A Unified Framework for Analyzing Meta-algorithms in Online Convex Optimization

Mohammad Pedramfar
Mila - Quebec AI Institute/McGill University
Montreal, QC, Canada
mohammad.pedramfar@mila.quebec

Vaneet Aggarwal
Purdue University
West Lafayette, IN, USA
vaneet@purdue.edu

ABSTRACT

In this paper, we analyze the problem of online convex optimization in different settings, including different feedback types (full-information/semi-bandit/bandit/etc) in either stochastic or non-stochastic setting and different notions of regret (static adversarial regret/dynamic regret/adaptive regret). This is done through a framework which allows us to systematically propose and analyze meta-algorithms for the various settings described above. We show that any algorithm for online linear optimization with deterministic gradient feedback against fully adaptive adversaries is an algorithm for online convex optimization. We also show that any such algorithm that requires full-information feedback may be transformed to an algorithm with semi-bandit feedback with comparable regret bound. We further show that algorithms that are designed for fully adaptive adversaries using deterministic semi-bandit feedback can obtain similar bounds using only stochastic semi-bandit feedback when facing oblivious adversaries. We use this to describe general meta-algorithms to convert first order algorithms to zeroth order algorithms with comparable regret bounds. Our framework allows us to analyze online optimization in various settings, recovers several results in the literature with a simplified proof technique, and provides new results.

CCS CONCEPTS

• **Theory of computation → Design and analysis of algorithms**.

KEYWORDS

Online Convex Optimization; Meta-algorithms; Regret Minimization; Bandit Feedback; Gradient Estimation

ACM Reference Format:
Mohammad Pedramfar and Vaneet Aggarwal. 2026. A Unified Framework for Analyzing Meta-algorithms in Online Convex Optimization. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 8 pages. https://doi.org/10.65109/

1 INTRODUCTION

Online optimization problems represent a class of problems where the decision-maker, referred to as an agent, aims to make sequential decisions in the face of an adversary with incomplete information [10, 24]. This setting mirrors a repeated game, where the agent and the adversary engage in a strategic interplay over a finite time horizon, commonly denoted as T. The dynamics of this game involve the agent's decision-making process, the adversary's strategy, and the information exchange through a query oracle.

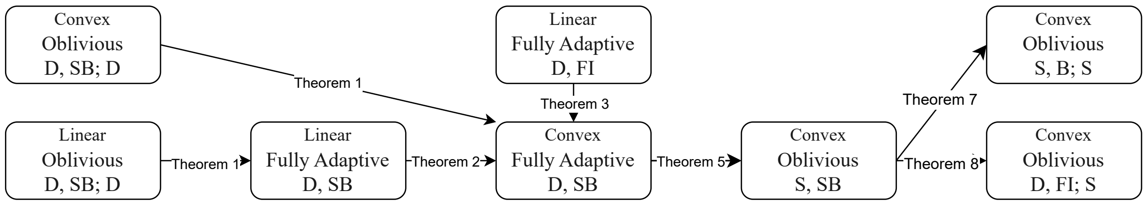
In this paper, we present a comprehensive approach to solve online convex optimization problems, addressing scenarios where the adversary can choose a function from the class of μ−strongly convex functions (or convex functions for $\mu = 0$) at each time step. Our approach encompasses different feedback types, including bandit, semi-bandit and full-information feedback in each iteration. It also encompasses different types of regret, such as adversarial regret, stochastic regret, and various forms of non-stationary regret. While the problem has been studied in many special cases, this comprehensive approach sheds light on relations between different cases and the powers and limitations of some of the approaches used in different settings.

One of our key contribution lies in establishing that any semi-bandit feedback online linear (or quadratic) optimization algorithm for fully adaptive adversary is also an online convex (or strongly convex) optimization algorithm. While the above result is for semi-bandit feedback, we then show that in online convex optimization for fully adaptive adversary, semi-bandit feedback is generally enough and more information is not needed. Further, we show that algorithms that are designed for fully adaptive adversaries using deterministic semi-bandit feedback can obtain similar bounds using only stochastic semi-bandit feedback when facing oblivious adversaries. Finally, we introduce meta algorithms, based on the variant of classical result of [6, 26], that convert algorithms for semi-bandit feedback, such as the ones mentioned above, to algorithms for bandit or zeroth-order feedback. In addition to recovering many results in the literature with simplified proofs, we give new static regret guarantees for online strongly convex optimization and adaptive regret guarantees for convex optimization with deterministic zeroth order feedback.

The key results of this paper are encapsulated in Figure 1, where each arrow represents a procedure or meta-algorithm that transforms an algorithm from one setting to another. The primary contribution is three-fold:
(I) We provide a framework that simplifies analysis of meta-algorithms in online learning.
(II) Using this framework, we formulate several meta-algorithms that facilitate the conversion of algorithms across various settings.
(III) In particular, we convert several insights in the literature into precise theorems, for example the idea of using 1-point gradient estimator in online optimization is formalized in Meta-algorithms 2 and 3 and Theorems 6 and 7. Similarly, the insight that "knowing

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/

Figure 1: Summary of the main results



Each arrow describes a procedure or meta-algorithm to convert an algorithm from one setting into another. The first line describes the function class - linear/convex. The second lines specifies the type of the adversary - fully adaptive or oblivious. The last line starts with D or S, to denote deterministic or stochastic oracles. After that, SB is used to denotes semi-bandit, B to denote bandit and FI to denote full information feedback. Finally, the last D or S, if exists, is used to denotes deterministic or stochastic algorithm.

the value of each loss function at two points is almost as useful as knowing the value of each function everywhere" (See [2]) is formalized in Meta-algorithm 4 and Theorem 8.

This framework not only simplifies proof analysis for numerous existing problems in the field (see Section 11) but also provides a tool for easier extension of multiple results in the literature. In the following, we go over some of the main meta-algorithms that we discuss in this paper.

(i) The first key result in this paper is Theorem 2. This result shows that any algorithm for online linear optimization with deterministic subgradient feedback results in a comparable regret when the function class is replaced by a class of convex functions. This is the key result that allows us to go from online linear optimization to convex optimization in Figure 1 with semi-bandit feedback. We note that this result recovers the result for online convex game in [1], where the authors analyzed online convex optimization with deterministic gradient feedback and fully adaptive adversary and showed that the min-max regret bounds of online convex optimization is equal to that of online linear optimization (or online quadratic optimization is all convex functions are μ-strongly convex).

While the above result is the key for going from online linear to online convex optimization, the main issue for its applicability is that it requires semi-bandit feedback. In the remaining results, we provide meta-algorithms to increase the applicability of this result.

(ii) The second key result, given in Theorem 3, is that for any full-information algorithm for online convex optimization with fully adaptive adversary, there is a corresponding algorithm with regret bound of the same order which only requires semi-bandit feedback. This allows us to get results for semi-bandit feedback from that of full information. This combined with Theorem 2 allows us to go from any full-information algorithm for online linear optimization with fully adaptive adversary to an algorithm for online convex optimization with semi-bandit feedback and fully adaptive adversary. This combination has been marked as Theorem 3 in Figure 1.

(iii) We note that the above results assume that the feedback oracle is deterministic. In the next key result, given in Theorem 5, we relax this assumption. We show that any online optimization algorithm for fully adaptive adversaries using deterministic semi-bandit feedback obtains similar regret bounds when facing an oblivious adversary using only stochastic semi-bandit feedback. Thus, this result allows us to get the results for online convex optimization with stochastic semi-bandit feedback from the results with deterministic semi-bandit feedback, as can be seen in Fig. 1. This result gives Theorem 6.5 in [10] as a special case. Further, this allows us to have a result that generalizes both adversarial regret and stochastic regret, as they are commonly defined in the literature.

(iv) The above discussion was mostly focused on full information and semi-bandit feedback settings. However, in many practical setups, we only have access to zeroth order or bandit feedback. In the next key result, given in Theorems 6-7, we use the smoothing trick [6, 18, 21], in order to estimate the gradient from the value oracle. Using this estimated gradient, the results for zeroth order and bandit feedback are derived from that for first order and semi-bandit feedback, respectively. These results apply to any optimization problem, not necessarily convex. This approach recovers the results in [6] with the base algorithm of Online Gradient Descent (`OGD`). In particular, this approach also matches the SOTA dynamic and adaptive regret bounds for online convex optimization with bandit feedback [2, 7, 31] with a simplified proof.

(v) In the above results, we assumed that we only have access to an unbiased estimate of the zeroth order (or bandit) feedback. In [2], the setting of multi-point bandit feedback is introduced. Even though their setting and notion of regret is different from our work, one insight is relevant. Their analysis suggests that, in some cases, if we have access to exact values of the functions, knowing the value of each loss function at two points is almost as useful as knowing the value of each function everywhere. We formalize this insight by designing a meta-algorithm based on the two-point gradient estimator [2, 26] that allows us to convert algorithms for stochastic first order feedback into algorithms for deterministic zeroth order feedback with the same order of regret bound (Theorem 8). Note that this result holds for convex and non-convex optimization as it only relies on the function class being Lipschitz. This result is shown to recover the result for dynamic regret for Improved Ader [31], while achieving two new results for adaptive regret for convex function of $O(\sqrt{T})$ and static regret for strongly convex function of $O(\log T)$.

Our work sheds light on the relation between different problems, algorithms and techniques used in online optimization and provides general meta-algorithms that allow conversion between different problem settings. For instance, we can use the results for a deterministic algorithm for online linear optimization with oblivious adversaries and deterministic semi-bandit feedback to that for a stochastic algorithm for online convex optimization with oblivious adversaries and stochastic bandit feedback using the different arrows as in Fig. 1. The key technical novelty is the recognition that many results, previously understood only in very limited contexts, actually apply more broadly. This broader applicability not only simplifies the analysis of numerous existing results within the field but also enables the derivation of multiple new results. For instance, results in [1, 2, 6, 10, 26, 31] as mentioned above can be demonstrated as corollaries of these more generalized results. Further, we provide new results for deterministic zeroth-order feedback case, showing that adaptive regret for convex function is $O(\sqrt{T})$ and static regret for strongly convex function is $O(\log T)$.

Related works: The term "online convex optimization" was first defined in [33], where Online Gradient Descent (OGD) was proposed, while this setting was first introduced in [8]. The Follow-The-Perturbed-Leader (FTPL) for online linear optimization was introduced by [15], inspired by Follow-The-Leader algorithm originally suggested by [9]. Follow-The-Regularized-Leader (FTRL) in the context of online convex optimization was coined in [23, 25]. The equivalence of FTRL and Online Mirror Descent (OMD) was observed by [11]. Non-stationary measures of regret have been studied in [3, 5, 7, 13, 16, 27, 29–33] and many other papers. We refer the reader to surveys [24] and [10] for more details.

2 BACKGROUND AND NOTATIONS

For a set $\mathcal{S} \subseteq \mathbb{R}^d$, we define its *affine hull* $\text{aff}(\mathcal{S})$ to be the set of all points of the form $\sum_{i=1}^m \alpha_i \mathbf{x}_i$ for integer $m \geq 1$, points $(\mathbf{x}_i)_{i=1}^m$ in $\mathcal{S}$, and real numbers $(\alpha_i)_{i=1}^m$. The *relative interior* of $\mathcal{S}$ is defined as

$$\text{relint}(\mathcal{S}) := \{\mathbf{x} \in \mathcal{S} \mid \exists r > 0, \mathbb{B}_r(\mathbf{x}) \cap \text{aff}(\mathcal{S}) \subseteq \mathcal{S}\}.$$

We use $\mathbb{B}_r(\mathcal{S})$ to denote the union of balls of radius r centered at $\mathcal{S}$ and use $\text{diam}(\mathcal{S}) := \sup_{\mathbf{x},\mathbf{y}\in\mathcal{S}} \|\mathbf{x}-\mathbf{y}\|$ to denote the diameter of $\mathcal{S}$. A *function class* is simply a set of real-valued functions. Given a set $\mathcal{S}$, a *function class* over $\mathcal{S}$ is simply a subset of all real-valued functions on $\mathcal{S}$. Given a real number $\mu \geq 0$ and a set $\mathcal{S}$, we use $\mathbf{Q}_\mu$ to denote the class of functions of the form $q_{\mathbf{x},\mathbf{o}}(\mathbf{y}) = \langle \mathbf{o}, \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}\|^2$, where $\mathbf{o} \in \mathbb{R}^d$ and $\mathbf{x} \in \mathcal{S}$. A set $\mathcal{K} \subseteq \mathbb{R}^d$ is called a *convex set* if for all $\mathbf{x}, \mathbf{y} \in \mathcal{K}$ and $\alpha \in [0, 1]$, we have $\alpha\mathbf{x} + (1-\alpha)\mathbf{y} \in \mathcal{K}$. Given a convex set $\mathcal{K}$, a function $f : \mathcal{K} \to \mathbb{R}$ is called *convex* if, for all $\mathbf{x}, \mathbf{y} \in \mathcal{K}$ and $\alpha \in [0, 1]$, we have $f(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1 - \alpha)f(\mathbf{y})$. All convex functions are continuous on any point in the relative interior of their domains, but not necessarily at their relative boundary. In this work, we will only focus on continuous functions. If $\mathbf{x} \in \text{relint}(\mathcal{K})$ and f is convex and is differentiable at $\mathbf{x}$, then we have $f(\mathbf{y}) - f(\mathbf{x}) \geq \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle$, for all $\mathbf{y} \in \mathcal{K}$. A vector $\mathbf{o} \in \mathbb{R}^d$ is called a *subgradient* of f at $\mathbf{x}$ if $f(\mathbf{y}) - f(\mathbf{x}) \geq \langle \mathbf{o}, \mathbf{y} - \mathbf{x} \rangle$, for all $\mathbf{y} \in \mathcal{K}$. More generally, given $\mu \geq 0$, we say a vector $\mathbf{o} \in \mathbb{R}^d$ is a μ*-subgradient* of f at $\mathbf{x}$ if $f(\mathbf{y}) - f(\mathbf{x}) \geq \langle \mathbf{o}, \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}\|^2$, for all $\mathbf{y} \in \mathcal{K}$. Given a convex set $\mathcal{K}$, a Lipschitz continuous function $f : \mathcal{K} \to \mathbb{R}$ is convex (resp. μ-strongly convex) if and only if it has a (μ-)subgradient at all points $\mathbf{x} \in \mathcal{K}$. We use $\nabla^* f$ to denote the set of μ-subgradients of f.

3 PROBLEM SETUP

Online optimization problems can be formalized as a repeated game between an agent and an adversary. The game lasts for T rounds on a convex domain $\mathcal{K}$ where T and $\mathcal{K}$ are known to both players. In t-th round, the agent chooses an action $\mathbf{x}_t$ from an action set $\mathcal{K} \subseteq \mathbb{R}^d$, then the adversary chooses a loss function $f_t \in \mathbf{F}$ and a query oracle for the function f_t. Then, for $1 \leq i \leq k_t$, the agent chooses a points $\mathbf{y}_{t,i}$ and receives the output of the query oracle. Here k_t denotes the total number of queries made by the agent at time-step t, which may or may not be known in advance.

To be more precise, an agent consists of a tuple $(\Omega^{\mathcal{A}}, \mathcal{A}^{\text{action}}, \mathcal{A}^{\text{query}})$, where $\Omega^{\mathcal{A}}$ is a probability space that captures all the randomness of $\mathcal{A}$. We assume that, before the first action, the agent samples $\omega \in \Omega$. The next element in the tuple, $\mathcal{A}^{\text{action}} = (\mathcal{A}_1^{\text{action}}, \cdots, \mathcal{A}_T^{\text{action}})$ is a sequence of functions such that $\mathcal{A}_t$ that maps the history $\Omega^{\mathcal{A}} \times \mathcal{K}^{t-1} \times \prod_{s=1}^{t-1}(\mathcal{K} \times O)^{k_s}$ to $\mathbf{x}_t \in \mathcal{K}$ where we use O to denote range of the query oracle. The last element in the tuple, $\mathcal{A}^{\text{query}}$, is the query policy. For each $1 \leq t \leq T$ and $1 \leq i \leq k_t$, $\mathcal{A}_{t,i}^{\text{query}} : \Omega^{\mathcal{A}} \times \mathcal{K}^t \times \prod_{s=1}^{t-1}(\mathcal{K} \times O)^{k_s} \times (\mathcal{K} \times O)^{i-1}$ is a function that, given previous actions and observations, either selects a point $\mathbf{y}_t^i \in \mathcal{K}$, i.e., query, or signals that the query policy at this time-step is terminated. We may drop ω as one of the inputs of the above functions when there is no ambiguity. We say the agent query function is *trivial* if $k_t = 1$ and $\mathbf{y}_{t,1} = \mathbf{x}_t$ for all $1 \leq t \leq T$. In this case, we simplify the notation and use the notation $\mathcal{A} = \mathcal{A}^{\text{action}} = (\mathcal{A}_1, \cdots, \mathcal{A}_T)$ to denote the agent action functions and assume that the domain of $\mathcal{A}_t$ is $\Omega^{\mathcal{A}} \times (\mathcal{K} \times O)^{t-1}$.

A query oracle is a function that provides the observation to the agent. Formally, a query oracle for a function f is a map Q defined on $\mathcal{K}$ such that for each $\mathbf{x} \in \mathcal{K}$, the $Q(\mathbf{x})$ is a random variable taking value in the observation space O. The query oracle is called a *stochastic value oracle* or *stochastic zeroth order oracle* if $O = \mathbb{R}$ and $f(\mathbf{x}) = \mathbb{E}[Q(\mathbf{x})]$. Similarly, it is called a *stochastic (sub)gradient oracle* or *stochastic first order oracle* if $O = \mathbb{R}^d$ and $\mathbb{E}[Q(\mathbf{x})]$ is a (sub)gradient of f at $\mathbf{x}$. In all cases, if the random variable takes a single value with probability one, we refer to it as a *deterministic* oracle. Note that, given a function, there is at most a single deterministic gradient oracle, but there may be many deterministic subgradient oracles. We will use ∇ to denote the deterministic gradient oracle. We say an oracle is bounded by B if its output is always within the Euclidean ball of radius B centered at the origin. We say the agent takes *semi-bandit feedback* if the oracle is first-order and the agent query function is trivial. Similarly, it takes *bandit feedback* if the oracle is zeroth-order and the agent query function is trivial [1]. If the agent query function is non-trivial, then we say the agent requires *full-information feedback.*

An adversary Adv is a set such that each element $\mathcal{B} \in \text{Adv}$, referred to as a *realized adversary*, is a sequence $(\mathcal{B}_1, \cdots, \mathcal{B}_T)$ of functions where each $\mathcal{B}_t$ maps a tuple $(\mathbf{x}_1, \cdots, \mathbf{x}_t) \in \mathcal{K}^t$ to a tuple (f_t, Q_t) where $f_t \in \mathbf{F}$ and Q_t is a query oracle for f_t [2]. We say an adversary Adv is *oblivious* if for any realization $\mathcal{B} = (\mathcal{B}_1, \cdots, \mathcal{B}_T)$, all functions $\mathcal{B}_t$ are constant, i.e., they are independent of $(\mathbf{x}_1, \cdots, \mathbf{x}_t)$. In this case, a realized adversary may be simply represented by a sequence of functions $(f_1, \cdots, f_T) \in \mathbf{F}^T$ and a sequence of query

[1] This is a slight generalization of the common use of the term bandit feedback. Usually, bandit feedback refers to the case where the oracle is a *deterministic* zeroth-order oracle and the agent query function is trivial, while our definition allows for *stochastic* oracles.

[2] Note that we do not assign a probability to each realized adversary since the notion of regret simply computes the supremum over all realizations.

oracles $(Q_1, \cdots, Q_T)$ for these functions. In this work we also consider adversaries that are *fully adaptive*, i.e., adversaries with no restriction. [3] Given a function class $\mathbf{F}$ and $i \in \{0, 1\}$, we use $\mathrm{Adv}_i^{\mathrm{f}}(\mathbf{F})$ to denote the set of all possible realized adversaries with deterministic i-th order oracles. If the oracle is instead stochastic and bounded by B, we use $\mathrm{Adv}_i^{\mathrm{f}}(\mathbf{F}, B)$ to denote such an adversary. Finally, we use $\mathrm{Adv}_i^{\mathrm{o}}(\mathbf{F})$ and $\mathrm{Adv}_i^{\mathrm{o}}(\mathbf{F}, B)$ to denote all oblivious realized adversaries with i-th order deterministic and stochastic oracles, respectively.

In order to handle different notions of regret with the same approach, for an agent $\mathcal{A}$, adversary Adv, compact set $\mathcal{U} \subseteq \mathcal{K}^T$ and $1 \leq a \leq b \leq T$, we define *regret* as

$$\mathcal{R}^{\mathcal{A}}_{\mathrm{Adv}}(\mathcal{U})[a, b] := \sup_{\mathcal{B} \in \mathrm{Adv}} \mathbb{E}\left[\sum_{t=a}^{b} f_t(\mathbf{x}_t) - \min_{\mathbf{u} \in \mathcal{U}} \sum_{t=a}^{b} f_t(\mathbf{u}_t)\right]$$

where $\mathbf{u} = (\mathbf{u}_1, \cdots, \mathbf{u}_T)$ and the expectation in the definition of the regret is over the randomness of the algorithm and the query oracles. We use the notation $\mathcal{R}^{\mathcal{A}}_{\mathcal{B}}(\mathcal{U})[a, b] := \mathcal{R}^{\mathcal{A}}_{\mathrm{Adv}}(\mathcal{U})[a, b]$ when $\mathrm{Adv} = \{\mathcal{B}\}$ is a singleton.

Static adversarial regret or simply *adversarial regret* corresponds to $a = 1$, $b = T$ and $\mathcal{U} = \mathcal{K}^T_\star := \{(\mathbf{x}, \cdots, \mathbf{x}) \mid \mathbf{x} \in \mathcal{K}\}$. When $a = 1$, $b = T$ and $\mathcal{U}$ contains only a single element then it is referred to as the *dynamic regret* [29, 33]. This notion of regret allows us to provide an upper bound on regret that may depend on the comparator. In Section 11, we will discuss algorithms that provide guarantees depending on the path-length, defined with $P_T := \mathbf{u} \mapsto \sum_{t=1}^{T-1} \|\mathbf{u}_t - \mathbf{u}_{t+1}\| : \mathcal{K}^T \to \mathbb{R}$. *Adaptive regret*, is defined as $\max_{1 \leq a \leq b \leq T} \mathcal{R}^{\mathcal{A}}_{\mathrm{Adv}}(\mathcal{K}^T_\star)[a, b]$ [13]. A crucial point in this definition is that the comparison is with respect to a (potentially) different optimum for any interval. We drop a, b and $\mathcal{U}$ when the statement is independent of their value or their value is clear from the context.

Remark 1. As a special case of static adversarial regret, if Adv is oblivious and every $\mathcal{B} \in \mathrm{Adv}$ corresponds to $f_1 = f_2 = \cdots = f_T = f$ and $Q_1 = Q_2 \cdots = Q_T = Q$ for some $f \in \mathbf{F}$ and some stochastic oracle Q for f, then the regret is referred to as the *stochastic regret*.

Another metric for comparing algorithms is *high probability regret bounds* which we define for stochastic algorithms. Given $\omega \in \Omega$, we may consider $\mathcal{A}_\omega$ as a deterministic algorithm. We say $h : [0, 1] \to [0, \infty)$ is a *high probability regret bound* for $(\mathcal{A}, \mathrm{Adv})$ if for each $\mathcal{B} \in \mathrm{Adv}$ and $\delta \in [0, 1]$, we have

$$\mathbb{P}\left(\{\omega \in \Omega \mid \mathcal{R}^{\mathcal{A}_\omega}_{\mathcal{B}} \leq h(\delta)\}\right) \geq 1 - \delta.$$

Note that the infimum of any family of high probability regret bounds is a high probability regret bound. Hence we use $\overline{\mathcal{R}}^{\mathcal{A}}_{\mathrm{Adv}} : [0, 1] \to [0, \infty)$ to denote the smallest high probability regret bound for $(\mathcal{A}, \mathrm{Adv})$. In other words, $\overline{\mathcal{R}}^{\mathcal{A}}_{\mathcal{B}}$ is the quantile function for the random variable $\omega \mapsto \mathcal{R}^{\mathcal{A}_\omega}_{\mathcal{B}}$.

[3]Another form of adversary considered in literature is a *weakly adaptive* adversary where each function $\mathcal{B}_t$ described above does not depend on $\mathbf{x}_t$ and therefore may be represented as a map defined on $\mathcal{K}^{t-1}$. Clearly any oblivious adversary is a weakly adaptive adversary and any weakly adaptive adversary is a fully adaptive adversary.

4 RE-STATEMENT OF PREVIOUS RESULT: OBLIVIOUS TO FULLY ADAPTIVE ADVERSARY

The following theorem states a result that is well-known in the literature. For example, [10] only defined the notion of adaptive adversary when discussing stochastic algorithms and mentions that the results of all previous chapters, which were focused on deterministic algorithms, apply to any type of adversary. Here we explicitly mention it as a theorem for completion.

Theorem 1. *Let $i \in \{0, 1\}$ and assume $\mathcal{A}$ is a deterministic online algorithm designed for i-th order feedback and $\mathbf{F}$ is a function class. Then we have*[4]

$$\mathcal{R}^{\mathcal{A}}_{\mathrm{Adv}_i^f(\mathbf{F})} = \mathcal{R}^{\mathcal{A}}_{\mathrm{Adv}_i^o(\mathbf{F})}.$$

Corollary 1. *The* OGD, FTRL, *and* OMD *algorithms are deterministic and therefore have same regret bound in fully-adaptive setting as the oblivious setting.*

5 LINEAR TO CONVEX WITH FULLY ADAPTIVE ADVERSARY

Here we show that any semi-bandit feedback online linear optimization algorithm for fully adaptive adversary is also an online convex optimization algorithm. We start with a definition.

Definition 1. Let $\mathbf{F}$ be a function class over $\mathcal{K}$ and let $\mu \geq 0$. We define $\mathbf{F}_\mu$, namely *μ-quadratization of* $\mathbf{F}$, to be class of functions $q : \mathcal{K} \to \mathbb{R}$ such that there exists $f \in \mathbf{F}$, $\mathbf{x} \in \mathcal{K}$, and $\mathbf{o} \in \nabla^* f(\mathbf{x})$ such that

$$q(\mathbf{y}) = \langle \mathbf{o}, \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}\|^2 \in \mathbf{Q}_\mu.$$

When $\mu = 0$, we may also refer to $\mathbf{F}_\mu$ as the *linearization of* $\mathbf{F}$. We say $\mathbf{F}$ is closed under μ-quadratization if $\mathbf{F} \supseteq \mathbf{F}_\mu$. Similarly, for $B > 0$, we define $\mathbf{Q}_\mu[B]$ to be the the class of functions q defined above where instead we have $\mathbf{o} \in \mathbb{B}_B(\mathbf{0})$.

Theorem 2. *Let $\mathcal{K} \subseteq \mathbb{R}^d$ be a convex set, let $\mu \geq 0$ and let $\mathcal{A}$ be algorithm for online optimization with semi-bandit feedback. If $\mathbf{F}$ be a μ-strongly convex function class over $\mathcal{K}$, then we have*

$$\mathcal{R}^{\mathcal{A}}_{\mathrm{Adv}_1^f(\mathbf{F})} \leq \mathcal{R}^{\mathcal{A}}_{\mathrm{Adv}_1^f(\mathbf{F}_\mu)}, \quad \overline{\mathcal{R}}^{\mathcal{A}}_{\mathrm{Adv}_1^f(\mathbf{F})} \leq \overline{\mathcal{R}}^{\mathcal{A}}_{\mathrm{Adv}_1^f(\mathbf{F}_\mu)}$$

Moreover, if $\mathbf{F}$ is closed under μ-quadratization, then we have equality.[4]

Corollary 2 (Theorem 14 in [1]). *The min-max regret bounds of online linear optimization with deterministic feedback and online convex optimization with deterministic feedback are equal.*

To see why this is true, note that any algorithm has the same performance in both settings. So it follows that an optimal algorithm for one setting is optimal for the other and therefore the min-max regret bounds are equal.

6 FULL INFORMATION FEEDBACK TO SEMI-BANDIT FEEDBACK

Any algorithm designed for semi-bandit setting may be trivially applied in the first-order full-information feedback setting. In the

[4]Detailed proofs are deferred to the appendix, which is available in the extended version on arXiv [20].

Meta-algorithm 1: Full information to semi-bandit - FTS($\mathcal{A}$)

Input : horizon T, algorithm $\mathcal{A} = (\mathcal{A}^{\text{action}}, \mathcal{A}^{\text{query}})$, strong convexity coefficient $\mu \geq 0$
for $t = 1, 2, \ldots, T$ **do**
  Play the action $\mathbf{x}_t$ chosen by $\mathcal{A}^{\text{action}}$
  Let f_t be the function chosen by the adversary
  Query the oracle at the point $\mathbf{x}_t$ to get $\mathbf{o}_t$
  Let $q_t := \mathbf{y} \mapsto \langle \mathbf{o}_t, \mathbf{y} - \mathbf{x}_t \rangle + \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}_t\|^2$
  for *i starting from 1, while $\mathcal{A}^{query}$ is not terminated for this time-step* **do**
    Let $\mathbf{y}_{t,i}$ be the query chosen by $\mathcal{A}^{\text{query}}$
    Return $\nabla q_t(\mathbf{y}_{t,i}) = \mathbf{o}_t + \mu(\mathbf{y}_{t,i} - \mathbf{x}_t)$ as the output of the query oracle to $\mathcal{A}$
  end
end

following theorem, we show that in online convex optimization for fully adaptive adversary, semi-bandit feedback is generally enough. Specifically, we show that an algorithm that requires full information feedback could be converted to an algorithm that only requires semi-bandit feedback with the same regret bounds. The meta-algorithm that does this conversion is described in Meta-algorithm 1. Next we show that FTS($\mathcal{A}$) always performs at least as good as $\mathcal{A}$ when the oracle is deterministic.

Theorem 3. *Let $\mathcal{K} \subseteq \mathbb{R}^d$ be a convex set, let $\mu \geq 0$ and let $\mathcal{A}$ be algorithm for online optimization with full information first order feedback. If $\mathbf{F}$ is a μ-strongly convex function class over $\mathcal{K}$, then we have*

$$\mathcal{R}^{\mathcal{A}'}_{\text{Adv}_1^f(\mathbf{F})} \leq \mathcal{R}^{\mathcal{A}}_{\text{Adv}_1^f(\mathbf{F}_\mu)},$$

where $\mathcal{A}' =$ FTS($\mathcal{A}$) is Meta-algorithm 1. In particular, if $\mathbf{F}$ is closed under μ-quadratization, then

$$\mathcal{R}^{\mathcal{A}'}_{\text{Adv}_1^f(\mathbf{F})} \leq \mathcal{R}^{\mathcal{A}}_{\text{Adv}_1^f(\mathbf{F})}.$$

See Appendix in [20] for proof. Note that when the base algorithm $\mathcal{A}$ is semi-bandit, we have FTS($\mathcal{A}$) $= \mathcal{A}$ and the above theorem becomes trivial.

7 LIPSCHITZ TO NON-LIPSCHITZ

It is well-known that when a convex function is restricted to a domain smaller than its original domain, then the restricted function is Lipschitz (See Observation 3 in [6] and Lemma 1 in Appendix at [20]). Here we use a shrinking method described in [21]. We choose a point $\mathbf{c} \in \text{relint}(\mathcal{K})$ and a real number $r > 0$ such that $\mathbb{B}_r(\mathbf{c}) \cap \text{aff}(\mathcal{K}) \subseteq \mathcal{K}$. Then, for any shrinking parameter $0 \leq \alpha < r$, we define $\hat{\mathcal{K}}_\alpha := (1 - \frac{\alpha}{r})\mathcal{K} + \frac{\alpha}{r}\mathbf{c}$. We define $\mathbf{F}|_{\hat{\mathcal{K}}_\alpha}$ to function class $\{f|_{\hat{\mathcal{K}}_\alpha} \mid f \in \mathbf{F}\}$. Given an adversary Adv, we define Adv$|_{\hat{\mathcal{K}}_\alpha}$ to be the adversary constructed by restricting the output of Adv to the set $\hat{\mathcal{K}}_\alpha$. Recall that the the domain $\mathcal{K}$ of the adversary is known to the agent $\mathcal{A}$. For an online algorithm $\mathcal{A}$, we define $\mathcal{A}|_{\hat{\mathcal{K}}_\alpha}$ to be the online algorithm resulting from restricting the domain of $\mathcal{A}$ to $\hat{\mathcal{K}}_\alpha$.

Theorem 4. *Let $\mathcal{A}$ be an online algorithm and $\mathbf{F}$ be a convex function class bounded by M_0. Also let $\mathcal{U} \subseteq \mathcal{K}^T$ be a compact set and $\hat{\mathcal{U}} := (1 - \frac{\alpha}{r})\mathcal{U} + \frac{\alpha}{r}\mathbf{c}$. Then, for any adversary* Adv*, we have*

$$\mathcal{R}^{\mathcal{A}|_{\hat{\mathcal{K}}_\alpha}}_{\text{Adv}}(\mathcal{U}) \leq \mathcal{R}^{\mathcal{A}}_{\text{Adv}_{\hat{\mathcal{K}}_\alpha}}(\hat{\mathcal{U}}) + \frac{\alpha M_0 T}{r},$$
$$\overline{\mathcal{R}}^{\mathcal{A}|_{\hat{\mathcal{K}}_\alpha}}_{\text{Adv}}(\mathcal{U}) \leq \overline{\mathcal{R}}^{\mathcal{A}}_{\text{Adv}_{\hat{\mathcal{K}}_\alpha}}(\hat{\mathcal{U}}) + \frac{\alpha M_0 T}{r}.$$

See Appendix in [20] for proof.

8 DETERMINISTIC FEEDBACK TO STOCHASTIC FEEDBACK

In this section, we show that algorithms that are designed for fully adaptive adversaries using deterministic semi-bandit feedback can obtain similar bounds using only stochastic semi-bandit feedback when facing oblivious adversaries. In particular, there is no need for any variance reduction method, such as momentum [4, 17, 28], as long as we know that the algorithm is designed for a fully adaptive adversary.

Theorem 5. *Let $\mathcal{K} \subseteq \mathbb{R}^d$ be a convex set, let $\mu \geq 0$ and let $\mathcal{A}$ be an algorithm for online optimization with semi-bandit feedback. If $\mathbf{F}$ is an M_1-Lipschitz μ-strongly convex function class over $\mathcal{K}$ and $B_1 \geq M_1$, then we have*

$$\mathcal{R}^{\mathcal{A}}_{\text{Adv}_1^o(\mathbf{F}, B_1)} \leq \mathcal{R}^{\mathcal{A}}_{\text{Adv}_1^f(\mathbf{Q}_\mu[B_1])},$$
$$\overline{\mathcal{R}}^{\mathcal{A}}_{\text{Adv}_1^o(\mathbf{F}, B_1)} \leq \overline{\mathcal{R}}^{\mathcal{A}}_{\text{Adv}_1^f(\mathbf{Q}_\mu[B_1])}.$$

See Appendix in [20] for proof. Theorem 6.5 in [10] may be seen as a special case of the above theorem when $\mathcal{U}$ is a singleton and $\mathcal{A}$ is deterministic.

9 FIRST ORDER FEEDBACK TO ZEROTH ORDER FEEDBACK

Meta-algorithm 2: First order to zeroth order - FOTZO($\mathcal{A}$)

Input : Shrunk domain $\hat{\mathcal{K}}_\alpha$, Linear space $\mathcal{L}_0$, smoothing parameter $\delta \leq \alpha$, horizon T, algorithm $\mathcal{A}$
Pass $\hat{\mathcal{K}}_\alpha$ as the domain to $\mathcal{A}$
$k \leftarrow \dim(\mathcal{L}_0)$
for $t = 1, 2, \ldots, T$ **do**
  Play $\mathbf{x}_t$, where $\mathbf{x}_t$ is the action chosen by $\mathcal{A}$
  Let f_t be the function chosen by the adversary
  for *i starting from 1, while $\mathcal{A}^{query}$ is not terminated for this time-step* **do**
    Sample $\mathbf{v}_{t,i} \in \mathbb{S}^1 \cap \mathcal{L}_0$ uniformly
    Let $\mathbf{y}_{t,i}$ be the query chosen by $\mathcal{A}^{\text{query}}$
    Let $o_{t,i}$ be the output of the query oracle at the point $\mathbf{y}_{t,i} + \delta \mathbf{v}_{t,i}$
    Pass $\frac{k}{\delta} o_{t,i} \mathbf{v}_{t,i}$ as the oracle output to $\mathcal{A}$
  end
end

When we do not have access to a gradient oracle, we rely on samples from a value oracle to estimate the gradient. The "smoothing trick" is a classical idea in convex optimization (See [18]) which was first used in online convex optimization in [6]. This idea involves averaging through spherical sampling around a given point. Here we use a variant that was introduced in [21] which does not require extra assumptions on the convex set $\mathcal{K}$.

For a function $f : \mathcal{K} \to \mathbb{R}$ defined on a convex set $\mathcal{K} \subseteq \mathbb{R}^d$ and a smoothing parameter $0 < \delta \le \alpha$, its δ-smoothed version $\hat{f}_\delta : \hat{\mathcal{K}}_\alpha \to \mathbb{R}$ is given as

$$\hat{f}_\delta(\mathbf{x}) := \mathbb{E}_{\mathbf{z} \sim \mathbb{B}_\delta(\mathbf{x}) \cap \mathrm{aff}(\mathcal{K})}[f(\mathbf{z})] = \mathbb{E}_{\mathbf{v} \sim \mathbb{B}_1(\mathbf{0}) \cap \mathcal{L}_0}[f(\mathbf{x} + \delta \mathbf{v})],$$

Meta-algorithm 3: Semi-bandit to bandit - STB($\mathcal{A}$)

Input : Shrunk domain $\hat{\mathcal{K}}_\alpha$, Linear space $\mathcal{L}_0$, smoothing parameter $\delta \le \alpha$, horizon T, algorithm $\mathcal{A}$
Pass $\hat{\mathcal{K}}_\alpha$ as the domain to $\mathcal{A}$
$k \leftarrow \dim(\mathcal{L}_0)$
for $t = 1, 2, \ldots, T$ **do**
  Sample $\mathbf{v}_t \in \mathbb{S}^1 \cap \mathcal{L}_0$ uniformly
  Play $\mathbf{x}_t + \delta \mathbf{v}_t$, where $\mathbf{x}_t$ is the action chosen by $\mathcal{A}$
  Let f_t be the function chosen by the adversary
  Let o_t be the output of the value oracle
  Pass $\frac{k}{\delta} o_t \mathbf{v}_t$ as the oracle output to $\mathcal{A}$
end

where $\mathcal{L}_0 = \mathrm{aff}(\mathcal{K}) - \mathbf{x}$, for any $\mathbf{x} \in \mathcal{K}$ is the linear space that is a translation of the affine hull of $\mathcal{K}$ and $\mathbf{v}$ is sampled uniformly at random from the $k = \dim(\mathcal{L}_0)$-dimensional ball $\mathbb{B}_1(\mathbf{0}) \cap \mathcal{L}_0$. Thus, the function value $\hat{f}_\delta(\mathbf{x})$ is obtained by "averaging" f over a sliced ball of radius δ around $\mathbf{x}$. For a function class $\mathbf{F}$ over $\mathcal{K}$, we use $\hat{\mathbf{F}}_\delta$ to denote $\{\hat{f}_\delta \mid f \in \mathbf{F}\}$. We will drop the subscripts α and δ when there is no ambiguity.

Theorem 6. *Let* $\mathbf{F}$ *be a function class over a convex set* $\mathcal{K}$ *and choose* $\mathbf{c}$ *and* r *as described above and let* $\delta \le \alpha < r$. *Let* $\mathcal{U} \subseteq \mathcal{K}^T$ *be a compact set and let* $\hat{\mathcal{U}} = (1 - \frac{\alpha}{r})\mathcal{U} + \frac{\alpha}{r}\mathbf{c}$. *Assume* $\mathcal{A}$ *is an algorithm for online optimization with first order feedback. Then, if* $\mathcal{A}' =$ FOTZO($\mathcal{A}$) *where* FOTZO *is described by Meta-algorithm 2, we have:*

- *if* $\mathbf{F}$ *is convex and bounded by* M_0*, we have*
$$\mathcal{R}^{\mathcal{A}'}_{\mathrm{Adv}^o_0(\mathbf{F}, B_0)}(\mathcal{U}) \le \mathcal{R}^{\mathcal{A}}_{\mathrm{Adv}^o_1(\hat{\mathbf{F}}, \frac{k}{\delta} B_0)}(\hat{\mathcal{U}}) + \left(\frac{6\delta}{\alpha} + \frac{\alpha}{r}\right) M_0 T.$$
- *if* $\mathbf{F}$ *is* M_1*-Lipschitz (but not necessarily convex), then we have*
$$\mathcal{R}^{\mathcal{A}'}_{\mathrm{Adv}^o_0(\mathbf{F}, B_0)}(\mathcal{U}) \le \mathcal{R}^{\mathcal{A}}_{\mathrm{Adv}^o_1(\hat{\mathbf{F}}, \frac{k}{\delta} B_0)}(\hat{\mathcal{U}}) + \left(3 + \frac{2D\alpha}{r\delta}\right) \delta M_1 T,$$
where $D = \mathrm{diam}(\mathcal{K})$.

Theorem 7. *Under the assumptions of Theorem 6, if we further assume that* $\mathcal{A}$ *is semi-bandit, then the same regret bounds hold with* $\mathcal{A}' =$ STB($\mathcal{A}$)*, where* STB *is described by Meta-algorithm 3.*

See Appendix in [20] for proof.

Remark 2. While STB turns a semi-bandit algorithm into a bandit algorithm, FOTZO turns a general first order algorithm into a zeroth order algorithm. However, we note that even when $\mathcal{A}$ is semi-bandit, FOTZO($\mathcal{A}$) is not a bandit algorithm. Instead it is a full-information zeroth order algorithm that uses a single query per iteration, but its query is not the same as its action. This is because FOTZO($\mathcal{A}$) plays the same point $\mathbf{x}_t$ as the base algorithm $\mathcal{A}$, but then adds a noise before querying according to $\mathcal{A}^{\text{query}}$.

Meta-algorithm 4: First order to zeroth order with two-point gradient estimator - FOTZO-2P($\mathcal{A}$)

Input : Shrunk domain $\hat{\mathcal{K}}_\delta$, Linear space $\mathcal{L}_0$, smoothing parameter $\delta < r$, horizon T, algorithm $\mathcal{A}$
Pass $\hat{\mathcal{K}}_\delta$ as the domain to $\mathcal{A}$
$k \leftarrow \dim(\mathcal{L}_0)$
for $t = 1, 2, \ldots, T$ **do**
  Play $\mathbf{x}_t$, where $\mathbf{x}_t$ is the action chosen by $\mathcal{A}$
  Let f_t be the function chosen by the adversary
  for *i starting from 1, while* $\mathcal{A}^{query}$ *is not terminated for this time-step* **do**
    Sample $\mathbf{v}_{t,i} \in \mathbb{S}^1 \cap \mathcal{L}_0$ uniformly
    Let $\mathbf{y}_{t,i}$ be the query chosen by $\mathcal{A}^{\text{query}}$
    Query the deterministic oracle at the points $\mathbf{y}_{t,i} + \delta \mathbf{v}_{t,i}$ and $\mathbf{y}_{t,i} - \delta \mathbf{v}_{t,i}$
    Pass $\frac{k}{2\delta}\left(f_t(\mathbf{y}_{t,i} + \delta \mathbf{v}_{t,i}) - f_t(\mathbf{y}_{t,i} - \delta \mathbf{v}_{t,i})\right)\mathbf{v}_{t,i}$ as the oracle output to $\mathcal{A}$
  end
end

Remark 3. We have shown in the previous sections that for online convex optimization, semi-bandit feedback is enough. However, we have included the FOTZO meta-algorithm, which converts a full-information algorithm to another full-information algorithm, since the results in this section apply also to non-convex online optimization problems.

10 FIRST ORDER FEEDBACK TO DETERMINISTIC ZEROTH ORDER FEEDBACK

When we have access to a deterministic zeroth order feedback, we may use the two point gradient estimator [2, 26] to significantly improve the results of the previous section.

Theorem 8. *Let* $\mathbf{F}$ *be an* M_1*-Lipschitz function class over a convex set* $\mathcal{K}$ *and choose* $\mathbf{c}$ *and* r *as described above and let* $\delta < r$. *Let* $\mathcal{U} \subseteq \mathcal{K}^T$ *be a compact set and let* $\hat{\mathcal{U}} = (1 - \frac{\delta}{r})\mathcal{U} + \frac{\delta}{r}\mathbf{c}$. *Assume* $\mathcal{A}$ *is an algorithm for online optimization with first order feedback. Then, if* $\mathcal{A}' =$ FOTZO-2P($\mathcal{A}$) *where* FOTZO-2P *is described by Meta-algorithm 4, we have*

$$\mathcal{R}^{\mathcal{A}'}_{\mathrm{Adv}^o_0(\mathbf{F})}(\mathcal{U}) \le \mathcal{R}^{\mathcal{A}}_{\mathrm{Adv}^o_1(\hat{\mathbf{F}}, k M_1)}(\hat{\mathcal{U}}) + \left(2 + \frac{2D}{r}\right) \delta M_1 T,$$

where $D = \mathrm{diam}(\mathcal{K})$.

Note that here we may choose $\delta = T^{-1}$ to see that the order of regret of $\mathcal{A}'$ remains the same as that of $\mathcal{A}$. See Appendix in [20] for proof.

11 APPLICATIONS

In this section, we discuss applications of meta-algorithms discussed above to some specific base algorithms. Note that all the results stated here are immediate applications of our framework to existing results and no extra proof is needed.

Online Gradient Descent: Recall that OGD algorithm [33] is a deterministic algorithm designed for deterministic semi-bandit feedback that obtains a regret bound of $O\left(M_1 T^{1/2}\right)$ over M_1-Lipschitz convex class $\mathbf{F}$ and $O\left(M_1^2 \log(T)\right)$ over strongly convex M_1-Lipschitz convex class $\mathbf{F}'$ (see [10]). [5] Since OGD is deterministic, using Theorem 1, we see that

$$\mathcal{R}^{\texttt{OGD}}_{\text{Adv}_1^{\text{f}}(\mathbf{F})}(\mathcal{K}_*^T)[1,T] = O\left(M_1 T^{1/2}\right),$$
$$\mathcal{R}^{\texttt{OGD}}_{\text{Adv}_1^{\text{f}}(\mathbf{F}')}(\mathcal{K}_*^T)[1,T] = O\left(M_1^2 \log(T)\right).$$

Hence we may apply Theorems 5 to see that, if we have access to stochastic gradient oracle bounded by $B_1 \geq M_1$, then we have

$$\mathcal{R}^{\texttt{OGD}}_{\text{Adv}_1^{\text{o}}(\mathbf{F},B_1)}(\mathcal{K}_*^T)[1,T] = O\left(B_1 T^{1/2}\right),$$
$$\mathcal{R}^{\texttt{OGD}}_{\text{Adv}_1^{\text{o}}(\mathbf{F}',B_1)}(\mathcal{K}_*^T)[1,T] = O\left(B_1^2 \log(T)\right).$$

For convex functions, we may apply Theorem 7, with $\alpha = \delta = T^{-1/4}$. Thus, if we have access to stochastic value oracle bounded by $B_0 \geq M_1$, we have

$$\mathcal{R}^{\texttt{STB(OGD)}}_{\text{Adv}_0^{\text{o}}(\mathbf{F},B_0)}(\mathcal{K}_*^T)[1,T] \leq \mathcal{R}^{\texttt{OGD}}_{\text{Adv}_1^{\text{o}}(\mathbf{F},\frac{k}{\delta}B_0)}(\hat{\mathcal{K}}_*^T)[1,T] + O\left(\delta M_1 T\right)$$
$$= O\left(\frac{k}{\delta} B_0 T^{1/2} + \delta M_1 T\right) = O\left(B_0 T^{3/4}\right),$$

which generalizes the result of [6] to allow the feedback of the value oracle to be stochastic. Similarly, for strongly convex functions, we may apply Theorem 7, with $\alpha = \delta = T^{-1/3}\log(T)^{1/3}$ to see that

$$\mathcal{R}^{\texttt{STB(OGD)}}_{\text{Adv}_0^{\text{o}}(\mathbf{F}',B_0)}(\mathcal{K}_*^T)[1,T] \leq \mathcal{R}^{\texttt{OGD}}_{\text{Adv}_1^{\text{o}}(\mathbf{F}',\frac{k}{\delta}B_0)}(\hat{\mathcal{K}}_*^T)[1,T] + O\left(\delta M_1 T\right)$$
$$= O\left(\frac{k^2}{\delta^2} B_0^2 \log(T) + \delta M_1 T\right) = O\left(B_0^2 T^{2/3} \log(T)^{1/3}\right).$$

This provides a new algorithm for bandit strongly convex optimization, matching the result of [2]. However, if we further assume that the functions are smooth, then the regret bound could be improved to $\widetilde{O}(\sqrt{T})$ (See [12, 14]).

On the other hand, using Theorem 8 on OGD, we see that

$$\mathcal{R}^{\texttt{FOTZO-2P(OGD)}}_{\text{Adv}_0^{\text{o}}(\mathbf{F})}(\mathcal{K}_*^T)[1,T] = O\left(M_1 T^{1/2}\right),$$
$$\mathcal{R}^{\texttt{FOTZO-2P(OGD)}}_{\text{Adv}_0^{\text{o}}(\mathbf{F}')}(\mathcal{K}_*^T)[1,T] = O\left(M_1^2 \log(T)\right),$$

where the first result is proven in [26] and the second one is novel.

Ader and Improved Ader: In [29], Algorithm 2, i.e., "Ader", is a full-information deterministic algorithm designed for deterministic first order feedback. Algorithm 3, i.e., "Improved Ader", is simply the result of applying the FTS meta-algorithm to Ader. We may apply Theorems 5 and 7, with $\alpha = \delta = T^{-1/4}$, to see that

$$\mathcal{R}^{\texttt{STB(IA)}}_{\text{Adv}_0^{\text{o}}(\mathbf{F},B_0)}(\mathbf{u})[1,T] = O\left(B_0 \left(1 + P_T(\mathbf{u})\right)^{1/2} T^{3/4}\right), \quad (1)$$

where $\mathbf{u} \in \mathcal{K}^T$ and $P_T := \mathbf{u} \mapsto \sum_{t=1}^{T-1} \|\mathbf{u}_t - \mathbf{u}_{t+1}\| : \mathcal{K}^T \to \mathbb{R}$. This matches the SOTA result of [31] for dynamic regret in the bandit feedback setting and generalizes it to allow for stochastic feedback.

[5]Here we assume $\text{diam}(\mathcal{K}) \leq 1$ for simplicity. General bounds follow similarly and may depend on $\text{diam}(\mathcal{K})$ and r, as described in Theorems 6 and 8.

Moreover, by applying Theorem 8 to Improved Ader, we recover another result of [31]

$$\mathcal{R}^{\texttt{FOTZO-2P(IA)}}_{\text{Adv}_0^{\text{o}}(\mathbf{F})}(\mathbf{u})[1,T] = O\left(M_1 \left(1 + P_T(\mathbf{u})\right)^{1/2} T^{1/2}\right).$$

Online Gradient Descent with Separation Oracles: Similarly, the SO-OGD algorithm [7] is a deterministic algorithm designed for deterministic semi-bandit feedback obtains a adaptive regret bound of $O(M_1 T^{1/2})$ (Theorem 14 in [7]) Hence we may apply Theorems 5 and 7, with $\alpha = \delta = T^{-1/4}$, to see that

$$\max_{1 \leq a \leq b \leq T} \mathcal{R}^{\texttt{STB(SO-OGD)}}_{\text{Adv}_0^{\text{o}}(\mathbf{F},B_0)}(\mathcal{K}_*^T)[a,b] = O\left(B_0 T^{3/4}\right).$$

This matches the result of Theorem 15 in [7] and generalizes it to allow for stochastic feedback. Moreover, using Theorem 8, we see that

$$\max_{1 \leq a \leq b \leq T} \mathcal{R}^{\texttt{FOTZO-2P(SO-OGD)}}_{\text{Adv}_0^{\text{o}}(\mathbf{F})}(\mathcal{K}_*^T)[a,b] = O\left(M_1 T^{1/2}\right),$$

which is a novel result for adaptive regret of convex functions with deterministic zeroth order feedback.

Remark 4. A careful review of Algorithms 6 and 7 in [7], together with their Lemmas 12 and 13 and Theorems 14 and 15 reveals that they added an extra layer of complexity to their base algorithm in order to be able to move from semi-bandit to bandit feedback. Specifically, if we instead use our Theorem 7, we can just drop δ' from Algorithms 6 and 7 and Lemmas 12 and 13 (by setting it to zero) to obtain simpler and clearer algorithms, statements and proofs. Then their Theorem 15 will be an immediate corollary of applying our Theorem 7 to their previous results and the 3 pages of proof in their Appendix I will not be needed.

12 CONCLUSIONS

This paper presents a comprehensive framework for the analysis of meta-algorithms in online learning, offering a simplified approach to understanding and extending existing results in the field. By demonstrating the equivalence between online linear and convex optimization algorithms, particularly in the context of semi-bandit feedback, we provide a versatile tool for addressing a wide range of optimization problems. Moreover, our findings pave the way for the development of novel meta-algorithms, enabling the seamless conversion of algorithms across different settings. In addition to recovering many results in the literature with simplified proofs, for convex function with deterministic zeroth order feedback, we obtain an adaptive regret of $O(\sqrt{T})$ (with FOTZO-2P(SO-OGD)) and static regret of $O(\log T)$ for strongly convex case (with FOTZO-2P(OGD)).

This work opens various research opportunities to delve into the interplay between online linear and convex optimization algorithms. While it streamlines results across numerous scenarios, we anticipate that this approach will yield novel insights into some unresolved problems moving forward. After this work, the meta-algorithm framework in this work was extended further from convex functions to linearizable/quadratizable functions in [19, 22].

13 ACKNOWLEDGMENTS

The authors want to acknowledge the useful feedback on the paper and its ideas provided by Yuanyu Wan.

This work was supported in part by Cisco Inc., and National Science Foundation under grant CCF-2149588

REFERENCES

[1] Jacob Abernethy, Peter L Bartlett, Alexander Rakhlin, and Ambuj Tewari. 2008. Optimal strategies and minimax lower bounds for online convex games. In *Proceedings of the 21st annual conference on learning theory*. 414–424.

[2] Alekh Agarwal, Ofer Dekel, and Lin Xiao. 2010. Optimal Algorithms for Online Convex Optimization with Multi-Point Bandit Feedback.. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT 2010)*, Adam Tauman Kalai and Mehryar Mohri (Eds.). 28–40.

[3] Omar Besbes, Yonatan Gur, and Assaf Zeevi. 2015. Non-Stationary Stochastic Optimization. *Operations Research* 63, 5 (2015), 1227–1244.

[4] Lin Chen, Christopher Harshaw, Hamed Hassani, and Amin Karbasi. 2018. Projection-Free Online Optimization with Stochastic Gradient: From Convexity to Submodularity. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 814–823.

[5] Amit Daniely, Alon Gonen, and Shai Shalev-Shwartz. 2015. Strongly Adaptive Online Learning. In *Proceedings of the 32nd International Conference on Machine Learning* (2015-06-01). PMLR, 1405–1411.

[6] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. 2005. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. 385–394.

[7] Dan Garber and Ben Kretzu. 2022. New Projection-free Algorithms for Online Convex Optimization with Adaptive Regret Guarantees. In *Proceedings of Thirty Fifth Conference on Learning Theory* (2022-06-28). PMLR, 2326–2359.

[8] Geoffrey J. Gordon. 1999. Regret bounds for prediction problems. In *Proceedings of the twelfth annual conference on Computational learning theory* (New York, NY, USA, 1999-07-06) *(COLT '99)*. Association for Computing Machinery, 29–40.

[9] James Hannan. 1957. Approximation to bayes risk in repeated play. In *M. Dresher, A. W. Tucker, and P. Wolfe, editors, Contributions to the Theory of Games*. Princeton University Press, 97–140.

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

[11] Elad Hazan and Satyen Kale. 2010. Extracting certainty from uncertainty: regret bounded by variation in costs. *Machine Learning* 80, 2 (2010), 165–188.

[12] Elad Hazan and Kfir Levy. 2014. Bandit Convex Optimization: Towards Tight Bounds. In *Advances in Neural Information Processing Systems*, Vol. 27.

[13] Elad Hazan and C. Seshadhri. 2009. Efficient learning algorithms for changing environments. In *Proceedings of the 26th Annual International Conference on Machine Learning* (New York, NY, USA, 2009-06-14) *(ICML '09)*. Association for Computing Machinery, 393–400.

[14] Shinji Ito. 2020. An Optimal Algorithm for Bandit Convex Optimization with Strongly-Convex and Smooth Loss. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics* (2020-06-03). PMLR, 2229–2239.

[15] Adam Kalai and Santosh Vempala. 2005. Efficient algorithms for online decision problems. *J. Comput. System Sci.* 71, 3 (2005), 291–307.

[16] Zhou Lu, Nataly Brukhim, Paula Gradu, and Elad Hazan. 2023. Projection-free Adaptive Regret with Membership Oracles. In *Proceedings of The 34th International Conference on Algorithmic Learning Theory* (2023-02-13). PMLR, 1055–1073.

[17] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. 2020. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *The Journal of Machine Learning Research* 21, 1 (2020), 4232–4280.

[18] Arkadiĭ Semenovich Nemirovskiĭ and David Berkovich ÎÛdin. 1983. *Problem Complexity and Method Efficiency in Optimization.* Wiley.

[19] Mohammad Pedramfar and Vaneet Aggarwal. 2024. From linear to linearizable optimization: a novel framework with applications to stationary and non-stationary DR-submodular optimization. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*. 37626–37664.

[20] Mohammad Pedramfar and Vaneet Aggarwal. 2024. A unified framework for analyzing meta-algorithms in online convex optimization. *arXiv preprint arXiv:2402.08621* (2024).

[21] Mohammad Pedramfar, Christopher John Quinn, and Vaneet Aggarwal. 2023. A Unified Approach for Maximizing Continuous DR-submodular Functions. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[22] Mohammad Pedramfar, Christopher John Quinn, and Vaneet Aggarwal. 2025. Uniform wrappers: Bridging concave to quadratizable functions in online optimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[23] Shai Shalev-Shwartz. 2007. *Online learning: theory, algorithms and applications*. Ph.D. Dissertation. Hebrew University.

[24] Shai Shalev-Shwartz. 2012. Online Learning and Online Convex Optimization. *Foundations and Trends® in Machine Learning* 4, 2 (2012), 107–194.

[25] Shai Shalev-Shwartz and Yoram Singer. 2007. A primal-dual perspective of online learning algorithms. *Machine Learning* 69, 2 (2007), 115–142.

[26] Ohad Shamir. 2017. An Optimal Algorithm for Bandit and Zero-Order Convex Optimization with Two-point Feedback. *Journal of Machine Learning Research* 18, 52 (2017), 1–11.

[27] Yibo Wang, Wenhao Yang, Wei Jiang, Shiyin Lu, Bing Wang, Haihong Tang, Yuanyu Wan, and Lijun Zhang. 2024. Non-stationary Projection-Free Online Learning with Dynamic and Adaptive Regret Guarantees. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 14 (2024), 15671–15679.

[28] Jiahao Xie, Zebang Shen, Chao Zhang, Boyu Wang, and Hui Qian. 2020. Efficient Projection-Free Online Methods with Stochastic Recursive Gradient. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 4 (2020), 6446–6453.

[29] Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. 2018. Adaptive Online Learning in Dynamic Environments. In *Advances in Neural Information Processing Systems*, Vol. 31. Curran Associates, Inc.

[30] Lijun Zhang, Tianbao Yang, jin, and Zhi-Hua Zhou. 2018. Dynamic Regret of Strongly Adaptive Methods. In *Proceedings of the 35th International Conference on Machine Learning* (2018-07-03). PMLR, 5882–5891.

[31] Peng Zhao, Guanghui Wang, Lijun Zhang, and Zhi-Hua Zhou. 2021. Bandit Convex Optimization in Non-stationary Environments. *Journal of Machine Learning Research* 22, 125 (2021), 1–45.

[32] Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. 2020. Dynamic Regret of Convex and Smooth Functions. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 12510–12520.

[33] Martin Zinkevich. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*. 928–936.