From Linear to Linearizable Optimization: A Novel Framework with Applications to Stationary and Non-stationary DR-submodular Optimization

Mohammad Pedramfar
McGill University and Mila *
mohammad.pedramfar@mila.quebec

Vaneet Aggarwal
Purdue University
vaneet@purdue.edu

Abstract

This paper introduces the notion of upper-linearizable/quadratizable functions, a class that extends concavity and DR-submodularity in various settings, including monotone and non-monotone cases over different types of convex sets. A general meta-algorithm is devised to convert algorithms for linear/quadratic maximization into ones that optimize upper-linearizable/quadratizable functions, offering a unified approach to tackling concave and DR-submodular optimization problems. The paper extends these results to multiple feedback settings, facilitating conversions between semi-bandit/first-order feedback and bandit/zeroth-order feedback, as well as between first/zeroth-order feedback and semi-bandit/bandit feedback. Leveraging this framework, new algorithms are derived using existing results as base algorithms for convex optimization, improving upon state-of-the-art results in various cases. Dynamic and adaptive regret guarantees are obtained for DR-submodular maximization, marking the first algorithms to achieve such guarantees in these settings. Notably, the paper achieves these advancements with fewer assumptions compared to existing state-of-the-art results, underscoring its broad applicability and theoretical contributions to non-convex optimization.

1 Introduction

Overview: The prominence of optimizing continuous adversarial γ-weakly up-concave functions (with DR-submodular and concave functions as special cases) has surged in recent years, marking a crucial subset within the realm of non-convex optimization challenges, particularly in the forefront of machine learning and statistics. This problem has numerous real-world applications, such as revenue maximization, mean-field inference, recommendation systems [4, 20, 30, 12, 24, 18, 26]. This problem is modeled as a repeated game between an optimizer and an adversary. In each round, the optimizer selects an action, and the adversary chooses a γ-weakly up-concave reward function. Depending on the scenario, the optimizer can then query this reward function either at any arbitrary point within the domain (called full information feedback) or specifically at the chosen action (called semi-bandit/bandit feedback), where the feedback can be noisy/deterministic. The performance metric of the algorithm is measured with multiple regret notions - static adversarial regret, dynamic regret, and adaptive regret. The algorithms for the problem are separated into the ones that use a projection operator to project the point to the closest point in the domain, and the projection-free methods that replace the projection with an alternative such as Linear Optimization Oracles (LOO) or Separation Oracles (SO). This interactive framework introduces a range of significant challenges, influenced by the characteristics of the up-concave function (monotone/non-monotone), the constraints imposed, the nature of the queries, projection-free/projection-based algorithms, and the different regret definitions.

*Work done while at Purdue University

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

Table 1: Online up-concave maximization

F	Set	Feedback			Reference	Appx.	# of queries	$\log_T(\alpha\text{-regret})$
Monotone	$0 \in \mathcal{K}$	∇F	Full Information	stoch.	[46] ‡ (*)	$1-e^{-\gamma}$	1	$1/2$
					[35]	$1-e^{-1}$	$T^\theta(\theta \in [0,1/2])$	$2/3-\theta/3$
					Corollary 7-c	$1-e^{-\gamma}$	1	$1/2$
			Semi-bandit	stoch.	[35]	$1-e^{-1}$	-	$3/4$
					Corollary 7-c	$1-e^{-\gamma}$	-	$2/3$
		F	Full Information	det.	Corollary 7-c	$1-e^{-1}$	2	$1/2$
				stoch.	[35]	$1-e^{-1}$	$T^\theta(\theta \in [0,1/4])$	$4/5-\theta/5$
					Corollary 7-c	$1-e^{-\gamma}$	1	$3/4$
			Bandit	det.	[39] ‡‡	$1-e^{-1}$	-	$3/4$
					[48] ‡(*)	$1-e^{-\gamma}$	-	$4/5$
				stoch.	[35]	$1-e^{-1}$	-	$5/6$
					Corollary 7-c	$1-e^{-\gamma}$	-	$4/5$
	general	∇F	Full Information	stoch.	[35]	$1/2$	$T^\theta(\theta \in [0,1/2])$	$2/3-\theta/3$
			Semi-bandit	stoch.	[8]‡(*)	$\gamma^2/(1+\gamma^2)$	-	$1/2$
					[35]	$1/2$	-	$3/4$
					Corollary 7-b	$\gamma^2/(1+c\gamma^2)$	-	$1/2$
		F	Full Information	det.	Corollary 7-b	$\gamma^2/(1+c\gamma^2)$	2	$1/2$
				stoch.	[35]	$1/2$	$T^\theta(\theta \in [0,1/4])$	$4/5-\theta/5$
			Bandit	stoch.	[35]	$1/2$	-	$5/6$
					Corollary 7-b	$\gamma^2/(1+c\gamma^2)$	-	$3/4$
Non-Monotone	general	∇F	Full Information	stoch.	[35]	$(1-h)/4$	$T^\theta(\theta \in [0,1/2])$	$2/3-\theta/3$
					[48] ‡(*)	$(1-h)/4$	1	$1/2$
					Corollary 7-d	$(1-h)/4$	1	$1/2$
			Semi-bandit	stoch.	[35]	$(1-h)/4$	-	$3/4$
					Corollary 7-d	$(1-h)/4$	-	$2/3$
		F	Full Information	det.	Corollary 7-d	$(1-h)/4$	2	$1/2$
				stoch.	[35]	$(1-h)/4$	$T^\theta(\theta \in [0,1/4])$	$4/5-\theta/5$
					Corollary 7-d	$(1-h)/4$	1	$3/4$
			Bandit	det.	[48] ‡(*)	$(1-h)/4$	-	$4/5$
				stoch.	[35]	$(1-h)/4$	-	$5/6$
					Corollary 7-d	$(1-h)/4$	-	$4/5$

This table compares different static regret results for the online up-concave maximization. The logarithmic terms in regret are ignored. Here $h := \min_{\mathbf{z} \in \mathcal{K}} \|\mathbf{z}\|_\infty$. Our algorithm is projection-free and use a separation oracle. The rows marked with ‡ use gradient ascent, requiring potentially computationally expensive projections. The rows marked with $(*)$ denote results that could be considered special case of our framework. In particular, we obtain those results if we use Online Gradient Ascent instead of `SO-OGA` as the base algorithm in Corollary 7. Note that the result of [39], marked by ‡‡, uses a convex optimization subroutine in each iteration, which could potentially be more expensive than projection and therefore not considered a projection-free result. **It is also the only existing result, in all the tables, that outperforms ours.** We note that stochastic results can be used in deterministic, and bandit/semi-bandit in full information and thus cases where our result is not added, our result improves state-of-the-art projection free result because of the result in less information setup. All results assume that functions are Lipschitz. Except for our results on monotone functions over general convex sets, all results also assume differentiability. All previous results assume that functions are DR-submodular, while we only require up-concavity. Results of [35] and [39] also assume functions are smooth, i.e., their gradients are Lipschitz.

In this paper, we present a comprehensive approach to solving adversarial up-concave optimization problems, encompassing different feedback types (including bandit, semi-bandit and full-information feedback), characteristics of the up-concave function and constraint region, projection-free/projection-based algorithms, and regret definitions. While the problem has been studied in many special cases, the main contribution of this work is a framework that is based on a novel notion of the function class being upper-linearizable (or upper-quadratizable). We design a meta-algorithm that converts certain algorithms designed for online linear maximization to algorithms capable of handling upper-linearizable function classes. This allows us to reduce the problem of up-concave maximization in three different settings to online linear maximization and obtain corresponding regret bounds. In particular, our results include monotone γ-weakly up-concave functions over general convex set, monotone γ-weakly up-concave functions over convex sets containing the origin and non-monotone up-concave functions. While the above result is for first order feedback, we then derive multiple results that increase the applicability of the above results. We extend the applicability of `FOTZO` and `STB` algorithm introduced in [34] to our setting which allows us to convert algorithms for first-order/semi-bandit feedback into algorithms for zeroth-order/bandit feedback. We also design a meta-algorithm that allows us to convert algorithms that require full-information feedback into algorithms that only require semi-bandit/bandit feedback.

We demonstrate the usefulness of results through two applications as described in the following. In the first application, we use the `SO-OGD` Algorithm in [17] as the base algorithm for online linear optimization, which is a projection-free algorithm. Using this, we first obtain the adaptive regret (and therefore also static regret) guarantees for the three setups of DR-submodular (or more generally, up-concave) optimization with semi-bandit feedback/first order feedback in the respective cases. Then, the meta-algorithms for conversion of first-order/semi-bandit to zeroth-order/bandit are used to get result with zeroth-order/bandit feedback. In the cases where the algorithms are full-information and not (semi-)bandit, we use another meta-algorithm to obtain algorithms in (semi-)bandit feedback setting. In the next application, we use the "Improved Ader" algorithm of [43] which is a projection based algorithm providing dynamic regret guarantees for the convex optimization. Afterwards, the same approach as above are used to obtain the results in the three scenarios of up-concave optimization with first-order feedback.

Technical Novelty: The main technical novelties in this work are as follows.

1. We proposes a novel notion of linearizable/quadratizable functions and extend the meta-algorithm framework of [34] from convex functions to linearizable/quadratizable functions. This allows us to relates a large class of algorithms and regret guarantees for optimization of linear/quadratic functions to that for linearizable/quadratizable functions.
2. We show that the class of quadratizable function optimization is general, and includes not only concave, but up-concave optimization in several cases. For some of the cases, this proof uses a generalization of the idea of boosting ([46, 48]) which was proposed for DR-submodular maximization, as mentioned in Corollaries 2 and 3.
3. We design a new meta-algorithm, namely `SFTT`, that captures the idea of random permutations (sometimes referred to as blocking) as used in several papers such as [45, 47, 35]. While previous works used this idea in specific settings, our meta-algorithm is applicable in general settings.
4. We note the generality of the above results in this paper. Our results are general in the following three aspects:

   a) In this work, we improve results for projection-free static regret guarantees for DR-submodular optimization in all considered cases and obtain the first results for dynamic and adaptive regret. Moreover, these guarantees follow from existing algorithms for the linear optimization, using only the statement of the regret bounds and simple properties of the algorithms.

   b) We consider 3 classes of DR-submodular functions in this work. However, to extend these results to another function class, all one needs to do is to (i) prove that the function class is quadratizable; and (ii) provide an unbiased estimator of $\mathfrak{g}$ (as described in Equation 1).

   c) We consider 2 different feedback types in offline setting (first/zero order) and 4 types of feedback in the online setting (first/zero order and full-information/trivial query). Converting results between different cases is obtained through meta-algorithms and guarantees for the meta-algorithms which only relies on high level properties of the base algorithms (See Theorems 5, 7, 6 and 8)

Key contributions: The key contributions in this work are summarized as follows.

1. We formulate the notion of *upper-quadratizable/upper-linearizeble* functions, which is a class that generalizes the notion of strong-concavity/concavity and also DR-submodularity in several settings. In particular, we demonstrate the the following function classes are upper-quadratizable: (i) monotone γ-weakly μ-strongly DR-submodular functions with curvature c over general convex sets, (ii) monotone γ-weakly DR-submodular functions over convex sets containing the origin, and (iii) non-monotone DR-submodular optimization over general convex sets.
2. We provide a general meta-algorithm that converts algorithms for linear/quadratic maximization to algorithms that maximize upper-quadratizable functions. This results is a unified approach to maximize both concave functions and DR-submodular functions in several settings.
3. While the above provides results for semi-bandit feedback (for monotone DR-submodular optimization over general convex sets) and first-order feedback (for monotone DR-submodular optimization over convex sets containing the origin, and non-monotone DR-submodular optimization over general convex sets), the results could be extended to more general feedback settings. Four meta algorithms are provided that relate semi-bandit/first-order feedback to bandit/zeroth order feedback; that relate; first order to deterministic zeroth order; and that relate

first/zeroth order feedback to semi-bandit/bandit feedback. Together they allow us to obtain results in 5 feedback settings (first/zeroth order full-information and semi-bandit/bandit; and deterministic zeroth order). We also discuss a meta-algorithm to convert online results to offline guarantees.

4. The above framework is applied using different algorithms as the base algorithms for linear optimization. `SO-OGD` [17] is a projection-free algorithm using separation oracles that provides adaptive regret guarantees for online convex optimization. We use our framework to cover the 18 cases in Table 1. We improve the regret guarantees for the previous SOTA projection-free algorithms in all the cases. If we also allow comparisons with the algorithms that are not projection-free, we still improve the SOTA results in 12 cases and match the SOTA in 5 cases.
5. Using our framework, we convert online results using `SO-OGD` to offline results to obtain 6 projection free algorithms described in Table 2. We improve the regret guarantees for the previous SOTA projection-free algorithms in all the cases, except for deterministic first order feedback where existing results are already SOTA. If we also allow comparisons with the algorithms that are not projection-free, we still improve the SOTA results in 6 cases and match the SOTA in the remaining 3 cases.
6. We use our framework to convert the adaptive regret guarantees of `SO-OGD` to obtain projection-free algorithms with adaptive regret bounds that cover all cases in Table 3. Our results are first algorithms with adaptive regret guarantee for online DR-submodular maximization.
7. "Improved Ader" [43] is an algorithm providing dynamic regret guarantees for online convex optimization. We use our framework to obtain 6 algorithms which provide the dynamic regret guarantees as shown in Table 3. Our results are first algorithms with dynamic regret guarantee for online DR-submodular maximization.
8. For monotone γ-weakly functions with bounded curvature over general convex sets, we improve the approximation ratio (See Lemma 1).
9. As mentioned in the descriptions of the tables, in all cases considered, whenever there is another existing result, we obtain our results using fewer assumptions than the existing SOTA.

2 Problem Setup and Definitions

For a set $\mathcal{D} \subseteq \mathbb{R}^d$, we define its *affine hull* $\operatorname{aff}(\mathcal{D})$ to be the set of $\alpha\mathbf{x} + (1-\alpha)\mathbf{y}$ for all $\mathbf{x}, \mathbf{y}$ in $\mathcal{K}$ and $\alpha \in \mathbb{R}$. The *relative interior* of $\mathcal{D}$ is defined as $\operatorname{relint}(\mathcal{D}) := \{\mathbf{x} \in \mathcal{D} \mid \exists r > 0, \mathbb{B}_r^d(\mathbf{x}) \cap \operatorname{aff}(\mathcal{D}) \subseteq \mathcal{D}\}$. For any $\mathbf{u} \in \mathcal{K}^T$, we define the path length $P_T(\mathbf{u}) := \sum_{i=1}^{T-1} \|\mathbf{u}_i - \mathbf{u}_{i+1}\|$. Given $\mu \geq 0$ and $0 < \gamma \leq 1$, we say a differentiable function $f : \mathcal{K} \to \mathbb{R}$ is *μ-strongly γ-weakly up-concave* if it is μ-strongly γ-weakly concave along positive directions. Specifically if, for all $\mathbf{x} \leq \mathbf{y}$ in $\mathcal{K}$, we have

$$\gamma \left(\langle \nabla f(\mathbf{y}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}\|^2 \right) \leq f(\mathbf{y}) - f(\mathbf{x}) \leq \frac{1}{\gamma} \left(\langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle - \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}\|^2 \right).$$

We say $\tilde{\nabla} f : \mathcal{K} \to \mathbb{R}^d$ is a *μ-strongly γ-weakly up-super-gradient* of f if for all $\mathbf{x} \leq \mathbf{y}$ in $\mathcal{K}$, the above holds with $\tilde{\nabla}$ instead of ∇. We say f is μ-strongly γ-weakly up-concave if it is continuous and it has a μ-strongly γ-weakly up-super-gradient. When it is clear from the context, we simply refer to $\tilde{\nabla} f$ as an up-super-gradient for f. When $\gamma = 1$ and the above inequality holds for all $\mathbf{x}, \mathbf{y} \in \mathcal{K}$, we say f is μ-strongly concave. A differentiable function $f : \mathcal{K} \to \mathbb{R}$ is called *γ-weakly continuous DR-submodular* if for all $\mathbf{x} \leq \mathbf{y}$, we have $\nabla f(\mathbf{x}) \geq \gamma \nabla f(\mathbf{y})$. It follows that any γ-weakly continuous DR-submodular functions is γ-weakly up-concave. We refer to Appendix B for more details.

Given a continuous monotone function $f : \mathcal{K} \to \mathbb{R}$, its curvature is defined as the smallest number $c \in [0, 1]$ such that $f(\mathbf{y} + \mathbf{z}) - f(\mathbf{y}) \geq (1 - c)(f(\mathbf{x} + \mathbf{z}) - f(\mathbf{x}))$, for all $\mathbf{x}, \mathbf{y} \in \mathcal{K}$ and $\mathbf{z} \geq 0$ such that $\mathbf{x} + \mathbf{z}, \mathbf{y} + \mathbf{z} \in \mathcal{K}$. We define the curvature of a function class $\mathbf{F}$ as the supremum of the curvature of functions in $\mathbf{F}$.

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 $1 \leq i \leq k_t$,

Table 2: Offline up-concave maximization

F	Set	Feedback		Reference	Appx.	Complexity
Monotone	$0 \in \mathcal{K}$	∇F	stoch.	[31]	$1 - e^{-\gamma}$	$O(1/\epsilon^3)$
				[19]	$1 - e^{-\gamma}$	$O(1/\epsilon^2)$
				[46] ‡	$1 - e^{-\gamma}$	$O(1/\epsilon^2)$
				Corollary 7-c	$1 - e^{-\gamma}$	$O(1/\epsilon^2)$
		F	det.	[36]	$1 - e^{-\gamma}$	$O(1/\epsilon^3)$
				Corollary 7-c	$1 - e^{-\gamma}$	$O(1/\epsilon^2)$
			stoch.	[36]	$1 - e^{-\gamma}$	$O(1/\epsilon^5)$
				Corollary 7-c	$1 - e^{-\gamma}$	$O(1/\epsilon^4)$
	general	∇F	stoch.	[20]‡	$\gamma^2/(1+\gamma^2)$	$O(1/\epsilon^2)$
				[36]	$\gamma^2/(1+\gamma^2)$	$\tilde{O}(1/\epsilon^3)$
				Corollary 7-b	$\gamma^2/(1+c\gamma^2)$	$O(1/\epsilon^2)$
		F	det.	[37]	$\gamma^2/(1+\gamma^2)$	$\tilde{O}(1/\epsilon^3)$
				Corollary 7-b	$\gamma^2/(1+c\gamma^2)$	$O(1/\epsilon^2)$
			stoch.	[37]	$\gamma^2/(1+\gamma^2)$	$\tilde{O}(1/\epsilon^5)$
				Corollary 7-b	$\gamma^2/(1+c\gamma^2)$	$O(1/\epsilon^4)$
Non-Monotone	general	∇F	stoch.	[36]	$\frac{\gamma(1-\gamma h)}{\gamma'-1}\left(\frac{1}{2}-\frac{1}{2^{\gamma'}}\right)$	$O(1/\epsilon^3)$
				[48] ‡	$(1-h)/4$	$O(1/\epsilon^2)$
				Corollary 7-d	$(1-h)/4$	$O(1/\epsilon^2)$
		F	det.	[36]	$\frac{\gamma(1-\gamma h)}{\gamma'-1}\left(\frac{1}{2}-\frac{1}{2^{\gamma'}}\right)$	$O(1/\epsilon^3)$
				Corollary 7-d	$(1-h)/4$	$O(1/\epsilon^2)$
			stoch.	[36]	$\frac{\gamma(1-\gamma h)}{\gamma'-1}\left(\frac{1}{2}-\frac{1}{2^{\gamma'}}\right)$	$O(1/\epsilon^5)$
				Corollary 7-d	$(1-h)/4$	$O(1/\epsilon^4)$

This table compares the different results for the number of oracle calls (complexity) *within the constraint set* for up-concave maximization. We refer to [36] for a more comprehensive table that includes results for deterministic first order feedback. Here $h := \min_{\mathbf{z} \in \mathcal{K}} \|\mathbf{z}\|_\infty$ and $\gamma' := \gamma + 1/\gamma$.
‡ [20], [46] and [48] use gradient ascent, requiring potentially computationally expensive projections.
All previous results assume that functions are differentiable, DR-submodular, Lipschitz and smooth (i.e., their gradients are Lipschitz). Result of [19] also requires the function Hessians to be Lipschitz. It also requires the density of the stochastic oracle to be known and the log of density to be 4 times differentiable with bounded 4th derivatives. We only require the functions to be up-concave, differentiable and Lipschitz, except for results on monotone functions over general convex sets where we do not need differentiability.

the agent chooses a points $\mathbf{y}_{t,i}$ and receives the output of the query oracle. The precise definition of agent $(\Omega^{\mathcal{A}}, \mathcal{A}^{\text{action}}, \mathcal{A}^{\text{query}})$ is given in Appendix B, with the query oracle being any of stochastic/deterministic first/zeroth order or semi-bandit/bandit.

An adversary Adv is a set of realized adversaries $\mathcal{B} = (\mathcal{B}_1, \cdots, \mathcal{B}_T)$, where each $\mathcal{B}_t$ maps $(\mathbf{x}_1, \cdots, \mathbf{x}_t) \in \mathcal{K}^T$ to $(f_t, \mathcal{Q}_t)$ where $f_t \in \mathbf{F}$ and $\mathcal{Q}_t$ is a query oracle for f_t. Adversaries can be oblivious ($\mathcal{B}_t$ are constant and independent of $(\mathbf{x}_1, \cdots, \mathbf{x}_t)$), weakly adaptive ($\mathcal{B}_t$ are independent of $\mathbf{x}_t$), or fully adaptive (no restrictions). We use $\text{Adv}_i^{\text{f}}(\mathbf{F})$ 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 $\text{Adv}_i^{\text{f}}(\mathbf{F}, B)$ to denote such an adversary. Finally, we use $\text{Adv}_i^{\text{o}}(\mathbf{F})$ and $\text{Adv}_i^{\text{o}}(\mathbf{F}, B)$ 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 $\mathcal{U} \subseteq \mathcal{K}^T$, approximation coefficient $0 < \alpha \leq 1$ and $1 \leq a \leq b \leq T$, we define *regret* as $\mathcal{R}^{\mathcal{A}}_{\alpha,\text{Adv}}(\mathcal{U})[a, b] := \sup_{\mathcal{B} \in \text{Adv}} \mathbb{E}\left[\alpha \max_{\mathbf{u}=(\mathbf{u}_1,\cdots,\mathbf{u}_T) \in \mathcal{U}} \sum_{t=a}^{b} f_t(\mathbf{u}_t) - \sum_{t=a}^{b} f_t(\mathbf{x}_t)\right]$, where the expectation in the definition of the regret is over the randomness of the algorithm and the query oracle. We use the notation $\mathcal{R}^{\mathcal{A}}_{\alpha,\mathcal{B}}(\mathcal{U})[a, b] := \mathcal{R}^{\mathcal{A}}_{\alpha,\text{Adv}}(\mathcal{U})[a, b]$ when $\text{Adv} = \{\mathcal{B}\}$ is a singleton. We may drop α when it is equal to 1. When $\alpha < 1$, we often assume that the functions are non-negative. *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* [51, 43]. *Adaptive regret* is defined as $\max_{1 \leq a \leq b \leq T} \mathcal{R}^{\mathcal{A}}_{\alpha,\text{Adv}}(\mathcal{K}^T_\star)[a, b]$ [23]. We drop a, b and $\mathcal{U}$ when the statement is independent of their value or their value is clear from the context.

3 Formulation of Upper-Quadratizable Functions and Regret Relation to that of Quadratic Functions

Let $\mathcal{K} \subseteq \mathbb{R}^d$ be a convex set, $\mathbf{F}$ be a function class over $\mathcal{K}$. We say the function class $\mathbf{F}$ is *upper-quadratizable* if there are maps $\mathfrak{g} : \mathbf{F} \times \mathcal{K} \to \mathbb{R}^d$ and $h : \mathcal{K} \to \mathcal{K}$ and constants $\mu \geq 0$, $0 < \alpha \leq 1$

Table 3: Non-stationary up-concave maximization

F	Set	Feedback			Reference	Appx.	regret type	α-regret
Monotone	$0 \in \mathcal{K}$	∇F	Full Information	stoch.	Corollary 8-c	$1-e^{-\gamma}$	dynamic	$T^{1/2}(1+P_T)^{1/2}$
					Corollary 7-c	$1-e^{-\gamma}$	adaptive	$T^{1/2}$
			Semi-bandit	stoch.	Corollary 7-c	$1-e^{-\gamma}$	adaptive	$T^{2/3}$
		F	Full Information	det.	Corollary 8-c	$1-e^{-\gamma}$	dynamic	$T^{1/2}(1+P_T)^{1/2}$
					Corollary 7-c	$1-e^{-\gamma}$	adaptive	$T^{1/2}$
				stoch.	Corollary 8-c	$1-e^{-\gamma}$	dynamic	$T^{3/4}(1+P_T)^{1/2}$
					Corollary 7-c	$1-e^{-\gamma}$	adaptive	$T^{3/4}$
			Bandit	stoch.	Corollary 7-c	$1-e^{-\gamma}$	adaptive	$T^{4/5}$
	general	∇F	Semi-bandit	stoch.	Corollary 8-b	$\gamma^2/(1+c\gamma^2)$	dynamic	$T^{1/2}(1+P_T)^{1/2}$
					Corollary 7-b	$\gamma^2/(1+c\gamma^2)$	adaptive	$T^{1/2}$
		F	Full Information	det.	Corollary 8-c	$\gamma^2/(1+c\gamma^2)$	dynamic	$T^{1/2}(1+P_T)^{1/2}$
					Corollary 7-c	$\gamma^2/(1+c\gamma^2)$	adaptive	$T^{1/2}$
			Bandit	stoch.	Corollary 8-b	$\gamma^2/(1+c\gamma^2)$	dynamic	$T^{3/4}(1+P_T)^{1/2}$
					Corollary 7-b	$\gamma^2/(1+c\gamma^2)$	adaptive	$T^{3/4}$
Non-Monotone	general	∇F	Full Information	stoch.	Corollary 8-d	$(1-h)/4$	dynamic	$T^{1/2}(1+P_T)^{1/2}$
					Corollary 7-d	$(1-h)/4$	adaptive	$T^{1/2}$
			Semi-bandit	stoch.	Corollary 7-d	$(1-h)/4$	adaptive	$T^{2/3}$
		F	Full Information	det.	Corollary 8-d	$(1-h)/4$	dynamic	$T^{1/2}(1+P_T)^{1/2}$
					Corollary 7-d	$(1-h)/4$	adaptive	$T^{1/2}$
				stoch.	Corollary 8-d	$(1-h)/4$	dynamic	$T^{3/4}(1+P_T)^{1/2}$
					Corollary 7-d	$(1-h)/4$	adaptive	$T^{3/4}$
			Bandit	stoch.	Corollary 7-d	$(1-h)/4$	adaptive	$T^{4/5}$

This table includes different results for non-stationary up-concave maximization, while no prior results exist in this setup to the best of our knowledge. The results for adaptive regret are projection-free and use a separation oracle while results for dynamic regret use convex projection. Note that full-information algorithms with deterministic feedback require 2 queries per function while the ones with stochastic feedback only require one, at the cost of higher regret.

and $\beta > 0$ such that [2]

$$\alpha f(\mathbf{y}) - f(h(\mathbf{x})) \leq \beta \left(\langle \mathfrak{g}(f, \mathbf{x}), \mathbf{y} - \mathbf{x} \rangle - \frac{\mu}{2} \|\mathbf{y} - \mathbf{x}\|^2 \right), \tag{1}$$

As a special case, when $\mu = 0$, we say $\mathbf{F}$ is *upper-linearizable*. We use the notation $\mathbf{F}_{\mu,\mathfrak{g}}$ to denote the class of functions $q(\mathbf{y}) := \langle \mathfrak{g}(f, \mathbf{x}), \mathbf{y} - \mathbf{x} \rangle - \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}\|^2 : \mathcal{K} \to \mathbb{R}$, for all $f \in \mathbf{F}$ and $\mathbf{x} \in \mathcal{K}$. Similarly, for any $B_1 > 0$, we use the notation $\mathbf{Q}_\mu[B_1]$ to denote the class of functions $q(\mathbf{y}) := \langle \mathbf{o}, \mathbf{y} - \mathbf{x} \rangle - \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}\|^2 : \mathcal{K} \to \mathbb{R}$, for all $\mathbf{x} \in \mathcal{K}$ and $\mathbf{o} \in \mathbb{B}^d_{B_1}(\mathbf{0})$. A similar notion of *lower-quadratizable/linearizable* may be similarly defined for minimization problems such as convex minimization. [3]

We say an algorithm $\mathcal{G}$ is a first order query algorithm for $\mathfrak{g}$ if, given a point $\mathbf{x} \in \mathcal{K}$ and a first order query oracle for f, it returns (a possibly unbiased estimate of) $\mathfrak{g}(f, \mathbf{x})$. We say $\mathcal{G}$ is bounded by B_1 if the output of $\mathcal{G}$ is always within the ball $\mathbb{B}^d_{B_1}(\mathbf{0})$ and we call it trivial if it simply returns the output of the query oracle at $\mathbf{x}$.

Recall that an online agent $\mathcal{A}$ is composed of action function $\mathcal{A}^{\text{action}}$ and query function $\mathcal{A}^{\text{query}}$. Informally, given an online algorithm $\mathcal{A}$ with semi-bandit feedback, we may think of $\mathcal{A}' := \texttt{OMBQ}(\mathcal{A}, \mathcal{G}, h)$ as the online algorithm with $(\mathcal{A}')^{\text{action}} \approx h(\mathcal{A}^{\text{action}})$ and $(\mathcal{A}')^{\text{query}} \approx \mathcal{G}$. As a special case, when $h = \text{Id}$ and $\mathcal{G}$ is trivial, we have $\mathcal{A}' = \mathcal{A}$.

Algorithm 1: Online Maximization By Quadratization - $\texttt{OMBQ}(\mathcal{A}, \mathcal{G}, h)$

Input : horizon T, semi-bandit algorithm $\mathcal{A}$, query algorithm $\mathcal{G}$ for $\mathfrak{g}$, the map $h : \mathcal{K} \to \mathcal{K}$

for $t = 1, 2, \ldots, T$ **do**

  Play $h(\mathbf{x}_t)$ where $\mathbf{x}_t$ is the action chosen by $\mathcal{A}$

  The adversary selects f_t and a first order query oracle for f_t

  Run $\mathcal{G}$ with access to $\mathbf{x}_t$ and the query oracle for f_t to calculate $\mathbf{o}_t$

  Return $\mathbf{o}_t$ as the output of the query oracle to $\mathcal{A}$

end

[2]Note that, without any loss in generality, we may replace $(\beta, \mu, \mathfrak{g})$ with $(1, \beta\mu, \beta\mathfrak{g})$ and therefore assume $\beta = 1$. However, we keep β as a separate variable as it makes some expressions in future sections simpler.

[3]Specifically, we say $\mathbf{F}$ is lower-quadratizable if $\alpha f(\mathbf{y}) - f(h(\mathbf{x})) \geq \beta \left(\langle \mathfrak{g}(f, \mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}\|^2 \right)$. Note that this is equivalent to $-\mathbf{F}$ being upper-quadratizable with the same α,β, and μ, but with $-\mathfrak{g}(-f, \mathbf{x})$.

Theorem 1. *Let $\mathcal{A}$ be algorithm for online optimization with semi-bandit feedback. Also let $\mathbf{F}$ be a function class over $\mathcal{K}$ that is quadratizable with $\mu \geq 0$ and maps $\mathfrak{g} : \mathbf{F} \times \mathcal{K} \to \mathbb{R}^d$ and $h : \mathcal{K} \to \mathcal{K}$, let $\mathcal{G}$ be a query algorithm for $\mathfrak{g}$ and let $\mathcal{A}' = \texttt{OMBQ}(\mathcal{A}, \mathcal{G}, h)$. Then the following are true.*

1. *If $\mathcal{G}$ returns the exact value of $\mathfrak{g}$, then we have $\mathcal{R}^{\mathcal{A}'}_{\alpha, \mathrm{Adv}^f_1(\mathbf{F})} \leq \beta \mathcal{R}^{\mathcal{A}}_{1, \mathrm{Adv}^f_1(\mathbf{F}_{\mu, \mathfrak{g}})}$.*
2. *On the other hand, if $\mathcal{G}$ returns an unbiased estimate of $\mathfrak{g}$ and the output of $\mathcal{G}$ is bounded by B_1, then we have $\mathcal{R}^{\mathcal{A}'}_{\alpha, \mathrm{Adv}^o_1(\mathbf{F}, B_1)} \leq \beta \mathcal{R}^{\mathcal{A}}_{1, \mathrm{Adv}^f_1(\mathbf{Q}_\mu[B_1])}$.*

As a special case, when f is concave, we may choose $\alpha = \beta = 1$, $h = \mathrm{Id}$, and $\mathfrak{g}(f, \mathbf{x})$ to be a super-gradient of f at $\mathbf{x}$. In this case, Theorem 1 reduces to the concave version of Theorems 2 and 5 in [34].

4 Up-concave function optimization is upper-quadratizable function optimization

In this section, we study three classes of up-concave functions and show that they are upper-quadratizable. We further use this property to obtain meta-algorithms that convert algorithms for quadratic optimization into algorithms for up-concave maximization.

4.1 Monotone up-concave optimization over general convex sets

For differentiable DR-submodular functions, the following lemma is proven for the case $\gamma = 1$ in Lemma 2 in [13] and for the case $\mu = 0$ in [20] (See Inequality 7.5 in the arXiv version). We show the result for general μ-strongly γ-weakly up-concave function with curvature bounded by c, See Appendix D for proof.

Lemma 1. *Let $f : [0,1]^d \to \mathbb{R}$ be a non-negative monotone μ-strongly γ-weakly up-concave function with curvature bounded by c. Then, for all $\mathbf{x}, \mathbf{y} \in [0,1]^d$, we have*

$$\frac{\gamma^2}{1 + c\gamma^2} f(\mathbf{y}) - f(\mathbf{x}) \leq \frac{\gamma}{1 + c\gamma^2} \left(\langle \tilde{\nabla} f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle - \frac{\mu}{2} \|\mathbf{y} - \mathbf{x}\|^2 \right),$$

where $\tilde{\nabla} f$ is an up-super-gradient for f.

Further, we show that any semi-bandit feedback online linear optimization algorithm for fully adaptive adversary is also an online up-concave optimization algorithm.

Theorem 2. *Let $\mathcal{K} \subseteq [0,1]^d$ be a convex set, let $\mu \geq 0$, $\gamma \in (0,1]$, $c \in [0,1]$ and let $\mathcal{A}$ be algorithm for online optimization with semi-bandit feedback. Also let $\mathbf{F}$ be an M_1-Lipschitz function class over $\mathcal{K}$ where every $f \in \mathbf{F}$ may be extended to a monotone μ-strongly γ-weakly up-concave function curvature bounded by c defined over $[0,1]^d$. Then, for any $B_1 \geq M_1$, we have*

$$\mathcal{R}^{\mathcal{A}}_{\frac{\gamma^2}{1+c\gamma^2}, \mathrm{Adv}^f_1(\mathbf{F})} \leq \frac{\gamma}{1 + c\gamma^2} \mathcal{R}^{\mathcal{A}}_{1, \mathrm{Adv}^f_1(\mathbf{Q}_\mu[M_1])}, \quad \mathcal{R}^{\mathcal{A}}_{\frac{\gamma^2}{1+c\gamma^2}, \mathrm{Adv}^o_1(\mathbf{F}, B_1)} \leq \frac{\gamma}{1 + c\gamma^2} \mathcal{R}^{\mathcal{A}}_{1, \mathrm{Adv}^f_1(\mathbf{Q}_\mu[B_1])}$$

These results follows immediately from Theorem 1 and Lemma 1. Note that it is important to assume that every function in $\mathbf{F}$ may be extended to a non-negative up-concave function over $[0,1]^d$ for Lemma 1 to be applied.

Corollary 1. *The results of [20], [8] and [13] on monotone continuous DR-submodular maximization over general convex sets may be thought of as special cases of Theorem 2 when $\mathcal{A}$ is the online gradient ascent algorithm.*

Algorithm 2: Boosted Query oracle for Monotone up-concave functions over convex sets containing the origin – `BQMO`

Input : First order query oracle, point $\mathbf{x}$
Sample $z \in [0,1]$ according to Equation (2)
Return the output of the query oracle at $z * \mathbf{x}$

4.2 Monotone up-concave optimization over convex sets containing the origin

The following lemma is proven for differentiable DR-submodular functions in Theorem 2 and Proposition 1 of [46]. The proof works for general up-concave functions as well. We include a proof in Appendix E for completeness.

Lemma 2. *Let $f : [0,1]^d \to \mathbb{R}$ be a non-negative monotone γ-weakly up-concave differentiable function and let $F : [0,1]^d \to \mathbb{R}$ be the function defined by*

$$F(\mathbf{x}) := \int_0^1 \frac{\gamma e^{\gamma(z-1)}}{(1-e^{-\gamma})z}(f(z * \mathbf{x}) - f(\mathbf{0}))dz.$$

Then F is differentiable and, if the random variable $\mathcal{Z} \in [0,1]$ is defined by the law

$$\forall z \in [0,1], \quad \mathbb{P}(\mathcal{Z} \leq z) = \int_0^z \frac{\gamma e^{\gamma(u-1)}}{1-e^{-\gamma}} du, \tag{2}$$

then we have $\mathbb{E}\left[\nabla f(\mathcal{Z} * \mathbf{x})\right] = \nabla F(\mathbf{x})$. Moreover, we have $(1-e^{-\gamma})f(\mathbf{y}) - f(\mathbf{x}) \leq \frac{1-e^{-\gamma}}{\gamma}\langle \nabla F(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle$.

Theorem 3. *Let $\mathcal{K} \subseteq [0,1]^d$ be a convex set containing the origin, let $\gamma \in (0,1]$ and let $\mathcal{A}$ be algorithm for online optimization with semi-bandit feedback. Also let $\mathbf{F}$ be a function class over $\mathcal{K}$ where every $f \in \mathbf{F}$ is the restriction of a monotone γ-weakly up-concave function defined over $[0,1]^d$ to the set $\mathcal{K}$. Assume $\mathbf{F}$ is differentiable and M_1-Lipschitz for some $M_1 > 0$. Then, for any $B_1 \geq M_1$, we have*

$$\mathcal{R}^{\mathcal{A}'}_{1-e^{-\gamma}, \mathrm{Adv}_1^o(\mathbf{F}, B_1)} \leq \frac{1-e^{-\gamma}}{\gamma} \mathcal{R}^{\mathcal{A}}_{1, \mathrm{Adv}_1^f(\mathbf{Q}_0[B_1])}$$

where $\mathcal{A}' = \mathtt{OMBQ}(\mathcal{A}, \mathtt{BQMO}, \mathrm{Id})$.

This result now follows immediately from Theorem 1 and Lemma 2.

Corollary 2. *The result of [46] in the online setting (when there is no delay) may be seen as an application of Theorem 3 when $\mathcal{A}$ is chosen to be online gradient ascent.*

4.3 Non-monotone up-concave optimization over general convex sets

The following lemma is proven for differentiable DR-submodular functions in Corollary 2, Theorem 4 and Proposition 2 of [48]. The arguments works for general up-concave functions as well. We include a proof in Appendix F for completeness.

Lemma 3. *Let $f : [0,1]^d \to \mathbb{R}$ be a non-negative continuous up-concave differentiable function and let $\underline{\mathbf{x}} \in \mathcal{K}$. Define $F : [0,1]^d \to \mathbb{R}$ as the function $F(\mathbf{x}) := \int_0^1 \frac{2}{3z(1-\frac{z}{2})^3}\left(f\left(\frac{z}{2} * (\mathbf{x} - \underline{\mathbf{x}}) + \underline{\mathbf{x}}\right) - f(\underline{\mathbf{x}})\right)dz$. Then F is differentiable and, if the random variable $\mathcal{Z} \in [0,1]$ is defined by the law*

$$\forall z \in [0,1], \quad \mathbb{P}(\mathcal{Z} \leq z) = \int_0^z \frac{1}{3(1-\frac{u}{2})^3} du, \tag{3}$$

then we have $\mathbb{E}\left[\nabla f\left(\frac{\mathcal{Z}}{2} * (\mathbf{x} - \underline{\mathbf{x}}) + \underline{\mathbf{x}}\right)\right] = \nabla F(\mathbf{x})$. Moreover, we have

$$\frac{1 - \|\underline{\mathbf{x}}\|_\infty}{4} f(\mathbf{y}) - f\left(\frac{\mathbf{x} + \underline{\mathbf{x}}}{2}\right) \leq \frac{3}{8}\langle \nabla F(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle.$$

Theorem 4. *Let $\mathcal{K} \subseteq [0,1]^d$ be a convex set, $\underline{\mathbf{u}} \in \mathcal{K}$, $h := \|\underline{\mathbf{u}}\|_\infty$ and $\mathcal{A}$ be algorithm for online optimization with semi-bandit feedback. Also let $\mathbf{F}$ be a function class over $\mathcal{K}$ where every $f \in \mathbf{F}$ is the restriction of an up-concave function defined over $[0,1]^d$ to the set $\mathcal{K}$. Assume $\mathbf{F}$ is differentiable and M_1-Lipschitz for some $M_1 > 0$. Then, for any $B_1 \geq M_1$ and $\mathcal{A}' = \mathtt{OMBQ}(\mathcal{A}, \mathtt{BQN}, \mathbf{x} \mapsto \frac{\mathbf{x}_t + \underline{\mathbf{x}}}{2})$, we have $\mathcal{R}^{\mathcal{A}'}_{\frac{1-h}{4}, \mathrm{Adv}_1^o(\mathbf{F}, B_1)} \leq \frac{3}{8}\mathcal{R}^{\mathcal{A}}_{1, \mathrm{Adv}_1^f(\mathbf{Q}_0[B_1])}$.*

These results now follows immediately from Theorem 1 and Lemma 3.

Corollary 3. *The result of [48] in the online setting without delay may be seen as an application of Theorem 4 when $\mathcal{A}$ is chosen to be online gradient ascent.*

Algorithm 3: Boosted Query oracle for Non-monotone up-concave functions over general convex sets – $\mathtt{BQN}$

Input : First order query oracle, point $\mathbf{x}$
Sample $z \in [0,1]$ according to Equation 3
Return the output of the query oracle at $\frac{z}{2} * (\mathbf{x} - \underline{\mathbf{x}}) + \underline{\mathbf{x}}$

5 Meta algorithms for other feedback cases

In this section, we study several meta-algorithms that allow us to convert between different feedback types and also convert results from the online setting to the offline setting.

First order/semi-bandit to zeroth order/bandit feedback: In this section we discuss meta-algorithms that convert algorithms designed for first order feedback into algorithms that can handle zeroth order feedback. These algorithms and results are generalization of similar results in [34] to the case where $\alpha < 1$.

We choose a point $\mathbf{c} \in \mathrm{relint}(\mathcal{K})$ and a real number $r > 0$ such that $\mathrm{aff}(\mathcal{K}) \cap \mathbb{B}_r^d(\mathbf{c}) \subseteq \mathcal{K}$. Then, for any shrinking parameter $0 \leq \delta < r$, we define $\hat{\mathcal{K}}_\delta := (1 - \frac{\delta}{r})\mathcal{K} + \frac{\delta}{r}\mathbf{c}$. For a function $f : \mathcal{K} \to \mathbb{R}$ defined on a convex set $\mathcal{K} \subseteq \mathbb{R}^d$, its δ-smoothed version $\hat{f}_\delta : \hat{\mathcal{K}}_\delta \to \mathbb{R}$ is given as

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

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 $\mathcal{L}_0 \cap \mathbb{B}_1^d(\mathbf{0})$. 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 subscript δ when there is no ambiguity (See Appendix G for the description of the algorithms and the proof.).

Theorem 5. *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 $\mathcal{A}' = \mathtt{FOTZO}(\mathcal{A})$ where* FOTZO *is described by Algorithm 5 and $0 < \alpha \leq 1$, we have*

$$\mathcal{R}^{\mathcal{A}'}_{\alpha, \mathrm{Adv}_0^o(\mathbf{F}, B_0)}(\mathcal{U}) \leq \mathcal{R}^{\mathcal{A}}_{\alpha, \mathrm{Adv}_1^o(\hat{\mathbf{F}}, \frac{k}{\delta} B_0)}(\hat{\mathcal{U}}) + \left(3 + \frac{2D}{r}\right) \delta M_1 T.$$

On the other hand, if we assume that $\mathcal{A}$ is semi-bandit, then the same regret bounds hold with $\mathcal{A}' = \mathtt{STB}(\mathcal{A})$, where STB *is described by Algorithm 6.*

Theorem 6. *Under the assumptions of Theorem 5, if $\mathcal{A}' = \mathtt{FOTZO\text{-}2P}(\mathcal{A})$ where* FOTZO-2P *is described by Algorithm 7 and $0 < \alpha \leq 1$, we have*

$$\mathcal{R}^{\mathcal{A}'}_{\alpha, \mathrm{Adv}_0^o(\mathbf{F})}(\mathcal{U}) \leq \mathcal{R}^{\mathcal{A}}_{\alpha, \mathrm{Adv}_1^o(\hat{\mathbf{F}}, k M_1)}(\hat{\mathcal{U}}) + \left(3 + \frac{2D}{r}\right) \delta M_1 T.$$

Full information to trivial query: In this section, we discuss a meta-algorithm that converts algorithms that require full-information feedback into algorithms that have a trivial query oracle. In particular, it converts algorithms that require first-order full-information feedback into semi-bandit algorithms and algorithms that require zeroth-order full-information feedback into bandit algorithms.

Here we assume that $\mathcal{A}^{\text{query}}$ does not depend on the observations in the current round. If the number of queries k_t is not constant for each time-step, we simply assume that $\mathcal{A}$ queries extra points and then discards them, so that we obtain an algorithm that queries exactly K points at each time-step, where K does not depend on t. We say a function class $\mathbf{F}$ is closed under convex combination if for any $f_1, \cdots, f_k \in \mathbf{F}$ and any $\delta_1, \cdots, \delta_k \geq 0$ with $\sum_i \delta_i = 1$, we have $\sum_i \delta_i f_i \in \mathbf{F}$.

Theorem 7. *Let $\mathcal{A}$ be an online optimization algorithm with full-information feedback and with K queries at each time-step where $\mathcal{A}^{query}$ does not depend on the observations in the current round and $\mathcal{A}' = \mathtt{SFTT}(\mathcal{A})$. Then, for any M_1-Lipschitz function class $\mathbf{F}$ that is closed under convex combination and any $B_1 \geq M_1$, $0 < \alpha \leq 1$ and $1 \leq a \leq b \leq T$, let $a' = \lfloor (a-1)/L \rfloor + 1$, $b' = \lceil b/L \rceil$, $D = \mathrm{diam}(\mathcal{K})$ and let $\{T\}$ and $\{T/L\}$ denote the horizon of the adversary. Then, we have*

$$\mathcal{R}^{\mathcal{A}'}_{\alpha, \mathrm{Adv}_1^o(\mathbf{F}, B_1)\{T\}}(\mathcal{K}_\star^T)[a, b] \leq M_1 D K (b' - a' + 1) + L \mathcal{R}^{\mathcal{A}}_{\alpha, \mathrm{Adv}_1^o(\mathbf{F}, B_1)\{T/L\}}(\mathcal{K}_\star^{T/L})[a', b'],$$

This result (proof in Appendix H) is based on the idea of random permutations used in [45, 47, 35].

Online to Offline: An offline optimization problem can be though of as an instance of online optimization where the adversary picks the same function and query oracle at each round. Moreover, instead of regret, the performance of the algorithm is measured by sample complexity, i.e., the minimum number of queries required so that the expected error from the α-approximation of the optimal value is less than ϵ.

Conversions of online algorithms to offline are referred to online-to-batch techniques and are well-known in the literature (See [38]). A simple approach is to simply run the online algorithm and if the actions chosen by the algorithm are $\mathbf{x}_1, \cdots, \mathbf{x}_T$, return $\mathbf{x}_t$ for $1 \leq t \leq T$ with probability $1/T$. We use OTB to denote the meta-algorithm that uses this approach to convert online algorithms to offline algorithms. The following theorem is a corollary which we include for completion. (See Appendix I for the proof.)

Theorem 8. *Let $\mathcal{A}$ be an online algorithm that queries no more than $K = T^\theta$ times per time-step that obtains an α-regret bound of $O(T^\eta)$ over an oblivious adversary* Adv. *Then the sample complexity of* $\mathtt{OTB}(\mathcal{A})$ *over* $\{(f, \mathcal{Q}_f) \mid ((f, \mathcal{Q}_f), \cdots, (f, \mathcal{Q}_f)) \in \mathrm{Adv}\}$ *is* $O(\epsilon^{-\frac{1+\theta}{1-\eta}})$.

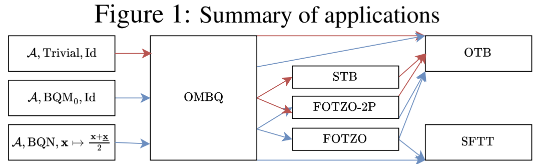
Algorithm 4: Stochastic Full-information To Trivial query - $\mathtt{SFTT}(\mathcal{A})$

Input : base algorithm $\mathcal{A}$, horizon T, block size $L > K$.
for $q = 1, 2, \ldots, T/L$ **do**
  Let $\hat{\mathbf{x}}_q$ be the action chosen by $\mathcal{A}^{\text{action}}$
  Let $(\hat{\mathbf{y}}_q^i)_{i=1}^K$ be the queries selected by $\mathcal{A}^{\text{query}}$
  Let $(t_{q,1}, \ldots, t_{q,L})$ be a random permutation of $\{(q-1)L + 1, \ldots, qL\}$
  for $t = (q-1)L + 1, \ldots, qL$ **do**
    if $t = t_{q,i}$ *for some* $1 \leq i \leq K$ **then**
      Play the action $\mathbf{x}_t = \hat{\mathbf{y}}_q^i$
      Return the observation to the query oracle as the response to the i-th query
    else
      Play the action $\mathbf{x}_t = \hat{\mathbf{x}}_q$
    end
  end
end

6 Applications

Figure 6 captures the applications that are mentioned in Tables 1, 2 and 3. The exact statements are stated in Corollaries 7 and 8 in the Appendix. To obtain a result from the graph, let $\mathcal{A}$ be one of SO-OGA or IA and select a directed path that has the following properties: (i) The path starts at one of the three nodes on the left. (ii) The path must be at least of length 1 and the edges must be the same color. (iii) If $\mathcal{A}$ is IA, the path should not contain SFTT or OTB.

Figure 1: Summary of applications



For example, if $\mathcal{A}$ = SO-OGA and the path starts at the middle node on the left, then passes through OMBQ, FOTZO, SFTT, we get $\mathtt{SFTT}(\mathtt{FOTZO}(\mathtt{OMBQ}(\mathtt{SO\text{-}OGA}, \mathtt{BQM0}, \mathrm{Id})))$, which is a projection-free algorithm (using separation oracles) with bandit feedback for monotone up-concave functions over convex sets that contain the origin. As mentioned in Table 3 and Corollary 7-(c), the adaptive regret of this algorithm is of order $O(T^{4/5})$. Note that the text written in the three nodes on the left correspond to the inputs of the meta-algorithm OMBQ. Also note that the color red corresponds to the setting where $\mathcal{G}$ is a trivial query algorithm which means that the output of OMBQ is semi-bandit.

7 Conclusions

In this work, we have presented a comprehensive framework for addressing optimization problems involving upper-quadratizable functions, encompassing both concave and DR-submodular functions across various settings and feedback types. Our contributions include the formulation of upper-quadratizable functions as a generalized class, the development of meta-algorithms for algorithmic conversions, and the derivation of new algorithms with improved static/ dynamic/ adaptive regret guarantees. Exploring more subset of classes of upper-quadratizable functions where such a framework could be applied is an important future direction.

8 Acknowledgement

This research was supported in part by the National Science Foundation under grant CCF-2149588.

References

[1] Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations Research*, 63(5):1227–1244, 2015.

[2] An Bian, Kfir Levy, Andreas Krause, and Joachim M Buhmann. Continuous DR-submodular maximization: Structure and algorithms. In *Advances in Neural Information Processing Systems*, 2017.

[3] Andrew An Bian, Baharan Mirzasoleiman, Joachim Buhmann, and Andreas Krause. Guaranteed Non-convex Optimization: Submodular Maximization over Continuous Domains. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, April 2017.

[4] Yatao Bian, Joachim Buhmann, and Andreas Krause. Optimal continuous DR-submodular maximization and applications to provable mean field inference. In *Proceedings of the 36th International Conference on Machine Learning*, June 2019.

[5] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[6] Chandra Chekuri, T.S. Jayram, and Jan Vondrak. On multiplicative weight updates for concave and submodular function maximization. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 201–210, 2015.

[7] Lin Chen, Christopher Harshaw, Hamed Hassani, and Amin Karbasi. Projection-free online optimization with stochastic gradient: From convexity to submodularity. In *Proceedings of the 35th International Conference on Machine Learning*, July 2018.

[8] Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, April 2018.

[9] Lin Chen, Mingrui Zhang, and Amin Karbasi. Projection-free bandit convex optimization. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pages 2047–2056. PMLR, 2019.

[10] Shengminjie Chen, Donglei Du, Wenguo Yang, Dachuan Xu, and Suixiang Gao. Continuous non-monotone DR-submodular maximization with down-closed convex constraint. *arXiv preprint arXiv:2307.09616*, July 2023.

[11] Amit Daniely, Alon Gonen, and Shai Shalev-Shwartz. Strongly adaptive online learning. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1405–1411. PMLR, 2015.

[12] Josip Djolonga and Andreas Krause. From map to marginals: Variational inference in Bayesian submodular models. *Advances in Neural Information Processing Systems*, 2014.

[13] Maryam Fazel and Omid Sadeghi. Fast first-order methods for monotone strongly dr-submodular maximization. In *SIAM Conference on Applied and Computational Discrete Algorithms (ACDA23)*, 2023.

[14] Yuval Filmus and Justin Ward. A tight combinatorial algorithm for submodular maximization subject to a matroid constraint. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 659–668, 2012.

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

[16] Dan Garber and Ben Kretzu. Improved regret bounds for projection-free bandit convex optimization. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 2196–2206. PMLR, 2020.

[17] Dan Garber and Ben Kretzu. New projection-free algorithms for online convex optimization with adaptive regret guarantees. In *Proceedings of Thirty Fifth Conference on Learning Theory*, pages 2326–2359. PMLR, 2022.

[18] Shuyang Gu, Chuangen Gao, Jun Huang, and Weili Wu. Profit maximization in social networks and non-monotone DR-submodular maximization. *Theoretical Computer Science*, 957:113847, 2023.

[19] Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Zebang Shen. Stochastic conditional gradient++: (non)convex minimization and continuous submodular maximization. *SIAM Journal on Optimization*, 30(4):3315–3344, 2020.

[20] Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient methods for submodular maximization. In *Advances in Neural Information Processing Systems*, 2017.

[21] Elad Hazan and Satyen Kale. Projection-free online learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1843–1850. Omnipress, 2012.

[22] Elad Hazan and Edgar Minasyan. Faster projection-free online learning. In *Proceedings of Thirty Third Conference on Learning Theory*, pages 1877–1893. PMLR, 2020.

[23] Elad Hazan and C. Seshadhri. Efficient learning algorithms for changing environments. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 393–400. Association for Computing Machinery, 2009.

[24] Shinji Ito and Ryohei Fujimaki. Large-scale price optimization via network flow. *Advances in Neural Information Processing Systems*, 2016.

[25] Duksang Lee, Nam Ho-Nguyen, and Dabeen Lee. Non-smooth, h\"older-smooth, and robust submodular maximization. *arXiv preprint arXiv:2210.06061*, 2023.

[26] Yuanyuan Li, Yuezhou Liu, Lili Su, Edmund Yeh, and Stratis Ioannidis. Experimental design networks: A paradigm for serving heterogeneous learners under networking constraints. *IEEE/ACM Transactions on Networking*, 2023.

[27] Yucheng Liao, Yuanyu Wan, Chang Yao, and Mingli Song. Improved Projection-free Online Continuous Submodular Maximization. *arXiv preprint arXiv:2305.18442*, May 2023.

[28] Zhou Lu, Nataly Brukhim, Paula Gradu, and Elad Hazan. Projection-free adaptive regret with membership oracles. In *Proceedings of The 34th International Conference on Algorithmic Learning Theory*, pages 1055–1073. PMLR, 2023.

[29] Zakaria Mhammedi. Efficient projection-free online convex optimization with membership oracle. In *Proceedings of Thirty Fifth Conference on Learning Theory*, pages 5314–5390. PMLR, 2022.

[30] Siddharth Mitra, Moran Feldman, and Amin Karbasi. Submodular+ concave. *Advances in Neural Information Processing Systems*, 2021.

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

[32] Loay Mualem and Moran Feldman. Resolving the approximability of offline and online non-monotone DR-submodular maximization over general convex sets. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, April 2023.

[33] Rad Niazadeh, Negin Golrezaei, Joshua Wang, Fransisca Susan, and Ashwinkumar Badanidiyuru. Online learning via offline greedy algorithms: Applications in market design and optimization. *Management Science*, 69(7):3797–3817, July 2023.

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

[35] Mohammad Pedramfar, Yididiya Y. Nadew, Christopher John Quinn, and Vaneet Aggarwal. Unified projection-free algorithms for adversarial DR-submodular optimization. In *The Twelfth International Conference on Learning Representations*, 2024.

[36] Mohammad Pedramfar, Christopher Quinn, and Vaneet Aggarwal. A unified approach for maximizing continuous γ-weakly DR-submodular functions. *optimization-online preprint optimization-online:25915*, 2024.

[37] Mohammad Pedramfar, Christopher John Quinn, and Vaneet Aggarwal. A unified approach for maximizing continuous DR-submodular functions. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[38] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[39] Zongqi Wan, Jialin Zhang, Wei Chen, Xiaoming Sun, and Zhijie Zhang. Bandit multi-linear dr-submodular maximization and its applications on adversarial submodular bandits. In *International Conference on Machine Learning*, 2023.

[40] Yibo Wang, Wenhao Yang, Wei Jiang, Shiyin Lu, Bing Wang, Haihong Tang, Yuanyu Wan, and Lijun Zhang. Non-stationary projection-free online learning with dynamic and adaptive regret guarantees. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(14):15671–15679, 2024.

[41] Bryan Wilder. Equilibrium computation and robust optimization in zero sum games with submodular structure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[42] Jiahao Xie, Zebang Shen, Chao Zhang, Boyu Wang, and Hui Qian. Efficient projection-free online methods with stochastic recursive gradient. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(4):6446–6453, 2020.

[43] Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive online learning in dynamic environments. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[44] Lijun Zhang, Tianbao Yang, jin, and Zhi-Hua Zhou. Dynamic regret of strongly adaptive methods. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5882–5891. PMLR, 2018.

[45] Mingrui Zhang, Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization: From full-information to bandit feedback. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[46] Qixin Zhang, Zengde Deng, Zaiyi Chen, Haoyuan Hu, and Yu Yang. Stochastic continuous submodular maximization: Boosting via non-oblivious function. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

[47] Qixin Zhang, Zengde Deng, Zaiyi Chen, Kuangqi Zhou, Haoyuan Hu, and Yu Yang. Online learning for non-monotone DR-submodular maximization: From full information to bandit feedback. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, April 2023.

[48] Qixin Zhang, Zongqi Wan, Zengde Deng, Zaiyi Chen, Xiaoming Sun, Jialin Zhang, and Yu Yang. Boosting gradient ascent for continuous DR-submodular maximization. *arXiv preprint arXiv:2401.08330*, 2024.

[49] Peng Zhao, Guanghui Wang, Lijun Zhang, and Zhi-Hua Zhou. Bandit convex optimization in non-stationary environments. *Journal of Machine Learning Research*, 22(125):1–45, 2021.

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

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

A Related works

DR-submodular maximization Two of the main methods for continuous DR-submodular maximization are *Frank-Wolfe type methods* and *Boosting based methods*. This division is based on how the approximation coefficient appears in the proof.

In Frank-Wolfe type algorithms, the approximation coefficient appears by specific choices of the Frank-Wolfe update rules. (See Lemma 8 in [35]) The specific choices of the update rules for different settings have been proposed in [3, 2, 32, 37, 10]. The momentum technique of [31] has been used to convert algorithms designed for deterministic feedback to stochastic feedback setting. [19] proposed a Frank-Wolfe variant with access to a stochastic gradient oracle with *known distribution*. Frank-Wolfe type algorithms been adapted to the online setting using Meta-Frank-Wolfe [8, 9] or using Blackwell approachablity [33]. Later [45] used a Meta-Frank-Wolfe with random permutation technique to obtain full-information results that only require a single query per function and also bandit results. This was extended to another settings by [47] and generalized to many different settings with improved regret bounds by [35].

Another approach, referred to as boosting, is to construct an alternative function such that maximization of this function results in approximate maximization of the original function. Given this definition, we may consider the result of [20, 8, 13] as the first boosting based results. However, in these cases (i.e., the case of monotone DR-submodular functions over general convex sets), the alternative function is identical to the original function. The term boosting in this context was first used in [46] for monotone functions over convex sets containing the origin, based on ideas presented in [14, 30]. This idea was used later in [39, 27] in bandit and projection-free full-information settings. Finally, in [48] a boosting based method was introduced for non-monotone functions over general convex sets.

Up-concave maximization Not all continuous DR-submodular functions are concave and not all concave functions are continuous DR-submodular. [30] considers functions that are the sum of a concave and a continuous DR-submodular function. It is well-known that continuous DR-submodular functions are concave along positive directions [5, 3]. Based on this idea, [41] defined an up-concave function as a function that is concave along positive directions. Up-concave maximization has been considered in the offline setting before, e.g. [25], but not in online setting. In this work, we focus on up-concave maximization which is a generalization of DR-submodular maximization.

Projection-free optimization In the past decade, numerous projection-free online convex optimization algorithms have emerged to tackle the computational limitations of their projection-based counterparts [21, 7, 42, 9, 22, 16, 29, 17]. In the context of DR-submodular maximization, the Frank-Wolfe type methods discussed above are projection-free.

Non-stationary regret Dynamic regret was first analyzed in [51] for first order deterministic feedback. Later [43] obtained the lower bound and optimal algorithm in this setting. This was later expanded to bandit setting in [49]. Adaptive regret was first analyzed in [23] and the first optimal algorithm for projection-free adaptive regret was proposed in [17]. We refer to [23, 1, 11, 44, 43, 50, 49, 28, 40, 17] and references therein for more details.

Optimization by quadratization The framework discussed here for analyzing online algorithms is based on the convex optimization framework introduced in [34]. We extend the framework to allows us to work with α-regret.Moreover, [34] also demonstrates that algorithms that are designed for quadratic/linear optimization with fully adaptive adversary obtain a similar regret in the convex setting. In this paper we introduce the notion of quadratizable functions generalizes this idea beyond convex functions to all quadratizable functions. (see Theorem 1) This allows us to integrate the boosting method with our framework to obtain various meta-algorithms for continuous DR-submodular maximization.

B Problem Setup in Detail

In this section, we further expand on the description in Section 2.

A *function class* is a set of real-valued functions. Given a set $\mathcal{D}$, a *function class* over $\mathcal{D}$ is a subset of all real-valued functions on $\mathcal{D}$. 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 $\alpha\mathbf{x} + (1-\alpha)\mathbf{y} \in \mathcal{K}$. For any $\mathbf{u} \in \mathcal{K}^T$, we define the path length $P_T(\mathbf{u}) := \sum_{i=1}^{T-1} \|\mathbf{u}_i - \mathbf{u}_{i+1}\|$.

A real-valued differentiable function f is called concave if $f(y) - f(x) \leq f'(x)(y-x)$, for all $x, y \in \mathrm{Dom}(f)$. More generally, given $\mu \geq 0$ and $0 < \gamma \leq 1$, we say a real-valued differentiable function is *μ-strongly γ-weakly concave* if

$$f(y) - f(x) \leq \frac{1}{\gamma}\left(f(x)'(y-x) - \frac{\mu}{2}|y-x|^2\right)$$

for all $x, y \in \mathrm{Dom}(f)$.

We say a differentiable function $f : \mathcal{K} \to \mathbb{R}$ is *μ-strongly γ-weakly up-concave* if it is μ-strongly γ-weakly concave along positive directions. Specifically if, for all $\mathbf{x} \leq \mathbf{y}$ in $\mathcal{K}$, we have

$$\gamma\left(\langle \nabla f(\mathbf{y}), \mathbf{y} - \mathbf{x}\rangle + \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}\|^2\right) \leq f(\mathbf{y}) - f(\mathbf{x}) \leq \frac{1}{\gamma}\left(\langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle - \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}\|^2\right).$$

This notion could be generalized in the following manner. We say $\tilde{\nabla} f : \mathcal{K} \to \mathbb{R}^d$ is a *μ-strongly γ-weakly up-super-gradient* of f if for all $\mathbf{x} \leq \mathbf{y}$ in $\mathcal{K}$, we have

$$\gamma\left(\langle \tilde{\nabla} f(\mathbf{y}), \mathbf{y} - \mathbf{x}\rangle + \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}\|^2\right) \leq f(\mathbf{y}) - f(\mathbf{x}) \leq \frac{1}{\gamma}\left(\langle \tilde{\nabla} f(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle - \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}\|^2\right).$$

Then we say f is μ-strongly γ-weakly up-concave if it is continuous and it has a μ-strongly γ-weakly up-super-gradient. When it is clear from the context, we simply refer to $\tilde{\nabla} f$ as an up-super-gradient for f. When $\gamma = 1$ and the above inequality holds for all $\mathbf{x}, \mathbf{y} \in \mathcal{K}$, we say f is μ-strongly concave.

A differentiable function $f : \mathcal{K} \to \mathbb{R}$ is called *continuous DR-submodular* if for all $\mathbf{x} \leq \mathbf{y}$, we have $\nabla f(\mathbf{x}) \geq \nabla f(\mathbf{y})$. More generally, we say f is *γ-weakly continuous DR-submodular* if for all $\mathbf{x} \leq \mathbf{y}$, we have $\nabla f(\mathbf{x}) \geq \gamma \nabla f(\mathbf{y})$. It follows that any γ-weakly continuous DR-submodular functions is γ-weakly up-concave.

Given a continuous monotone function $f : \mathcal{K} \to \mathbb{R}$, its curvature is defined as the smallest number $c \in [0,1]$ such that

$$f(\mathbf{y} + \mathbf{z}) - f(\mathbf{y}) \geq (1-c)(f(\mathbf{x} + \mathbf{z}) - f(\mathbf{x})),$$

for all $\mathbf{x}, \mathbf{y} \in \mathcal{K}$ and $\mathbf{z} \geq 0$ such that $\mathbf{x} + \mathbf{z}, \mathbf{y} + \mathbf{z} \in \mathcal{K}$. [4] We define the curvature of a function class $\mathbf{F}$ as the supremum of the curvature of functions in $\mathbf{F}$.

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 $\Omega^{\mathcal{A}} \times \mathcal{K}^{t-1} \times \prod_{s=1}^{t-1} (\mathcal{K} \times \mathcal{O})^{k_s}$ to $\mathbf{x}_t \in \mathcal{K}$ where we use $\mathcal{O}$ 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 $1 \leq i \leq k_t$, $\mathcal{A}_{t,i}^{\text{query}} : \Omega^{\mathcal{A}} \times \mathcal{K}^t \times \prod_{s=1}^{t-1} (\mathcal{K} \times \mathcal{O})^{k_s} \times (\mathcal{K} \times \mathcal{O})^{i-1}$ 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

[4] In the literature, the curvature is often defined for differentiable functions. When f is differentiable, we have

$$c = 1 - \inf_{\mathbf{x}, \mathbf{y} \in \mathcal{K}, 1 \leq i \leq d} \frac{[\nabla f(\mathbf{y})]_i}{[\nabla f(\mathbf{x})]_i}.$$

$k_t = 1$ 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 $\Omega^{\mathcal{A}} \times (\mathcal{K} \times \mathcal{O})^{t-1}$.

A query oracle is a function that provides the observation to the agent. Formally, a query oracle for a function f is a map $\mathcal{Q}$ defined on $\mathcal{K}$ such that for each $\mathbf{x} \in \mathcal{K}$, the $\mathcal{Q}(\mathbf{x})$ is a random variable taking value in the observation space $\mathcal{O}$. The query oracle is called a *stochastic value oracle* or *stochastic zeroth order oracle* if $\mathcal{O} = \mathbb{R}$ and $f(\mathbf{x}) = \mathbb{E}[\mathcal{Q}(\mathbf{x})]$. Similarly, it is called a *stochastic up-super-gradient oracle* or *stochastic first order oracle* if $\mathcal{O} = \mathbb{R}^d$ and $\mathbb{E}[\mathcal{Q}(\mathbf{x})]$ is a up-super-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 up-super-gradient 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. [5] 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 $\mathcal{B} \in \mathrm{Adv}$, 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 $(f_t, \mathcal{Q}_t)$ where $f_t \in \mathbf{F}$ and $\mathcal{Q}_t$ is a query oracle for f_t. 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 oracles $(\mathcal{Q}_1, \cdots, \mathcal{Q}_T)$ for these functions. We say an adversary is a *weakly adaptive* adversary if each function $\mathcal{B}_t$ described above does not depend on $\mathbf{x}_t$ and therefore may be represented as a map defined on $\mathcal{K}^{t-1}$. In this work we also consider adversaries that are *fully adaptive*, i.e., adversaries with no restriction. Clearly any oblivious adversary is a weakly adaptive adversary and any weakly adaptive adversary is a fully adaptive adversary. 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 $\mathcal{U} \subseteq \mathcal{K}^T$, approximation coefficient $0 < \alpha \leq 1$ and $1 \leq a \leq b \leq T$, we define *regret* as

$$\mathcal{R}^{\mathcal{A}}_{\alpha, \mathrm{Adv}}(\mathcal{U})[a, b] := \sup_{\mathcal{B} \in \mathrm{Adv}} \mathbb{E}\left[\alpha \max_{\mathbf{u} = (\mathbf{u}_1, \cdots, \mathbf{u}_T) \in \mathcal{U}} \sum_{t=a}^{b} f_t(\mathbf{u}_t) - \sum_{t=a}^{b} f_t(\mathbf{x}_t)\right],$$

where the expectation in the definition of the regret is over the randomness of the algorithm and the query oracle. We use the notation $\mathcal{R}^{\mathcal{A}}_{\alpha, \mathcal{B}}(\mathcal{U})[a, b] := \mathcal{R}^{\mathcal{A}}_{\alpha, \mathrm{Adv}}(\mathcal{U})[a, b]$ when $\mathrm{Adv} = \{\mathcal{B}\}$ is a singleton. We may drop α when it is equal to 1. When $\alpha < 1$, we often assume that the functions are non-negative.

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* [51, 43]. *Adaptive regret*, is defined as $\max_{1 \leq a \leq b \leq T} \mathcal{R}^{\mathcal{A}}_{\alpha, \mathrm{Adv}}(\mathcal{K}^T_\star)[a, b]$ [23]. We drop a, b and $\mathcal{U}$ when the statement is independent of their value or their value is clear from the context.

C Proof of Theorem 1

The proof is similar to the proof of Theorems 2 and 5 in [34].

Proof.

Deterministic oracle:

[5] 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.

We first consider the case where $\mathcal{G}$ is a deterministic query oracle for $\mathfrak{g}$. Let $\mathbf{o}_t = \mathfrak{g}(f_t, \mathbf{x}_t)$ denote the output of $\mathcal{G}$ at time-step t. For any realization $\mathcal{B} = (\mathcal{B}_1, \cdots, \mathcal{B}_T) \in \mathrm{Adv}_1^{\mathrm{f}}(\mathbf{F})$, we define $\mathcal{B}'_t(\mathbf{x}_1, \cdots, \mathbf{x}_t)$ to be the tuple (q_t, ∇) where

$$\mathcal{B}'_t(\mathbf{x}_1, \cdots, \mathbf{x}_t) := q_t := \mathbf{y} \mapsto \langle \mathbf{o}_t, \mathbf{y} - \mathbf{x}_t \rangle - \frac{\mu}{2} \|\mathbf{y} - \mathbf{x}_t\|^2,$$

and $\mathcal{B}' = (\mathcal{B}'_1, \cdots, \mathcal{B}'_T)$. Note that each $\mathcal{B}'_t$ is a deterministic function of $\mathbf{x}_1, \cdots, \mathbf{x}_t$ and therefore $\mathcal{B}' \in \mathrm{Adv}_1^{\mathrm{f}}(\mathbf{F}_{\mu,\mathfrak{g}})$. Since the algorithm uses semi-bandit feedback, the sequence of random vectors $(\mathbf{x}_1, \cdots, \mathbf{x}_T)$ chosen by $\mathcal{A}$ is identical between the game with $\mathcal{B}$ and $\mathcal{B}'$. Therefore, according to definition of quadratizable functions, for any $\mathbf{y} \in \mathcal{K}$, we have

$$\beta \left(q_t(\mathbf{y}) - q_t(\mathbf{x}_t) \right) = \beta \left(\langle \mathbf{o}_t, \mathbf{y} - \mathbf{x}_t \rangle - \frac{\mu}{2} \|\mathbf{y} - \mathbf{x}_t\|^2 \right) \geq \alpha f_t(\mathbf{y}) - f_t(h(\mathbf{x}_t)),$$

Therefore, we have

$$\max_{\mathbf{u} \in \mathcal{U}} \left(\alpha \sum_{t=a}^{b} f_t(\mathbf{u}_t) - \sum_{t=a}^{b} f_t(h(\mathbf{x}_t)) \right) \leq \beta \max_{\mathbf{u} \in \mathcal{U}} \left(\sum_{t=a}^{b} q_t(\mathbf{u}_t) - \sum_{t=a}^{b} q_t(\mathbf{x}_t) \right), \tag{4}$$

Hence

$$\begin{aligned}
\mathcal{R}^{\mathcal{A}'}_{\alpha, \mathrm{Adv}_1^{\mathrm{f}}(\mathbf{F})} &= \sup_{\mathcal{B} \in \mathrm{Adv}_1^{\mathrm{f}}(\mathbf{F})} \mathcal{R}^{\mathcal{A}'}_{\alpha, \mathcal{B}} \\
&= \sup_{\mathcal{B} \in \mathrm{Adv}_1^{\mathrm{f}}(\mathbf{F})} \mathbb{E} \left[\max_{\mathbf{u} \in \mathcal{U}} \left(\alpha \sum_{t=a}^{b} f_t(\mathbf{u}_t) - \sum_{t=a}^{b} f_t(h(\mathbf{x}_t)) \right) \right] \\
&\leq \sup_{\mathcal{B} \in \mathrm{Adv}_1^{\mathrm{f}}(\mathbf{F})} \mathbb{E} \left[\beta \max_{\mathbf{u} \in \mathcal{U}} \left(\sum_{t=a}^{b} q_t(\mathbf{u}_t) - \sum_{t=a}^{b} q_t(\mathbf{x}_t) \right) \right] \\
&\leq \beta \sup_{\mathcal{B}' \in \mathrm{Adv}_1^{\mathrm{f}}(\mathbf{F}_{\mu,\mathfrak{g}})} \mathcal{R}^{\mathcal{A}}_{1, \mathcal{B}'} = \beta \mathcal{R}^{\mathcal{A}}_{1, \mathrm{Adv}_1^{\mathrm{f}}(\mathbf{F}_{\mu,\mathfrak{g}})}.
\end{aligned}$$

Stochastic oracle:

Next we consider the case where $\mathcal{G}$ is a stochastic query oracle for $\mathfrak{g}$.

Let $\Omega^{\mathcal{Q}} = \Omega_1^{\mathcal{Q}} \times \cdots \times \Omega_T^{\mathcal{Q}}$ capture all sources of randomness in the query oracles of $\mathrm{Adv}_1^{\mathrm{o}}(\mathbf{F}, B_1)$, i.e., for any choice of $\theta \in \Omega^{\mathcal{Q}}$, the query oracle is deterministic. Hence for any $\theta \in \Omega^{\mathcal{Q}}$ and realized adversary $\mathcal{B} \in \mathrm{Adv}_1^{\mathrm{o}}(\mathbf{F}, B_1)$, we may consider $\mathcal{B}_\theta$ as an object similar to an adversary with a deterministic oracle. However, note that $\mathcal{B}_\theta$ does not satisfy the unbiasedness condition of the oracle, i.e., the returned value of the oracle is not necessarily the gradient of the function at that point. Recall that $\mathcal{B}_t$ maps a tuple $(\mathbf{x}_1, \cdots, \mathbf{x}_t)$ to a tuple of f_t and a stochastic query oracle for f_t. We will use $\mathbb{E}_{\Omega^{\mathcal{Q}}}$ to denote the expectation with respect to the randomness of query oracle and $\mathbb{E}_{\Omega_t^{\mathcal{Q}}}[\cdot] := \mathbb{E}_{\Omega^{\mathcal{Q}}}[\cdot | f_t, \mathbf{x}_t]$ to denote the expectation conditioned the action of the agent and the adversary. Similarly, let $\mathbb{E}_{\Omega^{\mathcal{A}}}$ denote the expectation with respect to the randomness of the agent. Let $\mathbf{o}_t$ be the random variable denoting the output of $\mathcal{G}$ at time-step t and let

$$\bar{\mathbf{o}}_t := \mathbb{E}[\mathbf{o}_t \mid f_t, \mathbf{x}_t] = \mathbb{E}_{\Omega_t^{\mathcal{Q}}}[\mathbf{o}_t] = \mathfrak{g}(f_t, \mathbf{x}_t).$$

Similar to the deterministic case, for any realization $\mathcal{B} = (f_1, \cdots, f_T) \in \mathrm{Adv}^{\mathrm{o}}(\mathbf{F})$ and any $\theta \in \Omega^{\mathcal{Q}}$, we define $\mathcal{B}'_{\theta,t}(\mathbf{x}_1, \cdots, \mathbf{x}_t)$ to be the pair (q_t, ∇) where

$$q_t := \mathbf{y} \mapsto \langle \mathbf{o}_t, \mathbf{y} - \mathbf{x}_t \rangle - \frac{\mu}{2} \|\mathbf{y} - \mathbf{x}_t\|^2.$$

We also define $\mathcal{B}'_\theta := (\mathcal{B}'_{\theta,1}, \cdots, \mathcal{B}'_{\theta,T})$. Note that a specific choice of θ is necessary to make sure that the function returned by $\mathcal{B}'_{\theta,t}$ is a deterministic function of $\mathbf{x}_1, \cdots, \mathbf{x}_t$ and not a random variable and therefore $\mathcal{B}'_\theta$ belongs to $\mathrm{Adv}_1^{\mathrm{f}}(\mathbf{Q}_\mu[B_1])$.

Since the algorithm uses (semi-)bandit feedback, given a specific value of θ, the sequence of random vectors $(\mathbf{x}_1, \cdots, \mathbf{x}_T)$ chosen by $\mathcal{A}$ is identical between the game with $\mathcal{B}_\theta$ and $\mathcal{B}'_\theta$. Therefore, for any $\mathbf{u}_t \in \mathcal{K}$, we have

$$\begin{aligned}
\alpha f_t(\mathbf{u}_t) - f_t(h(\mathbf{x}_t)) &\le \beta \left(\langle \bar{\mathbf{o}}_t, \mathbf{u}_t - \mathbf{x}_t \rangle - \frac{\mu}{2} \|\mathbf{u}_t - \mathbf{x}_t\|^2 \right) \\
&= \beta \left(\langle \mathbb{E}\left[\mathbf{o}_t \mid f_t, \mathbf{x}_t\right], \mathbf{u}_t - \mathbf{x}_t \rangle - \frac{\mu}{2} \|\mathbf{u}_t - \mathbf{x}_t\|^2 \right) \\
&= \beta \left(\mathbb{E}\left[\langle \mathbf{o}_t, \mathbf{u}_t - \mathbf{x}_t \rangle - \frac{\mu}{2} \|\mathbf{u}_t - \mathbf{x}_t\|^2 \mid f_t, \mathbf{x}_t \right] \right) \\
&= \beta \left(\mathbb{E}\left[q_t(\mathbf{u}_t) - q_t(\mathbf{x}_t) \mid f_t, \mathbf{x}_t \right] \right),
\end{aligned}$$

where the first inequality follows from the fact that f_t is up-quadratizable and $\bar{\mathbf{o}}_t = \mathfrak{g}(f_t, \mathbf{x}_t)$. Therefore we have

$$\begin{aligned}
\mathbb{E}\left[\alpha \sum_{t=a}^{b} f_t(\mathbf{u}_t) - \sum_{t=a}^{b} f_t(h(\mathbf{x}_t)) \right] &\le \beta \mathbb{E}\left[\sum_{t=a}^{b} \mathbb{E}\left[q_t(\mathbf{u}_t) - q_t(\mathbf{x}_t) | f_t, \mathbf{x}_t \right] \right] \\
&= \beta \mathbb{E}\left[\sum_{t=a}^{b} q_t(\mathbf{u}_t) - q_t(\mathbf{x}_t) \right].
\end{aligned}$$

Since $\mathcal{B}$ is oblivious, the sequence $(f_1, \cdots, f_T)$ is not affected by the randomness of query oracles or the agent. Therefore we have

$$\begin{aligned}
\mathcal{R}^{\mathcal{A}}_{\alpha,\mathcal{B}} &= \mathbb{E}\left[\alpha \max_{\mathbf{u} \in \mathcal{U}} \sum_{t=a}^{b} f_t(\mathbf{u}_t) - \sum_{t=a}^{b} f_t(h(\mathbf{x}_t)) \right] \\
&= \max_{\mathbf{u} \in \mathcal{U}} \mathbb{E}\left[\alpha \sum_{t=a}^{b} f_t(\mathbf{u}_t) - \sum_{t=a}^{b} f_t(h(\mathbf{x}_t)) \right] \\
&\le \beta \max_{\mathbf{u} \in \mathcal{U}} \mathbb{E}\left[\sum_{t=a}^{b} q_t(\mathbf{u}_t) - \sum_{t=a}^{b} q_t(\mathbf{x}_t) \right] \\
&\le \beta \mathbb{E}\left[\max_{\mathbf{u} \in \mathcal{U}} \left(\sum_{t=a}^{b} q_t(\mathbf{u}_t) - \sum_{t=a}^{b} q_t(\mathbf{x}_t) \right) \right] = \beta \mathbb{E}\left[\mathcal{R}^{\mathcal{A}}_{1,\mathcal{B}'_\theta} \right],
\end{aligned}$$

where the second inequality follows from Jensen's inequality. Hence we have

$$\begin{aligned}
\mathcal{R}^{\mathcal{A}}_{\alpha,\mathrm{Adv}^{\mathrm{o}}_1(\mathbf{F},B_1)} &= \sup_{\mathcal{B} \in \mathrm{Adv}^{\mathrm{o}}_1(\mathbf{F},B_1)} \mathcal{R}^{\mathcal{A}}_{\alpha,\mathcal{B}} \le \sup_{\mathcal{B} \in \mathrm{Adv}^{\mathrm{o}}_1(\mathbf{F},B_1), \theta \in \Omega^{\mathcal{Q}}} \beta \mathcal{R}^{\mathcal{A}}_{1,\mathcal{B}'_\theta} \\
&\le \sup_{\mathcal{B}' \in \mathrm{Adv}^{\mathrm{f}}_1(\mathbf{Q}_\mu[B_1])} \beta \mathcal{R}^{\mathcal{A}}_{1,\mathcal{B}'} = \beta \mathcal{R}^{\mathcal{A}}_{1,\mathrm{Adv}^{\mathrm{f}}_1(\mathbf{Q}_\mu[B_1])}
\end{aligned}$$

□

D Proof of Lemma 1

Proof. We have $\mathbf{x} \vee \mathbf{y} + \mathbf{x} \wedge \mathbf{y} = \mathbf{x} + \mathbf{y}$. Therefore, following the definition of curvature, we have

$$f(\mathbf{x} \vee \mathbf{y}) - f(\mathbf{y}) \ge (1 - c)(f(\mathbf{x}) - f(\mathbf{x} \wedge \mathbf{y})).$$

Since f is non-negative, this implies that

$$\begin{aligned}
&(f(\mathbf{x} \vee \mathbf{y}) - f(\mathbf{x})) + \frac{1}{\gamma^2}(f(\mathbf{x} \wedge \mathbf{y}) - f(\mathbf{x})) \\
&\qquad = (f(\mathbf{y}) - f(\mathbf{x})) + (f(\mathbf{x} \vee \mathbf{y}) - f(\mathbf{y})) + \frac{1}{\gamma^2}(f(\mathbf{x} \wedge \mathbf{y}) - f(\mathbf{x})) \\
&\qquad \geq (f(\mathbf{y}) - f(\mathbf{x})) + (1 - c - \frac{1}{\gamma^2})(f(\mathbf{x}) - f(\mathbf{x} \wedge \mathbf{y})) \qquad (5) \\
&\qquad = f(\mathbf{y}) - (c + \frac{1}{\gamma^2})f(\mathbf{x}) + (-1 + c + \frac{1}{\gamma^2})f(\mathbf{x} \wedge \mathbf{y}) \\
&\qquad \geq f(\mathbf{y}) - (c + \frac{1}{\gamma^2})f(\mathbf{x}).
\end{aligned}$$

On the other hand, according to the definition, we have

$$\begin{aligned}
f(\mathbf{x} \vee \mathbf{y}) - f(\mathbf{x}) &\leq \frac{1}{\gamma}\left(\langle \tilde{\nabla} f(\mathbf{x}), \mathbf{x} \vee \mathbf{y} - \mathbf{x} \rangle - \frac{\mu}{2}\|\mathbf{x} \vee \mathbf{y} - \mathbf{x}\|^2\right), \\
f(\mathbf{x}) - f(\mathbf{x} \wedge \mathbf{y}) &\geq \gamma\left(\langle \tilde{\nabla} f(\mathbf{x}), \mathbf{x} - \mathbf{x} \wedge \mathbf{y} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{x} \wedge \mathbf{y}\|^2\right).
\end{aligned}$$

Therefore, using Inequality 5 and the fact that $f(\mathbf{x} \wedge \mathbf{y}) \geq 0$, we see that

$$\begin{aligned}
f(\mathbf{y}) - \frac{1 + c\gamma^2}{\gamma^2} f(\mathbf{x}) &\leq (f(\mathbf{x} \vee \mathbf{y}) - f(\mathbf{x})) + \frac{1}{\gamma^2}(f(\mathbf{x} \wedge \mathbf{y}) - f(\mathbf{x})) \\
&\leq \frac{1}{\gamma}\Big(\langle \tilde{\nabla} f(\mathbf{x}), \mathbf{x} \vee \mathbf{y} - \mathbf{x} \rangle - \frac{\mu}{2}\|\mathbf{x} \vee \mathbf{y} - \mathbf{x}\|^2 + \langle \tilde{\nabla} f(\mathbf{x}), \mathbf{x} \wedge \mathbf{y} - \mathbf{x} \rangle \\
&\qquad - \frac{\mu}{2}\|\mathbf{x} - \mathbf{x} \wedge \mathbf{y}\|^2\Big) \\
&= \frac{1}{\gamma}\left(\langle \tilde{\nabla} f(\mathbf{x}), \mathbf{x} \vee \mathbf{y} + \mathbf{x} \wedge \mathbf{y} - 2\mathbf{x} \rangle - \frac{\mu}{2}\|\mathbf{x} \vee \mathbf{y} - \mathbf{x}\|^2 - \frac{\mu}{2}\|\mathbf{x} - \mathbf{x} \wedge \mathbf{y}\|^2\right) \\
&= \frac{1}{\gamma}\left(\langle \tilde{\nabla} f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle - \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2\right),
\end{aligned}$$

where we used $\mathbf{x} \vee \mathbf{y} + \mathbf{x} \wedge \mathbf{y} = \mathbf{x} + \mathbf{y}$ and

$$\begin{aligned}
\|\mathbf{x} \vee \mathbf{y} - \mathbf{x}\|^2 + \|\mathbf{x} - \mathbf{x} \wedge \mathbf{y}\|^2 &= \sum_{[y]_i \geq [x]_i} (y_i - x_i)^2 + \sum_{[y]_i < [x]_i} (x_i - y_i)^2 \\
&= \sum_i (x_i - y_i)^2 = \|\mathbf{x} - \mathbf{y}\|^2
\end{aligned}$$

in the last equality. The claim now follows from multiply both sides by $\frac{\gamma^2}{1+c\gamma^2}$. $\square$

E Proof of Lemma 2

Proof. Clearly we have $F(\mathbf{0}) = 0$. For any $\mathbf{x} \neq \mathbf{0}$, the integrand in the definition of F is a continuous non-negative function of z that is bounded by

$$\frac{\gamma e^{\gamma(z-1)}}{(1 - e^{-\gamma})z}(f(z * \mathbf{x}) - f(\mathbf{0})) \leq \frac{\gamma e^{\gamma(z-1)}}{(1 - e^{-\gamma})z} M_1 \|z * \mathbf{x}\| \leq \frac{\gamma}{1 - e^{-\gamma}} M_1 \|\mathbf{x}\|.$$

Therefore F is well-defined on $[0, 1]^d$. Moreover, we have [6]

$$\int_0^1 \frac{\gamma e^{\gamma(z-1)}}{(1 - e^{-\gamma})} \nabla f(z * \mathbf{x}) dz = \nabla \int_0^1 \frac{\gamma e^{\gamma(z-1)}}{(1 - e^{-\gamma})z}(f(z * \mathbf{x}) - f(\mathbf{0})) dz = \nabla F(\mathbf{x}).$$

[6]Note that we do not require the gradient ∇f to be defined everywhere for this equality to hold. It is sufficient for ∇f to exist at Lebesgue almost every point on every line segment. This is satisfied when the 1-dimensional Hausdorff measure of the set $\{\mathbf{x} \in [0, 1]^d \mid \nabla f \text{ is undefined}\}$ is zero.

It follows that F is differentiable everywhere and $\mathbb{E}\left[\nabla f(\mathcal{Z} * \mathbf{x})\right] = \nabla F(\mathbf{x})$. To prove the last claim, first we note that

$$\begin{aligned}
\frac{1-e^{-\gamma}}{\gamma}\langle \nabla F(\mathbf{x}), \mathbf{x}\rangle &= \left\langle \int_0^1 e^{\gamma(z-1)} \nabla f(z * \mathbf{x}) dz, \mathbf{x} \right\rangle \\
&= \int_0^1 e^{\gamma(z-1)} \langle \nabla f(z * \mathbf{x}), \mathbf{x}\rangle dz \\
&= \int_0^1 e^{\gamma(z-1)} df(z * \mathbf{x}) \\
&= e^{\gamma(z-1)} f(z * \mathbf{x})|_{z=0}^{z=1} - \int_0^1 f(z * \mathbf{x}) \frac{de^{\gamma(z-1)}}{dz} dz \\
&= f(\mathbf{x}) - f(\mathbf{0}) - \int_0^1 \gamma e^{\gamma(z-1)} f(z * \mathbf{x}) dz.
\end{aligned}$$

On the other hand, using monotonicity and up-concavity of f, we have

$$\begin{aligned}
\frac{1-e^{-\gamma}}{\gamma}\langle \nabla F(\mathbf{x}), \mathbf{y}\rangle &= \int_0^1 e^{\gamma(z-1)} \langle \nabla f(z * \mathbf{x}), \mathbf{y}\rangle dz \\
&\geq \int_0^1 e^{\gamma(z-1)} \langle \nabla f(z * \mathbf{x}), \mathbf{y} \vee (z * \mathbf{x}) - z * \mathbf{x}\rangle dz \\
&\geq \int_0^1 \gamma e^{\gamma(z-1)} \left(f(\mathbf{y} \vee (z * \mathbf{x})) - f(z * \mathbf{x}) \right) dz \\
&\geq \int_0^1 \gamma e^{\gamma(z-1)} \left(f(\mathbf{y}) - f(z * \mathbf{x}) \right) dz \\
&= (1 - e^{-\gamma}) f(\mathbf{y}) - \left(\int_0^1 \gamma e^{\gamma(z-1)} f(z * \mathbf{x}) dz \right),
\end{aligned}$$

where we used $\int_0^1 e^{\gamma(z-1)} \gamma dz = 1 - e^{-\gamma}$ in the last equality. Therefore

$$\frac{1-e^{-\gamma}}{\gamma}\langle \nabla F(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle \geq (1 - e^{-\gamma}) f(\mathbf{y}) - f(\mathbf{x}) + f(\mathbf{0}) \geq (1 - e^{-\gamma}) f(\mathbf{y}) - f(\mathbf{x}). \qquad \square$$

F Proof of Lemma 3

We start by extending a useful lemma from the literature. Versions of the following lemma for DR-submodular functions appeared in [2, 6, 32] and it was later extended to γ-weakly DR-submodular functions in [36]. Here we further extend it to γ-weakly up-concave functions. The proof is similar, but we include it for completeness.

Lemma 4. *For any two vectors $\mathbf{x}, \mathbf{y} \in [0, 1]^d$ and any continuously differentiable non-negative γ-weakly up-concave function f we have*

$$f(\mathbf{x} \vee \mathbf{y}) \geq (1 - \gamma \|\mathbf{x}\|_\infty) f(\mathbf{y}).$$

Proof of Lemma 4. If $\|\mathbf{x}\|_\infty = 0$, then $\mathbf{x}$ is the zero vector, and the lemma is trivially true. On the other hand, if $\mathbf{x} \vee \mathbf{y} = \mathbf{y}$, the lemma follows from non-negativity of f. Thus, we may assume that $\mathbf{z} := \mathbf{x} \vee \mathbf{y} - \mathbf{y} > \mathbf{0}$ and $\|\mathbf{x}\|_\infty > 0$. We have

$$\begin{aligned}
f(\mathbf{x} \vee \mathbf{y}) - f(\mathbf{y}) &= \int_0^1 \left. \frac{df(\mathbf{y} + r \cdot \mathbf{z})}{dr} \right|_{r=t} dt = \int_0^1 \langle \mathbf{z}, \nabla f(\mathbf{y} + t \cdot \mathbf{z})\rangle dt \\
&= \|\mathbf{x}\|_\infty \cdot \int_0^{1/\|\mathbf{x}\|_\infty} \langle \mathbf{z}, \nabla f(\mathbf{y} + \|\mathbf{x}\|_\infty \cdot t' \cdot \mathbf{z})\rangle dt' \\
&\geq \|\mathbf{x}\|_\infty \cdot \int_0^{1/\|\mathbf{x}\|_\infty} \langle \mathbf{z}, \gamma \nabla f(\mathbf{y} + t' \cdot \mathbf{z})\rangle dt', \qquad (6)
\end{aligned}$$

where (6) holds by changing the integration variable to $t' = t/\|\mathbf{x}\|_\infty$, and the inequality follows from γ-weakly up-concavity of f, in particular because f is γ-weakly concave along the line segment $[\mathbf{y}, \mathbf{y} + t'.\mathbf{z}] \subseteq [0,1]^d$. To see that the last inclusion holds, note that, for every $1 \le i \le d$, if $x_i \le y_i$, then $y_i + t' \cdot z_i = y_i \le 1$, and if $x_i \ge y_i$, then

$$y_i + t' \cdot z_i \le y_i + \frac{z_i}{\|\mathbf{x}\|_\infty} = y_i + \frac{x_i - y_i}{\|\mathbf{x}\|_\infty} \le \frac{x_i}{\|\mathbf{x}\|_\infty} \le 1.$$

Next we see that

$$\begin{aligned}\int_0^{1/\|\mathbf{x}\|_\infty} \langle \mathbf{z}, \gamma \nabla f(\mathbf{y} + t' \cdot \mathbf{z})\rangle dt' &= \gamma \int_0^{1/\|\mathbf{x}\|_\infty} \langle \mathbf{z}, \nabla f(\mathbf{y} + t' \cdot \mathbf{z})\rangle dt' \\ &= \gamma \int_0^{1/\|\mathbf{x}\|_\infty} \left. \frac{df(\mathbf{y} + r \cdot \mathbf{z})}{dr} \right|_{r=t'} dt' \\ &= \gamma f\left(\mathbf{y} + \frac{\mathbf{z}}{\|\mathbf{x}\|_\infty}\right) - \gamma f(\mathbf{y}) \ge -\gamma f(\mathbf{y}),\end{aligned}$$

where the inequality follows from non-negativity of f. The lemma now follows by plugging this inequality into Inequality (6) and rearranging the terms. $\square$

Proof of Lemma 3. Clearly we have $F(\underline{\mathbf{x}}) = 0$. For any $\mathbf{x} \neq \underline{\mathbf{x}}$, the integrand in the definition of F is a continuous function of z that is bounded by

$$\left| \frac{2}{3z(1-\frac{z}{2})^3} \left(f\left(\frac{z}{2} * (\mathbf{x} - \underline{\mathbf{x}}) + \underline{\mathbf{x}}\right) - f(\underline{\mathbf{x}}) \right) \right| \le \frac{2}{3z(1-\frac{z}{2})^3} M_1 \|\frac{z}{2} * (\mathbf{x} - \underline{\mathbf{x}})\| \le \frac{8}{3} M_1 \|\mathbf{x} - \underline{\mathbf{x}}\|.$$

Therefore F is well-defined on $[0,1]^d$. Moreover, we have [7]

$$\begin{aligned}\int_0^1 \frac{1}{3(1-\frac{z}{2})^3} \nabla f\left(\frac{z}{2} * (\mathbf{x} - \underline{\mathbf{x}}) + \underline{\mathbf{x}}\right) dz &= \nabla \int_0^1 \frac{2}{3z(1-\frac{z}{2})^3} \left(f\left(\frac{z}{2} * (\mathbf{x} - \underline{\mathbf{x}}) + \underline{\mathbf{x}}\right) - f(\underline{\mathbf{x}}) \right) dz \\ &= \nabla F(\mathbf{x})\end{aligned}$$

It follows that F is differentiable everywhere and $\mathbb{E}\left[\nabla f\left(\frac{\mathcal{Z}}{2} * (\mathbf{x} - \underline{\mathbf{x}}) + \underline{\mathbf{x}}\right)\right] = \nabla F(\mathbf{x})$.

To prove the last claim, let $\mathbf{x}_z := \frac{z}{2} * (\mathbf{x} - \underline{\mathbf{x}}) + \underline{\mathbf{x}}$ and $\omega(z) = \frac{1}{8(1-\frac{z}{2})^3}$. We have

$$\begin{aligned}\frac{3}{8}\langle \nabla F(\mathbf{x}), \mathbf{y}\rangle &= \int_0^1 \omega(z) \langle \nabla f(\mathbf{x}_z), \mathbf{y}\rangle dz \\ &= \int_0^1 \omega(z) \big(\langle \nabla f(\mathbf{x}_z), \mathbf{y} - \mathbf{x}_z \wedge \mathbf{y}\rangle + \langle \nabla f(\mathbf{x}_z), \mathbf{x}_z \wedge \mathbf{y} - \mathbf{x}_z\rangle + \langle \nabla f(\mathbf{x}_z), \mathbf{x}_z\rangle \big) dz \\ &= \int_0^1 \omega(z) \big(\langle \nabla f(\mathbf{x}_z), \mathbf{x}_z \vee \mathbf{y} - \mathbf{x}_z\rangle + \langle \nabla f(\mathbf{x}_z), \mathbf{x}_z \wedge \mathbf{y} - \mathbf{x}_z\rangle + \langle \nabla f(\mathbf{x}_z), \mathbf{x}_z\rangle \big) dz \\ &\ge \int_0^1 \omega(z) \big((f(\mathbf{x}_z \vee \mathbf{y}) - f(\mathbf{x}_z)) + (f(\mathbf{x}_z \wedge \mathbf{y}) - f(\mathbf{x}_z)) + \langle \nabla f(\mathbf{x}_z), \mathbf{x}_z\rangle \big) dz \\ &\ge \int_0^1 \omega(z) \big(f(\mathbf{x}_z \vee \mathbf{y}) - 2f(\mathbf{x}_z) + \langle \nabla f(\mathbf{x}_z), \mathbf{x}_z\rangle \big) dz.\end{aligned}$$

[7] Similar to Lemma 2, we do not require the gradient ∇f to be defined everywhere for this equality to hold. It is sufficient for ∇f to exist at Lebesgue almost every point on every line segment. This is satisfied when the 1-dimensional Hausdorff measure of the set $\{\mathbf{x} \in [0,1]^d \mid \nabla f \text{ is undefined}\}$ is zero.

Using Lemma 4, we have

$$
\begin{aligned}
f(\mathbf{x}_z \vee \mathbf{y}) &\geq (1 - \|\mathbf{x}_z\|_\infty) f(y) \\
&\geq \left(1 - \left(\left(1 - \frac{z}{2}\right)\|\underline{\mathbf{x}}\|_\infty + \frac{z}{2}\|\mathbf{x}\|_\infty\right)\right) f(y) \\
&\geq \left(1 - \left(\left(1 - \frac{z}{2}\right)\|\underline{\mathbf{x}}\|_\infty + \frac{z}{2}\right)\right) f(y) \\
&= \left(1 - \frac{z}{2}\right)\left(1 - \|\underline{\mathbf{x}}\|_\infty\right) f(y).
\end{aligned}
$$

Therefore

$$
\frac{3}{8}\langle \nabla F(\mathbf{x}), \mathbf{y}\rangle \geq \int_0^1 \omega(z)\left(\left(1 - \frac{z}{2}\right)\left(1 - \|\underline{\mathbf{x}}\|_\infty\right) f(y) - 2f(\mathbf{x}_z) + \langle \nabla f(\mathbf{x}_z), \mathbf{x}_z\rangle\right) dz. \tag{7}
$$

Next we bound $\langle \nabla F(\mathbf{x}), \mathbf{x}\rangle$. We have

$$
\frac{3}{8}\langle \nabla F(\mathbf{x}), \mathbf{x}\rangle = \int_0^1 \omega(z)\langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{x}_z\rangle dz + \int_0^1 \omega(z)\langle \nabla f(\mathbf{x}), \mathbf{x}_z\rangle dz
$$

For the first term, we have

$$
\begin{aligned}
\int_0^1 \omega(z)\langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{x}_z\rangle dz &= \int_0^1 \omega(z)\left\langle \nabla f(\mathbf{x}), \left(1 - \frac{z}{2}\right)(\mathbf{x} - \underline{\mathbf{x}})\right\rangle dz \\
&= \int_0^1 (2 - z)\omega(z)\left\langle \nabla f(\mathbf{x}), \frac{\mathbf{x} - \underline{\mathbf{x}}}{2}\right\rangle dz \\
&= \int_0^1 (2 - z)\omega(z) df(\mathbf{x}_z) \\
&= (2 - z)\omega(z) f(\mathbf{x}_z)|_{z=0}^1 - \int_0^1 ((2 - z)\omega'(z) - \omega(z)) f(\mathbf{x}_z) dz \\
&= f(\mathbf{x}_1) - \frac{1}{4} f(\underline{\mathbf{x}}) - \int_0^1 \frac{1}{4(1 - \frac{z}{2})^3} f(\mathbf{x}_z) dz,
\end{aligned}
$$

which implies that

$$
\begin{aligned}
\frac{3}{8}\langle \nabla F(\mathbf{x}), \mathbf{x}\rangle &= f(\mathbf{x}_1) - \frac{1}{4} f(\underline{\mathbf{x}}) - \int_0^1 \frac{1}{4(1 - \frac{z}{2})^3} f(\mathbf{x}_z) dz + \int_0^1 \omega(z)\langle \nabla f(\mathbf{x}), \mathbf{x}_z\rangle dz \\
&\leq f(\mathbf{x}_1) - \int_0^1 2\omega(z) f(\mathbf{x}_z) dz + \int_0^1 \omega(z)\langle \nabla f(\mathbf{x}), \mathbf{x}_z\rangle dz
\end{aligned} \tag{8}
$$

Combining Equations 7 and 8, we have

$$
\begin{aligned}
\frac{3}{8}\langle \nabla F(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle &\geq \int_0^1 \omega(z)\left(\left(1 - \frac{z}{2}\right)\left(1 - \|\underline{\mathbf{x}}\|_\infty\right) f(y) - 2f(\mathbf{x}_z) + \langle \nabla f(\mathbf{x}_z), \mathbf{x}_z\rangle\right) dz \\
&\quad - f(\mathbf{x}_1) + \int_0^1 2\omega(z) f(\mathbf{x}_z) dz - \int_0^1 \omega(z)\langle \nabla f(\mathbf{x}), \mathbf{x}_z\rangle dz \\
&= \frac{1 - \|\underline{\mathbf{x}}\|_\infty}{4} f(\mathbf{y}) \int_0^1 4\left(1 - \frac{z}{2}\right)\omega(z) dz - f(\mathbf{x}_1) \\
&= \frac{1 - \|\underline{\mathbf{x}}\|_\infty}{4} f(\mathbf{y}) - f\left(\frac{\mathbf{x} + \underline{\mathbf{x}}}{2}\right).
\end{aligned}
$$

□

G Proof of Theorem 5

The algorithms are special cases of Algorithms 2 and 3 in [34] where the shrinking parameter and the smoothing parameter are equal. We include a description of the algorithms for completion.

Algorithm 5: First order to zeroth order - `FOTZO`$(\mathcal{A})$

Input : Shrunk domain $\hat{\mathcal{K}}_\delta$, Linear space $\mathcal{L}_0$, smoothing parameter $\delta \leq r$, 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**
  $\mathbf{x}_t \leftarrow$ the action chosen by $\mathcal{A}$
  Play $\mathbf{x}_t$
  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 oracle at the point $\mathbf{y}_{t,i} + \delta \mathbf{v}_{t,i}$ to get $o_{t,i}$
    Pass $\frac{k}{\delta} o_t \mathbf{v}_t$ as the oracle output to $\mathcal{A}$
  end
end

Algorithm 6: Semi-bandit to bandit - `STB`$(\mathcal{A})$

Input : Shrunk domain $\hat{\mathcal{K}}_\delta$, Linear space $\mathcal{L}_0$, smoothing parameter $\delta \leq r$, 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**
  Sample $\mathbf{v}_t \in \mathbb{S}^1 \cap \mathcal{L}_0$ uniformly
  $\mathbf{x}_t \leftarrow$ the action chosen by $\mathcal{A}$
  Play $\mathbf{x}_t + \delta \mathbf{v}_t$
  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

Algorithm 7: 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 $\delta \leq r$, 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**
  $\mathbf{x}_t \leftarrow$ the action chosen by $\mathcal{A}$
  Play $\mathbf{x}_t$
  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 $\frac{k}{2\delta} \left(f_t(\mathbf{y}_{t,i} + \delta \mathbf{v}_{t,i}) - f_t(\mathbf{y}_{t,i} - \delta \mathbf{v}_{t,i}) \right) \mathbf{v}_t$ as the oracle output to $\mathcal{A}$
  end
end

The proof of Theorems 5 and 6 are similar to the proof of Theorems 6, 7 and 8 in [34]. The only difference being that we prove the result for α-regret instead of regret. We include a proof for completion.

Proof of Theorem 5.

Regret bound for `STB`:

Note that any realized adversary $\mathcal{B} \in \mathrm{Adv}_0^{\mathrm{o}}(\mathbf{F}, B_0)$ may be represented as a sequence of functions $(f_1, \cdots, f_T)$ and a corresponding sequence of query oracles $(\mathcal{Q}_1, \cdots, \mathcal{Q}_T)$. For such realized adversary $\mathcal{B}$, we define $\hat{B}$ to be the realized adversary corresponding to $(\hat{f}_1, \cdots, \hat{f}_T)$ with the stochastic gradient oracles

$$\hat{\mathcal{Q}}_t(\mathbf{x}) := \frac{k}{\delta}\mathcal{Q}_t(\mathbf{x} + \delta\mathbf{v})\mathbf{v}, \tag{9}$$

where $\mathbf{v}$ is a random vector, taking its values uniformly from $\mathbb{S}^1 \cap \mathcal{L}_0 = \mathbb{S}^1 \cap (\mathrm{aff}(\mathcal{K}) - \mathbf{z})$, for any $\mathbf{z} \in \mathcal{K}$ and $k = \dim(\mathcal{L}_0)$. Since $\mathcal{Q}_t$ is a stochastic value oracle for f_t, according to Remark 4 in [37], $\hat{\mathcal{Q}}_t(\mathbf{x})$ is an unbiased estimator of $\nabla \hat{f}_t(\mathbf{x})$. [8] Hence we have $\hat{B} \in \mathrm{Adv}_1^{\mathrm{o}}(\hat{\mathbf{F}}, \frac{k}{\delta}B_0)$. Using Equation 9 and the definition of the Algorithm 6, we see that the responses of the queries are the same between the game $(\mathcal{A}, \hat{\mathcal{B}})$ and $(\mathcal{A}', \mathcal{B})$. It follows that the sequence of actions $(\mathbf{x}_1, \cdots, \mathbf{x}_T)$ in $(\mathcal{A}, \hat{\mathcal{B}})$ corresponds to the sequence of actions $(\mathbf{x}_1 + \delta\mathbf{v}_1, \cdots, \mathbf{x}_T + \delta\mathbf{v}_T)$ in $(\mathcal{A}', \mathcal{B})$.

Let $\mathbf{u} \in \mathrm{argmax}_{\mathbf{u}\in\mathcal{U}} \sum_{t=a}^{b} f_t(\mathbf{u}_t)$ and $\hat{\mathbf{u}} \in \mathrm{argmax}_{\mathbf{u}\in\hat{\mathcal{U}}} \sum_{t=a}^{b} \hat{f}_t(\mathbf{u}_t)$. We have

$$\begin{aligned}
\mathcal{R}_{\alpha,\mathcal{B}}^{\mathcal{A}'} - \mathcal{R}_{\alpha,\hat{\mathcal{B}}}^{\mathcal{A}} &= \mathbb{E}\left[\alpha\sum_{t=a}^{b} f_t(\mathbf{u}_t) - \sum_{t=a}^{b} f_t(\mathbf{x}_t + \delta\mathbf{v}_t)\right] - \mathbb{E}\left[\alpha\sum_{t=a}^{b} \hat{f}_t(\hat{\mathbf{u}}_t) - \sum_{t=a}^{b} \hat{f}_t(\mathbf{x}_t)\right] \\
&= \mathbb{E}\left[\left(\sum_{t=a}^{b} \hat{f}_t(\mathbf{x}_t) - \sum_{t=a}^{b} f_t(\mathbf{x}_t + \delta\mathbf{v}_t)\right) + \alpha\left(\sum_{t=a}^{b} f_t(\mathbf{u}_t) - \sum_{t=a}^{b} \hat{f}_t(\hat{\mathbf{u}}_t)\right)\right].
\end{aligned} \tag{10}$$

According to Lemma 3 in [37], we have $|\hat{f}_t(\mathbf{x}_t) - f_t(\mathbf{x}_t)| \leq \delta M_1$. By using Lipschitz property for the pair $(\mathbf{x}_t, \mathbf{x}_t + \delta\mathbf{v}_t)$, we see that

$$|f_t(\mathbf{x}_t + \delta\mathbf{v}_t) - \hat{f}_t(\mathbf{x}_t)| \leq |f_t(\mathbf{x}_t + \delta\mathbf{v}_t) - f_t(\mathbf{x}_t)| + |f_t(\mathbf{x}_t) - \hat{f}_t(\mathbf{x}_t)| \leq 2\delta M_1. \tag{11}$$

On the other hand, we have

$$\begin{aligned}
\sum_{t=a}^{b} \hat{f}_t(\hat{\mathbf{u}}_t) &= \max_{\hat{\mathbf{u}}\in\hat{\mathcal{U}}} \sum_{t=a}^{b} \hat{f}_t(\hat{\mathbf{u}}_t) \\
&\geq -\delta M_1 T + \max_{\hat{\mathbf{u}}\in\hat{\mathcal{U}}} \sum_{t=a}^{b} f_t(\hat{\mathbf{u}}_t) && \text{(Lemma 3 in [37])} \\
&= -\delta M_1 T + \max_{\mathbf{u}\in\mathcal{U}} \sum_{t=a}^{b} f_t\left(\left(1 - \frac{\delta}{r}\right)\mathbf{u}_t + \frac{\delta}{r}\mathbf{c}\right) && \text{(Definition of } \hat{\mathcal{U}}\text{)} \\
&= -\delta M_1 T + \max_{\mathbf{u}\in\mathcal{K}} \sum_{t=a}^{b} f_t\left(\mathbf{u}_t + \frac{\delta}{r}(\mathbf{c} - \mathbf{x})\right) \\
&\geq -\delta M_1 T + \max_{\mathbf{u}\in\mathcal{U}} \sum_{t=a}^{b} \left(f_t(\mathbf{u}_t) - \frac{2\delta M_1 D}{r}\right) && \text{(Lipschitz)} \\
&= -\left(1 + \frac{2D}{r}\right)\delta M_1 T + \sum_{t=a}^{b} f_t(\mathbf{u}_t)
\end{aligned}$$

Therefore, using Equation 10, we see that

$$\mathcal{R}_{\mathcal{B}}^{\mathcal{A}'} - \mathcal{R}_{\hat{\mathcal{B}}}^{\mathcal{A}} \leq 2\delta M_1 T + \alpha\left(1 + \frac{2D}{r}\right)\delta M_1 T \leq \left(3 + \frac{2D}{r}\right)\delta M_1 T.$$

[8]When using a spherical estimator, it was shown in [15] that $\hat{f}$ is differentiable even when f is not. When using a sliced spherical estimator as we do here, differentiability of $\hat{f}$ is not proved in [37]. However, their proof is based on the proof for the spherical case and therefore the results carry forward to show that $\hat{f}$ is differentiable.

Therefore, we have

$$\begin{aligned}
\mathcal{R}^{\mathcal{A}'}_{\mathrm{Adv}^{\mathrm{o}}_0(\mathbf{F},B_0)} &= \sup_{\mathcal{B}\in\mathrm{Adv}^{\mathrm{o}}_0(\mathbf{F},B_0)} \mathcal{R}^{\mathcal{A}'}_{\mathcal{B}} \\
&\le \sup_{\mathcal{B}\in\mathrm{Adv}^{\mathrm{o}}_0(\mathbf{F},B_0)} \mathcal{R}^{\mathcal{A}}_{\hat{\mathcal{B}}} + \left(3 + \frac{2D}{r}\right)\delta M_1 T \\
&\le \sup_{\mathcal{B}\in\mathrm{Adv}^{\mathrm{o}}_0(\mathbf{F},B_0)} \mathcal{R}^{\mathcal{A}}_{\hat{\mathcal{B}}} + \left(3 + \frac{2D}{r}\right)\delta M_1 T \\
&\le \mathcal{R}^{\mathcal{A}}_{\mathrm{Adv}^{\mathrm{o}}_1(\hat{\mathbf{F}},\frac{k}{\delta}B_0)} + \left(3 + \frac{2D}{r}\right)\delta M_1 T.
\end{aligned}$$

Regret bound for FOTZO:

The proof of the bounds for this case is similar to the previous case. As before, we see that the responses of the queries are the same between the game $(\mathcal{A},\hat{\mathcal{B}})$ and $(\mathcal{A}',\mathcal{B})$. It follows from the description of Algorithm 5 that the sequence of actions $(\mathbf{x}_1,\cdots,\mathbf{x}_T)$ in $(\mathcal{A},\hat{\mathcal{B}})$ corresponds to the same sequence of actions in $(\mathcal{A}',\mathcal{B})$.

Let $\mathbf{u} \in \operatorname{argmax}_{\mathbf{u}\in\mathcal{U}} \sum_{t=a}^{b} f_t(\mathbf{u}_t)$ and $\hat{\mathbf{u}} \in \operatorname{argmax}_{\mathbf{u}\in\hat{\mathcal{U}}} \sum_{t=a}^{b} \hat{f}_t(\mathbf{u}_t)$. We have

$$\begin{aligned}
\mathcal{R}^{\mathcal{A}'}_{\mathcal{B}} - \mathcal{R}^{\mathcal{A}}_{\hat{\mathcal{B}}} &= \mathbb{E}\left[\alpha\sum_{t=a}^{b} f_t(\mathbf{u}_t) - \sum_{t=a}^{b} f_t(\mathbf{x}_t)\right] - \mathbb{E}\left[\alpha\sum_{t=a}^{b} \hat{f}_t(\hat{\mathbf{u}}_t) - \sum_{t=a}^{b} \hat{f}_t(\mathbf{x}_t)\right] \\
&= \mathbb{E}\left[\left(\sum_{t=a}^{b} \hat{f}_t(\mathbf{x}_t) - \sum_{t=a}^{b} f_t(\mathbf{x}_t)\right) + \alpha\left(\sum_{t=a}^{b} f_t(\mathbf{u}_t) - \sum_{t=a}^{b} \hat{f}_t(\hat{\mathbf{u}}_t)\right)\right]. \qquad (12)
\end{aligned}$$

To obtain the same bound as before, instead of Inequality 11, we have

$$|f_t(\mathbf{x}_t) - \hat{f}_t(\mathbf{x}_t)| \le \delta M_1 < 2\delta M_1.$$

The rest of the proof follows verbatim. □

Proof of Theorem 6. Note that any realized adversary $\mathcal{B} \in \mathrm{Adv}^{\mathrm{o}}_0(\mathbf{F})$ may be represented as a sequence of functions $(f_1,\cdots,f_T)$. For such realized adversary $\mathcal{B}$, we define $\hat{B}$ to be the realized adversary corresponding to $(\hat{f}_1,\cdots,\hat{f}_T)$ with the stochastic gradient oracles

$$\hat{\mathcal{Q}}_t(\mathbf{x}) := \frac{k}{2\delta}\big(f_t(\mathbf{x}+\delta\mathbf{v}) - f_t(\mathbf{x}-\delta\mathbf{v})\big)\,\mathbf{v}, \qquad (13)$$

where $\mathbf{v}$ is a random vector, taking its values uniformly from $\mathbb{S}^1 \cap \mathcal{L}_0 = \mathbb{S}^1 \cap (\mathrm{aff}(\mathcal{K}) - \mathbf{z})$, for any $\mathbf{z}\in\mathcal{K}$ and $k = \dim(\mathcal{L}_0)$. Since $\mathcal{Q}_t$ is a stochastic value oracle for f_t, according to Lemma 5 in [37], $\hat{\mathcal{Q}}_t(\mathbf{x})$ is an unbiased estimator of $\nabla\hat{f}_t(\mathbf{x})$. Moreover, we have

$$\|\frac{k}{2\delta}\big(f_t(\mathbf{x}+\delta\mathbf{v}) - f_t(\mathbf{x}-\delta\mathbf{v})\big)\,\mathbf{v}\| \le \frac{k}{2\delta}M_1\|(\mathbf{x}+\delta\mathbf{v}) - (\mathbf{x}-\delta\mathbf{v})\| \le kM_1.$$

Hence we have $\hat{B} \in \mathrm{Adv}^{\mathrm{o}}_1(\hat{\mathbf{F}}, kM_1)$. Using Equation 13 and the definition of the Algorithm 5, we see that the responses of the queries are the same between the game $(\mathcal{A},\hat{\mathcal{B}})$ and $(\mathcal{A}',\mathcal{B})$. It follows that the sequence of actions $(\mathbf{x}_1,\cdots,\mathbf{x}_T)$ in $(\mathcal{A},\hat{\mathcal{B}})$ corresponds to the same sequence of actions in $(\mathcal{A}',\mathcal{B})$.

Let $\mathbf{u} \in \operatorname{argmax}_{\mathbf{u}\in\mathcal{U}} \sum_{t=a}^{b} f_t(\mathbf{u}_t)$ and $\hat{\mathbf{u}} \in \operatorname{argmax}_{\mathbf{u}\in\hat{\mathcal{U}}} \sum_{t=a}^{b} \hat{f}_t(\mathbf{u}_t)$. We have

$$\begin{aligned}
\mathcal{R}^{\mathcal{A}'}_{\mathcal{B}} - \mathcal{R}^{\mathcal{A}}_{\hat{\mathcal{B}}} &= \mathbb{E}\left[\alpha\sum_{t=a}^{b} f_t(\mathbf{u}_t) - \sum_{t=a}^{b} f_t(\mathbf{x}_t)\right] - \mathbb{E}\left[\alpha\sum_{t=a}^{b} \hat{f}_t(\hat{\mathbf{u}}_t) - \sum_{t=a}^{b} \hat{f}_t(\mathbf{x}_t)\right] \\
&= \mathbb{E}\left[\left(\sum_{t=a}^{b} \hat{f}_t(\mathbf{x}_t) - \sum_{t=a}^{b} f_t(\mathbf{x}_t)\right) + \alpha\left(\sum_{t=a}^{b} f_t(\mathbf{u}_t) - \sum_{t=a}^{b} \hat{f}_t(\hat{\mathbf{u}}_t)\right)\right]. \qquad (14)
\end{aligned}$$

According to Lemma 3 in [37], we have $|\hat{f}_t(\mathbf{x}_t) - f_t(\mathbf{x}_t)| \leq \delta M_1$. On the other hand, we have

$$
\begin{aligned}
\sum_{t=a}^{b} \hat{f}_t(\hat{\mathbf{u}}_t) &= \max_{\hat{\mathbf{u}} \in \hat{\mathcal{U}}} \sum_{t=a}^{b} \hat{f}_t(\hat{\mathbf{u}}_t) \\
&\geq -\delta M_1 T + \max_{\hat{\mathbf{u}} \in \hat{\mathcal{U}}} \sum_{t=a}^{b} f_t(\hat{\mathbf{u}}_t) && \text{(Lemma 3 in [37])} \\
&= -\delta M_1 T + \max_{\mathbf{u} \in \mathcal{U}} \sum_{t=a}^{b} f_t\left(\left(1 - \frac{\delta}{r}\right)\mathbf{u}_t + \frac{\delta}{r}\mathbf{c}\right) && \text{(Definition of } \hat{\mathcal{U}}\text{)} \\
&= -\delta M_1 T + \max_{\mathbf{u} \in \mathcal{K}} \sum_{t=a}^{b} f_t\left(\mathbf{u}_t + \frac{\delta}{r}(\mathbf{c} - \mathbf{x})\right) \\
&\geq -\delta M_1 T + \max_{\mathbf{u} \in \mathcal{U}} \sum_{t=a}^{b} \left(f_t(\mathbf{u}_t) - \frac{2\delta M_1 D}{r}\right) && \text{(Lipschitz)} \\
&= -\left(1 + \frac{2D}{r}\right)\delta M_1 T + \sum_{t=a}^{b} f_t(\mathbf{u}_t)
\end{aligned}
$$

Therefore, using Equation 14, we see that

$$
\mathcal{R}_{\mathcal{B}}^{\mathcal{A}'} - \mathcal{R}_{\hat{\mathcal{B}}}^{\mathcal{A}} \leq \delta M_1 T + \alpha\left(1 + \frac{2D}{r}\right)\delta M_1 T \leq \left(2 + \frac{2D}{r}\right)\delta M_1 T.
$$

Therefore, we have

$$
\begin{aligned}
\mathcal{R}_{\mathrm{Adv}_0^{\mathrm{o}}(\mathbf{F})}^{\mathcal{A}'} &= \sup_{\mathcal{B} \in \mathrm{Adv}_0^{\mathrm{o}}(\mathbf{F})} \mathcal{R}_{\mathcal{B}}^{\mathcal{A}'} \\
&\leq \sup_{\mathcal{B} \in \mathrm{Adv}_0^{\mathrm{o}}(\mathbf{F})} \mathcal{R}_{\hat{\mathcal{B}}}^{\mathcal{A}} + \left(2 + \frac{2D}{r}\right)\delta M_1 T \\
&\leq \sup_{\mathcal{B} \in \mathrm{Adv}_0^{\mathrm{o}}(\mathbf{F})} \mathcal{R}_{\hat{\mathcal{B}}}^{\mathcal{A}} + \left(2 + \frac{2D}{r}\right)\delta M_1 T \\
&\leq \mathcal{R}_{\mathrm{Adv}_1^{\mathrm{o}}(\hat{\mathbf{F}}, k M_1)}^{\mathcal{A}} + \left(2 + \frac{2D}{r}\right)\delta M_1 T. \qquad \square
\end{aligned}
$$

Corollary 4. *Under the assumptions of Theorem 5, if we have* $\mathcal{R}_{\alpha,\mathrm{Adv}_1^{o}(\mathbf{F},B_1)}^{\mathcal{A}} = O(B_1 T^{\eta})$ *and* $\delta = T^{(\eta-1)/2}$*, then we have*

$$
\mathcal{R}_{\alpha,\mathrm{Adv}_0^{o}(\mathbf{F},B_0)}^{\mathcal{A}'} = O(B_0 T^{(1+\eta)/2}).
$$

Corollary 5. *Under the assumptions of Theorem 6, if we have* $\delta = T^{-1}$*, then* $\mathcal{R}_{\alpha,\mathrm{Adv}_0^{o}(\mathbf{F})}^{\mathcal{A}'}$ *has the same order of regret as that of* $\mathcal{R}_{\alpha,\mathrm{Adv}_1^{o}(\mathbf{F},B_1)}^{\mathcal{A}}$ *with* B_1 *replaced with* kM_1*.*

H Proof of Theorem 7

Proof. Given a realized adversary $\mathcal{B} \in \mathrm{Adv}_i^{\mathrm{o}}(\mathbf{F}, B)\{T\}$, we may define $\hat{\mathcal{B}} = \mathrm{Adv}_i^{\mathrm{o}}(\mathbf{F}, B)\{T/L\}$ to be the realized adversary constructed by averaging each T/L block of length L. Specifically, if the functions chosen by $\mathcal{B}$ are $f_1, \cdots, f_T$, the functions chosen by $\hat{\mathcal{B}}$ are $\hat{f}_q := \frac{1}{L}\sum_{t=(q-1)L+1}^{qL} f_t$ for $1 \leq q \leq T/L$. Note that, for any $\mathbf{x} \in \mathcal{K}$ and $(q-1)L < t \leq qL$, we have $\mathbb{E}[f_t(\mathbf{x})] = \hat{f}_q(\mathbf{x})$ and if each f_t is differentiable at $\mathbf{x}$, then $\mathbb{E}[\nabla f_t(\mathbf{x})] = \nabla \hat{f}_q(\mathbf{x})$. If the query oracles selected by $\mathcal{B}$ are $\mathcal{Q}_1, \cdots, \mathcal{Q}_T$, then for any $1 \leq q \leq T/L$ we define the query oracle $\hat{\mathcal{Q}}_q$ as the algorithm that first selects an integer $(q-1)L + 1 \leq t \leq qL$ with uniform probability and then returns the output of $\mathcal{Q}_t$. It follows that $\hat{\mathcal{Q}}_q$ is a query oracle for $\hat{f}_q$. It is clear from the description of Algorithm 4 that, when the adversary is $\mathcal{B}$, the output returned to the base algorithm corresponds to $\hat{\mathcal{B}}$. We have $1 \le (a'-1)L+1 \le a \le b \le b'L \le T$. Hence

$$\begin{aligned}
\mathcal{R}^{\mathcal{A}'}_{\alpha,\mathcal{B}}(\mathcal{K}^T_\star)[a,b] &= \mathbb{E}\left[\alpha \max_{\mathbf{u}_0\in\mathcal{K}} \sum_{t=a}^{b} f_t(\mathbf{u}_0) - \sum_{t=a}^{b} f_t(\mathbf{x}_t)\right] \\
&= \mathbb{E}\left[L\left(\alpha \max_{\mathbf{u}_0\in\mathcal{K}} \frac{1}{L}\sum_{t=a}^{b} f_t(\mathbf{u}_0) - \frac{1}{L}\sum_{t=a}^{b} f_t(\mathbf{x}_t)\right)\right] \\
&\le \mathbb{E}\left[L\left(\alpha \max_{\mathbf{u}_0\in\mathcal{K}} \frac{1}{L}\sum_{t=(a'-1)L+1}^{b'L} f_t(\mathbf{u}_0) - \frac{1}{L}\sum_{t=(a'-1)L+1}^{b'L} f_t(\mathbf{x}_t)\right)\right] \\
&= \mathbb{E}\left[\sum_{q=a'}^{b'}\sum_{t=(q-1)L+1}^{qL}\big(f_t(\hat{\mathbf{x}}_q) - f_t(\mathbf{x}_t)\big) + L\left(\alpha \max_{\mathbf{u}_0\in\mathcal{K}} \sum_{q=a'}^{b'} \hat{f}_q(\mathbf{u}_0) - \sum_{q=a'}^{b'} \hat{f}_q(\hat{\mathbf{x}}_q)\right)\right] \\
&\le \sum_{q=a'}^{b'} KM_1D + L\mathbb{E}\left[\alpha \max_{\mathbf{u}_0\in\mathcal{K}} \sum_{q=a'}^{b'} \hat{f}_q(\mathbf{u}_0) - \sum_{q=a'}^{b'} \hat{f}_q(\hat{\mathbf{x}}_q)\right] \\
&\le (b'-a'+1)KM_1D + L\mathcal{R}^{\mathcal{A}}_{\alpha,\hat{\mathcal{B}}}(\mathcal{K}^{T/L}_\star)[a',b']
\end{aligned}$$

Therefore

$$\begin{aligned}
\mathcal{R}^{\mathcal{A}'}_{\alpha,\mathrm{Adv}^{\mathrm{o}}_i(\mathbf{F},B)\{T\}}(\mathcal{K}^T_\star)[a,b] &= \sup_{\mathcal{B}\in\mathrm{Adv}^{\mathrm{o}}_i(\mathbf{F},B)\{T\}} \mathcal{R}^{\mathcal{A}'}_{\alpha,\mathcal{B}}(\mathcal{K}^T_\star)[a,b] \\
&\le M_1DK(b'-a'+1) + L \sup_{\hat{\mathcal{B}}\in\mathrm{Adv}^{\mathrm{o}}_1(\mathbf{F},B)\{T/L\}} \mathcal{R}^{\mathcal{A}}_{\alpha,\hat{\mathcal{B}}}(\mathcal{K}^{T/L}_\star)[a',b'] \\
&= M_1DK(b'-a'+1) + L\mathcal{R}^{\mathcal{A}}_{\alpha,\mathrm{Adv}^{\mathrm{o}}_i(\mathbf{F},B)\{T/L\}}(\mathcal{K}^{T/L}_\star)[a',b']. \qquad \square
\end{aligned}$$

Remark 1. Note that in the above proof, we did not need to assume that the query oracles are bounded. Specifically, what we require is that the set of query oracles to be closed under convex combinations. This holds when all query oracles are bounded by B, but it also holds under many other assumptions, e.g., if we assume all query oracles variances are bounded by some $\sigma^2 > 0$.

Corollary 6. *Under the assumptions of Theorem 7, if we have* $\mathcal{R}^{\mathcal{A}'}_{\alpha,\mathrm{Adv}^o_i(\mathbf{F},B)}(\mathcal{K}^T_\star)[a,b] = O(BT^\eta)$, $K = O(T^\theta)$ *and* $L = T^{\frac{1+\theta-\eta}{2-\eta}}$, *then we have*

$$\mathcal{R}^{\mathcal{A}'}_{\alpha,\mathrm{Adv}^o_i(\mathbf{F},B)}(\mathcal{K}^T_\star)[a,b] = O\left(BT^{\frac{(1+\theta)(1-\eta)+\eta}{2-\eta}}\right).$$

As a special case, when $K = O(1)$, *then we have*

$$\mathcal{R}^{\mathcal{A}'}_{\alpha,\mathrm{Adv}^o_i(\mathbf{F},B)}(\mathcal{K}^T_\star)[a,b] = O\left(BT^{\frac{1}{2-\eta}}\right).$$

Proof. We have

$$\begin{aligned}
\mathcal{R}^{\mathcal{A}'}_{\alpha,\mathrm{Adv}^{\mathrm{o}}_i(\mathbf{F},B)\{T\}}(\mathcal{K}^T_\star)[a,b] &\le M_1DK(b'-a'+1) + L\mathcal{R}^{\mathcal{A}}_{\alpha,\mathrm{Adv}^{\mathrm{o}}_i(\mathbf{F},B)\{T/L\}}(\mathcal{K}^{T/L}_\star)[a',b'] \\
&\le M_1DK(T/L) + L\mathcal{R}^{\mathcal{A}}_{\alpha,\mathrm{Adv}^{\mathrm{o}}_i(\mathbf{F},B)\{T/L\}}(\mathcal{K}^{T/L}_\star)[a',b'] \\
&= O(KT/L) + O(LB(T/L)^\eta) \\
&= O\left(BT^{\frac{(1+\theta)(1-\eta)+\eta}{2-\eta}}\right). \qquad \square
\end{aligned}$$

I Proof of Theorem 8

Proof. Since $\mathcal{A}$ requires T^θ queries per time-step, it requires a total of $T^{1+\theta}$ queries. The expected error is bounded by the regret divided by time. Hence we have $\epsilon = O(T^{\eta-1})$ after $T^{1+\theta}$ queries. Therefore, the total number of queries to keep the error bounded by ϵ is $O(\epsilon^{-\frac{1+\theta}{1-\eta}})$. $\square$

J Projection-free adaptive regret

The SO-OGD algorithm in [17] is a deterministic algorithm with semi-bandit feedback, designed for online convex optimization with a deterministic gradient oracle. Here *we assume that the separation oracle is deterministic.*

Here we use the notation $\mathbf{c}$, r and $\hat{\mathcal{K}}_\delta$ described in Section 5.

Algorithm 8: Online Gradient Ascent via a Separation Oracle - SO-OGA

Input : horizon T, constraint set $\mathcal{K}$, step size η

$\mathbf{x}_1 \leftarrow \mathbf{c} \in \hat{\mathcal{K}}_\delta$

for $t = 1, 2, \ldots, T$ **do**

  Play $\mathbf{x}_t$ and observe $\mathbf{o}_t = \nabla f_t(\mathbf{x}_t)$

  $\mathbf{x}'_{t+1} = \mathbf{x}_t + \eta \mathbf{o}_t$

  Set $\mathbf{x}_{t+1} = \texttt{SO-IP}_{\mathcal{K}}(\mathbf{x}'_{t+1})$, the output of Algorithm 9 with initial point $\mathbf{x}'_{t+1}$

end

Note that here we use a maximization version of the algorithm, which we denote by SO-OGA. Here $\mathbf{P}_{\mathcal{K}}$ denotes projection into the convex set $\mathcal{K}$. The original version, which is designed for minimization, uses the update rule $\mathbf{x}'_{t+1} = \mathbf{x}_t - \eta \mathbf{o}_t$ in Algorithm 8 instead.

Algorithm 9: Infeasible Projection via a Separation Oracle - $\texttt{SO-IP}_{\mathcal{K}}(\mathbf{y}_0)$

Input : Constraint set $\mathcal{K}$, shrinking parameter $\delta < r$, initial point $\mathbf{y}_0$

$\mathbf{y}_1 \leftarrow \mathbf{P}_{\mathrm{aff}(\mathcal{K})}(\mathbf{y}_0)$

$\mathbf{y}_2 \leftarrow \mathbf{c} + \frac{\mathbf{y}_1 - \mathbf{c}}{\max\{1, \|\mathbf{y}_1\|/D\}}$ /* $\mathbf{y}_1$ is projection of $\mathbf{y}_0$ over $\mathbb{B}^d_D(\mathbf{c}) \cap \mathrm{aff}(\mathcal{K})$ */

for $i = 1, 2, \ldots$ **do**

  Call $\mathrm{SO}_{\mathcal{K}}$ with input $\mathbf{y}_i$

  if $\mathbf{y}_i \notin \mathcal{K}$ **then**

    Set $\mathbf{g}_i$ to be the hyperplane returned by $\mathrm{SO}_{\mathcal{K}}$ /* $\forall \mathbf{x} \in \mathcal{K}, \quad \langle \mathbf{y}_i - \mathbf{x}, \mathbf{g}_i \rangle > 0$ */

    $\mathbf{g}'_i \leftarrow \mathbf{P}_{\mathrm{aff}(\mathcal{K}) - \mathbf{c}}(\mathbf{g}_i)$

    Update $\mathbf{y}_{i+1} \leftarrow \mathbf{y}_i - \delta \frac{\mathbf{g}'_i}{\|\mathbf{g}'_i\|}$

  else

    Return $\mathbf{y} \leftarrow \mathbf{y}_i$

  end

end

Lemma 5. *Algorithm 9 stops after at most* $(\mathrm{dist}(\mathbf{y}_0, \hat{\mathcal{K}}_\delta)^2 - \mathrm{dist}(\mathbf{y}, \hat{\mathcal{K}}_\delta)^2)/\delta^2 + 1$ *iterations and returns* $\mathbf{y} \in \mathcal{K}$ *such that* $\forall \mathbf{z} \in \hat{\mathcal{K}}_\delta$*, we have* $\|\mathbf{y} - \mathbf{z}\| \leq \|\mathbf{y}_0 - \mathbf{z}\|$.

Proof. We first note that this algorithm is invariant under translations. Hence it is sufficient to prove the result when $\mathbf{c} = 0$.

Let $\mathrm{SO}'_{\mathcal{K}}$ denote the following separation oracle. If $\mathbf{y} \in \mathcal{K}$ or $\mathbf{y} \notin \mathrm{aff}(\mathcal{K})$, then $\mathrm{SO}'_{\mathcal{K}}$ returns the same output as $\mathrm{SO}_{\mathcal{K}}$. Otherwise, it returns $\mathbf{P}_{\mathrm{aff}(\mathcal{K})}(\mathbf{g})$ where $\mathbf{g} \in \mathbb{R}^d$ is the output of $\mathrm{SO}_{\mathcal{K}}$. To prove that this is indeed a separation oracle, we only need to consider the case where $\mathbf{y} \in \mathrm{aff}(\mathcal{K}) \setminus \mathcal{K}$. We know that $\mathbf{g}$ is a vector such that

$$\forall \mathbf{x} \in \mathcal{K}, \quad \langle \mathbf{y} - \mathbf{x}, \mathbf{g} \rangle > 0.$$

Since $\mathbf{P}_{\mathrm{aff}(\mathcal{K})}$ is an orthogonal projection, we have

$$\langle \mathbf{y} - \mathbf{x}, \mathbf{P}_{\mathrm{aff}(\mathcal{K})}(\mathbf{g}) \rangle = \langle \mathbf{P}_{\mathrm{aff}(\mathcal{K})}(\mathbf{y} - \mathbf{x}), \mathbf{g} \rangle = \langle \mathbf{y} - \mathbf{x}, \mathbf{g} \rangle > 0.$$

for all $\mathbf{x} \in \mathcal{K}$, which implies that $\mathrm{SO}'_{\mathcal{K}}$ is a separation oracle.

Now we see that Algorithm 9 is an instance of Algorithm 6 in [17] applied to the initial point $\mathbf{y}_1$ using the separation oracle $\mathrm{SO}'_{\mathcal{K}}$. Hence we may use Lemma 13 in [17] directly to see that Algorithm 9 stops after at most $(\mathrm{dist}(\mathbf{y}_1, \hat{\mathcal{K}}_\delta)^2 - \mathrm{dist}(\mathbf{y}, \hat{\mathcal{K}}_\delta)^2)/\delta^2 + 1$ iterations and returns $\mathbf{y} \in \mathcal{K}$ such that $\forall \mathbf{z} \in \hat{\mathcal{K}}_\delta$, we have $\|\mathbf{y} - \mathbf{z}\| \leq \|\mathbf{y}_1 - \mathbf{z}\|$ Since $\mathbf{y}_1$ is the projection of $\mathbf{y}$ over $\mathrm{aff}(\mathcal{K})$, we see that Algorithm 9 stops after at most

$$\begin{aligned}
\frac{\mathrm{dist}(\mathbf{y}_1, \hat{\mathcal{K}}_\delta)^2 - \mathrm{dist}(\mathbf{y}, \hat{\mathcal{K}}_\delta)^2}{\delta^2} + 1 &= \frac{\mathrm{dist}(\mathbf{y}_0, \hat{\mathcal{K}}_\delta)^2 - \mathrm{dist}(\mathbf{y}, \hat{\mathcal{K}}_\delta)^2}{\delta^2} - \frac{\|\mathbf{y}_0 - \mathbf{y}_1\|^2}{\delta^2} + 1 \\
&\leq \frac{\mathrm{dist}(\mathbf{y}_0, \hat{\mathcal{K}}_\delta)^2 - \mathrm{dist}(\mathbf{y}, \hat{\mathcal{K}}_\delta)^2}{\delta^2} + 1
\end{aligned}$$

steps and

$$\forall \mathbf{z} \in \hat{\mathcal{K}}_\delta \subseteq \mathrm{aff}(\mathcal{K}), \quad \|\mathbf{y} - \mathbf{z}\| \leq \|\mathbf{y}_1 - \mathbf{z}\| \leq \|\mathbf{y}_0 - \mathbf{z}\|. \qquad \square$$

In the following, we use the notation

$$\mathcal{AR}^{\mathcal{A}}_{\alpha,\mathrm{Adv}} := \max_{1 \leq a \leq b \leq T} \mathcal{R}^{\mathcal{A}}_{\alpha,\mathrm{Adv}}(\mathcal{K}^T_*)[a, b],$$

to denote the adaptive regret.

Theorem 9. *Let* $\mathbf{L}$ *be a class of linear functions over* $\mathcal{K}$ *such that* $\|l\| \leq M_1$ *for all* $l \in \mathbf{L}$ *and let* $D = \mathrm{diam}(\mathcal{K})$. *Fix* $v > 0$ *such that* $\delta = vT^{-1/2} \in (0, 1)$ *and set* $\eta = \frac{vr}{2M_1} T^{-1/2}$. *Then we have*

$$\mathcal{AR}^{\texttt{SO-OGA}}_{1,\mathrm{Adv}^f_1(\mathbf{L})} = O(M_1 T^{1/2}).$$

Proof. Since the algorithm is deterministic, according to Theorem 1 in [34], it is sufficient to prove this regret bound against the oblivious adversary $\mathrm{Adv}^{\mathrm{o}}_1(\mathbf{L})$.

Note that this algorithm is invariant under translations. Hence it is sufficient to prove the result when $\mathbf{c} = 0$. If $\mathrm{aff}(\mathcal{K}) = \mathbb{R}^d$, then we have $\mathbb{B}^d_r(\mathbf{0}) \subseteq \mathcal{K} \subseteq \mathbb{B}^d_R(\mathbf{0})$ and we may use Theorem 14 from [17] to obtain the desired result for the oblivious adversary $\mathrm{Adv}^{\mathrm{o}}_1(\mathbf{L})$. On the other hand, the assumption $\mathbb{B}^d_r(\mathbf{0}) \subseteq \mathcal{K}$ is only used in the proof of Lemma 13 in [17]. Here we use Lemma 5 instead which does not require this assumption. $\square$

The following corollary is an immediate consequence of the above theorem and Theorems 2, 3, 4, 8 and Corollaries 4, 5 and 6.

Corollary 7. *Let* `SO-OGA` *denote the algorithm described above. Then the following are true.*

a) *Under the assumptions of Theorem 2, we have:*

$$\begin{aligned}
\mathcal{AR}^{\texttt{SO-OGA}}_{\frac{\gamma^2}{1+c\gamma^2},\mathrm{Adv}^f_1(\mathbf{F})} &\leq O(M_1 T^{1/2}), \\
\mathcal{AR}^{\texttt{SO-OGA}}_{\frac{\gamma^2}{1+c\gamma^2},\mathrm{Adv}^o_1(\mathbf{F},B_1)} &\leq O(B_1 T^{1/2}), \\
\mathcal{AR}^{\texttt{FOTZO(SO-OGA)}}_{\frac{\gamma^2}{1+c\gamma^2},\mathrm{Adv}^o_0(\mathbf{F})} &\leq O(M_1 T^{1/2}).
\end{aligned}$$

If we also assume $\mathbf{F}$ *is bounded by* M_0 *and* $B_0 \geq M_0$, *then*

$$\mathcal{AR}^{\texttt{STB(SO-OGA)}}_{\frac{\gamma^2}{1+c\gamma^2},\mathrm{Adv}^o_0(\mathbf{F},B_0)} \leq O(B_0 T^{3/4}).$$

b) *Under the assumptions of Theorem 3, we have:*

$$\begin{aligned}
\mathcal{AR}^{\mathcal{A}}_{1-e^{-\gamma},\mathrm{Adv}^o_1(\mathbf{F},B_1)} &\leq O(B_1 T^{1/2}), \\
\mathcal{AR}^{\texttt{FOTZO}(\mathcal{A})}_{1-e^{-\gamma},\mathrm{Adv}^o_0(\mathbf{F})} &\leq O(M_1 T^{1/2}),
\end{aligned}$$

where $\mathcal{A} = \texttt{OMBQ}(\texttt{SO-OGA}, \texttt{BQMO}, \mathrm{Id})$. *Note that* $\mathcal{A}$ *is a first order full-information algorithm that requires a single query per time-step. If we also assume* $\mathbf{F}$ *is bounded by* M_0 *and* $B_0 \geq M_0$*, then*

$$\mathcal{AR}^{\mathcal{A}_{\text{semi-bandit}}}_{1-e^{-\gamma}, \mathrm{Adv}_1^o(\mathbf{F}, B_1)} \leq O(B_1 T^{2/3}),$$
$$\mathcal{AR}^{\mathcal{A}_{\text{full-info-0}}}_{1-e^{-\gamma}, \mathrm{Adv}_0^o(\mathbf{F}, B_0)} \leq O(B_0 T^{3/4})$$
$$\mathcal{AR}^{\mathcal{A}_{\text{bandit}}}_{1-e^{-\gamma}, \mathrm{Adv}_0^o(\mathbf{F}, B_0)} \leq O(B_0 T^{4/5})$$

where

$$\mathcal{A}_{\text{semi-bandit}} = \texttt{SFTT}(\mathcal{A}), \quad \mathcal{A}_{\text{full-info-0}} = \texttt{FOTZO}(\mathcal{A}), \quad \mathcal{A}_{\text{bandit}} = \texttt{SFTT}(\mathcal{A}_{\text{full-info-0}}).$$

c) *Under the assumptions of Theorem 4, we have:*

$$\mathcal{AR}^{\mathcal{A}}_{\frac{1-h}{4}, \mathrm{Adv}_1^o(\mathbf{F}, B_1)} \leq O(B_1 T^{1/2}),$$
$$\mathcal{AR}^{\texttt{FOTZO}(\mathcal{A})}_{\frac{1-h}{4}, \mathrm{Adv}_0^o(\mathbf{F})} \leq O(M_1 T^{1/2}),$$

where $\mathcal{A} = \texttt{OMBQ}(\texttt{SO-OGA}, \texttt{BQN}, \mathbf{x} \mapsto \frac{\mathbf{x}_t + \mathbf{x}}{2})$. *Note that* $\mathcal{A}$ *is a first order full-information algorithm that requires a single query per time-step. If we also assume* $\mathbf{F}$ *is bounded by* M_0 *and* $B_0 \geq M_0$*, then*

$$\mathcal{AR}^{\mathcal{A}_{\text{semi-bandit}}}_{\frac{1-h}{4}, \mathrm{Adv}_1^o(\mathbf{F}, B_1)} \leq O(B_1 d^{1/2} T^{2/3}),$$
$$\mathcal{AR}^{\mathcal{A}_{\text{full-info-0}}}_{\frac{1-h}{4}, \mathrm{Adv}_0^o(\mathbf{F}, B_0)} \leq O(B_0 d^{1/2} T^{3/4}),$$
$$\mathcal{AR}^{\mathcal{A}_{\text{bandit}}}_{\frac{1-h}{4}, \mathrm{Adv}_0^o(\mathbf{F}, B_0)} \leq O(B_0 d^{1/2} T^{4/5}),$$

where

$$\mathcal{A}_{\text{semi-bandit}} = \texttt{SFTT}(\mathcal{A}), \quad \mathcal{A}_{\text{full-info-0}} = \texttt{FOTZO}(\mathcal{A}), \quad \mathcal{A}_{\text{bandit}} = \texttt{SFTT}(\mathcal{A}_{\text{full-info-0}}).$$

K Dynamic regret

Improved Ader (`IA`) algorithm [43] is a deterministic algorithm with semi-bandit feedback, designed for online convex optimization with a deterministic gradient oracle.

Algorithm 10: Improved Ader - `IA`

Input : horizon T, constraint set $\mathcal{K}$, step size λ, a set $\mathcal{H}$ containing step sizes for experts
Activate a set of experts $\{E^\eta \mid \eta \in \mathcal{H}\}$ by invoking Algorithm 11 for each step size $\eta \in \mathcal{H}$
Sort step sizes in ascending order $\eta_1 \leq \cdots \leq \eta_N$, and set $w_1^{\eta_i} = \frac{C}{i(i+1)}$ where $C = 1 + \frac{1}{|\mathcal{H}|}$
for $t = 1, 2, \ldots, T$ **do**
    Receive $\mathbf{x}_t^\eta$ from each expert E^η
    Play the action $\mathbf{x}_t = \sum_{\eta \in \mathcal{H}} w_t^\eta \mathbf{x}_t^\eta$ and observe $\mathbf{o}_t = \nabla f_t(\mathbf{x}_t)$
    Define $l_t(\mathbf{y}) := \langle \mathbf{o}_t, \mathbf{y} - \mathbf{x}_t \rangle$
    Update the weight of each expert by $w_{t+1}^\eta = \frac{w_t^\eta e^{-\lambda l_t(\mathbf{x}_t^\eta)}}{\sum_{\mu \in \mathcal{H}} w_t^\mu e^{-\lambda l_t(\mathbf{x}_t^\mu)}}$.
    Send the gradient $\mathbf{o}_t$ to each expert E^η
end

Algorithm 11: Improved Ader : Expert algorithm

Input : horizon T, constraint set $\mathcal{K}$, step size η
Let $\mathbf{x}_1^\eta$ be any point in $\mathcal{K}$
for $t = 1, 2, \ldots, T$ **do**
    Send $\mathbf{x}_t^\eta$ to the main algorithm
    Receive $\mathbf{o}_t$ from the main algorithm
    $\mathbf{x}_{t+1}^\eta = \mathbf{P}_\mathcal{K}(\mathbf{x}_t^\eta + \eta \mathbf{o}_t)$
end

In Algorithm 11, $\mathbf{P}_{\mathcal{K}}$ denotes projection into the convex set $\mathcal{K}$. Note that here we used the maximization version of this algorithm. The original version, which is designed for minimization, uses the update rule $\mathbf{x}_{t+1}^{\eta} = \mathbf{P}_{\mathcal{K}}(\mathbf{x}_t^{\eta} - \eta \mathbf{o}_t)$ in Algorithm 11 instead.

Theorem 10. *Let $\mathbf{L}$ be a class of linear functions over $\mathcal{K}$ such that $\|l\| \leq M_1$ for all $l \in \mathbf{L}$ and let $D = \operatorname{diam}(\mathcal{K})$. Set $\mathcal{H} := \{\eta_i = \frac{2^{i-1}D}{M_1}\sqrt{\frac{7}{2T}} \mid 1 \leq i \leq N\}$ where $N = \lceil \frac{1}{2}\log_2(1 + 4T/7) \rceil + 1$ and $\lambda = \sqrt{2/(TM_1^2 D^2)}$. Then for any comparator sequence $\mathbf{u} \in \mathcal{K}^T$, we have*

$$\mathcal{R}^{\mathtt{IA}}_{1,\mathrm{Adv}_1^f(\mathbf{L})}(\mathbf{u}) = O(M_1\sqrt{T(1 + P_T(\mathbf{u}))}).$$

Proof. If we use the oblivious adversary $\mathrm{Adv}_1^{\mathrm{o}}(\mathbf{L})$ instead, this theorem is simply a restatement of the special case (i.e. when the functions are linear) of Theorem 4 in [43]. [9] Since the algorithm is deterministic, according to Theorem 1 in [34], the regret bound remains unchanged when we replace $\mathrm{Adv}_1^{\mathrm{o}}(\mathbf{L})$ with $\mathrm{Adv}_1^{\mathrm{f}}(\mathbf{L})$. $\square$

The following corollary is an immediate consequence of the above theorem and Theorems 2, 3, 4, and Corollaries 4 and 5.

Note that we do not use the meta-algorithm `OTB` since Improved Ader is designed for non-stationary regret and does not offer any advantages in the offline case. On the other hand, we do not use the meta-algorithm `SFTT` in this case since Theorem 7 is only for the setting where the comparator is $\mathcal{K}_*^T$ and does not allow us to convert bounds for dynamic regret.

Corollary 8. *Let `IA` denote "Improved Ader" described above. Then the following are true.*

a) Under the assumptions of Theorem 2, we have:

$$\mathcal{R}^{\mathtt{IA}}_{\frac{\gamma^2}{1+c\gamma^2},\mathrm{Adv}_1^f(\mathbf{F})}(\mathbf{u}) = O(M_1\sqrt{T(1 + P_T(\mathbf{u}))}),$$
$$\mathcal{R}^{\mathtt{IA}}_{\frac{\gamma^2}{1+c\gamma^2},\mathrm{Adv}_1^o(\mathbf{F},B_1)}(\mathbf{u}) = O(B_1\sqrt{T(1 + P_T(\mathbf{u}))}),$$
$$\mathcal{R}^{\mathtt{FOTZO(IA)}}_{\frac{\gamma^2}{1+c\gamma^2},\mathrm{Adv}_0^o(\mathbf{F})}(\mathbf{u}) = O(M_1\sqrt{T(1 + P_T(\mathbf{u}))}).$$

If we also assume $\mathbf{F}$ is bounded by M_0 and $B_0 \geq M_0$, then

$$\mathcal{R}^{\mathtt{STB(IA)}}_{\frac{\gamma^2}{1+c\gamma^2},\mathrm{Adv}_0^o(\mathbf{F},B_0)}(\mathbf{u}) = O(B_0 T^{3/4}(1 + P_T(\mathbf{u}))^{1/2}).$$

b) Under the assumptions of Theorem 3, we have:

$$\mathcal{R}^{\mathcal{A}}_{1-e^{-\gamma},\mathrm{Adv}_1^o(\mathbf{F},B_1)}(\mathbf{u}) = O(B_1\sqrt{T(1 + P_T(\mathbf{u}))})$$
$$\mathcal{R}^{\mathtt{FOTZO(IA)}}_{1-e^{-\gamma},\mathrm{Adv}_0^o(\mathbf{F})}(\mathbf{u}) = O(M_1\sqrt{T(1 + P_T(\mathbf{u}))}).$$

where $\mathcal{A} = \mathtt{OMBQ}(\mathtt{IA}, \mathtt{BQMO}, \mathrm{Id})$. Note that $\mathcal{A}$ is a first order full-information algorithm that requires a single query per time-step. If we also assume $\mathbf{F}$ is bounded by M_0 and $B_0 \geq M_0$, then

$$\mathcal{R}^{\mathcal{A}_{\text{full-info-0}}}_{1-e^{-\gamma},\mathrm{Adv}_0^o(\mathbf{F},B_0)}(\mathbf{u}) = O(B_0 T^{3/4}(1 + P_T(\mathbf{u}))^{1/2}).$$

where $\mathcal{A}_{\text{full-info-0}} = \mathtt{FOTZO}(\mathcal{A})$.

c) Under the assumptions of Theorem 4, we have:

$$\mathcal{R}^{\mathcal{A}}_{\frac{1-h}{4},\mathrm{Adv}_1^o(\mathbf{F},B_1)}(\mathbf{u}) = O(B_1\sqrt{T(1 + P_T(\mathbf{u}))}),$$
$$\mathcal{R}^{\mathtt{FOTZO(IA)}}_{\frac{1-h}{4},\mathrm{Adv}_0^o(\mathbf{F})}(\mathbf{u}) = O(M_1\sqrt{T(1 + P_T(\mathbf{u}))}).$$

[9] We note that although Theorem 4 in [43] assumes that the convex set contains the origin, this assumption is not really needed. In fact, for any arbitrary convex set, we may first translate it to contain the origin, apply Theorem 4 and then translate it back to obtain the results for the original convex set.

where $\mathcal{A} = \texttt{OMBQ}(\texttt{IA}, \texttt{BQN}, \mathbf{x} \mapsto \frac{\mathbf{x}_t + \mathbf{x}}{2})$*. Note that* $\mathcal{A}$ *is a first order full-information algorithm that requires a single query per time-step. If we also assume* $\mathbf{F}$ *is bounded by* M_0 *and* $B_0 \geq M_0$*, then*

$$\mathcal{R}^{\mathcal{A}_{\text{full-info-0}}}_{\frac{1-h}{4}, \text{Adv}^o_0(\mathbf{F}, B_0)}(\mathbf{u}) = O(B_0 T^{3/4}(1 + P_T(\mathbf{u}))^{1/2}).$$

where $\mathcal{A}_{\text{full-info-0}} = \texttt{FOTZO}(\mathcal{A})$*.*

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the paper's contribution and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The statements of the results have the precise assumptions. Further, some limitations are also discussed in the conclusions section as a future work direction.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [Yes]

Justification: We have clearly stated the required assumptions and an accompanying complete proof in the appendix for each theory result.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

   Answer: [NA]

   Justification: Our paper is primarily of theoretical nature and does not include experiments.

   Guidelines:

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

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

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

Answer: [NA]

Justification: Our paper is primarily of theoretical nature and does not include experiments.

Guidelines:

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

6. **Experimental Setting/Details**

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

Answer: [NA]

Justification: Our paper is primarily of theoretical nature and does not include experiments.

Guidelines:

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

7. **Experiment Statistical Significance**

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

Answer: [NA]

Justification: Our paper is primarily of theoretical nature and does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

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

   Answer: [NA]

   Justification: Our paper is primarily of theoretical nature and does not include experiments.

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: Our research conforms, in every respect, to the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [NA]

    Justification: Our work is primarily of theoretical nature and has no immediate societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: No high risk data or model have been used.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [NA]

    Justification: No existing asset has been used in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

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

    Answer: [NA]

    Justification: No new asset is introduced in the paper.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: No experiments with human subjects were conducted.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: We conducted no experiments with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.