtalab, and P. Jaillet, "Online learning with a hint," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5299–5308.
[44] A. Bhaskara, A. Cutkosky, R. Kumar, and M. Purohit, "Online learning with imperfect hints," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 822–831.
[45] A. Lesage-Landry, I. Shames, and J. A. Taylor, "Predictive online convex optimization," *Automatica*, vol. 113, 2020, Art. no. 108771.
[46] R. J. Ravier, A. R. Calderbank, and V. Tarokh, "Prediction in online convex optimization for parametrizable objective functions," in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 2455–2460.
[47] T.-J. Chang and S. Shahrampour, "On online optimization: Dynamic regret analysis of strongly convex and smooth problems," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 8, 2021, pp. 6966–6973.
[48] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Mach. Learn.*, vol. 69, no. 2, pp. 169–192, 2007.
[49] P. Zattoni Scroccaro, A. S. Kolarijani, and P. Mohajerin Esfahani, "Adaptive composite online optimization: Predictions in static and dynamic environments," 2022, *arXiv:2205.00446*.
[50] M. Mohri and S. Yang, "Accelerating online convex optimization via adaptive prediction," *Artif. Intell. Statist.*, pp. 848–856, 2016.
[51] Y. Nesterov, *Lectures on Convex Optimization*, vol. 137. Springer: Berlin, Germany, 2018.
[52] S. Bubeck, *Convex Optimization: Algorithms and Complexity, Ser. Foundations and Trends in Machine Learning*. Norwell, MA, USA: Now Publishers, 2015.
[53] B. Li, S. C. Hoi, D. Sahoo, and Z.-Y. Liu, "Moving average reversion strategy for on-line portfolio selection," *Artif. Intell.*, vol. 222, pp. 104–123, 2015.

Pedro Zattoni Scroccaro received the B.Eng degree in control and automation from the Federal University of Technology - Parana, Curitiba, Brazil, in 2017, and the M.Sc. degree in systems and control from the Delft University of Technology, Delft, The Netherlands, in 2020.

He is currently a Ph.D. Researcher with the Delft University of Technology, Delft, The Netherlands. His current research interests include online convex optimization, inverse optimization, and optimization algorithms, with applications to control and transportation systems.

Arman Sharifi Kolarijani received the M.Sc. degree in applied mechanics from the Sharif University of Technology, Tehran, Iran, in 2011, and the M.Sc. and Ph.D. degrees in systems and control from the Delft University of Technology, Delft, The Netherlands, in 2014 and 2019, respectively.

He is currently a Postdoctoral Researcher with the Delft University of Technology. From 2008 to 2012, he was an Associate Researcher with Actuators Lab, Khaj-e-Nasir University of Technology, Tehran, Iran, working on the design and construction of rehabilitation device for lower limbs. His current research interests include optimization algorithms, optimal control, and symbolic verification of networked control systems.

Peyman Mohajerin Esfahani received the B.Sc. and M.Sc. degrees in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2005 and 2007, respectively, and the Ph.D. degree in control from ETH Zurich, Zurich, Switzerland, in 2014. He joined TU Delft in October 2016 as an Assistant Professor. He is currently an Associate Professor with the Delft Center for Systems and Control and a co-Director of the Delft-AI Energy Lab. Prior to joining TU Delft, he held several research appointments at EPFL, ETH Zurich, and MIT between 2014 and 2016. His research interests include theoretical and practical aspects of decision-making problems in uncertain and dynamic environments, with applications to control and security of large-scale and distributed systems.

He is an Associate Editor of *Operations Research*, *Transactions on Automatic Control, and Open Journal of Mathematical Optimization*. He was one of the three finalists for the Young Researcher Prize in Continuous Optimization awarded by the Mathematical Optimization Society in 2016, and a recipient of the 2016 George S. Axelby Outstanding Paper Award from the IEEE Control Systems Society. He received the ERC Starting Grant and the INFORMS Frederick W. Lanchester Prize in 2020. He is the recipient of the 2022 European Control Award.