UNIFIED PROJECTION-FREE ALGORITHMS FOR ADVERSARIAL DR-SUBMODULAR OPTIMIZATION

Mohammad Pedramfar
School of Industrial Engineering
Purdue University
West Lafayette, IN 47907, USA
mpedramf@purdue.edu

Yididiya Y. Nadew
Department of Computer Science
Iowa State University
Ames, IA 50010, USA
yididiya@iastate.edu

Christopher J. Quinn
Department of Computer Science
Iowa State University
Ames, IA 50010, USA
cjquinn@iastate.edu

Vaneet Aggarwal
School of Industrial Engineering
Purdue University
West Lafayette, IN 47907, USA
vaneet@purdue.edu

ABSTRACT

This paper introduces unified projection-free Frank-Wolfe type algorithms for adversarial continuous DR-submodular optimization, spanning scenarios such as full information and (semi-)bandit feedback, monotone and non-monotone functions, different constraints, and types of stochastic queries. For every problem considered in the non-monotone setting, the proposed algorithms are either the first with proven sub-linear α-regret bounds or have better α-regret bounds than the state of the art, where α is a corresponding approximation bound in the offline setting. In the monotone setting, the proposed approach gives state-of-the-art sub-linear α-regret bounds among projection-free algorithms in 7 of the 8 considered cases while matching the result of the remaining case. Additionally, this paper addresses semi-bandit and bandit feedback for adversarial DR-submodular optimization, advancing the understanding of this optimization area.

1 INTRODUCTION

The optimization of continuous adversarial DR-submodular functions has become increasingly prominent in recent years. This form of optimization represents an important subset of non-convex optimization problems at the forefront of machine learning and statistics. These challenges have numerous real-world applications like revenue maximization, mean-field inference, and recommendation systems, among others (Bian et al., 2019; Hassani et al., 2017; Mitra et al., 2021; Djolonga & Krause, 2014; Ito & Fujimaki, 2016; Gu et al., 2023; Li et al., 2023). The problems at hand can be conceptualized as a recurring game played between an optimizer and an adversary. In each round of this game, the optimizer makes an action selection, while the adversary selects a reward function. The optimizer is then allowed to query this reward function, either at any arbitrary point within the domain (full information) or specifically at the chosen action (in the case of semi-bandit/bandit scenarios). The adversary provides a noisy version of the gradient/value at the queried point. This framework gives rise to a set of significant challenges, varying based on the properties of the DR-submodular function, the constraint set, and the types of queries involved.

This paper presents a comprehensive investigation into online continuous adversarial DR-submodular optimization. There have been significant advances in recent years, though most research has predominantly focused on monotone (i.e., non-decreasing) objective functions and/or full information feedback via stochastic gradient queries. In contrast, our study encompasses a broader spectrum, addressing combinations of: (i) monotone or non-monotone functions, (ii) optimization under downward closed (or convex sets containing the origin) or general convex sets, (iii) gradient or value queries, and (iv) queries at arbitrary points or only the current action. See Table 1 for an enumeration of the cases along with corresponding approximation ratios α, query complexities (for full-information feedback), and α-regret bounds for prior works and ours.

In this paper, we propose Frank-Wolfe based algorithms for this diverse family of problems. For many cases in Table 1, our algorithms are the first to achieve sub-linear regret or improve on the state of the art. For non-monotone objective functions (i.e., the bottom half of Table 1), our algorithms beat the state of the art for almost every combination of convex feasible regions (downward-closed or general), feedback models (full-information (with T^β queries for various β), semi-bandit, bandit), and feedback type (exact/noisy gradient/value). See Fig. 1 for a visual depiction of the improved regret bound and query complexity trade offs for non-monotone functions with a downward closed feasible region and full-information feedback with (noisy) gradient queries. The single case for non-monotone objectives that our algorithms do not strictly improve on the prior works is for general convex sets with full-information feedback of $T^{1/2}$ gradients (per round), in which case the regret bound of our algorithm and that proposed by (Mualem & Feldman, 2023) both have $\tilde{O}(\sqrt{T})$ dependence. We note for T^β queries with $0 \leq \beta < \frac{1}{2}$, our algorithm's regret bound is strictly better.

For monotone objective functions (i.e., the top half of Table 1), our algorithms achieve the first sublinear regret for general convex sets with full information value feedback and for general convex sets with bandit feedback. Our algorithms also achieve the first or better regret bounds than prior projection-free methods (those without "‡" symbols in Table 1) for all but one case, i.e. for general convex feasible regions with bandit feedback where we match the results of (Niazadeh et al., 2021).[1] See Appendix A for more discussion.

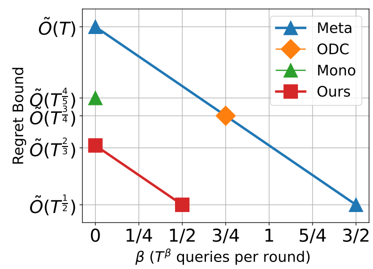

Figure 1: $\frac{1}{e}$-regret bounds for non-monotone functions with a downward-closed feasible region and full-information (noisy) gradient feedback, as a function of query-complexity. ODC is from (Thang & Srivastav, 2021). Meta and Mono are from (Zhang et al., 2023). Better performance corresponds to the bottom left corner. Our algorithm's regret bounds dominate the state of the art.

Our algorithms and most prior Frank-Wolfe based methods can rely on solving only linear optimization problems as subroutines (hence referred to as "projection-free"). Some of the prior works that use projected gradient ascent, in particular (Zhang et al., 2022; Chen et al., 2018b) (marked with a "‡" in Table 1), achieve superior regret bounds of $\tilde{O}(\sqrt{T})$ to other prior works and our algorithms. In some instances, solving a projection (or other non-linear optimization problems like (Wan et al., 2023; Thang & Srivastav, 2021)) as a sub-routine can be computationally expensive. Braun et al. (2022) identify matrix completion, routing and graph problems, and problems with matroid polytopes as example problems for which it is efficient to solve linear optimization problems but solving projections can be expensive. (Chen et al., 2018a) showed projected gradient ascent took between five to eight times longer than several Frank-Wolfe based methods even for small matrix completion tasks. Thus, for some large-scale problems, if the agent has limited per-round computational resources, the regret bounds achieved by projection-free methods (for which our methods match or improve on the state of the art) could represent the best "practically-achievable" regret-bounds.

The key contributions of this work can be summarized as follows:

1. We propose a unified framework for Frank-Wolfe type (projection-free) algorithms for adversarial continuous DR-submodular optimization, spanning scenarios with different types of feedback (full information, semi-bandit, and bandit; exact or stochastic), objective function classes (monotone and non-monotone), and convex feasible region geometries (downward-closed, origin feasible, general). In particular, we provide the first projection-free algorithm for online adversarial maximization of monotone functions over general convex sets for any feedback and oracle type.
2. For the class of non-monotone DR-submodular functions, our algorithms achieve the best (in some cases the first) sublinear α-regret bounds for all feedback models and feasible region geometries considered.

[1] The result of (Niazadeh et al., 2021) is proven for monotone functions over d-dimensional downward-closed convex sets given a deterministic value oracle. However, as we discuss in Appendix C, replacing their shrunk constraint set with the construction presented in (Pedramfar et al., 2023) together with a more detailed analysis of their algorithms can be used to obtain the same regret bounds for monotone functions over all convex sets containing the origin when we only have access to a stochastic value oracle.

Table 1: Online DR-submodular optimization results.

F	Set	Feedback			Reference	Appx.	# of queries	$\log_T(\alpha\text{-regret})$
Monotone	$0 \in \mathcal{K}$	∇F	Full Information	det.	(Chen et al., 2018b), (*)	$1-e^{-1}$	$T^\beta(\beta \in [0,1/2])$	$1-\beta$
					(Niazadeh et al., 2021)	$1-e^{-1}$	$T^\beta(\beta \in [0,1/2])$	$1-\beta$
				stoch.	(Chen et al., 2018a),	$1-e^{-1}$	$T^\beta(\beta \in [0,3/2])$	$1-\beta/3$
					(Zhang et al., 2019)	$1-e^{-1}$	1	$4/5$
					(Liao et al., 2023)	$1-e^{-1}$	1	$3/4$
					(Zhang et al., 2022) ‡	$1-e^{-1}$	1	$1/2$
					This paper	$1-e^{-1}$	$T^\beta(\beta \in [0,1/2])$	$2/3-\beta/3$
			Semi-bandit	stoch.	This paper	$1-e^{-1}$	-	$3/4$
		F	Full Information	stoch.	This paper	$1-e^{-1}$	$T^\beta(\beta \in [0,1/2])$	$3/5-\beta/5$
			Bandit	det.	(Zhang et al., 2019)	$1-e^{-1}$	-	$8/9$
					(Niazadeh et al., 2021)	$1-e^{-1}$	-	$5/6$
					(Wan et al., 2023) ‡‡	$1-e^{-1}$	-	$3/4$
				stoch.	This paper	$1-e^{-1}$	-	$5/6$
	general	∇F	Full Information	stoch.	This paper	$1/2$	$T^\beta(\beta \in [0,1/2])$	$2/3-\beta/3$
			Semi-bandit	stoch.	(Chen et al., 2018b)‡	$1/2$	-	$1/2$
					This paper	$1/2$	-	$3/4$
		F	Full Information	stoch.	This paper	$1/2$	$T^\beta(\beta \in [0,1/2])$	$3/5-\beta/5$
			Bandit	stoch.	This paper	$1/2$	-	$5/6$
Non-Monotone	d.c.	∇F	Full Information	stoch.	(Thang & Srivastav, 2021)	e^{-1}	$T^\beta(\beta \in [0,3/4])$	$1-\beta/3$
					(Zhang et al., 2023)	e^{-1}	$T^\beta(\beta \in [0,3/2])$	$1-\beta/3$
					(Zhang et al., 2023)	e^{-1}	1	$4/5$
					This paper	e^{-1}	$T^\beta(\beta \in [0,1/2])$	$2/3-\beta/3$
			Semi-bandit	stoch.	This paper	e^{-1}	-	$3/4$
		F	Full Information	stoch.	This paper	e^{-1}	$T^\beta(\beta \in [0,1/2])$	$3/5-\beta/5$
			Bandit	det.	(Zhang et al., 2023)	e^{-1}	-	$8/9$
				stoch.	This paper	e^{-1}	-	$5/6$
	general	∇F	Full Information	stoch.	(Thang & Srivastav, 2021)	$(1-h)/3\sqrt{3}$	$T^\beta(\beta > 0)$	1
					(Mualem & Feldman, 2023), (*)	$(1-h)/4$	$T^\beta(\beta \in [0,1/2])$	$1-\beta$
					This paper	$(1-h)/4$	$T^\beta(\beta \in [0,1/2])$	$2/3-\beta/3$
			Semi-bandit	stoch.	This paper	$(1-h)/4$	-	$3/4$
		F	Full Information	stoch.	This paper	$(1-h)/4$	$T^\beta(\beta \in [0,1/2])$	$3/5-\beta/5$
			Bandit	stoch.	This paper	$(1-h)/4$	-	$5/6$

Here $h := \min_{\mathbf{z}\in\mathcal{K}} \|\mathbf{z}\|_\infty$. The rows marked with (*) are special cases of our algorithms with appropriate hyperparameters. The rows marked with ‡ use gradient ascent, requiring potentially computationally expensive projections. ‡‡ (Wan et al., 2023) uses a convex optimization subroutine in each iteration. The logarithmic terms in regret are ignored. See Appendix A.2 for details.

3. For the class of monotone (i.e., non-decreasing) DR-submodular functions, our algorithms achieve the state of the art α-regret bounds in 4 of the 8 cases.[2] Moreover, if we only compare with other projection-free algorithms, then we obtain the state of the art in 7 out of the 8 cases and match the result of (Niazadeh et al., 2021) in the last case.[1]

In addition to the enumerated list above, our technical novelties include (i) a novel combination of the idea of meta-actions and random permutations to obtain a new algorithm in full-information setting; (ii) handling stochastic (gradient/value) feedback *without* using variance reduction techniques like momentum, which in turn leads to state of the art regret bounds; and (iii) a unified approach that specializes to multiple scenarios considered in this paper. See Sections 3.1 and 3.2 and Appendix A.2 for more details. Table 1 describes the key comparisons of our works, where the related works are expanded on in Appendix A.

2 Background and Notation

We introduce some basic notions, concepts and assumptions which will be used throughout the paper. For any vector $\mathbf{x} \in \mathbb{R}^d$, $[\mathbf{x}]_i$ is the i-th entry of $\mathbf{x}$. We consider the partial order on $\mathbb{R}^d$ where $\mathbf{x} \le \mathbf{y}$ if and only if $[\mathbf{x}]_i \le [\mathbf{y}]_i$ for all $1 \le i \le d$. For two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, the *join* of $\mathbf{x}$ and $\mathbf{y}$, denoted by $\mathbf{x} \vee \mathbf{y}$ and the *meet* of $\mathbf{x}$ and $\mathbf{y}$, denoted by $\mathbf{x} \wedge \mathbf{y}$, are defined

$$\mathbf{x} \vee \mathbf{y} := (\max\{[\mathbf{x}]_i, [\mathbf{y}]_i\})_{i=1}^d \quad \text{and} \quad \mathbf{x} \wedge \mathbf{y} := (\min\{[\mathbf{x}]_i, [\mathbf{y}]_i\})_{i=1}^d, \tag{1}$$

respectively. Clearly, we have $\mathbf{x} \wedge \mathbf{y} \le \mathbf{x} \le \mathbf{x} \vee \mathbf{y}$. We use $\mathbf{x} \odot \mathbf{y}$ for coordinate-wise multiplication. We use $\|\cdot\|$ to denote the Euclidean norm, and $\|\cdot\|_\infty$ to denote the supremum norm. In the paper, we consider a bounded convex domain $\mathcal{K}$ and w.l.o.g. assume that $\mathcal{K} \subseteq [0,1]^d$. We say that $\mathcal{K}$ is *downward-closed* (d.c.) if there is a point $\mathbf{u} \in \mathcal{K}$ such that for all $\mathbf{z} \in \mathcal{K}$, we have $\{\mathbf{x} \mid \mathbf{u} \le \mathbf{x} \le \mathbf{z}\} \subseteq \mathcal{K}$. Unless explicitly stated, we will assume that downward-closed convex sets contain the

[2] Algorithms for (semi-)bandit feedback can be used in full-information setting. Therefore (Chen et al., 2018b) obtains the state of the art in both semi-bandit and full-information setting for monotone functions over general convex set when given access to gradient oracles.

origin. The *diameter* D of the convex domain $\mathcal{K}$ is defined as $D := \sup_{\mathbf{x},\mathbf{y}\in\mathcal{K}}\|\mathbf{x}-\mathbf{y}\|$. We use $\mathbb{B}_r(\mathbf{x})$ to denote the open ball of radius r centered at $\mathbf{x}$. More generally, for a subset $X \subseteq \mathbb{R}^d$, we define $\mathbb{B}_r(X) := \bigcup_{\mathbf{x}\in X}\mathbb{B}_r(\mathbf{x})$. For an affine subspace A of $\mathbb{R}^d$, we define $\mathbb{B}_r^A(X) := A \cap \mathbb{B}_r(X)$. We will use $\mathbb{R}_+^d$ to denote the set $\{\mathbf{x} \in \mathbb{R}^d | \mathbf{x} \geq 0\}$. For any set $X \subseteq \mathbb{R}^d$, the affine hull of X, denoted by $\mathrm{aff}(X)$, is defined to be the intersection of all affine subsets of $\mathbb{R}^d$ that contain X. The *relative interior* of a set X is defined by

$$\mathrm{relint}(X) := \{\mathbf{x} \in X \mid \exists\varepsilon > 0, \mathbb{B}_\varepsilon^{\mathrm{aff}(X)}(\mathbf{x}) \subseteq X\}.$$

It is well known that for any non-empty convex set $\mathcal{K}$, the set $\mathrm{relint}(\mathcal{K})$ is always non-empty. We will always assume that the feasible set contains at least two points and therefore the dimension $d' := \dim(\mathcal{K}) = \dim(\mathrm{aff}(\mathcal{K})) \geq 1$, otherwise the optimization problem is trivial.

A set function $f : \{0,1\}^d \to \mathbb{R}$ is called *submodular* if for all $\mathbf{x},\mathbf{y} \in \{0,1\}^d$ with $\mathbf{x} \geq \mathbf{y}$,

$$f(\mathbf{x}\vee\mathbf{a}) - f(\mathbf{x}) \leq f(\mathbf{y}\vee\mathbf{a}) - f(\mathbf{y}), \qquad \forall\mathbf{a} \in \{0,1\}^d. \tag{2}$$

Submodular functions can be generalized over continuous domains. A function $F : [0,1]^d \to \mathbb{R}$ is called *DR-submodular* if for all vectors $\mathbf{x},\mathbf{y} \in [0,1]^d$ with $\mathbf{x} \leq \mathbf{y}$, any basis vector $\mathbf{e}_i = (0,\cdots,0,1,0,\cdots,0)$ and any constant $c > 0$ such that $\mathbf{x} + c\mathbf{e}_i \in [0,1]^d$ and $\mathbf{y} + c\mathbf{e}_i \in [0,1]^d$,

$$F(\mathbf{x}+c\mathbf{e}_i) - F(\mathbf{x}) \geq F(\mathbf{y}+c\mathbf{e}_i) - F(\mathbf{y}). \tag{3}$$

Note that if a function F is differentiable then the diminishing-return (DR) property (3) is equivalent to $\nabla F(\mathbf{x}) \geq \nabla F(\mathbf{y})$ for $\mathbf{x} \leq \mathbf{y}$ with $\mathbf{x},\mathbf{y} \in [0,1]^d$. A function $F : \mathcal{D} \to \mathbb{R}$ is M_1*-Lipschitz continuous* if for all $\mathbf{x},\mathbf{y} \in \mathcal{D}$, $\|F(\mathbf{x}) - F(\mathbf{y})\| \leq M_1\|\mathbf{x}-\mathbf{y}\|$. A differentiable function $F : \mathcal{D} \to \mathbb{R}$ is M_2*-smooth* if for all $\mathbf{x},\mathbf{y} \in \mathcal{D}$, $\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\| \leq M_2\|\mathbf{x}-\mathbf{y}\|$. A *DR-submodular* F is monotone if $F(\mathbf{x}) \geq F(\mathbf{y})$ for all $\mathbf{x} \geq \mathbf{y}$.

2.1 Problem Setup

Adversarial bandit optimization problems can be formalized as a repeated game between an optimizer and an adversary. The game lasts for T rounds and T is known to both players. In t-th round, the optimizer chooses an action $\mathbf{x}_t$ from an action set $\mathcal{K}$, then the adversary chooses a reward function $F_t \in \mathcal{F}$. We assume that the function class $\mathcal{F}$ has functions that map $\mathcal{K}$ to a bounded interval $[0, M_0] \subseteq \mathbb{R}$. We consider the setting with *oblivious adversary* where the choice of the sequence of functions F_t is not affected by the choice of the optimizer. In other words, we may assume that the adversary chooses the sequence $\{F_t\}_{t=1}^T$ before the first action of the optimizer.

Before we discuss different forms of feedback, we first formally define the notion of oracle. A stochastic *non-oblivious* value oracle for the function $F : \mathcal{K} \to \mathbb{R}$ is a tuple $(\mathfrak{Z}_0, p_0, \tilde{F})$ where $\mathfrak{Z}_0$ is an arbitrary measure space, $p_0 : \mathfrak{Z}_0 \times \mathcal{K} \to \mathbb{R}$ is a non-negative measurable function such that $\int_{\mathfrak{Z}_0} p_0(\mathbf{z};\mathbf{x})d\mathbf{z} = 1$ for each $\mathbf{x} \in \mathcal{K}$ and $\tilde{F} : \mathfrak{Z}_0 \times \mathcal{K} \to \mathbb{R}$ is a measurable function such that

$$F(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim p_0(\cdot;\mathbf{x})}[\tilde{F}(\mathbf{z},\mathbf{x})],$$

for all $\mathbf{x} \in \mathcal{K}$. We will use $\tilde{F}(\mathbf{x})$ to denote the random variable $\tilde{F}(\mathbf{x},\mathbf{z})$ where $\mathbf{z}$ is a random variable samples according to the distribution $p_0(\cdot;\mathbf{x})$. Such an oracle would be called an *oblivious* oracle when $p_0(\cdot;\mathbf{x})$ is independent of the choice of $\mathbf{x}$. We consider the more general setting where we only have access to a non-oblivious oracle, i.e., where $p_0(\cdot;\mathbf{x})$ may depend on $\mathbf{x}$.

Similarly, a non-oblivious gradient oracle for the function $F : \mathcal{K} \to \mathbb{R}$ is a tuple $(\mathfrak{Z}_1, p_1, \tilde{\nabla}F)$ where $\mathfrak{Z}_1$ is an arbitrary measure space, $p_1 : \mathfrak{Z}_1 \times \mathcal{K} \to \mathbb{R}$ is a non-negative measurable function such that $\int_{\mathfrak{Z}_1} p_1(\mathbf{z};\mathbf{x})d\mathbf{z} = 1$ for each $\mathbf{x} \in \mathcal{K}$ and $\tilde{\nabla}F : \mathfrak{Z}_1 \times \mathcal{K} \to \mathbb{R}$ is a measurable function such that

$$\nabla F(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim p_1(\cdot;\mathbf{x})}[\tilde{\nabla}F(\mathbf{z},\mathbf{x})],$$

for all $\mathbf{x} \in \mathcal{K}$. Similarly, we will use $\tilde{\nabla}F(\mathbf{x})$ to denote the random variable $\tilde{\nabla}F(\mathbf{x},\mathbf{z})$ where $\mathbf{z}$ is a random variable sampled according to the distribution $p_1(\cdot;\mathbf{x})$.

Assumption 1. *We assume that the functions $F_t : [0,1]^d \to \mathbb{R}$ are DR-submodular, first-order differentiable, non-negative, bounded by M_0, M_1-Lipschitz, and M_2-smooth for some values of $M_0, M_1, M_2 < \infty$. Note that this implies that $\|\nabla F_t(\mathbf{x})\| \leq M_1$. Moreover, we also assume that we either have access to a value oracle bounded by B_0 or a gradient oracle bounded by B_1 for some values of for some $B_0, B_1 < \infty$.*

Remark 1. *The proposed algorithm does not need to know the values of M_0, M_1, M_2, B_0 or B_1, in advance. However these variables appear in the regret bounds. Note that we always have $B_0 \geq M_0$ and $B_1 \geq M_1$. Exact oracles are special cases of stochastic oracles with $B_0 = M_0$ and $B_1 = M_1$. When we have access to exact oracles, the performance of the proposed algorithms does not change beyond the replacement of B_0 with M_0 and B_1 with M_1.*

We consider different forms of feedback to the optimizer:

1. **Full Information with gradient query oracle**: In this feedback model, the optimizer is allowed to query a stochastic non-oblivious gradient oracle $\tilde{\nabla} F_t(\mathbf{x})$ for $F_t : \mathcal{K} \to \mathbb{R}$ at multiple points. We assume that the optimizer can query F_t a total of T^β times, where $\beta \geq 0$ gives a range from constant queries to infinite queries.
2. **Full Information with value query oracle**: Same as the previous case, but the adversary reveals a stochastic non-oblivious value oracle $\tilde{F}_t(\mathbf{x})$.
3. **Semi-Bandit**: In this feedback model, the adversary reveals a gradient sample $\tilde{\nabla} F_t(\mathbf{y}_t)$ for the specific action $\mathbf{y}_t$ taken, where $\tilde{\nabla} F_t$ is a stochastic non-oblivious gradient oracle for F_t.
4. **Noisy Bandit:** In this feedback model, the optimizer can only observe a sample of $\tilde{F}_t(\mathbf{y}_t)$, where $\tilde{F}_t$ is a stochastic non-oblivious value oracle for F_t. Such feedback model is a generalization of what is called *full-bandit feedback* in the literature. In the full-bandit feedback setting, the optimizer observes the exact value of $F_t(\mathbf{y}_t)$.

We note that the full information feedback can query at any point in $\mathcal{K}$, while the semi-bandit and bandit feedback can only query at $\mathbf{y}_t$ in time t. Also note that the distributions p_0 and p_1, described in the definition of stochastic oracles, may depend on the function F_t. We will use a superscript, i.e., p_0^t and p_1^t, to specify the function in question.

Please note that even when dealing with offline scenarios, it is NP-hard to solve a DR-submodular maximization problem (Bian et al., 2017b). For the problems we consider, however, there are polynomial time approximation algorithms. We let α denote corresponding approximation ratios. Thus, the goal of this work is to minimize the α-regret, which is defined as:

$$\mathcal{R}_\alpha = \alpha \max_{\mathbf{y} \in \mathcal{K}} \sum_{t=1}^{T} F_t(\mathbf{y}) - \sum_{t=1}^{T} F_t(\mathbf{y}_t) \tag{4}$$

In Appendix A.1, we discuss the best known approximation ratios in different settings.

Remark 2. *Any algorithm designed for semi-bandit setting may be trivially applied in full-information setting with a gradient oracle. Similarly, any algorithm designed for bandit setting may be applied in full-information setting with a value oracle.*

Remark 3. *As a special case, if all of the functions F_t are equal, then the semi-bandit setting we consider reduces to online stochastic continuous DR-submodular maximization. See Table 2 in Appendix for the list of previous results in this setting for (i) monotone/non-monotone function, (ii) constraint set choices, or (iii) bandit/semi-bandit feedback. These results achieve the same regret guarantees as in (Pedramfar et al., 2023), and thus match the state of art for projection-free algorithms in all cases.*

3 Proposed Algorithms

In this section, we describe the proposed algorithm with different forms of feedback and the different problem setups (based on the properties of the functions and the feasible set). For efficient description of the algorithm, we first divide the problem setup into four categories:

(A) The functions $\{F_t\}_{t=1}^T$ are monotone DR-submodular and $\mathbf{0} \in \mathcal{K}$.

(B) The functions $\{F_t\}_{t=1}^T$ are non-monotone DR-submodular and $\mathcal{K}$ is a downward closed set containing 0.

(C) The functions $\{F_t\}_{t=1}^T$ are monotone DR-submodular and $\mathcal{K}$ is a general convex set.

(D) The functions $\{F_t\}_{t=1}^T$ are non-monotone DR-submodular and $\mathcal{K}$ is a general convex set.

The four divisions here for the problem provide different approximation ratios α for the problem. Thus, the algorithm steps and the proofs change with these cases. For combining definitions in the different forms of feedback, we define

$$\text{oracle-adv}(\mathbf{d}, \mathbf{x}) := \begin{cases} \mathbf{d} \odot (1 - \mathbf{x}) & \text{(B);} \\ \mathbf{d} & \text{otherwise,} \end{cases}, \text{update}(\mathbf{x}, \mathbf{v}, \underline{\mathbf{u}}) = \begin{cases} \mathbf{x} + \frac{1}{K}(\mathbf{v} - \underline{\mathbf{u}}) & \text{(A);} \\ \mathbf{x} + \frac{1}{K}(\mathbf{v} - \underline{\mathbf{u}}) \odot (\mathbf{1} - \mathbf{x}) & \text{(B);} \\ (1 - \varepsilon_C)\mathbf{x} + \varepsilon_C \mathbf{v} & \text{(C);} \\ (1 - \varepsilon_D)\mathbf{x} + \varepsilon_D \mathbf{v} & \text{(D),} \end{cases}$$

where $\varepsilon_C = \frac{\log(K)}{2K}$ and $\varepsilon_D = \frac{\log(2)}{K}$. We also define the approximation ratio α as $1 - \frac{1}{e}$ for case (A), $\frac{1}{e}$ for case (B), $\frac{1}{2}$ for case (C), and $\frac{1-h}{4}$ for case (D), where $h := \min_{\mathbf{z} \in \mathcal{K}} \|\mathbf{z}\|_\infty$. Further,

$$\text{grad-estimate}(F, \mathbf{x}, \mathbf{u}) := \begin{cases} \tilde{\nabla} F(\mathbf{x}) & \text{gradient oracle;} \\ \frac{d'}{\delta} \tilde{F}(\mathbf{x} + \delta \mathbf{u})\mathbf{u} & \text{value oracle.} \end{cases}$$

In the following subsections, we divide the proposed algorithm into different feedback scenarios.

3.1 Full Information

There are four main ideas used in Algorithm 1 that allows us to obtain the desired regret bounds.

1. Offline bounds *Given a submodular function F, sequence of vectors $(\mathbf{v}^{(k)})_{k=1}^K$ in the convex set $\mathcal{K}$ and a sequence of points $\mathbf{x}^{(k+1)} = \text{update}(\mathbf{x}^{(k)}, \mathbf{v}^{(k)})$ for $k \in [K]$, for any $\mathbf{x}^* \in \mathcal{K}$, we may bound $\alpha F(\mathbf{x}^*) - F(\mathbf{x}^{(K+1)})$ from above by the sum of a known term and a linear combination of $\langle \text{oracle-adv}(\nabla F(\mathbf{x}^{(k)}), \mathbf{x}^{(k)}), \mathbf{v}^{(k)} - \mathbf{x}^* \rangle$, for $k \in [K]$.*

See Lemma 8 for the exact statement for different cases. This lemma captures the core idea behind all Frank-Wolfe type algorithms for DR-submodular maximization. In particular, if we choose

$$\mathbf{v}^{(k)} \in \text{argmax}_{\mathbf{v} \in \mathcal{K}} \langle \text{oracle-adv}(\nabla F(\mathbf{x}^{(k)}), \mathbf{x}^{(k)}), \mathbf{v} \rangle,$$

we recover the results in offline setting with access to a deterministic gradient oracle. Lemma 8 is a reformulation of some of the ideas presented in (Bian et al., 2017b;a; Zhang et al., 2023; Du, 2022; Mualem & Feldman, 2023) and (Pedramfar et al., 2023).

2. Meta actions Having $L = 1$, $\delta = 0$, and access to gradient oracles corresponds to the idea of meta-actions (without using Ideas 3 and 4). The idea of meta-actions, proposed in (Streeter & Golovin, 2008) for discrete submodular functions, was first used for continuous DR-submodular maximization in (Chen et al., 2018b). This idea allows us to convert offline algorithm into online algorithms. To be precise, let us consider the first iteration and the first objective function F_1 of our online optimization setting. Note that F_1 remains unknown until the algorithm commits to a choice. If we were in the offline setting, we could have chosen

$$\mathbf{v}^{(k)} \in \text{argmax}_{\mathbf{v} \in \mathcal{K}} \langle \text{oracle-adv}(\nabla F_1(\mathbf{x}^{(k)}), \mathbf{x}^{(k)}), \mathbf{v} \rangle,$$

to obtain the desired regret bounds. The idea of meta-actions is to mimic this process in an online setting as follows. We run K instances of an online linear optimization (OLO) algorithm, $\{\mathcal{E}^{(k)}\}_{k=1}^K$.

```
Input : smoothing radius δ, shrunk constraint set K̂, horizon T, block size L, number of linear
        maximization oracles K, online linear maximization oracles on K̂: E^(1), ···, E^(K),
        number of blocks Q = T/L.
for q = 1, 2, ..., Q do
    Pick any u ∈ argmin_{x∈K̂} ||x||_∞
    x_q^(1) ← u
    for k = 1, 2, ..., K do
        Let v_q^(k) ∈ K̂ be the output of E^(k) in round q.
        x_q^(k+1) ← update(x_q^(k), v_q^(k), u)
    end
    x_q ← x_q^(K+1)
    Let (t_{q,1}, ..., t_{q,L}) be a random permutation of {(q−1)L+1, ..., qL}
    for t = (q−1)L+1, ..., qL do
        Play y_t = x_q and obtain the reward F_t(y_t)
        Find the corresponding l ∈ [L] such that t = t_{q,l}
        for k ∈ [K] such that k ≡ l (mod L) do
            If we have a value oracle, sample u_q^(k) ~ S^{d−1} ∩ L_0 uniformly, otherwise
              u_q^(k) ← 0
            d_q^(k) ← grad-estimate(F_{t_{q,l}}, x_q^(k), u_q^(k))
            g_q^(k) ← oracle-adv(d_q^(k), x_q^(k))
            Pass g_q^(k) as the adversarially chosen vector to E^(k)
        end
    end
end
```

Algorithm 1: Generalized Meta-Frank-Wolfe

Here the number K denotes the number of iterations of the offline Frank-Wolfe algorithm that we intend to mimic. Thus, to maximize $\langle \text{oracle-adv}(\nabla F_1(\mathbf{x}^{(k)}), \mathbf{x}^{(k)}), \cdot \rangle$, we simply use $\mathcal{E}^{(k)}$. Once the function F_1 is revealed to the algorithm, it knows each linear maximization oracle "adversaries" $\{\text{oracle-adv}(\nabla F_1(\mathbf{x}^{(k)}), \mathbf{x}^{(k)})\}_{k=1}^K$. Now, we simply feed each online algorithm $\mathcal{E}^{(k)}$ with the reward $\{\langle \text{oracle-adv}(\nabla F_1(\mathbf{x}^{(k)}), \mathbf{x}^{(k)}), \cdot \rangle\}_{k=1}^K$. We repeat this process for each subsequent function $\{F_t\}_{t \geq 2}$. This idea, combined with Idea 1, allows us to obtain the desired α-regret bounds.

Remark 4. *We assume that every instance $\mathcal{E}^{(k)}$ has the following behavior and guarantee. In every block $1 \leq q \leq Q$, the oracle $\mathcal{E}^{(k)}$ selects a vector $\mathbf{v}_q^{(k)}$ and then the adversary reveals a vector $\mathbf{g}_q^{(k)}$ to the oracle that was chosen independently of $\mathbf{v}_q^{(k)}$. The OLO oracle guarantees that $\sum_{q=1}^Q \langle \mathbf{g}_q^{(k)}, \mathbf{x}^* - \mathbf{v}_q^{(k)} \rangle \leq \mathcal{R}_Q^{\mathcal{E}^{(k)}}$, for some regret function $\mathcal{R}_Q^{\mathcal{E}^{(k)}}$. One possible choice for such an oracle is Follow-the-Perturbed-Leader by (Kalai & Vempala, 2005) that guarantees $\mathcal{R}_Q^{\mathcal{E}^{(k)}} \leq CDB\sqrt{Q}$ where D is the diameter of $\mathcal{K}$, $B = \max_{q,k} \|\mathbf{g}_q^{(k)}\|$ and $C > 0$ is a constant. It follows from the definition of* grad-estimate *that if we have access to gradient oracles, then $B \leq B_1$, while if we have access to value oracles, then $B \leq \frac{d'}{\delta} B_0$.*

3. Random permutations Using random permutations allows us to use less queries at the cost of increased regret. In the context of DR-submodular maximization, this idea was first used in Mono-Frank-Wolfe algorithm in (Zhang et al., 2019). The Mono-Frank-Wolfe corresponds to Algorithm 1 when $K = L$ and we have access to a gradient oracle. Here we describe this idea in the general setting where we allow $K \neq L$, while we still assume access to a gradient oracle. We start by dividing the T functions into $Q = T/L$ blocks of length L. We define $\bar{F}_q$ as the average of functions in the q-th block. For each block q, we pick a random permutation $(t_{q,1}, \ldots, t_{q,L})$ of $\{(q-1)L+1, \ldots, qL\}$ uniformly from the set of all of its permutations. The key insight is that for all $(q-1)L < t \leq qL$, the expected value of F_t is $\bar{F}_q$. Therefor we can estimate $\nabla \bar{F}_q$ using information obtained from functions F_t for $(q-1)L < t \leq qL$ which allows us to apply the idea of meta-actions on the sequence of functions $\{\bar{F}_q\}_{q=1}^Q$.

Input : smoothing radius δ, shrunk constraint set $\hat{\mathcal{K}}$, horizon T, block size L, the number of exploration steps per block $K \leq L$, online linear maximization oracles on $\hat{\mathcal{K}}$: $\mathcal{E}^{(1)}, \cdots, \mathcal{E}^{(K)}$, number of blocks $Q = T/L$.

- **for** $q = 1, 2, \ldots, Q$ **do**
  - Pick any $\underline{\mathbf{u}} \in \operatorname{argmin}_{\mathbf{x} \in \hat{\mathcal{K}}} \|\mathbf{x}\|_\infty$
  - $\mathbf{x}_q^{(1)} \leftarrow \underline{\mathbf{u}}$
  - **for** $k = 1, 2, \ldots, K$ **do**
    - Let $\mathbf{v}_q^{(k)} \in \hat{\mathcal{K}}$ be the output of $\mathcal{E}^{(k)}$ in round q
    - $\mathbf{x}_q^{(k+1)} \leftarrow \text{update}(\mathbf{x}_q^{(k)}, \mathbf{v}_q^{(k)}, \underline{\mathbf{u}})$
  - **end**
  - $\mathbf{x}_q \leftarrow \mathbf{x}_q^{(K+1)}$
  - 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** $t \in \{t_{q,1}, \cdots, t_{q,K}\}$ **then**
      - Find the corresponding $k \in [K]$ such that $t = t_{q,k}$
      - If we have a value oracle, sample $\mathbf{u}_q^{(k)} \sim \mathbb{S}^{d-1} \cap \mathcal{L}_0$ uniformly, otherwise $\mathbf{u}_q^{(k)} \leftarrow \mathbf{0}$
      - Play $\mathbf{y}_t = \mathbf{y}_{t_{q,k}} = \mathbf{x}_q^{(k)} + \delta \mathbf{u}_q^{(k)}$ for F_t (i.e., $F_{t_{q,k}}$) `// Exploration`
      - $\mathbf{d}_q^{(k)} \leftarrow \text{grad-estimate}(F_{t_{q,k}}, \mathbf{x}_q^{(k)}, \mathbf{u}_q^{(k)})$
      - $\mathbf{g}_q^{(k)} \leftarrow \text{oracle-adv}(\mathbf{d}_q^{(k)}, \mathbf{x}_q^{(k)})$
      - Pass $\mathbf{g}_q^{(k)}$ as the adversarially chosen vector to $\mathcal{E}^{(k)}$
    - **else**
      - Play $\mathbf{y}_t = \mathbf{x}_q$ for F_t `// Exploitation`
    - **end**
  - **end**
- **end**

Algorithm 2: Generalized (Semi-)Bandit-Frank-Wolfe

4. Smoothing trick When we do not have access to a gradient oracle, we rely on samples from a value oracle to estimate the gradient. The "smoothing trick" (Flaxman et al., 2005; Hazan et al., 2016; Agarwal et al., 2010; Shamir, 2017; Zhang et al., 2019; Chen et al., 2020; Zhang et al., 2023; Niazadeh et al., 2021; Pedramfar et al., 2023) involves averaging through spherical sampling around a given point. Here we use a variant that was introduced in (Pedramfar et al., 2023).

Definition 1 (Smoothing Trick)**.** *For a function $F : \mathcal{D} \to \mathbb{R}$ defined on $\mathcal{D} \subseteq \mathbb{R}^d$, its δ-smoothed version $\hat{F}$ is given as*

$$\hat{F}_\delta(\mathbf{x}) := \mathbb{E}_{\mathbf{z} \sim \mathbb{B}_\delta^{\text{aff}(\mathcal{D})}(\mathbf{x})}[F(\mathbf{z})] = \mathbb{E}_{\mathbf{v} \sim \mathbb{B}_1^{\text{aff}(\mathcal{D}) - \mathbf{x}}(\mathbf{0})}[F(\mathbf{x} + \delta \mathbf{v})], \tag{5}$$

where $\mathbf{v}$ is chosen uniformly at random from the $\dim(\mathrm{aff}(\mathcal{D}))$*-dimensional ball* $\mathbb{B}_1^{\mathrm{aff}(\mathcal{D})-\mathbf{x}}(\mathbf{0})$*. Thus, the function value $\hat{F}_\delta(\mathbf{x})$ is obtained by "averaging" F over a sliced ball of radius δ around $\mathbf{x}$.*

The power of the smoothing trick lies in the facts that it is a good approximation of the original function (Lemma 1) and there is a simple one-point gradient estimator for the smoothed version of the function (Lemma 2). We will drop the subscript δ when there is no ambiguity.

Note that the domain of $\hat{F}$ is smaller than the domain of F. Therefore, our ability to estimate the gradient is limited to a smaller region compared to the entire feasible set. This limitation should be considered when developing an algorithm for DR-submodular maximization. The notion of "shrunk constraint set" (Zhang et al., 2019; 2023; Niazadeh et al., 2021; Pedramfar et al., 2023) was designed to address this concern. Here we use the variant designed by (Pedramfar et al., 2023). Formally, we choose a point $\mathbf{c} \in \mathrm{relint}(\mathcal{K})$ and a real number $r > 0$ such that $\mathbb{B}_r^{\mathrm{aff}(\mathcal{K})}(\mathbf{c}) \subseteq \mathcal{K}$. Then, for a given $\delta < r$, we define $\hat{\mathcal{K}}_\delta^{\mathbf{c},r} := (1 - \frac{\delta}{r})\mathcal{K} + \frac{\delta}{r}\mathbf{c}$. Clearly if $\mathcal{K}$ is downward-closed, then so is $\hat{\mathcal{K}}_\delta^{\mathbf{c},r}$. We will use the notation $\hat{\mathcal{K}}$ to denote this set when there is no ambiguity.

Putting these ideas together, in order to maximize F over $\mathcal{K}$ with a value oracle, we restrict ourselves to the shrunk constraint set $\hat{\mathcal{K}}$ and maximize $\hat{F}$ over this set using the one-point gradient estimator.

3.2 (Semi-)Bandit Feedback

In the (semi-)bandit feedback setting, we only observe the value or gradient at the point where the action is taken. To adapt Algorithm 1 to this setting, we start by assuming $K \leq L$. In each block, there are L functions that are used to update K linear maximization oracles. Therefore, we may choose K exploration steps in each block where we take actions according to what we queried in Algorithm 1. These actions are informative and allow us to carry out similar analysis to the full information setting. In the remaining $L - K$ steps, we exploit our knowledge of the best action so far to minimize total α-regret. See Algorithm 2 for pseudo-code.

4 α-Regret Guarantees

For brevity, we define $\mathcal{R}^u$ as

$$\mathcal{R}^u = \big(M_0 + (3+D)M_1\big)\,\frac{T\delta}{r} + \begin{cases} (8M_0 + M_2 D^2 \log(K)^2)\frac{T}{8K} + LCDB\sqrt{Q}\log(K) & (C); \\ (M_0 + 2M_2D^2)\frac{T}{4K} + LCDB\sqrt{Q} & \text{Otherwise,} \end{cases}$$

where $B \leq B_1$ if we have access to a gradient oracle and $B \leq \frac{d'}{\delta} B_0$ otherwise and C is the constant in Remark 4.

Theorem 1. *Using Algorithm 1, we have* $\mathbb{E}[\mathcal{R}_\alpha] \leq \mathcal{R}^u$.

The proof is in Appendix F. Note that the number of times any function F_t is queried in Algorithm 1 is K/L. Let β be a real number such that $T^\beta = K/L$. Given a gradient oracle, for any choice of $0 \leq \beta \leq \frac{1}{2}$, we may set $\delta = 0$, $L = T^{\frac{1-2\beta}{3}}$ and therefore $K = T^{\frac{1+\beta}{3}}$ and $Q = T/L = T^{\frac{2+2\beta}{3}}$, to obtain $\mathbb{E}[\mathcal{R}_\alpha] = O(T^{\frac{2-\beta}{3}} \log(T)^2)$ in case (C), and $O(T^{\frac{2-\beta}{3}})$, otherwise. The special case $L = 1$ corresponds to Meta-Frank-Wolfe (Chen et al., 2018b;a; Zhang et al., 2023; Thang & Srivastav, 2021; Mualem & Feldman, 2023) while the special case $L = K$ corresponds to Mono-Frank-Wolfe (Zhang et al., 2019; 2023). Similarly, given a value oracle, for any choice of $0 \leq \beta \leq \frac{1}{2}$, by setting $\delta = T^{-\frac{2+\beta}{5}}$, $L = T^{\frac{2-4\beta}{5}}$ and therefore $K = T^{\frac{2+\beta}{5}}$ and $Q = T/L = T^{\frac{3+4\beta}{3}}$, we see that $\mathbb{E}[\mathcal{R}_\alpha] = O(T^{\frac{3-\beta}{5}} \log(T)^2)$ in Case (C), and $O(T^{\frac{3-\beta}{5}})$, otherwise.

Theorem 2. *Using Algorithm 2, we have* $\mathbb{E}[\mathcal{R}_\alpha] \leq \mathcal{R}^u + 2M_0QK$.

The proof is in Appendix G. In particular, given a gradient oracle, we may set $\delta = 0$, $K = T^{1/4}$ and $L = T^{1/2}$ and therefore $Q = T^{1/2}$, to obtain $\mathbb{E}[\mathcal{R}_\alpha] = O(T^{3/4}\log(T)^2)$ in Case (C) and $O(T^{3/4})$, otherwise. Similarly, when given a value oracle, if we set $K = T^{1/6}$, $L = T^{1/3}$, $\delta = T^{-1/6}$ and therefore $Q = T^{2/3}$, we see that $\mathbb{E}[\mathcal{R}_\alpha] = O(T^{5/6}\log(T)^2)$ in Case (C) and $O(T^{5/6})$, otherwise.

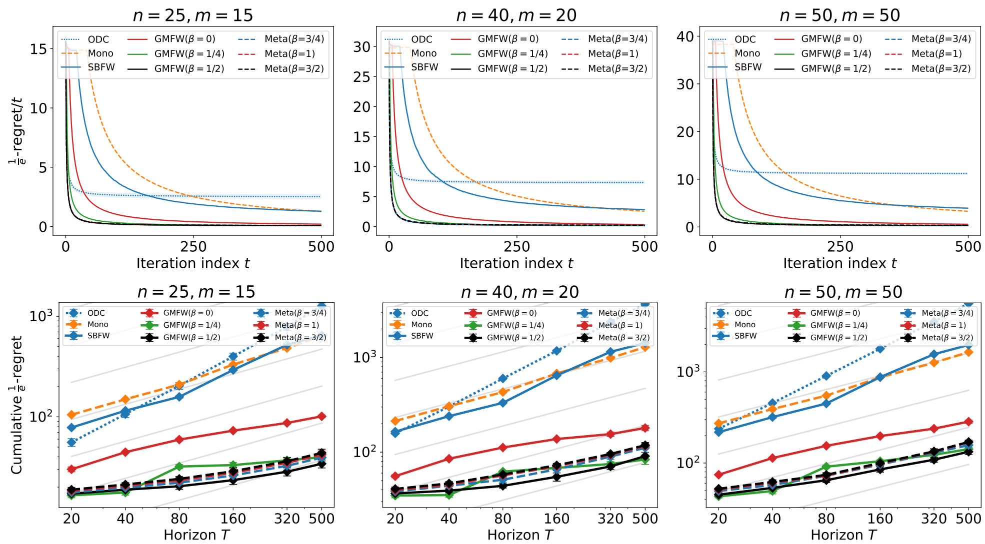

Figure 2: Empirical regret plots for the experiments. The top row depicts time-averaged regret for each round t for a horizon of $T = 500$. The bottom row depicts cumulative regret for multiple horizons with logarithmic scaling. Grey lines in the bottom row represent $y = aT^{1/2}$ curves for different a for visual reference. Colors correspond to regret bounds (i.e. black for $\tilde{O}(T^{1/2})$). Our methods (solid lines) use significantly fewer queries and less computation than baselines (dashed and dotted lines) with similar regret bounds and achieve better regret than baselines using similar numbers of queries and computation.

5 EXPERIMENTS

We test our online continuous DR-submodular maximization algorithms for non-monotone objectives, a downward-closed feasible region, and both full-information and semi-bandit gradient feedback. We briefly describe the setup and highlight key results. See Appendix H for more details. We use online non-convex/non-concave non-monotone quadratic maximization following (Bian et al., 2017a; Chen et al., 2018b; Zhang et al., 2023), randomly generating linear inequalities to form a downward closed feasible region and for each round t we generate a quadratic function $F_t(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{H}\mathbf{x} + \mathbf{h}^\top \mathbf{x} + c$. Similar to (Zhang et al., 2023), we considered three pairs (n, m) of dimensions n and number of constraints m, $\{(25, 15), (40, 20), (50, 50)\}$.

We ran three online algorithms from prior works, **ODC** from (Thang & Srivastav, 2021), **Mono** (full information single query) from (Zhang et al., 2023), and **Meta**(β) from (Zhang et al., 2023), where we used query parameters $\beta = \{3/4, 1, 3/2\}$; here and in the following we only explicitly mention the query parameter so that there are T^β queries per round and other algorithm parameters implicit. We ran our Algorithm 1 (**GMFW**(β) for short) with query parameter $\beta = \{0, 1/4, 1/2\}$ and our semi-bandit Algorithm 2 (**SBFW** for short). Fig. 1 depicts regret bound and query complexity trade offs for full-information methods.

Figure 2 shows both averaged regret within runs for a fixed horizon (top row) and cumulative regret for different horizons, averaged over 10 independent runs. See Fig. 2's caption for a description. Average run-times for a horizon of $T = 100$ are displayed in Table 3 in Appendix H. Major differences in run-times is in large part due to the number of online linear maximization oracles used, which is in part related to the number of per-round queries.

In each experiment, our **GMFW**($\beta = 1/2$) (black solid line) performs the best overall, despite using significantly fewer gradient queries and significantly less computation than any of the **Meta** algorithms. Our **GMFW**($\beta = 0$) (red solid line) performs better than the baseline **Mono** (orange dashed line; designed for the same amount of feedback).

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under grants CCF-2149588 and CCF-2149617. We also thank the authors of (Zhang et al., 2023) for sharing their code.

REPRODUCIBILITY STATEMENT

Source code for our algorithms is available at `https://github.com/yididiyan/unified-dr-submodular`. All assumptions are included in the main paper in Assumption 1. All proofs are included in the appendices.

REFERENCES

Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In Adam Tauman Kalai and Mehryar Mohri (eds.), *Proceedings of the 23rd Annual Conference on Learning Theory (COLT 2010)*, pp. 28–40, 2010.

An Bian, Kfir Levy, Andreas Krause, and Joachim M Buhmann. Continuous DR-submodular maximization: Structure and algorithms. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017a.

Andrew An Bian, Baharan Mirzasoleiman, Joachim Buhmann, and Andreas Krause. Guaranteed Non-convex Optimization: Submodular Maximization over Continuous Domains. In Aarti Singh and Jerry Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 111–120. PMLR, 20–22 Apr 2017b.

Yatao Bian, Joachim Buhmann, and Andreas Krause. Optimal continuous DR-submodular maximization and applications to provable mean field inference. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 644–653. PMLR, 09–15 Jun 2019.

Gábor Braun, Alejandro Carderera, Cyrille W. Combettes, Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Sebastian Pokutta. Conditional Gradient Methods. *arXiv preprint arXiv:2211.14103*, November 2022.

Lin Chen, Christopher Harshaw, Hamed Hassani, and Amin Karbasi. Projection-free online optimization with stochastic gradient: From convexity to submodularity. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 814–823. PMLR, 10–15 Jul 2018a.

Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization. In Amos Storkey and Fernando Perez-Cruz (eds.), *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1896–1905. PMLR, 09–11 Apr 2018b.

Lin Chen, Mingrui Zhang, Hamed Hassani, and Amin Karbasi. Black box submodular maximization: Discrete and continuous settings. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1058–1070. PMLR, 26–28 Aug 2020.

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*, 2023.

Alon Cohen and Tamir Hazan. Following the perturbed leader for online structured learning. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1034–1042, Lille, France, 07–09 Jul 2015. PMLR.

Josip Djolonga and Andreas Krause. From map to marginals: Variational inference in Bayesian submodular models. *Advances in Neural Information Processing Systems*, 27, 2014.

Donglei Du. Lyapunov function approach for approximation algorithm design and analysis: with applications in submodular maximization. *arXiv preprint arXiv:2205.12442*, 2022.

Donglei Du, Zhicheng Liu, Chenchen Wu, Dachuan Xu, and Yang Zhou. An improved approximation algorithm for maximizing a DR-submodular function over a convex set. *arXiv preprint arXiv:2203.14740*, 2022.

Christoph Dürr, Nguyen Kim Thang, Abhinav Srivastav, and Léo Tible. Non-monotone DR-submodular maximization: Approximation and regret guarantees. *arXiv preprint arXiv:1905.09595*, 2019.

Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

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*, pp. 385–394, 2005.

Shayan Oveis Gharan and Jan Vondrák. Submodular maximization by simulated annealing. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1098–1116. Society for Industrial and Applied Mathematics, 2011.

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.

Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient methods for submodular maximization. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

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

Shinji Ito and Ryohei Fujimaki. Large-scale price optimization via network flow. *Advances in Neural Information Processing Systems*, 29, 2016.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Amin Karbasi, Hamed Hassani, Aryan Mokhtari, and Zebang Shen. Stochastic continuous greedy ++: When upper and lower bounds match. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

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.

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

Siddharth Mitra, Moran Feldman, and Amin Karbasi. Submodular+ concave. *Advances in Neural Information Processing Systems*, 34:11577–11591, 2021.

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.

Loay Mualem and Moran Feldman. Resolving the approximability of offline and online non-monotone DR-submodular maximization over general convex sets. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 2542–2564. PMLR, 25–27 Apr 2023.

Rad Niazadeh, Tim Roughgarden, and Joshua R Wang. Optimal algorithms for continuous non-monotone submodular and DR-submodular maximization. *The Journal of Machine Learning Research*, 21(1):4937–4967, 2020.

Rad Niazadeh, Negin Golrezaei, Joshua R. Wang, Fransisca Susan, and Ashwinkumar Badanidiyuru. Online learning via offline greedy algorithms: Applications in market design and optimization. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, EC '21, pp. 737–738, New York, NY, USA, 2021. Association for Computing Machinery.

Mohammad Pedramfar, Christopher Quinn, and Vaneet Aggarwal. A unified approach for maximizing continuous DR-submodular functions. *Advances in Neural Information Processing Systems*, 36:61103–61114, 2023.

Ohad Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *Journal of Machine Learning Research*, 18(52):1–11, 2017.

Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. In *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008.

Nguyen Kim Thang and Abhinav Srivastav. Online non-monotone DR-submodular maximization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(11):9868–9876, May 2021.

Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM Journal on Computing*, 42(1):265–304, 2013.

Zongqi Wan, Jialin Zhang, Wei Chen, Xiaoming Sun, and Zhijie Zhang. Bandit Multi-linear DR-Submodular Maximization and Its Applications on Adversarial Submodular Bandits. *arXiv preprint arXiv:2305.12402*, May 2023.

Mingrui Zhang, Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization: From full-information to bandit feedback. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

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*, pp. 26116–26134. PMLR, 2022.

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 Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 3515–3537. PMLR, 25–27 Apr 2023.