Decentralized Projection-free Online Upper-Linearizable Optimization with Applications to DR-Submodular Optimization

Anonymous authors
Paper under double-blind review

Abstract

We introduce a novel framework for decentralized projection-free optimization, extending projection-free methods to a broader class of upper-linearizable functions. Our approach leverages decentralized optimization techniques with the flexibility of upper-linearizable function frameworks, effectively generalizing traditional DR-submodular function optimization. We obtain the regret of $O(T^{1-\theta/2})$ with communication complexity of $O(T^{\theta})$ and number of linear optimization oracle calls of $O(T^{2\theta})$ for decentralized upper-linearizable function optimization, for any $0 \leq \theta \leq 1$. This approach allows for the first results for monotone up-concave optimization with general convex constraints and non-monotone up-concave optimization with general convex constraints. Further, the above results for first order feedback are extended to zeroth order, semi-bandit, and bandit feedback.

1 Introduction

In recent years, the challenge of optimizing continuous adversarial γ-weakly up-concave functions, which includes DR-submodular and concave functions as special cases, has gained significant attention. This problem represents a critical subset of non-convex optimization problems, especially relevant in the fields of machine learning and statistics. Solutions to this problem have a wide range of practical applications, including revenue maximization, mean-field inference, and recommendation systems (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; Pedramfar et al., 2023; 2024). The optimization process is typically formulated as a repeated game between an optimizer and an adversary. During each iteration, the optimizer selects an action while the adversary chooses a γ-weakly up-concave reward function. The optimizer can either query the reward function at any arbitrary point within the domain (in the case of full information feedback) or only at the chosen action (in semi-bandit/bandit feedback scenarios), with the feedback potentially being noisy or deterministic. Recently, Pedramfar & Aggarwal (2024a) introduced the concept of *upper-linearizable* functions, which broadened the classical notions of concavity and DR-submodularity in various contexts. They also proposed a meta-algorithm that transforms linear maximization methods into algorithms capable of optimizing upper linearizable functions. Although the regret bounds for such algorithms have been studied in centralized settings (Pedramfar & Aggarwal, 2024a), this work extends the analysis to decentralized setting, which is motivated by applications in multi-agent systems and sensor networks (Li et al., 2002; Xiao et al., 2007; Mokhtari et al., 2018; Duchi et al., 2011). The objective of this paper is to establish regret bounds for optimizing upper-linearizable functions under different feedback types in a decentralized setup, where communication between agents is limited.

In decentralized optimization, each node in the network represents a local decision maker that must select its own decision and subsequently receive a local reward function. The objective of each local decision maker is to minimize its α-regret, which is determined by the average of local functions at each iteration. To achieve this, nodes are permitted to communicate with their neighbors and exchange local information. Decentralized optimization has been extensively studied in online convex optimization (Duchi et al., 2011; Sundhar Ram et al., 2010; Yan et al., 2012; Wang et al., 2022; Zhang et al., 2017; 2023b; Elgabli et al., 2020). Wan et al. (2022) proposed a projection-free distributed algorithm for online convex optimization with sublinear communication complexity. Recently, this problem has been explored within the scope of DR-submodular optimization (Xie et al., 2019; Gao et al., 2021; Zhang et al., 2022; Liao et al., 2023). Specifically, for monotone

Table 1: Decentralized Projection-free up-concave maximization

F	Set	Feedback		Reference	Appx.	$\log_T(\alpha\text{-Regret})$	$\log_T(\text{Communication})$	$\log_T(\text{LOO calls})$	Range of θ
Monotone	$0 \in \mathcal{K}$	∇F	full information	(Zhu et al., 2021)	$1-e^{-1}$	1/2	5/2	5/2	-
				(Zhang et al., 2023a)	$1-e^{-1}$	4/5	1	1	-
				(Liao et al., 2023)	$1-e^{-1}$	3/4	1/2	1	-
				Theorem 2	$1-e^{-\gamma}$	$1-\theta/2$	θ	2θ	[0, 1]
			semi-bandit	Theorem 4	$1-e^{-\gamma}$	$1-\theta/2$	θ	2θ	[0, 2/3]
		F	full information	Theorem 5	$1-e^{-\gamma}$	$1-\theta/4$	θ	2θ	[0, 1]
			bandit	Theorem 6	$1-e^{-\gamma}$	$1-\theta/4$	θ	2θ	[0, 4/5]
	general	∇F	semi-bandit	Theorem 2	$\gamma^2/(1+\gamma^2)$	$1-\theta/2$	θ	2θ	[0, 1]
		F	bandit	Theorem 3	$\gamma^2/(1+\gamma^2)$	$1-\theta/4$	θ	2θ	[0, 1]
Non-Mono	general	∇F	full information	Theorem 2	$(1-h)/4$	$1-\theta/2$	θ	2θ	[0, 1]
			semi-bandit	Theorem 4	$(1-h)/4$	$1-\theta/2$	θ	2θ	[0, 2/3]
		F	full information	Theorem 5	$(1-h)/4$	$1-\theta/4$	θ	2θ	[0, 1]
			bandit	Theorem 6	$(1-h)/4$	$1-\theta/4$	θ	2θ	[0, 4/5]

This table compares the different results for the regret for up-concave maximization. Here $h := \min_{\mathbf{z}\in\mathcal{K}} \|\mathbf{z}\|_\infty$. Communication refers to the communication complexity, and LOO calls refers to the number of calls to the Linear Optimization Oracle. Further, the results hold for any θ in the range specified in the last column.

DR-submodular functions over constraint sets that include the origin, algorithms with regret guarantees have been proposed. Notably, this is just one instance of an upper-linearizable function class proposed by Pedramfar & Aggarwal (2024a), a broader class that encompasses several cases, including (i) monotone γ-weakly up-concave functions (generalizing DR-submodular functions) over general convex sets, (ii) monotone γ-weakly up-concave functions over convex sets containing the origin, and (iii) non-monotone up-concave optimization over general convex sets. Therefore, in this work, we focus on decentralized algorithms for the more general class of upper-linearizable functions and provide regret guarantees for all the aforementioned cases.

Algorithms for minimizing the α-regret in DR-submodular optimization can be broadly categorized into two types. The first category consists of projection-based algorithms, such as online gradient ascent (OGA) (Chen et al., 2018b) and online boosting gradient ascent (OBGA) (Zhang et al., 2022), which require the computation of a projection onto the decision set at each iteration. The second category includes projection-free algorithms, such as Meta-Frank-Wolfe (Meta-FW) (Chen et al., 2018a;b) and Mono-Frank-Wolfe (Mono-FW) (Zhang et al., 2019), which bypass the projection step by employing linear optimization over the decision set. In many practical applications, including matrix completion (Chandrasekaran et al., 2009), network routing (Hazan & Luo, 2016), and structural SVMs (Lacoste-Julien et al., 2013), the projection operation can be significantly more computationally expensive than the linear optimization step, making projection-free algorithms particularly attractive in such scenarios. This motivates us to provide projection-free algorithms for the problem.

In this paper, we present an algorithm for optimizing upper-linearizable functions, achieving a regret bound of $O(T^{1-\theta/2})$ with a communication complexity of $O(T^{\theta})$ and number of linear optimization oracle calls of $O(T^{2\theta})$. We note that for $\theta = 1$, this achieves the optimal regret of $O(\sqrt{T})$ while having the number of linear optimization oracle calls as $O(T^2)$, while still improving the results in Zhu et al. (2021) reducing the communication complexity and number of linear optimization oracle calls in their special case. Further, for $\theta = 1$, the result matches that by Liao et al. (2023) in their special case and improves all metrics as compared to the result by Zhang et al. (2023a). The result in this paper gives a tradeoff between the metrics that shows that decreasing regret causes an increase in the communication complexity and the number of linear optimization oracle calls.

The algorithm requires a querying function $\mathcal{G}$, which corresponds to a semi-bandit query when dealing with monotone γ-weakly up-concave functions over general convex sets, and a first-order full-information query in two cases: (i) monotone γ-weakly up-concave functions over convex sets containing the origin, and (ii) non-monotone up-concave optimization over general convex sets. By obtaining results for these queries, we adapt the meta-algorithms by Pedramfar & Aggarwal (2024a) to derive new results for different feedback types across various settings, including first-order full information, zeroth-order full information, semi-bandit, and bandit feedback. A summary of the results across these cases is provided in Table 1. It is important to note that prior work only addressed the case of monotone 1-weakly DR-submodular functions over convex sets containing the origin with first-order full-information feedback, while the remaining results in the table are novel contributions to the literature.

We summarize the contributions and technical novelties of this work as follows.

Contributions:

1. First result for online monotone DR-submodular maximization over general convex sets and online non-monotone DR-submodular maximization over general convex sets in decentralized setting. Further, the result for monotone DR-submodular with $0 \in \mathcal{K}$ has been extended to γ-weakly DR-submodular functions while the previous work only considered 1-weakly DR-submodular functions (See Table 1 for the results).

2. Even in the 1-weakly DR-submodular functions with $0 \in \mathcal{K}$, there were results on multiple tradeoff points on regret-communication complexity tradeofff. We combined these results demonstrating a tradeoff between the two objectives (See Table 1 for such comparisons).

3. First results for the more general class of up-concave functions, which includes DR-submodular functions as a subset. DR-submodular functions satisfy $\nabla f(x) \leq \nabla f(y)$ for all $x \geq y$ while up-concave functions only satisfy $\langle \nabla f(x), x - y \rangle \leq \langle \nabla f(y), x - y \rangle$ for all $x \geq y$.

4. First result for online decentralized optimization problem for upper-linearizable functions, a function class much broader than monotone DR-submodular maximization over convex sets containing the origin, giving opportunity to solve problems beyond submodular maximization.

5. A meta-algorithm extending the framework of (Pedramfar & Aggarwal, 2024a) from centralized to decentralized setting, generating first results for other feedbacks. In total we proposed 10 algorithms, 9 of which are the first in their respective settings, including trivial vs non-trivial queries, gradient vs function value oracle, which handles different function instances of our upper-linearizable function class. Indeed, the approach allows us to come up with the first results on decentralized DR-submodular optimization with bandit feedback in any of the classes.

Technical Novelty

1. We note that previous works on decentralized DR-submodular optimization assumed monotone 1-weakly DR-submodular functions with $0 \in \mathcal{K}$. Thus, we needed a non-trivial approach to extend the setup to include non-monotonicity of the function, γ-weakly DR-submodular functions, and general convex set constraints. This is done using the notion of upper-linearizable functions, allowing us to obtain the first results for (i) monotone γ-weakly up-concave functions over convex sets with $0 \in \mathcal{K}$, (ii) monotone up-concave functions with general convex sets, and (iii) non-monotone up-concave functions with general convex sets.

2. We proved that the framework proposed by (Pedramfar & Aggarwal, 2024a) which was proposed for and examined in the centralized setting, can be extended to the decentralized setting with proper adaptations. The notion of regret changes in the decentralized setting, where we have to consider the average of loss function among all agents instead of just one function, as is the case in a the centralized setting. Note that the changes in the definition of online optimization between centralized and decentralized optimization makes applications of meta-algorithms used in (Pedramfar & Aggarwal, 2024a) non-trivial. The centralized version have no notion of communication between nodes and it requires nuance in how one goes about applying meta-algorithms designed for centralized setting to base algorithms that are decentralized.

3. In the cases of monotone functions over convex sets containing the origin and non-monotone functions, the main algorithm, i.e., `DOLCO`, requires first order full-information feedback. In the centralized setting, the `SFTT` meta-algorithm, described in (Pedramfar & Aggarwal, 2024a), is designed to convert the such algorithms into algorithms only requiring semi-bandit feedback. However, this algorithm does not directly work in the decentralized setup. We design Algorithms 5 and 7 using the idea of `SFTT` meta-algorithm. The challenge here is to ensure that the `SFTT` blocking mechanism interacts properly with the existing blocking mechanism of the base algorithm.

2 Related Works

Upper-Linearizable Function: Pedramfar & Aggarwal (2024a) introduced the concept of upper-linearizable functions, a class that generalizes concavity and DR-submodularity across various settings, including both monotone and non-monotone cases over different convex sets. They also explored projection-free algorithms in centralized settings for these setups. Additionally, they proposed several meta-algorithms that adapt the feedback type, converting full-information queries to trivial queries and transitioning from first-order to zeroth-order feedback.

Decentralized Online DR-Submodular Maximization: Zhu et al. (2021) introduced the first decentralized online projection-free algorithm for monotone DR-submodular maximization with a constraint set containing the origin, known as the Decentralized Meta-Frank-Wolfe (DMFW). This algorithm provides a $1 - 1/e$ approximation guarantee and achieves an expected regret bound of $O(\sqrt{T})$, where T denotes the time horizon. However, in each round, DMFW requires $T^{3/2}$ stochastic gradient estimates for each local objective function, followed by the transmission of these gradient messages across the network, resulting in significant computational and communication overhead, especially for large T (with a total communication complexity of $O(T^{5/2})$). To address these limitations, Zhang et al. (2023a) proposed the One-shot Decentralized Meta-Frank-Wolfe Algorithm, which achieves $(1 - 1/e)$-regret of $O(T^{4/5})$ while lowering the communication complexity and the number of linear optimization oracle (LOO) calls to $O(T)$. Additionally, Liao et al. (2023) introduced a projection-free online boosting gradient ascent algorithm that achieves $(1 - 1/e)$-regret of $O(T^{3/4})$ with a communication complexity of $O(T^{1/2})$ and number of LOO calls as $O(T)$. On the other hand, the projection-based Decentralized Online Boosting Gradient Ascent algorithm proposed in Zhang et al. (2023a) obtains $(1 - 1/e)$-regret of $O(\sqrt{T})$ with a communication complexity of $O(T)$.

In this paper, we present algorithms for optimizing upper-linearizable functions, achieving a regret bound of $O(T^{1-\theta/2})$, in first order feedback case, and $O(T^{1-\theta/4})$, in zeroth order feedback case, with a communication complexity of $O(T^{\theta})$ and number of LOO calls of $O(T^{2\theta})$. In first order full-information case, for $\theta = 1$, we show that the regret and the communication complexity matches the best known projection-based algorithm in (Zhang et al., 2023a) and for $\theta = 1/2$, the results match that in (Liao et al., 2023) in the special case of monotone 1-weakly DR-submodular functions with the convex set containing the origin.

3 Problem Setup and Definitions

This paper considers a decentralized setting involving N agents connected over a network represented by an undirected graph $G = (V, E)$, where $V = \{1, \cdots, N\}$ is the set of nodes, and $E \subseteq V \times V$ is the set of edges. Each agent $i \in V$ acts as a local decision maker and can communicate only with its neighbors, defined as $N_i = \{j \in V \mid (i, j) \in E\} \cup \{i\}$. To model the communication between agents, we introduce a non-negative weight matrix $A = [a_{i,j}] \in \mathbb{R}_+^{N\times N}$, which is supported on the graph G. The matrix A is symmetric and doubly stochastic, with $a_{i,j} > 0$ only if $(i, j) \in E$ or $i = j$. Since $A^\top = A$ and $A\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ denotes the vector of all ones, $\mathbf{1}$ is an eigenvector of A with eigen-value of 1. We further assume that the second largest eigenvalue of A, λ_2, is strictly less than 1.

We formalize online optimization as a repeated game between N agents and an adversary - a game lasting T rounds on a convex domain $\mathcal{K}$, and all players know T and $\mathcal{K}$. In a decentralized setting, in t^{th} round, the agent i chooses an action $\mathbf{x}_t^i$ from an action set $\mathcal{K} \subseteq \mathbb{R}^d$, then the adversary chooses a loss function $f_{t,i} \in \mathcal{F}$ and a query oracle for $f_{t,i}$. We assume that $f_{t,i}(\mathbf{x})$ is differentiable and continuous. Further, the agent i chooses a point $\mathbf{y}_t^i$ and receives the response of the oracle at the given point. [1] We say the agent query function is *trivial* if $\mathbf{y}_t^i = \mathbf{x}_t^i$ for all $1 \leq t \leq T$. We assume that the radius of the convex set $\mathcal{K} \in \mathcal{X}$ is bounded by R, i.e., $R = \max_{\mathbf{x}\in\mathcal{K}} \|\mathbf{x}\|$.

The goal for each agent i is to optimize the aggregate function over the network over time: $\sum_{t=1}^{T} \sum_{j=1}^{N} f_{t,j}(\mathbf{x}^i)$ (Zhu et al., 2021; Zhang et al., 2023a). Thus, with an approximation coefficient $0 < \alpha \leq 1$, we define the

[1] In general, the agent may query more than one point. However, in all algorithms considered in this paper, agents only require a single query.

α-regret for agent i to be:

$$\mathcal{R}_\alpha^i := \alpha \max_{\mathbf{x}\in\mathcal{K}} \frac{1}{N}\sum_{t=1}^{T}\sum_{j=1}^{N} f_{t,j}(\mathbf{x}) - \frac{1}{N}\sum_{t=1}^{T}\sum_{j=1}^{N} f_{t,j}(\mathbf{x}_t^i),$$

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. If the agent query function is non-trivial, then we say the agent requires *full-information feedback*. We further assume that the query function is stochastic, which means that the output of the query oracle is a noisy unbiased estimate of the desired query. Let the query function for function $f_{t,i}$ at point $\mathbf{x}$ be given as $\mathfrak{q}(f_{t,i}, \mathbf{x})$. Then, for any $\mathbf{x} \in \mathcal{K}$, $t = 1, \cdots, T$ and agent $i = 1, \cdots, N$, the oracle returns $\mathbf{o}_{t,i} = \tilde{\mathfrak{q}}(f_{t,i}, \mathbf{x})$, which we write as $\tilde{\mathfrak{q}}_{t,i}(\mathbf{x})$ for simplicity, is unbiased and its norm is bounded by G, i.e.,

$$\mathbb{E}[\tilde{\mathfrak{q}}_{t,i}(\mathbf{x})] = \mathfrak{q}(f_{t,i}, \mathbf{x}) \quad \text{and} \quad \|\tilde{\mathfrak{q}}_{t,i}(\mathbf{x})\| \le G.$$

The class of functions $\mathcal{F}$ is assumed to be the class of *upper-linearizable* functions proposed in Pedramfar & Aggarwal (2024a). The function class $\mathcal{F}$ is upper-linearizable if there exists $\mathfrak{g} : \mathcal{F} \times \mathcal{K} \to \mathbb{R}^d$, $h : \mathcal{K} \to \mathcal{K}$, and constants $0 < \alpha \le 1$ and $\beta > 0$ such that

$$\alpha f(\mathbf{y}) - f(h(\mathbf{x})) \le \beta \left(\langle \mathfrak{g}(f, \mathbf{x}), \mathbf{y} - \mathbf{x}\rangle\right).$$

For the proposed algorithm for upper-linearizable optimization, we will use query function as $\mathfrak{q} = \mathfrak{g}$. We further assume that the query function $\mathfrak{g}(f_{t,i}, \mathbf{x})$ is L_1-Lipschitz. We assume that $\mathcal{F}$ is M_1-Lipschitz continuous and first order query oracles for $\mathcal{F}$ are bounded by G. In all the examples of upper-linearizable functions and query algorithms considered in this paper, the first order query oracle being bounded by G implies that $\tilde{\mathfrak{g}}$ is also bounded by G. (See Algorithms 2 and 3)

Pedramfar & Aggarwal (2024a) demonstrated the generality of this class of functions by showing that this includes concave functions as well as up-concave functions in many scenarios. More precisely, this class includes (i) monotone γ-weakly up-concave functions over general convex sets, (ii) monotone γ-weakly up-concave functions over convex sets containing the origin, and (iii) non-monotone up-concave optimization over general convex sets. We also note that the query oracle is trivial in case (i), and is non-trivial in cases (ii) and (iii). We also note that the query function $\mathfrak{g}$ for these cases is Lipschitz continuous (assumed in this paper) if the up-concave function is L-smooth, which is a commonly used assumption in the literatures (Liao et al., 2023; Fazel & Sadeghi, 2023; Zhang et al., 2022; 2024). The details of the up-concave functions with the corresponding query oracles is provided in the Supplementary for completeness.

Lemma 1 (Lemma 1 in Liao et al. (2023))**.** *Let $\mathcal{B}$ be an unit ball centered at the origin. There exists an algorithm $\mathcal{O}_{IP}$ referred to as infeasible projection oracle over any convex set $\mathcal{K} \subseteq R\mathcal{B}$, which takes the set $\mathcal{K}$, a pair of points $(\mathbf{x}_0, \mathbf{y}_0) \in \mathcal{K} \times \mathbb{R}^n$, and an error tolerance parameter ϵ as the input, and can output*

$$(\mathbf{x}, \tilde{\mathbf{y}}) = \mathcal{O}_{IP}(\mathcal{K}, \mathbf{x}_0, \mathbf{y}_0, \epsilon)$$

such that $(\mathbf{x}, \tilde{\mathbf{y}}) \in \mathcal{K} \times R\mathcal{B}$, $\|\mathbf{x} - \tilde{\mathbf{y}}\|^2 \le 3\epsilon$, and $\forall \mathbf{z} \in \mathcal{K}$, $\|\tilde{\mathbf{y}} - \mathbf{z}\|^2 \le \|\mathbf{y}_0 - \mathbf{z}\|^2$. Moreover, such an oracle $\mathcal{O}_{IP}$ can be implemented by at most

$$\left\lceil \frac{27R^2}{\epsilon} - 2 \right\rceil \max\left(1, \frac{\|\mathbf{x}_0 - \mathbf{y}_0\|^2(\|\mathbf{x}_0 - \mathbf{y}_0\|^2 - \epsilon)}{4\epsilon^2} + 1\right) \tag{1}$$

LOO calls.

Having described the infeasible projection and its implementation with LOO calls, we introduce a `DOCLO` algorithm which is given in Algorithm 1. The algorithm uses the hyper-parameters $\mathcal{K}$, η, and ϵ, which will be described in the results. In addition, let $\tilde{\mathfrak{g}} = \tilde{\mathfrak{q}}$. Each agent i in the network stores two variables, decision variable $\mathbf{x}^i \in \mathcal{K}$ and an auxiliary variable $\tilde{\mathbf{y}}^i$ to help with infeasible projection, initialized in line 2 as $\mathbf{x}_1^i$ and $\tilde{\mathbf{y}}_1^i$ to be some $\mathbf{c} \in \mathcal{K}$. We divide time T into T/K blocks, each of size K. Line 3 loops over the blocks. In each time t inside block m, agent i plays $h(\mathbf{x}_m^i)$ (line 6). Further, each agent i runs the query algorithm $\mathcal{G}$ with access to $\mathbf{x}_m^i$ to obtain $\mathbf{o}_t^i$ (line 7). We note that $h(\cdot)$ and $\mathcal{G}$ are determined by the characteristics of the upper-linearizable function class.

At the end of each block, each agent i communicates $\mathbf{x}_m^i$ and $\tilde{\mathbf{y}}_m^i$ with neighbours (line 9), which are combined with the weight matrix A. Thus, each agent i receives $\sum_{j\in\mathcal{N}_i} a_{ij}\tilde{\mathbf{y}}_m^j$ and $\sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}_m^j$, where $\mathcal{N}_i$ denotes the neighbours of agent i. Based on the information received, each agent i computes $\mathbf{y}_{m+1}^i = \sum_{j\in\mathcal{N}_i} a_{ij}\tilde{\mathbf{y}}_m^j + \eta\sum_{t\in\mathcal{T}_m}\mathbf{o}_t^i$ (line 10), then performs an infeasible projection to compute $(\mathbf{x}_{m+1}^i, \tilde{\mathbf{y}}_{m+1}^i) = \mathcal{O}_{IP}(\mathcal{K}, \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}_m^j, \mathbf{y}_{m+1}^i, \epsilon)$ using the algorithm $\mathcal{O}_{IP}$ described in Lemma 1 (line 11).

Algorithm 1 Decentralized Online Continuous Upper-Linearizable Optimization - `DOCLO`

1: **Input:** decision set $\mathcal{K}$, horizon T, block size K, step size η, error tolerance ϵ, number of agents N, weight matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}_+^{N\times N}$, query algorithm $\mathcal{G}$, transformation map $h(\cdot)$
2: Set $\mathbf{x}_1^i = \tilde{\mathbf{y}}_1^i = \mathbf{c} \in \mathcal{K}$ for any $i = 1, \cdots, N$
3: **for** $m = 1, \cdots, T/K$ **do**
4:   **for** each agent $i = 1, \cdots, N$ in parallel **do**
5:     **for** $t = (m-1)K+1, \ldots, mK$ **do**
6:       Play $h(\mathbf{x}_m^i)$
7:       Run $\mathcal{G}$ with access to $\mathbf{x}_m^i$ and get $\mathbf{o}_t^i = \tilde{\mathfrak{g}}(f_{t,i}, \mathbf{x}_m^i)$
8:     **end for**
9:     Communicate $\mathbf{x}_m^i$ and $\tilde{\mathbf{y}}_m^i$ with neighbours
10:     $\mathbf{y}_{m+1}^i \leftarrow \sum_{j\in\mathcal{N}_i} a_{ij}\tilde{\mathbf{y}}_m^j + \eta\sum_{t\in\mathcal{T}_m}\mathbf{o}_t^i$
11:     $(\mathbf{x}_{m+1}^i, \tilde{\mathbf{y}}_{m+1}^i) \leftarrow \mathcal{O}_{IP}(\mathcal{K}, \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}_m^j, \mathbf{y}_{m+1}^i, \epsilon)$
12:   **end for**
13: **end for**

4 Analysis of the Proposed Algorithm

In this section, we will provide the key results for the proposed algorithm, including the regret, communication complexity, and the number of LOO calls used by the proposed algorithm. The result is given in the following Theorem.

Theorem 1. *Algorithm 1 ensures that the α-regret for agent i is bounded as*

$$\begin{aligned}\mathbb{E}\left[R_\alpha^i\right] &\le \beta\left[\frac{2R^2}{\eta} + \frac{18\epsilon T}{\eta K} + 7\eta TKG^2 + 13TG\sqrt{3\epsilon}\right] \\ &+ \frac{\beta}{1-\lambda_2}\left(\frac{12\epsilon T}{\eta K} + 9\eta TKG^2 + 12TG\sqrt{3\epsilon}\right) \\ &+ (G + 2L_1R)(N^{1/2}+1)\beta\left(3\sqrt{2\epsilon} + \frac{(3\eta KG + 2\sqrt{3\epsilon})}{1-\lambda_2}\right).\end{aligned}$$

Further, the communication complexity is $O(T/K)$. Finally, the number of LOO calls are upper bounded as $\frac{27TR^2}{\epsilon K}\left(8.5 + 5.5\frac{K^2\eta^2G^2}{\epsilon} + \frac{K^4\eta^4G^4}{\epsilon^2}\right)$. In particular, if we set $\epsilon = K^2\eta^2G^2$, then we have

$$\mathbb{E}\left[R_\alpha^i\right] = O\left(\frac{1}{\eta} + \eta TKG^2\right),$$

and the number of LOO calls is $O(\frac{T}{\epsilon K})$.

Proof. The detailed proof of Theorem 1 is provided in the Appendix. For the completeness of argument, we provide a high-level outline of proof for the regret bound and the communication complexity.

Regret: Note that instead of a 1-weakly DR-submodular function which has the nice property of $\nabla f(\mathbf{x}) \geq f(\mathbf{y}), \forall \mathbf{x} \leq \mathbf{y}$, we are dealing with upper linearizable functions, a much more generalized function class that includes any function that satisfies $\alpha f(\mathbf{y}) - f(h(\mathbf{x})) \leq \beta\left(\langle \mathfrak{g}(f, \mathbf{x}), \mathbf{y} - \mathbf{x}\rangle\right)$ for some functional $\mathfrak{g}$, function h, and constants $0 < \alpha \leq 1$ and $\beta > 0$ as previously described, of which 1-weakly DR-submodular function is an instance.

As is common to bound regret of convex algorithm with first order linear approximation (Orabona, 2019), we bound the regret using the inner product of the distance in action space and $\mathfrak{g}$ function space to approximate the function value, with the help of law of iterated expectation. Let $\mathbf{x}^* \in \operatorname{argmax}_{\mathbf{u}\in\mathcal{K}} \frac{1}{N}\sum_{t=1}^{T}\sum_{i=1}^{N} f_{t,i}(\mathbf{u})$ denote the optimal action, we have

$$\begin{aligned}
\mathbb{E}[\mathcal{R}_\alpha^j] &= \frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\mathbb{E}\left[\alpha f_{t,i}(\mathbf{x}^*) - f_{t,i}(h(\mathbf{x}_m^j))\right] \\
&\leq \frac{\beta}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\mathbb{E}\left[\langle \mathbf{x}^* - \mathbf{x}_m^j, \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^j)\rangle\right]
\end{aligned}$$

Rearranging terms to better leverage the communication structure of the decentralized network, we have:

$$\begin{aligned}
\frac{1}{\beta}\mathbb{E}[\mathcal{R}_\alpha^j] \leq{}& \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\langle \mathbf{x}^* - \mathbf{x}_m^i, \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\rangle\right] \\
&+ \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\langle \mathbf{x}_m^i - \mathbf{x}_m^j, \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\rangle\right] \\
&+ \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\langle \mathbf{x}^* - \mathbf{x}_m^j, \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^j) - \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\rangle\right]
\end{aligned} \tag{2}$$

Let the three parts in Equation (2) be respectively P_1, P_2, P_3. Through exploitation of properties of the loss functions and domain, the update rule (line 10) and the infeasible projection operation (line 11) in Algorithm 1, we obtain the upper bound of the expectation of each part:

$$\begin{aligned}
\mathbb{E}\left[P_1\right] &\leq \frac{R^2}{\eta} + \frac{18\epsilon T}{\eta K} + 7\eta T K G^2 + 13TG\sqrt{3\epsilon} \\
&\quad + \frac{1}{1-\lambda_2}\left(\frac{12\epsilon T}{\eta K} + 9\eta T K G^2 + 12TG\sqrt{3\epsilon}\right) \\
\mathbb{E}\left[P_2\right] &\leq G(N^{1/2}+1)\left(3\sqrt{2\epsilon} + \frac{3\eta K G + 2\sqrt{3\epsilon}}{1-\lambda_2}\right) \\
\mathbb{E}\left[P_3\right] &\leq 2L_1 R(N^{1/2}+1)\left(3\sqrt{2\epsilon} + \frac{3\eta K G + 2\sqrt{3\epsilon}}{1-\lambda_2}\right)
\end{aligned}$$

Adding P_1, P_2, P_3, we obtain the upper bound for α-regret for agent i as given in the statement of the Theorem.

LOO calls: Based on Lemma 1 for the infeasible projection oracle, we have the number of LOO calls for agent i in block m as:

$$\begin{aligned}
l_m^i \quad = \quad & \frac{27R^2}{\epsilon}\max\Bigg(\frac{1}{4\epsilon^2}(\|\mathbf{y}_{m+1}^i - \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}_m^j\|^2) \\
& (\|\mathbf{y}_{m+1}^i - \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}_m^j\|^2 - \epsilon) + 1, 1\Bigg)
\end{aligned} \tag{3}$$

Through exploitation of the update rule (line 10) and the infeasible projection operation (line 11) in Algorithm 1, we have

$$\left\|\mathbf{y}_{m+1}^i - \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}_m^j\right\|^2 \leq 2\eta^2 K^2 G^2 + 6\epsilon \tag{4}$$

Substituting (4) to (3), we obtain the total LOO calls, $\sum_{m=1}^{T/K} l_m^i$, as in the statement of the Theorem. $\square$

With appropriate selection of parameters block size K, update step η, and infeasible projection error tolerance ϵ, we have final results for the main Algorithm 1 in Theorem 2. Motivated by the trade-off between block size and the time complexity, we introduce a hyper parameter θ, through which users adjust block size accordingly, allowing resilience against practical communication limitations.

Theorem 2. *Choosing $K = T^{1-\theta}$, $\eta = \frac{1}{\sqrt{KT}}$, and $\epsilon = K^2\eta^2$, we get that for each agent i the*

$$\mathbb{E}\left[R_\alpha^i\right] = O(T^{1-\theta/2}) \tag{5}$$

Further, the communication complexity is $O(T^\theta)$ and the number of LOO calls is $O(T^{2\theta})$.

We note that in the special case of $\theta = 1$, there will be no block effect, and we achieve a regret of $O(\sqrt{T})$, with a communication complexity of $O(T)$ and number of LOO calls of $O(T^2)$. Further, in the special case of $\theta = 1/2$, we achieve a regret of $O(T^{3/4})$, with a communication complexity of $O(\sqrt{T})$ and number of LOO calls of $O(T)$.

5 Extension of the Result to Different Feedback Types for Up-Concave Optimization

In Theorem 1, we provided the result for a query function $\mathfrak{g}$. For the different cases of up-concave function summarized in Table 1, this query function is first order query oracle. Further, this query oracle is trivial in the case of monotone γ-weakly up-concave functions over general convex sets, thus making the feedback a semi-bandit feedback. In this section, we will consider different extensions for the proposed results, that provide the regret guarantees for the different feedback settings, following the meta-algorithms proposed by Pedramfar & Aggarwal (2024a).

5.1 Bandit Feedback for Trivial Query Oracle

Since with the trivial query oracle, we have the results for the semi-bandit feedback, we provide an algorithm that uses the semi-bandit to bandit feedback meta-algorithm, `STB`, by Pedramfar & Aggarwal (2024b) to get the results for bandit feedback. The algorithm designed to handle bandit feedback for upper-linearizable functions with trivial query is described in Algorithm 4.

Before detailing steps in Algorithm 4, we introduce several mathematical notations that are being used by the `STB` meta-algorithm. For a set $\mathcal{D} \subseteq \mathbb{R}^d$, we define its *affine hull* $\mathrm{aff}(\mathcal{D})$ 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 $\mathrm{relint}(\mathcal{D}) := \{\mathbf{x} \in \mathcal{D} \mid \exists r > 0, \mathbb{B}_r(\mathbf{x}) \cap \mathrm{aff}(\mathcal{D}) \subseteq \mathcal{D}\}$.

We choose a point $\mathbf{c} \in \mathrm{relint}(\mathcal{K})$ and a real number $r > 0$ such that $\mathrm{aff}(\mathcal{K}) \cap \mathbb{B}_r(\mathbf{c}) \subseteq \mathcal{K}$. 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

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

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 $\mathcal{L}_0 \cap \mathbb{B}_1(\mathbf{0})$. 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.

The important property of this notion is that it allows for construction of a one-point gradient estimator. Specifically, it is well-known that

$$\nabla \hat{f}_\delta(\mathbf{x}) = \frac{k}{\delta}\mathbb{E}_{\mathbf{v}\sim\mathcal{L}_0\cap\mathbb{S}^1}[f(\mathbf{x}+\delta\mathbf{v})].$$

This allows us to convert Algorithm 1 to allow for first order feedback. Specifically, we run Algorithm 1 on functions $\hat{f}_{t,i}$ instead of $f_{t,i}$ and when it requires an unbiased estimate of the gradient of $\hat{f}_{t,i}(\mathbf{x})$, we use $f_{t,i}(\mathbf{x}+\delta\mathbf{v})$ where $\mathbf{v}$ is sampled uniformly from $\mathcal{L}_0\cap\mathbb{S}^1$. More generally, if we have access to $o_{t,i}$, an unbiased estimate of $f_{t,i}(\mathbf{x}+\delta\mathbf{v})$, then $\frac{k}{\delta}o_{t,i}$ is an unbiased estimate of $\nabla\hat{f}_{t,i}(\mathbf{x})$. If the value oracle, from which $o_{t,i}$ is sampled, is bounded by B_0, then we see that this new one-point gradient estimator of $\nabla\hat{f}_{t,i}(\mathbf{x})$ is bounded by $G' := \frac{k}{\delta}B_0$. Therefore, if we set $\epsilon = K^2\eta^2(G')^2$, we may use Theorem 1 to see that the regret is bounded by $O\left(\frac{1}{\eta}+\eta TK(G')^2\right)$, and the number of LOO calls is $O(\frac{T}{\epsilon K})$. However, it should be noted that the functions $\hat{f}_{t,i}$ are defined over $\mathcal{K}_\delta$ and this regret is computed against the best point in $\mathcal{K}_\delta$ which can be $O(\delta)$ away from the best point in $\mathcal{K}$. Hence, we see that

$$\begin{aligned}\mathbb{E}\left[R^i_\alpha\right] &= O\left(\frac{1}{\eta}+\eta TK(G')^2+\delta T\right)\\ &= O\left(\frac{1}{\eta}+\eta TK\delta^{-2}+\delta T\right).\end{aligned}$$

Putting these results together, we obtain the following result.

Theorem 3. *If $\mathcal{G}$ is trivial, and the value oracle is bounded and we set $\epsilon = K^2\eta^2\delta^{-2}$, then Algorithm 4 ensures a regret bound of*

$$\mathbb{E}\left[R^i_\alpha\right] = O\left(\frac{1}{\eta}+\eta TK\delta^{-2}+\delta T\right),$$

with at most $O(\frac{T}{\epsilon K})$ LOO calls and $O(T/K)$ communication complexity. In particular, if we set $K = T^{1-\theta}$, $\delta = T^{-\theta/4}$ and $\eta = \frac{\delta}{\sqrt{KT}}$, we see that $\mathbb{E}\left[R^i_\alpha\right] = O(T^{1-\theta/4})$ with at most $O(T^{2\theta})$ LOO calls and $O(T^\theta)$ communication complexity.

5.2 Semi-Bandit Algorithm for Non-Trivial Query

When the query is non-trivial, Algorithm 1 is a first order full-information feedback. To have the algorithm with semi-bandit feedback, we use the idea used by the meta-algorithm "Stochastic Full-information To Trivial query" `SFTT`, proposed by Pedramfar & Aggarwal (2024a) to convert Algorithm 1 to one that can work with semi-bandit feedback. Note that `SFTT` itself uses a blocking mechanism and we must ensure that its blocking mechanism interacts properly with both the `DOLCO` blocking mechanism and the inter-node communication. The resulting algorithm, which is designed to handle semi-bandit feedback for upper-linearizable functions with non-trivial query, is summarized in Algorithm 5.

Let $L \geq 1$ be an integer. The main idea here is to consider the functions $(\bar{f}_{q,i})_{1\leq q\leq T/L, 1\leq i\leq N}$ where $\bar{f}_{q,i} = \frac{1}{L}\sum_{t=(q-1)L+1}^{qL} f_{t,i}$. We want to run Algorithm 1 against this sequence of functions. To do this, we need to construct unbiased estimates of the gradient of $\bar{f}_{q,i}$. This can be achieved by considering a random permutation $t'_1,\cdots,t'_L$ of $(q-1)L+1,\cdots,qL$ and picking $f_{t'_1,i}$. Since we want an algorithm with semi-bandit feedback, at time-step t'_1 we select the point where the original algorithm, i.e., Algorithm 1, needed to query. In the other $L-1$ time-steps within this block, we pick the action that Algorithm 1 wants to take and ignore the returned value of the query function. Thus, at one time-step per each block of length L, we have no control over the regret, which adds $O(T/L)$ to the total regret. In the remaining time-steps, the behavior is similar to Algorithm 1, with each action repeated $L-1$ times. We note that we are running Algorithm 1 against $\bar{f}_{q,i}$ with a horizon of $T' := T/L$. Hence, using the discussion above and Theorem 1, if we set $\epsilon = K^2\eta^2G^2$, then we see that the regret is bounded by

$$\begin{aligned}\mathbb{E}\left[R_\alpha^i\right] &= (L-1)O\left(\frac{1}{\eta} + \eta T' K G^2\right) + O\left(\frac{T}{L}\right) \\ &= O\left(\frac{L}{\eta} + \eta T K G^2 + \frac{T}{L}\right).\end{aligned}$$

the number of LOO calls is $O(\frac{T'}{\epsilon K})$, and the communication complexity is bounded by $O(T'/K)$.

The key result is summarized in the following theorem.

Theorem 4. *If $\mathcal{G}$ is non-trivial, and we set $\epsilon = K^2\eta^2G^2$, then Algorithm 5 ensures a regret bound of*

$$\mathbb{E}\left[R_\alpha^i\right] = O\left(\frac{L}{\eta} + \eta T K G^2 + \frac{T}{L}\right).$$

with at most $O(\frac{T}{\epsilon KL})$ LOO calls and $O(\frac{T}{KL})$ communication complexity. In particular, if $0 \leq \theta \leq 2/3$ and we set $K = T^{1-3\theta/2}$, $L = T^{\theta/2}$, and $\eta = T^{\theta-1}$, we see that $\mathbb{E}\left[R_\alpha^i\right] = O(T^{1-\theta/2})$ with at most $O(T^{2\theta})$ LOO calls and $O(T^\theta)$ communication complexity.

5.3 Zeroth Order Feedback for Non-Trivial Query

When the query is non-trivial, Algorithm 1 is a first order full-information feedback. To have the algorithm with zeroth order feedback, we use the "First Order To Zeroth Order" meta-algorithm, `FOTZO`, proposed by Pedramfar & Aggarwal (2024b) to obtain the result. The algorithm designed to handle zeroth-order full-information feedback for upper-linearizable functions with non-trivial query is summarized in Algorithm 6. The discussion and analysis of this meta-algorithm is quite similar to that of `STB`.

Theorem 5. *If $\mathcal{G}$ is non-trivial, and the value oracle is bounded and we set $\epsilon = K^2\eta^2\delta^{-2}$, then Algorithm 6 ensures a regret bound of*

$$\mathbb{E}\left[R_\alpha^i\right] = O\left(\frac{1}{\eta} + \eta T K \delta^{-2} + \delta T\right),$$

with at most $O(\frac{T}{\epsilon K})$ LOO calls and $O(T/K)$ communication complexity. In particular, if we set $K = T^{1-\theta}$, $\delta = T^{-\theta/4}$ and $\eta = \frac{\delta}{\sqrt{KT}}$, we see that $\mathbb{E}\left[R_\alpha^i\right] = O(T^{1-\theta/4})$ with at most $O(T^{2\theta})$ LOO calls and $O(T^\theta)$ communication complexity.

5.4 Bandit Algorithm for Non-Trivial Query

In this case, we apply the Stochastic Full-information To Trivial query meta-algorithm, `SFTT`, discussed above, to convert Algorithm 6, which requires zeroth order feedback, to an algorithm that works with bandit feedback. The resulting algorithm, given in Algorithm 7, can handle bandit feedback for upper-linearizable functions with non-trivial query.

Similar to the discussion in Section 5.2, we see that the regret bound is equal to $O(T/L)$ plus $L-1$ times the regret bound of Algorithm 6 when applied over a horizon of T/L. In other words, if $T' = T/L$ and $\epsilon = K^2\eta^2\delta^{-2}$, then we have

$$\begin{aligned}\mathbb{E}\left[R_\alpha^i\right] &= (L-1)O\left(\frac{1}{\eta} + \eta T' K \delta^{-2} + \delta T'\right) + O\left(\frac{T}{L}\right) \\ &= O\left(\frac{L}{\eta} + \eta T K \delta^{-2} + \delta T + \frac{T}{L}\right).\end{aligned}$$

Hence, we obtain the following result.

Theorem 6. *If $\mathcal{G}$ is non-trivial, and the value oracle is bounded and we set $\epsilon = K^2\eta^2\delta^{-2}$, then Algorithm 7 ensures a regret bound of*

$$\mathbb{E}\left[R_\alpha^i\right] = O\left(\frac{L}{\eta} + \eta T K \delta^{-2} + \delta T + \frac{T}{L}\right),$$

with at most $O(\frac{T}{\epsilon KL})$ LOO calls and $O(\frac{T}{KL})$ communication complexity. In particular, if $0 \leq \theta \leq 4/5$ and we set $K = T^{1-5\theta/4}$, $\delta = T^{-\theta/4}$, $L = T^{\theta/4}$, and $\eta = T^{\theta/2-1}$, we see that $\mathbb{E}\left[R^i_\alpha\right] = O(T^{1-\theta/4})$ with at most $O(T^{2\theta})$ LOO calls and $O(T^\theta)$ communication complexity.

6 Conclusions

In this paper, we presented a decentralized, projection-free approach to optimizing upper-linearizable functions, which extends the analysis of classical DR-submodular and concave functions. By incorporating projection-free methods, our framework provides efficient regret bounds of $O(T^{1-\theta/2})$, in first order feedback case, $O(T^{1-\theta/4})$, in zeroth order feedback case, with a communication complexity of $O(T^\theta)$ and number of linear optimization oracle calls of $O(T^{2\theta})$ for suitable choices of $0 \leq \theta \leq 1$, making it scalable for large decentralized networks. This illustrates a tradeoff between the regret and the communication complexity. The versatility of our approach allows it to handle a variety of feedback models, including full information, semi-bandit, and bandit settings. This is the first known result that provides such generalized guarantees for monotone and non-monotone up-concave functions over general convex sets.

References

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.

Venkat Chandrasekaran, Sujay Sanghavi, Pablo A Parrilo, and Alan S Willsky. Sparse and low-rank matrix decompositions. *IFAC Proceedings Volumes*, 42(10):1493–1498, 2009.

Lin Chen, Christopher Harshaw, Hamed Hassani, and Amin Karbasi. Projection-free online optimization with stochastic gradient: From convexity to submodularity. In *International Conference on Machine Learning*, pp. 814–823. PMLR, 2018a.

Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization. In *International Conference on Artificial Intelligence and Statistics*, pp. 1896–1905. PMLR, 2018b.

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

John C Duchi, Alekh Agarwal, and Martin J Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3):592–606, 2011.

Anis Elgabli, Jihong Park, Amrit Singh Bedi, Chaouki Ben Issaid, Mehdi Bennis, and Vaneet Aggarwal. Q-gadmm: Quantized group admm for communication efficient decentralized machine learning. *IEEE Transactions on Communications*, 69(1):164–181, 2020.

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)*, pp. 169–179. SIAM, 2023.

Hongchang Gao, Hanzi Xu, and Slobodan Vucetic. Sample efficient decentralized stochastic frank-wolfe methods for continuous dr-submodular maximization. In *IJCAI*, pp. 3501–3507, 2021.

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 *Advances in Neural Information Processing Systems*, 2017.

Elad Hazan and Haipeng Luo. Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, pp. 1263–1271. PMLR, 2016.

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

Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt, and Patrick Pletscher. Block-coordinate frank-wolfe optimization for structural svms. In *International Conference on Machine Learning*, pp. 53–61. PMLR, 2013.

Dan Li, Kerry D Wong, Yu Hen Hu, and Akbar M Sayeed. Detection, classification, and tracking of targets. *IEEE signal processing magazine*, 19(2):17–29, 2002.

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*, 2023.

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

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Decentralized submodular maximization: Bridging discrete and continuous settings. In *International conference on machine learning*, pp. 3616–3625. PMLR, 2018.

Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

Mohammad Pedramfar and Vaneet Aggarwal. From linear to linearizable optimization: A novel framework with applications to stationary and non-stationary dr-submodular optimization. *Advances in Neural Information Processing Systems*, 2024a.

Mohammad Pedramfar and Vaneet Aggarwal. A generalized approach to online convex optimization. *arXiv preprint arXiv:2402.08621*, 2024b.

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.

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.

S Sundhar Ram, Angelia Nedić, and Venugopal V Veeravalli. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of optimization theory and applications*, 147:516–545, 2010.

Yuanyu Wan, Guanghui Wang, Wei-Wei Tu, and Lijun Zhang. Projection-free distributed online learning with sublinear communication complexity. *Journal of Machine Learning Research*, 23(172):1–53, 2022. URL `http://jmlr.org/papers/v23/20-1239.html`.

Cong Wang, Shengyuan Xu, Deming Yuan, Baoyong Zhang, and Zhengqiang Zhang. Distributed online convex optimization with a bandit primal-dual mirror descent push-sum algorithm. *Neurocomputing*, 497: 204–215, 2022.

Lin Xiao, Stephen Boyd, and Seung-Jean Kim. Distributed average consensus with least-mean-square deviation. *Journal of parallel and distributed computing*, 67(1):33–46, 2007.

Jiahao Xie, Chao Zhang, Zebang Shen, Chao Mi, and Hui Qian. Decentralized gradient tracking for continuous dr-submodular maximization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2897–2906. PMLR, 2019.

Feng Yan, Shreyas Sundaram, SVN Vishwanathan, and Yuan Qi. Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties. *IEEE Transactions on Knowledge and Data Engineering*, 25(11):2483–2493, 2012.

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

Qixin Zhang, Zengde Deng, Zaiyi Chen, Haoyuan Hu, and Yu Yang. Stochastic continuous submodular maximization: Boosting via non-oblivious function. In *International Conference on Machine Learning*, pp. 26116–26134. PMLR, 2022.

Qixin Zhang, Zengde Deng, Xiangru Jian, Zaiyi Chen, Haoyuan Hu, and Yu Yang. Communication-efficient decentralized online continuous dr-submodular maximization. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 3330–3339, 2023a.

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.

Wenpeng Zhang, Peilin Zhao, Wenwu Zhu, Steven CH Hoi, and Tong Zhang. Projection-free distributed online learning in networks. In *International Conference on Machine Learning*, pp. 4054–4062. PMLR, 2017.

Wentao Zhang, Yang Shi, Baoyong Zhang, Kaihong Lu, and Deming Yuan. Quantized distributed online projection-free convex optimization. *IEEE Control Systems Letters*, 7:1837–1842, 2023b.

Junlong Zhu, Qingtao Wu, Mingchuan Zhang, Ruijuan Zheng, and Keqin Li. Projection-free decentralized online learning for submodular maximization over time-varying networks. *Journal of Machine Learning Research*, 22(51):1–42, 2021.

A Up-concave Functions are Linearizable

Given $0 < \gamma \leq 1$, we say a differentiable function $f : \mathcal{K} \to \mathbb{R}$ is *γ-weakly up-concave* if it is γ-weakly concave along positive directions. Specifically if, for all $\mathbf{x} \leq \mathbf{y}$ in $\mathcal{K}$, we have

$$\gamma \left(\langle \nabla f(\mathbf{y}), \mathbf{y} - \mathbf{x}\rangle\right) \leq f(\mathbf{y}) - f(\mathbf{x}) \leq \frac{1}{\gamma} \left(\langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle\right). \tag{6}$$

We say $\tilde{\nabla} f : \mathcal{K} \to \mathbb{R}^d$ is a *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 γ-weakly up-concave if it is continuous and it has a 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. 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.

In this section, we provide for completeness that three cases of common up-concave functions can be formulated as upper-linearizable functions, and we give the exact query algorithm to obtain estimates of $\mathfrak{g}$ functions. We further show that $\mathfrak{g}$ given by these algorithms is Lipschitz-continuous, if f is L-smooth.

A.1 Monotone Up-concave optimization over general convex set

Following Lemma 1 by Pedramfar & Aggarwal (2024a), we note that monotone up-concave optimization over general convex set can be formulated as an online maximization by quantization algorithm with trivial query $\mathcal{G}$.

Lemma 2 (Pedramfar & Aggarwal (2024a))**.** *Let $f : [0, 1]^d \to \mathbb{R}$ be a a non-negative monotone γ-weakly up-concave function with curvature bounded by c. Then, for all $\mathbf{x}, \mathbf{y} \in [0, 1]^d$, we have*

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

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

A.2 Monotone up-concave optimization over convex set containing the origin

Following Lemma 2 by Pedramfar & Aggarwal (2024a), we note that monotone up-concave optimization over convex set containing the origin can be formulated as an online maximization by quantization algorithm with non-trivial query $\mathcal{G} = \texttt{BQMO}$, which is described in Algorithm A.2.

Lemma 3 (Pedramfar & Aggarwal (2024a))**.** *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

$$\forall z \in [0, 1], \quad \mathbb{P}(\mathcal{Z} \leq z) = \int_0^z \frac{\gamma e^{\gamma(u-1)}}{1 - e^{-\gamma}} du, \tag{7}$$

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

$$(1 - e^{-\gamma}) f(\mathbf{y}) - f(\mathbf{x}) \leq \frac{1 - e^{-\gamma}}{\gamma} \langle \nabla F(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle.$$

In this case, $\mathfrak{g} = \nabla F(\mathbf{x})$, $h(\mathbf{x}) = \text{Id}$. Further, if f is smooth, $\mathfrak{g}$ is Lipschitz continuous, as shown in the following Lemma.

Lemma 4 (Theorem 2(iii), Zhang et al. (2022))**.** *If f is L-smooth and satisfies other assumptions of Lemma 3, F is L'-smooth, where $L' = L\frac{\gamma + e^{-\gamma} - 1}{\gamma(1 - e^{-\gamma})}$, i.e., $\nabla F(\mathbf{x})$ is L'-Lipschitz continuous.*

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

1: **Input:** first order query oracle, point $\mathbf{x}$
2: Sample $z \in [0, 1]$ according to Equation (7) in Lemma 3
3: **Return:** the output of the query oracle at $z * \mathbf{x}$

A.3 Non-monotone up-concave optimization over general convex set

Following Lemma 3 by Pedramfar & Aggarwal (2024a), we note that Non-monotone up-concave optimization over general convex set can be formulated as an online maximization by quantization algorithm with non-trivial query $\mathcal{G} = \texttt{BQN}$, as described in Algorithm A.3.

Lemma 5 (Pedramfar & Aggarwal (2024a)). *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

$$\forall z \in [0, 1], \quad \mathbb{P}(\mathcal{Z} \leq z) = \int_0^z \frac{1}{3(1 - \frac{u}{2})^3} du, \tag{8}$$

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

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

Algorithm 3 Boosted Query oracle for non-monotone up-concave functions over general convex sets - `BQN`

1: **Input:** First order query oracle, point $\mathbf{x}$
2: Sample $z \in [0, 1]$ according to Equation 8
3: **Return:** the output of the query oracle at $\frac{z}{2} * (\mathbf{x} - \underline{\mathbf{x}}) + \underline{\mathbf{x}}$

In this case, $\mathfrak{g} = \nabla F(\mathbf{x})$, $h(\mathbf{x}) : \mathbf{x} \mapsto \frac{\mathbf{x}_t + \underline{\mathbf{x}}}{2}$. Further, if f is L-smooth, $\mathfrak{g}$ is Lipschitz continuous, as given in the following Lemma.

Lemma 6 (Theorem 18, Zhang et al. (2024)). *If f is L-smooth, L_1-Lipschitz and f satisfies assumptions in Lemma 5, $\nabla F(\mathbf{x})$ is $\frac{1}{8}L$-smooth and $\frac{3}{8}L_1$-Lipschitz continuous.*

B Proof of Theorem 1

Let $\mathbf{x}^* = \mathrm{argmax}_{\mathbf{u}\in\mathcal{K}} \frac{1}{N}\sum_{t=1}^{T}\sum_{i=1}^{N} f_{t,i}(\mathbf{u})$, $\mathcal{T}_m = \{(m-1)K+1, \cdots, mK\}$. By the definition of α-regret for agent j, we have

$$
\begin{aligned}
\mathbb{E}[\mathcal{R}_\alpha^j] &= \frac{1}{N}\sum_{t=1}^{T}\sum_{i=1}^{N}\mathbb{E}\left[\alpha f_{t,i}(\mathbf{x}^*) - f_{t,i}(h(\mathbf{x}_t^j))\right] \\
&= \frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\mathbb{E}\left[\alpha f_{t,i}(\mathbf{x}^*) - f_{t,i}(h(\mathbf{x}_m^j))\right] \\
&\overset{(a)}{\le} \frac{\beta}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\mathbb{E}\left[\langle \mathbf{x}^* - \mathbf{x}_m^j, \mathfrak{g}_{t,i}(\mathbf{x}_m^j)\rangle\right] \\
&\overset{(b)}{=} \frac{\beta}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\mathbb{E}\left[\langle \mathbf{x}^* - \mathbf{x}_m^j, \mathbb{E}\left[\tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^j)|\mathbf{x}_m^j\right]\rangle\right] \\
&\overset{(c)}{=} \frac{\beta}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\mathbb{E}\left[\mathbb{E}\left[\langle \mathbf{x}^* - \mathbf{x}_m^j, \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^j)\rangle|\mathbf{x}_m^j\right]\right] \\
&\overset{(d)}{=} \frac{\beta}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\mathbb{E}\left[\langle \mathbf{x}^* - \mathbf{x}_m^j, \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^j)\rangle\right] \\
&\overset{(e)}{=} \mathbb{E}\left[\underbrace{\frac{\beta}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\langle \mathbf{x}^* - \mathbf{x}_m^i, \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\rangle}_{:=P_1}\right] + \mathbb{E}\left[\underbrace{\frac{\beta}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\langle \mathbf{x}_m^i - \mathbf{x}_m^j, \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\rangle}_{:=P_2}\right] \\
&\quad + \mathbb{E}\left[\underbrace{\frac{\beta}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\langle \mathbf{x}^* - \mathbf{x}_m^j, \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^j) - \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\rangle}_{:=P_3}\right] \qquad (9)
\end{aligned}
$$

where step (a) is because $f_{t,i}$'s are upper linearizable, step (b)is because $\tilde{\mathfrak{g}}_{t,i}(\cdot)$ is unbiased, step (c) is due to the linearity of conditional expectation, step (d) is due to law of iterated expectation, and step (e) rewrites $(\mathbf{x}^* - \mathbf{x}_m^j)$ as $[(\mathbf{x}^* - \mathbf{x}_m^i) + (\mathbf{x}_m^i - \mathbf{x}_m^j)]$ and $\tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^j)$ as $[(\tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^j) - \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)) + \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)]$.

As illustrated in step (e) in Equation 9, total regret can be divided into three parts, P_1, P_2, P_3, and we provide upper bound for each of these three parts with proof in the following sections, B.2, B.3, B.4, respectively. In Section B.1, we introduce some auxiliary variables and lemmas that are useful to the following proof.

B.1 Auxiliary variables and lemmas

In this section, we introduce some auxiliary variables and lemmas to better present the proof of bound for each of the three parts of the total regret presented in Equation 9.

Let $\mathbf{y}_1^i = \tilde{\mathbf{y}}_m^i = \mathbf{c}$, for any $i \in [N]$, since Algorithm 1 would only generate $\mathbf{y}_m^i$ for $m = 2, \cdots, T/K$. Let $\mathbf{r}_m^i = \tilde{\mathbf{y}}_m^i - \mathbf{y}_m^i$ for any $i \in [N]$ and $m \in [T/K]$. For any $m \in [T/K]$, we denote the averages by:

$$
\bar{\mathbf{x}}_m = \frac{\sum_{i=1}^{N}\mathbf{x}_m^i}{N},\ \bar{\mathbf{y}}_m = \frac{\sum_{i=1}^{N}\mathbf{y}_m^i}{N},\ \hat{\mathbf{y}}_m = \frac{\sum_{i=1}^{N}\tilde{\mathbf{y}}_m^i}{N},\ \text{and } \bar{\mathbf{r}}_m = \frac{\sum_{i=1}^{N}\mathbf{r}_m^i}{N}.
$$

There are two lemmas that would be useful to the proofs, and we provide proof of each of these in the following appendices.

Lemma 7. *For any $i \in [N]$ and $m \in [T/K]$, Algorithm 1 ensures*

$$\|\mathbf{r}_m^i\| \leq 2\sqrt{3\epsilon} + 2\eta K G.$$

Proof. See Appendix B.7 for complete proof for Lemma 7. □

Lemma 8. *For any $i \in [N]$ and $m \in [T/K]$, Algorithm 1 ensures*

$$\sqrt{\sum_{i=1}^{N} \|\hat{\mathbf{y}}_m - \tilde{\mathbf{y}}_m^i\|^2} \leq \frac{\sqrt{N}(3\eta K G + 2\sqrt{3\epsilon})}{1-\lambda_2}, \tag{10}$$

$$\sqrt{\sum_{i=1}^{N} \|\hat{\mathbf{y}}_m - \mathbf{y}_{m+1}^i\|^2} \leq \frac{\sqrt{N}(3\eta K G + 2\sqrt{3\epsilon})}{1-\lambda_2}, \text{ and} \tag{11}$$

$$\sum_{i=1}^{N} \|\mathbf{x}_m^i - \mathbf{x}_m^j\| \leq \left(3\sqrt{2\epsilon} + \frac{(3\eta K G + 2\sqrt{3\epsilon})}{1-\lambda_2}\right)(N^{3/2} + N). \tag{12}$$

Proof. See Appendix B.8 for complete proof of Lemma 8. □

B.2 Bound of P_1

By replacing $(\mathbf{x}^* - \mathbf{x}_m^i)$ with $[(\mathbf{x}^* - \hat{\mathbf{y}}_m) + (\hat{\mathbf{y}}_m - \tilde{\mathbf{y}}_m^i) + (\tilde{\mathbf{y}}_m^i - \mathbf{x}_m^i)]$, P_1 can be decomposed as:

$$\begin{aligned}
\frac{1}{\beta}\P_1 =& \frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\langle \hat{\mathbf{y}}_m - \tilde{\mathbf{y}}_m^i, \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i))\rangle + \frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\langle \tilde{\mathbf{y}}_m^i - \mathbf{x}_m^i, \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\rangle \\
&+ \underbrace{\frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\langle \mathbf{x}^* - \hat{\mathbf{y}}_m, \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\rangle}_{:=P_4} \\
\overset{(a)}{\leq}& \frac{G}{N}\left[\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\|\hat{\mathbf{y}}_m - \tilde{\mathbf{y}}_m^i\| + \sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\|\tilde{\mathbf{y}}_m^i - \mathbf{x}_m^i\|\right] + P_4 \\
\overset{(b)}{\leq}& G\left(\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\sqrt{\frac{1}{N}\sum_{i=1}^{N}\|\hat{\mathbf{y}}_m - \tilde{\mathbf{y}}_m^i\|^2} + T\sqrt{3\epsilon}\right) + \mathcal{R}_4 \\
\overset{(c)}{\leq}& TG\left(\frac{3\eta K G + 2\sqrt{3\epsilon}}{1-\lambda_2} + \sqrt{3\epsilon}\right) + \mathcal{R}_4
\end{aligned} \tag{13}$$

where step (a) follows from Cauchy-Schwartz inequality and the bound of $\tilde{\mathfrak{g}}_{t,i}(\cdot)$ function, step (b) follows from Arithmetic Mean-Quadratic Mean inequality and Lemma 1, and step (c) follows from Equation (10) in Lemma 8.

To attain upper bound on P_4, we notice that,

$$\begin{aligned}
\bar{\mathbf{y}}_{m+1} &= \frac{1}{N}\sum_{i=1}^{N}\mathbf{y}_{m+1}^i = \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{j\in\mathcal{N}_i} a_{ij}\tilde{\mathbf{y}}_m^j + \eta\sum_{t\in\mathcal{T}_m}\tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\right) \\
&\overset{(a)}{=} \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}\tilde{\mathbf{y}}_m^j + \frac{\eta}{N}\sum_{i=1}^{N}\sum_{t\in\mathcal{T}_m}\tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i) \\
&\overset{(b)}{=} \hat{\mathbf{y}}_m + \frac{\eta}{N}\sum_{i=1}^{N}\sum_{t\in\mathcal{T}_m}\tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)
\end{aligned} \tag{14}$$

where step (a) is because $a_{ij} = 0$ for any agent $j \notin \mathcal{N}_i$, and step (b) is because $\mathbf{A1} = \mathbf{1}$.

Substituting $\bar{\mathbf{y}}_{m+1}$ using Equation (13), we have for any $m \in [T/K]$,

$$\begin{aligned}\hat{\mathbf{y}}_{m+1} =& \hat{\mathbf{y}}_{m+1} - \bar{\mathbf{y}}_{m+1} + \bar{\mathbf{y}}_{m+1} \\ \overset{(14)}{=}& \bar{\mathbf{r}}_{m+1} + \hat{\mathbf{y}}_m + \frac{\eta}{N} \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_m} \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\end{aligned} \tag{15}$$

because $\bar{\mathbf{r}}_{m+1} = \hat{\mathbf{y}}_{m+1} - \bar{\mathbf{y}}_{m+1}$.

By replacing $\hat{\mathbf{y}}_{m+1}$ with Equation (15) and then expand the equation, we have

$$\begin{aligned}\|\hat{\mathbf{y}}_{m+1} - \mathbf{x}^*\|^2 \overset{(15)}{=}& \left\| \bar{\mathbf{r}}_{m+1} + \hat{\mathbf{y}}_m + \frac{\eta}{N} \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_m} \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i) - \mathbf{x}^* \right\|^2 \\ =& \|\hat{\mathbf{y}}_m - \mathbf{x}^*\|^2 + 2 \left\langle \hat{\mathbf{y}}_m - \mathbf{x}^*, \frac{\eta}{N} \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_m} \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i) \right\rangle \\ &+ 2\langle \hat{\mathbf{y}}_m - \mathbf{x}^*, \bar{\mathbf{r}}_{m+1} \rangle + \left\| \bar{\mathbf{r}}_{m+1} + \frac{\eta}{N} \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_m} \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i) \right\|^2 .\end{aligned} \tag{16}$$

Following Lemma 1, we deduce that for any $m \in [T/K]$,

$$\begin{aligned}\|\tilde{\mathbf{y}}_{m+1}^i - \mathbf{x}^*\|^2 \leq \|\mathbf{y}_{m+1}^i - \mathbf{x}^*\|^2 &= \|\mathbf{y}_{m+1}^i - \tilde{\mathbf{y}}_{m+1}^i + \tilde{\mathbf{y}}_{m+1}^i - \mathbf{x}^*\|^2 \\ &= \|\mathbf{y}_{m+1}^i - \tilde{\mathbf{y}}_{m+1}^i\|^2 + 2\langle \mathbf{y}_{m+1}^i - \tilde{\mathbf{y}}_{m+1}^i, \tilde{\mathbf{y}}_{m+1}^i - \mathbf{x}^* \rangle + \|\tilde{\mathbf{y}}_{m+1}^i - \mathbf{x}^*\|^2 \\ &= \|\mathbf{r}_{m+1}^i\|^2 - 2\langle \mathbf{r}_{m+1}^i, \tilde{\mathbf{y}}_{m+1}^i - \mathbf{x}^* \rangle + \|\tilde{\mathbf{y}}_{m+1}^i - \mathbf{x}^*\|^2\end{aligned}$$

where the last equality is due to the definition of $\mathbf{r}_{m+1}^i$.

Omitting $\|\tilde{\mathbf{y}}_{m+1}^i - \mathbf{x}^*\|^2$ on both sides and moving $\langle \tilde{\mathbf{y}}_{m+1}^i - \mathbf{x}^*, \mathbf{r}_{m+1}^i \rangle$ to the left, we have

$$\langle \tilde{\mathbf{y}}_{m+1}^i - \mathbf{x}^*, \mathbf{r}_{m+1}^i \rangle \leq \frac{1}{2} \|\mathbf{r}_{m+1}^i\|^2. \tag{17}$$

Thus, we can bound the term $\langle \hat{\mathbf{y}}_m - \mathbf{x}^*, \bar{\mathbf{r}}_{m+1} \rangle$ in Equation (16) as follows

$$\begin{aligned}\langle \hat{\mathbf{y}}_m - \mathbf{x}^*, \bar{\mathbf{r}}_{m+1} \rangle =& \frac{1}{N} \sum_{i=1}^{N} \langle \hat{\mathbf{y}}_m - \mathbf{x}^*, \mathbf{r}_{m+1}^i \rangle \\ \overset{(a)}{\leq}& \frac{1}{N} \sum_{i=1}^{N} \langle \hat{\mathbf{y}}_m - \mathbf{y}_{m+1}^i, \mathbf{r}_{m+1}^i \rangle + \frac{1}{N} \sum_{i=1}^{N} \langle \tilde{\mathbf{y}}_{m+1}^i - \mathbf{x}^*, \mathbf{r}_{m+1}^i \rangle \\ \overset{(b)}{\leq}& \frac{1}{N} \sum_{i=1}^{N} \|\hat{\mathbf{y}}_m - \mathbf{y}_{m+1}^i\| \|\mathbf{r}_{m+1}^i\| + \frac{1}{2N} \sum_{i=1}^{N} \|\mathbf{r}_{m+1}^i\|^2 \\ \overset{(c)}{\leq}& \frac{2\eta KG + 2\sqrt{3\epsilon}}{\sqrt{N}} \sqrt{\sum_{i=1}^{N} \|\hat{\mathbf{y}}_m - \mathbf{y}_{m+1}^i\|^2} + \frac{1}{2N} \sum_{i=1}^{N} \|\mathbf{r}_{m+1}^i\|^2 \\ \overset{(d)}{\leq}& \frac{1}{1 - \lambda_2} \left(6\eta^2 K^2 G^2 + 10\eta KG\sqrt{3\epsilon} + 12\epsilon \right) \\ &+ 2 \left(\eta^2 K^2 G^2 + 2\eta KG\sqrt{3\epsilon} + 3\epsilon \right)\end{aligned} \tag{18}$$

where step (a) replaces $\hat{\mathbf{y}}_m - \mathbf{x}^*$ as $(\hat{\mathbf{y}}_m - \mathbf{y}_{m+1}^i) + (\mathbf{y}_{m+1}^i - \tilde{\mathbf{y}}_{m+1}^i) + (\tilde{\mathbf{y}}_{m+1}^i - \mathbf{x}^*)$ and omits $\langle \mathbf{y}_{m+1}^i - \tilde{\mathbf{y}}_{m+1}^i, \mathbf{r}_{m+1}^i \rangle \leq 0$, step (b) is due to Cauchy-Schwartz inequality and Equation (17), step (c) comes from Lemma 7 and Arithmetic Mean-Quadratic Mean inequality, and step (d) follows from Equation 10 in Lemma 8.

Also, we can bound the last term in Equation (16) as follows

$$
\begin{aligned}
\left\|\bar{\mathbf{r}}_{m+1}+\frac{\eta}{N}\sum_{i=1}^{N}\sum_{t\in\mathcal{T}_m}\tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\right\|^2 &\overset{(a)}{\leq} 2\|\bar{\mathbf{r}}_{m+1}\|^2+2\left\|\frac{\eta}{N}\sum_{i=1}^{N}\sum_{t\in\mathcal{T}_m}\tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\right\|^2 \\
&\overset{(b)}{\leq}\frac{2}{N}\sum_{i=1}^{N}\|\mathbf{r}_{m+1}^i\|^2+\frac{2K\eta^2}{N}\sum_{i=1}^{N}\sum_{t\in\mathcal{T}_m}\|\tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\|^2 \\
&\overset{(c)}{\leq}8\left(\sqrt{3\epsilon}+\eta KG\right)^2+2\eta^2K^2G^2 \\
&=24\epsilon+10\eta^2K^2G^2+16\eta KG\sqrt{3\epsilon}
\end{aligned}
\tag{19}
$$

where both step (a) and step (b) utilize Cauchy-Schwartz inequality and step (c) is due to Lemma 7 and the bound of $\tilde{\mathfrak{g}}_{t,i}(\cdot)$ functions.

Substitute Equation (18) and Equation (19) into Equation (16) and taking expectation on both sides, we have

$$
\begin{aligned}
\mathbb{E}\left[\|\mathbf{x}^*-\hat{\mathbf{y}}_{m+1}\|^2\right]\leq\mathbb{E}\left[\|\mathbf{x}^*-\hat{\mathbf{y}}_{m}\|^2\right]&-\frac{2\eta}{N}\sum_{i=1}^{N}\sum_{t\in\mathcal{T}_m}\mathbb{E}\left[\langle\mathbf{x}^*-\hat{\mathbf{y}}_m,\tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\rangle\right] \\
&+\frac{1}{1-\lambda_2}\left(12\eta^2K^2G^2+20\eta KG\sqrt{3\epsilon}+24\epsilon\right) \\
&+\left(14\eta^2K^2G^2+24\eta KG\sqrt{3\epsilon}+36\epsilon\right).
\end{aligned}
\tag{20}
$$

Moving terms to different sides in Equation (20), we have

$$
\begin{aligned}
\mathbb{E}\left[P_4\right]&=\frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\mathbb{E}\left[\langle\mathbf{x}^*-\hat{\mathbf{y}}_m,\tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\rangle\right] \\
&\overset{(20)}{\leq}\sum_{m=1}^{T/K}\left[\frac{\mathbb{E}\left[\|\hat{\mathbf{y}}_m-\mathbf{x}^*\|^2\right]-\mathbb{E}\left[\|\hat{\mathbf{y}}_{m+1}-\mathbf{x}^*\|^2\right]}{2\eta}\right] \\
&\quad+\frac{T}{K}\left[\frac{18\epsilon}{\eta}+7\eta K^2G^2+12KG\sqrt{3\epsilon}\right] \\
&\quad+\frac{T}{K}\left[\frac{1}{1-\lambda_2}\left(6\eta K^2G^2+10KG\sqrt{3\epsilon}+\frac{12\epsilon}{\eta}\right)\right] \\
&\overset{(a)}{\leq}\frac{2R^2}{\eta}+\frac{18\epsilon T}{\eta K}+7\eta TKG^2+12TG\sqrt{3\epsilon} \\
&\quad+\frac{1}{1-\lambda_2}\left(\frac{12\epsilon T}{\eta K}+6\eta TKG^2+10TG\sqrt{3\epsilon}\right)
\end{aligned}
\tag{21}
$$

where the last inequality is due to $\mathbb{E}\left[\|\hat{\mathbf{y}}_{T/K+1}-\mathbf{x}^*\|^2\right]\geq 0$ and $\|\hat{\mathbf{y}}_1-\mathbf{x}^*\|^2\leq 4R^2$, which is derived by combining $\hat{\mathbf{y}}_1=\mathbf{c}\in\mathcal{K}$, $\mathbf{x}^*\in\mathcal{K}$, and $R=\max_{\mathbf{x}\in\mathcal{K}}\|\mathbf{x}\|$, and Cauchy-Schwarz Inequality.

Therefore, plugging the result for term $\mathbb{E}\left[P_4\right]$ from Equation (21) into Equation (13), we have

$$
\mathbb{E}\left[P_1\right]\leq TG\beta\left(\frac{3\eta KG+2\sqrt{3\epsilon}}{1-\lambda_2}+\sqrt{3\epsilon}\right)
\tag{22}
$$

$$
\begin{aligned}
&+\frac{2\beta R^2}{\eta}+\frac{18\beta\epsilon T}{\eta K}+7\beta\eta TKG^2+12TG\beta\sqrt{3\epsilon} \\
&+\frac{\beta}{1-\lambda_2}\left(\frac{12\epsilon T}{\eta K}+6\eta TKG^2+10TG\sqrt{3\epsilon}\right)
\end{aligned}
\tag{23}
$$

B.3 Bound of P_2

Next, we bound P_2 in (9)

$$\begin{aligned}
\P_2 &\overset{(9)}{=} \frac{\beta}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\langle \mathbf{x}_m^i - \mathbf{x}_m^j, \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\rangle \\
&\overset{(a)}{\leq} \frac{\beta}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\|\mathbf{x}_m^i - \mathbf{x}_m^j\|\|\tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\| \\
&\overset{(b)}{\leq} \frac{G\beta}{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\sum_{i=1}^{N}\|\mathbf{x}_m^i - \mathbf{x}_m^j\| \\
&\overset{(c)}{\leq} G\beta(N^{1/2} - +1)\left(3\sqrt{2\epsilon} + \frac{3\eta KG + 2\sqrt{3\epsilon}}{1-\lambda_2}\right).
\end{aligned} \tag{24}$$

where step (a) is due to Cauchy-Schwartz inequality, step (b) follows from the bound of $\tilde{\mathfrak{g}}_{t,i}(\cdot)$ functions, and step (c) uses the inequality (12) in Lemma 8.

B.4 Bound of P_3

Recall that $\mathfrak{g}_{t,i}(\cdot)$ are L_1-Lipschitz continuous, i.e.,

$$\|\mathfrak{g}_{t,i}(\mathbf{x}_m^j) - \mathfrak{g}_{t,i}(\mathbf{x}_m^i)\| \leq L_1\|\mathbf{x}_m^i - \mathbf{x}_m^j\|.$$

Thus, we have

$$\begin{aligned}
\mathbb{E}\left[P_3\right] &\overset{(9)}{=} \frac{\beta}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\mathbb{E}\langle \mathbf{x}^* - \mathbf{x}_m^j, \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^j) - \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\rangle \\
&\overset{(a)}{=} \frac{\beta}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\mathbb{E}\left[\left\langle \mathbf{x}^* - \mathbf{x}_m^j, \mathbb{E}\left[\tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^j)|\mathbf{x}_m^j\right] - \mathbb{E}\left[\tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)|\mathbf{x}_m^i\right]\right\rangle\right] \\
&\overset{(b)}{=} \frac{\beta}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\mathbb{E}\left[\langle \mathbf{x}^* - \mathbf{x}_m^j, \mathfrak{g}_{t,i}(\mathbf{x}_m^j) - \mathfrak{g}_{t,i}(\mathbf{x}_m^i)\rangle\right] \\
&\overset{(c)}{\leq} \frac{\beta}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\mathbb{E}\left[\|\mathbf{x}^* - \mathbf{x}_m^j\|\|\mathfrak{g}_{t,i}(\mathbf{x}_m^j) - \mathfrak{g}_{t,i}(\mathbf{x}_m^i)\|\right] \\
&\overset{(d)}{\leq} \frac{L_1\beta}{N}\sum_{i=1}^{N}\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\mathbb{E}\left[\|\mathbf{x}^* - \mathbf{x}_m^j\|\|\mathbf{x}_m^i - \mathbf{x}_m^j\|\right] \\
&\overset{(e)}{\leq} \frac{2L_1R\beta}{N}\mathbb{E}\left[\sum_{m=1}^{T/K}\sum_{t\in\mathcal{T}_m}\sum_{i=1}^{N}\|\mathbf{x}_m^i - \mathbf{x}_m^j\|\right] \\
&\overset{(f)}{\leq} 2L_1R\beta(N^{1/2}+1)\left(3\sqrt{2\epsilon} + \frac{3\eta KG + 2\sqrt{3\epsilon}}{1-\lambda_2}\right)
\end{aligned} \tag{25}$$

where step (a) and (b) is due to the law of iterated expectations, step (c) comes from Cauchy-Schwartz inequality, step (d) follows from continuity of $\mathfrak{g}_{t,i}(\cdot)$ functions, step (e) follows from the bound on $\mathcal{K}$ and Cauchy-Schwartz inequality, and step (f) is due to Equation (12) in Lemma 8.

B.5 Final Regret Bound

Plugging Equation (22), Equation (24) and Equation (25) into Equation (9), for any $j \in [N]$, we have

$$
\begin{aligned}
\mathbb{E}\left[\mathcal{R}^j_{T,\alpha}\right] \leq & \frac{2\beta R^2}{\eta} + \frac{18\epsilon\beta T}{\eta K} + 7\eta\beta TKG^2 + 13TG\beta\sqrt{3\epsilon} \\
& + \frac{\beta}{1-\lambda_2}\left(\frac{12\epsilon T}{\eta K} + 9\eta TKG^2 + 12TG\sqrt{3\epsilon}\right) \\
& + G\beta(N^{1/2}+1)\left(3\sqrt{2\epsilon} + \frac{(3\eta KG + 2\sqrt{3\epsilon})}{1-\lambda_2}\right) \\
& + 2L_1RT\beta(N^{1/2}+1)\left(3\sqrt{2\epsilon} + \frac{(3\eta KG + 2\sqrt{3\epsilon})}{1-\lambda_2}\right)
\end{aligned}
$$

B.6 Number of Linear Optimization Oracle Calls

Finally, we analyze the total number of linear optimization oracle calls for each agent i. In Lemma 1, the term $\mathcal{R}_5 = \left\|\mathbf{y}^i_{m+1} - \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}^j_m\right\|^2$ can be bounded as follows

$$
\begin{aligned}
\mathcal{R}_5 = \left\|\mathbf{y}^i_{m+1} - \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}^j_m\right\|^2 & \overset{(a)}{\leq} 2\left\|\mathbf{y}^i_{m+1} - \sum_{j\in\mathcal{N}_i} a_{ij}\tilde{\mathbf{y}}^j_m\right\|^2 + 2\left\|\sum_{j\in\mathcal{N}_i} a_{ij}\tilde{\mathbf{y}}^j_m - \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}^j_m\right\|^2 \\
& \overset{(b)}{\leq} 2\left\|\eta\sum_{t\in\mathcal{T}_m}\tilde{\mathfrak{g}}_{t,i}(\mathbf{x}^i_m)\right\|^2 + 2\sum_{j\in\mathcal{N}_i} a_{ij}\left\|\tilde{\mathbf{y}}^j_m - \mathbf{x}^j_m\right\|^2 \\
& \overset{(c)}{\leq} 2\eta^2K^2G^2 + 6\epsilon
\end{aligned} \tag{26}
$$

where both step (a) follows from Cauchy-Schwartz inequality, step (b) follows from Cauchy-Schwartz inequality and Line (10) in Algorithm 1

From Equation (1) in Lemma 1, in each block m, each agent i in Algorithm 1 at most utilizes

$$
\begin{aligned}
l^i_m &= \frac{27R^2}{\epsilon}\max\left(\frac{\|\mathbf{y}^i_{m+1} - \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}^j_m\|^2(\|\mathbf{y}^i_{m+1} - \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}^j_m\|^2 - \epsilon)}{4\epsilon^2} + 1, 1\right) \\
&= \frac{27R^2}{\epsilon}\max\left(\frac{\mathcal{R}_5(\mathcal{R}_5 - \epsilon)}{4\epsilon^2} + 1, 1\right) \\
&\overset{(26)}{\leq} \frac{27R^2}{\epsilon}\max\left(\frac{(2\eta^2K^2G^2 + 6\epsilon)(2\eta^2K^2G^2 + 6\epsilon - \epsilon)}{4\epsilon^2} + 1, 1\right) \\
&= \frac{27R^2}{\epsilon}\frac{(2\eta^2K^2G^2 + 6\epsilon)(2\eta^2K^2G^2 + 5\epsilon) + 4\epsilon^2}{4\epsilon^2}
\end{aligned} \tag{27}
$$

linear optimization oracle calls, where the last equality is due to the fact that $(2\eta^2K^2G^2+6\epsilon)(2\eta^2K^2G^2+5\epsilon) \geq 0$.

Thus, by summing Equation (27) over $\frac{T}{K}$ blocks, we have that the total number of linear optimization steps required by each agent i of Algorithm 1 is at most

$$
\sum_{m=1}^{T/K} l^i_m \leq \frac{27TR^2}{\epsilon K}\left(8.5 + 5.5\frac{K^2\eta^2G^2}{\epsilon} + \frac{K^4\eta^4G^4}{\epsilon^2}\right)
$$

B.7 Proof of Lemma 7

$$
\begin{aligned}
\|\mathbf{r}_{m+1}^i\| = \|\tilde{\mathbf{y}}_{m+1}^i - \mathbf{y}_{m+1}^i\| &\overset{(a)}{\leq} \left\|\tilde{\mathbf{y}}_{m+1}^i - \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}_m^j\right\| + \left\|\sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}_m^j - \mathbf{y}_{m+1}^i\right\| \\
&\overset{(b)}{\leq} 2\left\|\sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}_m^j - \mathbf{y}_{m+1}^i\right\| \overset{(c)}{=} 2\left\|\sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}_m^j - \sum_{j\in\mathcal{N}_i} a_{ij}\tilde{\mathbf{y}}_m^j - \eta\sum_{t=(m-1)K+1}^{mK}\tilde{\mathfrak{g}}_{t,i}(\mathbf{x}_m^i)\right\| \\
&\overset{(d)}{\leq} 2\sum_{j\in\mathcal{N}_i} a_{ij}\|\mathbf{x}_m^j - \tilde{\mathbf{y}}_m^j\| + 2\eta KG \\
&\overset{(e)}{\leq} 2\sqrt{3\epsilon} + 2\eta KG
\end{aligned}
$$

where step (a) comes from triangle inequality, step (b) follows by Lemma 1, step (c) replaces $\mathbf{y}_{m+1}^i$ with update rule described in Algorithm 1, step (d) comes from triangle inequality, and step (e) follows by Lemma 1.

Moreover, if $m = 1$, we can verify that

$$\|\mathbf{r}_1^i\| = \|\mathbf{0}\| \leq 2\sqrt{3\epsilon} + 2\eta KG.$$

B.8 Proof of Lemma 8

To prove Lemma 8, we introduce additional auxiliary variables as follows:

$$\mathbf{x}'_m = [\mathbf{x}_m^1; \cdots; \mathbf{x}_m^N] \in \mathbb{R}^{Nd},\ \mathbf{y}'_m = [\mathbf{y}_m^1; \cdots; \mathbf{y}_m^N] \in \mathbb{R}^{Nd},\ \tilde{\mathbf{y}}'_m = [\tilde{\mathbf{y}}_m^1; \cdots; \tilde{\mathbf{y}}_m^N] \in \mathbb{R}^{Nd}$$

and

$$\mathbf{r}'_m = [\mathbf{r}_m^1; \cdots; \mathbf{r}_m^N] \in \mathbb{R}^{Nd},\ \mathbf{g}'_m = \sum_{t=(m-1)K+1}^{mK} [\tilde{\mathfrak{g}}_{t,1}(\mathbf{x}_m^1); \cdots; \tilde{\mathfrak{g}}_{t,N}(\mathbf{x}_m^N)] \in \mathbb{R}^{Nd}.$$

According to step 10 in Algorithm 1, for any $m \in \{2, \dots, T/K\}$, we have

$$\mathbf{y}'_{m+1} = (\mathbf{A}\otimes\mathbf{I})\tilde{\mathbf{y}}'_m + \eta\mathbf{g}'_m = \sum_{k=1}^{m-1}(\mathbf{A}\otimes\mathbf{I})^{m-k}\mathbf{r}'_{k+1} + \sum_{k=1}^{m}(\mathbf{A}\otimes\mathbf{I})^{m-k}\eta\mathbf{g}'_k \tag{28}$$

where the notation $\otimes$ indicates the Kronecker product and $\mathbf{I}$ denotes the identity matrix of size $n \times n$.

In the same manner, for any $m \in [T/K]$, we have

$$
\begin{aligned}
\tilde{\mathbf{y}}'_{m+1} = \mathbf{r}'_{m+1} + \mathbf{y}'_{m+1} &= \mathbf{r}'_{m+1} + (\mathbf{A}\otimes\mathbf{I})\tilde{\mathbf{y}}'_m + \eta\mathbf{g}'_m \\
&= \sum_{k=1}^{m}(\mathbf{A}\otimes\mathbf{I})^{m-k}\mathbf{r}'_{k+1} + \sum_{k=1}^{m}(\mathbf{A}\otimes\mathbf{I})^{m-k}\eta\mathbf{g}'_k
\end{aligned} \tag{29}
$$

where the second equality follows the fact that $\mathbf{r}'_1 = \tilde{\mathbf{y}}'_1 - \mathbf{y}'_1 = \mathbf{0}$.

By the definition of $\hat{\mathbf{y}}_{m+1}$, for any $m \in [T/K]$, we have

$$[\hat{\mathbf{y}}_{m+1}; \cdots; \hat{\mathbf{y}}_{m+1}] = \left(\frac{\mathbf{1}\mathbf{1}^T}{N} \otimes \mathbf{I}\right)\tilde{\mathbf{y}}'_{m+1} \tag{30}$$

where the second equality comes from $\mathbf{1}^\top\mathbf{A} = \mathbf{1}^\top$.

B.8.1 Proof of Equation (10)

For any $m \in [T/K]$, we have

$$
\begin{aligned}
\sqrt{\sum_{i=1}^{N} \|\hat{\mathbf{y}}_{m+1} - \tilde{\mathbf{y}}^i_{m+1}\|^2} &\overset{(30)}{=} \left\| \left(\frac{\mathbf{1}\mathbf{1}^T}{N} \otimes \mathbf{I} \right) \tilde{\mathbf{y}}'_{m+1} - \tilde{\mathbf{y}}'_{m+1} \right\| \\
&\overset{(29)}{=} \left\| \sum_{k=1}^{m} \left(\left(\frac{\mathbf{1}\mathbf{1}^T}{N} - \mathbf{A}^{m-k} \right) \otimes \mathbf{I} \right) \mathbf{r}'_{k+1} + \sum_{k=1}^{m} \left(\left(\frac{\mathbf{1}\mathbf{1}^T}{N} - \mathbf{A}^{m-k} \right) \otimes \mathbf{I} \right) \eta \mathbf{g}'_k \right\| \\
&\overset{(a)}{\leq} \left\| \sum_{k=1}^{m} \left(\left(\frac{\mathbf{1}\mathbf{1}^T}{N} - \mathbf{A}^{m-k} \right) \otimes \mathbf{I} \right) \mathbf{r}'_{k+1} \right\| + \left\| \sum_{k=1}^{m} \left(\left(\frac{\mathbf{1}\mathbf{1}^T}{N} - \mathbf{A}^{m-k} \right) \otimes \mathbf{I} \right) \eta \mathbf{g}'_k \right\| \\
&\overset{(b)}{\leq} \sum_{k=1}^{m} \left\| \frac{\mathbf{1}\mathbf{1}^T}{N} - \mathbf{A}^{m-k} \right\| \|\mathbf{r}'_{k+1}\| + \sum_{k=1}^{m} \left\| \frac{\mathbf{1}\mathbf{1}^T}{N} - \mathbf{A}^{m-k} \right\| \|\eta \mathbf{g}'_k\| \\
&\overset{(c)}{\leq} \sqrt{N} \sum_{k=1}^{m} \lambda_2^{m-k} (3\eta K G + 2\sqrt{3\epsilon}) \overset{(d)}{\leq} \frac{\sqrt{N}(3\eta K G + 2\sqrt{3\epsilon})}{1 - \lambda_2},
\end{aligned}
$$

where step (a) is due to triangle inequality, and step (b) is due to Cauchy-Schwartz inequality and triangle inequality, step (c) comes from Lemma 7 and the fact that $\forall k \in [m], \|\frac{\mathbf{1}\mathbf{1}^T}{N} - \mathbf{A}^{m-k}\| \leq \lambda_2^{m-k}$ (see Mokhtari et al. (2018) for details), and step (d) is due to the property of geometric series.

By noticing $\hat{\mathbf{y}}_1 = \tilde{\mathbf{y}}_1 = \mathbf{c}$, we complete the proof of Equation (10) in Lemma 8.

B.8.2 Proof of Equation (11)

Similarly, for any $m \in \{2, \cdots, T/K\}$, we have

$$
\begin{aligned}
\sqrt{\sum_{i=1}^{N} \|\hat{\mathbf{y}}_{m} - \mathbf{y}^i_{m+1}\|^2} &\overset{(30)}{=} \left\| \left(\frac{\mathbf{1}\mathbf{1}^T}{N} \otimes \mathbf{I} \right) \tilde{\mathbf{y}}'_{m} - \mathbf{y}'_{m+1} \right\| \\
&\overset{(28)}{=} \left\| \sum_{k=1}^{m-1} \left(\left(\frac{\mathbf{1}\mathbf{1}^T}{N} - \mathbf{A}^{m-k} \right) \otimes \mathbf{I} \right) \mathbf{r}'_{k+1} + \sum_{k=1}^{m-1} \left(\left(\frac{\mathbf{1}\mathbf{1}^T}{N} - \mathbf{A}^{m-k} \right) \otimes \mathbf{I} \right) \eta \mathbf{g}'_k - \eta \mathbf{g}'_m \right\| \\
&\overset{(a)}{\leq} \left\| \sum_{k=1}^{m-1} \left(\left(\frac{\mathbf{1}\mathbf{1}^T}{N} - \mathbf{A}^{m-k} \right) \otimes \mathbf{I} \right) \mathbf{r}'_{k+1} \right\| + \left\| \sum_{k=1}^{m-1} \left(\left(\frac{\mathbf{1}\mathbf{1}^T}{N} - \mathbf{A}^{m-k} \right) \otimes \mathbf{I} \right) \eta \mathbf{g}'_k \right\| + \|\eta \mathbf{g}'_m\| \\
&\overset{(b)}{\leq} \sum_{k=1}^{m-1} \left\| \frac{\mathbf{1}\mathbf{1}^T}{N} - \mathbf{A}^{m-k} \right\| \|\mathbf{r}'_{k+1}\| + \sum_{k=1}^{m-1} \left\| \frac{\mathbf{1}\mathbf{1}^T}{N} - \mathbf{A}^{m-k} \right\| \|\eta \mathbf{g}'_k\| + \|\eta \mathbf{g}'_m\| \\
&\overset{(c)}{\leq} \sqrt{N} \sum_{k=1}^{m} \lambda_2^{m-k} (3\eta K G + 2\sqrt{3\epsilon}) \overset{(d)}{\leq} \frac{\sqrt{N}(3\eta K G + 2\sqrt{3\epsilon})}{1 - \lambda_2},
\end{aligned}
$$

where step (a) is due to triangle inequality, and step (b) is due to Cauchy-Schwartz inequality and triangle inequality, step (c) comes from Lemma 7 and the fact that $\forall k \in [m], \|\frac{\mathbf{1}\mathbf{1}^T}{N} - \mathbf{A}^{m-k}\| \leq \lambda_2^{m-k}$ (see Mokhtari et al. (2018) for details), and step (d) is due to the property of geometric series.

When $m = 1$, $\hat{\mathbf{y}}_1 = \tilde{\mathbf{y}}^i_1 = \sum_{j \in \mathcal{N}_i} a_{ij} \tilde{\mathbf{y}}^j_1 = \mathbf{c}$. Due to Line (10) in Algorithm 1, we have

$$
\sqrt{\sum_{i=1}^{N} \|\hat{\mathbf{y}}_1 - \mathbf{y}^i_2\|^2} = \sqrt{\sum_{i=1}^{N} \left\| \sum_{j \in \mathcal{N}_i} a_{ij} \tilde{\mathbf{y}}^j_1 - \mathbf{y}^i_2 \right\|^2} = \sqrt{\sum_{i=1}^{N} \left\| \sum_{t \in \mathcal{T}_m} \tilde{\mathfrak{g}}_{t,i}(\mathbf{x}^i_m) \right\|^2} \leq \sqrt{N} \eta K G.
$$

By noticing that $\sqrt{N}\eta K G < \frac{\sqrt{N}(3\eta K G + 2\sqrt{3\epsilon})}{1 - \lambda_2}$, we complete the proof of Equation (11).

B.8.3 Proof of Equation (12)

For any $m \in [T/K]$, we notice that

$$\sum_{i=1}^{N} \|\mathbf{x}_m^i - \mathbf{x}_m^j\| \leq \sum_{i=1}^{N} \|\mathbf{x}_m^i - \bar{\mathbf{x}}_m + \bar{\mathbf{x}}_m - \mathbf{x}_m^j\| \leq \sum_{i=1}^{N} \|\mathbf{x}_m^i - \bar{\mathbf{x}}_m\| + N\|\bar{\mathbf{x}}_m - \mathbf{x}_m^j\|$$
$$\leq \left(\sqrt{N} + N\right) \sqrt{\sum_{i=1}^{N} \|\mathbf{x}_m^i - \bar{\mathbf{x}}_m\|^2}. \tag{31}$$

Moreover, for any $i \in [N]$ and $m \in \{2, \ldots, T/K\}$, we have

$$\begin{aligned}
\|\bar{\mathbf{x}}_m - \mathbf{x}_m^i\|^2 &\leq \|\bar{\mathbf{x}}_m - \hat{\mathbf{y}}_m + \hat{\mathbf{y}}_m - \tilde{\mathbf{y}}_m^i + \tilde{\mathbf{y}}_m^i - \mathbf{x}_m^i\|^2 \\
&\overset{(a)}{\leq} 3\|\bar{\mathbf{x}}_m - \hat{\mathbf{y}}_m\|^2 + 3\|\hat{\mathbf{y}}_m - \tilde{\mathbf{y}}_m^i\|^2 + 3\|\tilde{\mathbf{y}}_m^i - \mathbf{x}_m^i\|^2 \\
&\overset{(b)}{\leq} \frac{3}{N}\sum_{j=1}^{N} \|\mathbf{x}_m^j - \tilde{\mathbf{y}}_m^j\|^2 + 3\|\hat{\mathbf{y}}_m - \tilde{\mathbf{y}}_m^i\|^2 + 3\|\tilde{\mathbf{y}}_m^i - \mathbf{x}_m^i\|^2 \\
&\overset{(c)}{\leq} 18\epsilon + 3\|\hat{\mathbf{y}}_m - \tilde{\mathbf{y}}_m^i\|^2,
\end{aligned}$$

where step (a) utilizes Cauchy-Schwarz inequality, step (b) utilizes Cauchy-Schwarz inequality and the definition of $\hat{\mathbf{y}}_m$ and $\bar{\mathbf{x}}_m$, and step (c) comes from Lemma 1. When $m = 1$, $\|\bar{\mathbf{x}}_1 - \mathbf{x}_1^i\|^2 = \mathbf{0} \leq 18\epsilon + 3\|\hat{\mathbf{y}}_m - \tilde{\mathbf{y}}_m^i\|^2$, which leads us to conclude that for any $m \in [T/K]$,

$$\|\bar{\mathbf{x}}_m - \mathbf{x}_m^i\|^2 \leq 18\epsilon + 3\|\hat{\mathbf{y}}_m - \tilde{\mathbf{y}}_m^i\|^2.$$

Thus, for any $m \in [T/K]$, we have

$$\begin{aligned}
\sqrt{\sum_{i=1}^{N} \|\bar{\mathbf{x}}_m - \mathbf{x}_m^i\|^2} &\leq \sqrt{\sum_{i=1}^{N} (18\epsilon + 3\|\hat{\mathbf{y}}_m - \tilde{\mathbf{y}}_m^i\|^2)} \\
&\overset{(a)}{\leq} 3\sqrt{2N\epsilon} + \sqrt{3\sum_{i=1}^{N} \|\hat{\mathbf{y}}_m - \tilde{\mathbf{y}}_m^i\|^2} \\
&\overset{(b)}{\leq} 3\sqrt{2N\epsilon} + \frac{\sqrt{3N}(3\eta KG + 2\sqrt{3\epsilon})}{1 - \lambda_2}
\end{aligned} \tag{32}$$

where step (a) is due to triangle inequality and step (b) follows by Equation (10) in Lemma 8.

Finally, by substituting Equation (32) into Equation (31), for any $m \in [T/K]$, we have

$$\begin{aligned}
\sum_{i=1}^{N} \|\mathbf{x}_m^i - \mathbf{x}_m^j\| &\leq (\sqrt{N} + N)\sqrt{\sum_{i=1}^{N} \|\mathbf{x}_m^i - \bar{\mathbf{x}}_m\|^2} \\
&\leq \left(3\sqrt{2\epsilon} + \frac{(3\eta KG + 2\sqrt{3\epsilon})}{1 - \lambda_2}\right)(N^{3/2} + N).
\end{aligned}$$

C Bandit Feedback for Trivial Query Oracle

In this section, we describe and discuss the variation of `DOCLO` to handle bandit feedback for functions with trivial query oracle. The detailed implementation is given in the Algorithm 4, followed by proof of Theorem 3. This algorithm requires additional input from the user: smoothing parameter $\delta \leq \alpha$, shrunk set $\hat{\mathcal{K}}_\delta$, linear space $\mathcal{L}_0$. Since queries are trivial, $h(\mathbf{x}) = \mathbf{x}$.

Algorithm 4 Bandit Algorithm for Trivial Query

Input: decision set $\mathcal{K}$, horizon T, block size K, step size η, error tolerance ϵ, number of agents N, weight matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}_+^{N\times N}$, Query Algorithm $\mathcal{G}$, transformation map $h(\cdot)$, smoothing parameter $\delta \le \alpha$, shrunk set $\hat{\mathcal{K}}_\delta$, linear space $\mathcal{L}_0$, value oracle
Let $k = \dim(\mathcal{L}_0)$
Set $\mathbf{x}_1^i = \tilde{\mathbf{y}}_1^i = \mathbf{c} \in \hat{\mathcal{K}}_\delta$ for any $i = 1, \cdots, N$
for $m = 1, \cdots, T/K$ **do**
  for each node $i = 1, \cdots, N$ in parallel **do**
    for $t = (m-1)K + 1, \ldots, mK$ **do**
      Sample $\mathbf{v}_t^i \in \mathbb{S}^1 \cap \mathcal{L}_0$ uniformly
      Play $h(\mathbf{x}_m^i) + \delta \mathbf{v}_t^i$
      Let o_t^i be the output of the value oracle (i.e., the output of value oracle for $f_{t,i}$ at $h(\mathbf{x}_m^i) + \delta \mathbf{v}_t^i$)
      $\mathbf{o}_t^i \leftarrow \frac{k}{\delta} o_t^i \mathbf{v}_t^i$
    end for
    Communicate $\mathbf{x}_m^i$ and $\tilde{\mathbf{y}}_m^i$ with neighbors
    $\mathbf{y}_{m+1}^i \leftarrow \sum_{j\in\mathcal{N}_i} a_{ij}\tilde{\mathbf{y}}_m^j + \eta \sum_{t\in\mathcal{T}_m} \mathbf{o}_t^i$
    $(\mathbf{x}_{m+1}^i, \tilde{\mathbf{y}}_{m+1}^i) \leftarrow O_{IP}\left(\hat{\mathcal{K}}_\delta, \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}_m^j, \mathbf{y}_{m+1}^i, \epsilon\right)$
  end for
end for

Proof. Let $\mathcal{A}'$ denote Algorithm 4 and $\mathcal{A}$ denote Algorithm 1. Let $\hat{f}_{t,j}$ denote a δ-smoothed version of $f_{t,i}$. Let $\mathbf{x}^* \in \operatorname{argmax}_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^T \sum_{j=1}^N f_{t,j}(\mathbf{x})$ and $\hat{\mathbf{x}}^* \in \operatorname{argmax}_{\mathbf{x}\in\hat{\mathcal{K}}_\delta} \sum_{t=1}^T \sum_{j=1}^N \hat{f}_{t,j}(\mathbf{x})$. Following our description in Section 5.1, Algorithm 4 is equivalent to running Algorithm 1 on $\hat{f}_{t,i}$, a δ smoothed version of $f_{t,i}$ over a shrunk set $\hat{\mathcal{K}}_\delta$.

By the definition of regret, we have

$$\begin{aligned}
\mathbb{E}\left[\mathcal{R}_\alpha^{i,\mathcal{A}'}\right] - \mathbb{E}\left[\mathcal{R}_\alpha^{i,\mathcal{A}}\right] &= \frac{1}{N}\mathbb{E}\left[\alpha\sum_{t=1}^T\sum_{j=1}^N f_{t,j}(\mathbf{x}^*) - \sum_{t=1}^T\sum_{j=1}^N f_{t,j}(h(\mathbf{x}_t^i) + \delta\mathbf{v}_t^i)\right] \\
&\quad - \frac{1}{N}\mathbb{E}\left[\alpha\sum_{t=1}^T\sum_{j=1}^N \hat{f}_{t,j}(\hat{\mathbf{x}}^*) - \sum_{t=1}^T\sum_{j=1}^N \hat{f}_{t,j}(h(\mathbf{x}_t^i))\right] \\
&= \frac{1}{N}\mathbb{E}\left[\left(\sum_{t=1}^T\sum_{j=1}^N \hat{f}_{t,j}(h(\mathbf{x}_t^i)) - \sum_{t=1}^T\sum_{j=1}^N f_{t,j}(h(\mathbf{x}_t^i) + \delta\mathbf{v}_t^i)\right)\right. \\
&\quad \left. +\alpha\left(\sum_{t=1}^T\sum_{j=1}^N f_{t,j}(\mathbf{x}^*) - \sum_{t=1}^T\sum_{j=1}^N \hat{f}_{t,j}(\hat{\mathbf{x}}^*)\right)\right].
\end{aligned} \tag{33}$$

Based on Lemma 3 proved by Pedramfar et al. (2023), we have $|\hat{f}_{t,j}(h(\mathbf{x}_t^i)) - f_{t,j}(h(\mathbf{x}_t^i))| \le \delta M_1$, and $\hat{f}_{t,j}$ is M_1-Lipschitz continuous as well. Thus, we have

$$|f_{t,j}(h(\mathbf{x}_t^i) + \delta\mathbf{v}_t^i) - \hat{f}_{t,j}(h(\mathbf{x}_t^i))| \le |f_{t,j}(h(\mathbf{x}_t^i) + \delta\mathbf{v}_t^i) - f_{t,j}(h(\mathbf{x}_t^i))| + |f_{t,j}(h(\mathbf{x}_t^i)) - \hat{f}_{t,j}(h(\mathbf{x}_t^i))| \le 2\delta M_1. \tag{34}$$

Meanwhile, for the second part of Equation 33, we have

$$
\begin{aligned}
\sum_{t=1}^{T}\sum_{j=1}^{N}\hat{f}_{t,j}(\hat{\mathbf{x}}^*) &= \max_{\hat{\mathbf{x}}\in\hat{\mathcal{K}}_\alpha}\sum_{t=1}^{T}\sum_{j=1}^{N}\hat{f}_{t,j}(\hat{\mathbf{x}}) \\
&\overset{(a)}{\geq} -N\delta M_1 T + \max_{\hat{\mathbf{x}}\in\hat{\mathcal{K}}_\alpha}\sum_{t=1}^{T}\sum_{j=1}^{N} f_{t,j}(\hat{\mathbf{x}}) \\
&\overset{(b)}{=} -N\delta M_1 T + \max_{\mathbf{x}\in\mathcal{K}}\sum_{t=1}^{T}\sum_{j=1}^{N} f_{t,j}\left(\left(1-\frac{\delta}{r}\right)\mathbf{x}+\frac{\delta}{r}\mathbf{c}\right) \\
&= -N\delta M_1 T + \max_{\mathbf{x}\in\mathcal{K}}\sum_{t=1}^{T}\sum_{j=1}^{N} f_{t,j}\left(\mathbf{x}+\frac{\delta}{r}(\mathbf{c}-\mathbf{x})\right) \\
&\overset{(c)}{\geq} -N\delta M_1 T + \max_{\mathbf{x}\in\mathcal{K}}\sum_{t=1}^{T}\sum_{j=1}^{N}\left(f_{t,j}(\mathbf{x})-\frac{4\delta M_1 R}{r}\right) \\
&= -\left(1+\frac{4R}{r}\right)N\delta M_1 T + \sum_{t=1}^{T}\sum_{j=1}^{N} f_{t,j}(\mathbf{x}^*)
\end{aligned}
$$

where step (a) follows from Lemma 3 by Pedramfar et al. (2023), step (b) follows from the definition of $\hat{\mathcal{K}}_\delta$, and step (c) is due to the M_1-Lipschitz continuity of $f_{t,i}$'s.

Putting it together with Equation 33 and Equation 34, we have

$$
\mathcal{R}_\alpha^{i,\mathcal{A}'} - \mathcal{R}_\alpha^{i,\mathcal{A}} \leq \frac{1}{N}\left(2N\delta M_1 T + \left(1+\frac{4R}{r}\right)N\delta M_1 T\right) = \left(3+\frac{4R}{r}\right)\delta M_1 T.
$$

Thus we have

$$
\mathcal{R}_\alpha^{i,\mathcal{A}'} \leq \mathcal{R}_\alpha^{i,\mathcal{A}} + \left(3+\frac{4R}{r}\right)\delta M_1 T.
$$

Assuming the value oracle is bounded by B_0, from Line 10 of Algorithm 4 we see that the gradient sample that is being passed to $\mathcal{A}$ is bounded by $G = \frac{k}{\delta}B_0 = O(\delta^{-1})$. Substituting results from Theorem 1, we see that

$$
\mathbb{E}\left[\mathcal{R}_\alpha^{i,\mathcal{A}'}\right] = O\left(\mathcal{R}_\alpha^{i,\mathcal{A}} + \delta T\right) = O\left(\frac{1}{\eta} + \eta T K \delta^{-2} + \delta T\right).
$$

Since we are doing same amount of infeasible projection operation and communication operation in Algorithm 4, LOO calls and communication complexity for Algorithm 4 remains the same as Algorithm 1. $\square$

D Semi-Bandit Feedback for Non-Trivial Query Oracle

To transform `DOCLO` into an algorithm that can handle semi-bandit feedback when we are dealing with functions with non-trivial queries, we pass $\frac{T}{L}$ as time horizon to `DOCLO`. In each of those $\frac{T}{L}$ blocks, we consider functions $\left(\hat{f}_{q,i}\right)_{1\leq q\leq T/L, 1\leq i\leq N}$, where $\hat{f}_{q,i} = \frac{1}{L}\sum_{t=(q-1)L+1}^{qL} f_{t,i}$. We note that in Algorithm 5, for any $\mathbf{x}\in\mathcal{K}$ and $1\leq q\leq T/L$, we have $\mathbb{E}[f_{t'_q}(\mathbf{x})] = \hat{f}_q(\mathbf{x})$, and if f_t are differentiable, $\mathbb{E}[\nabla f_{t'_q}(\mathbf{x})] = \nabla\hat{f}_q(\mathbf{x})$. This way, the transformed algorithm only queries once at the point of action per block, thus semi-bandit.

Note that, in order to sample from $\mathcal{G}(\mathbf{x})$, in the two cases considered, $\mathcal{G}(\mathbf{x})$ "selects" a point and queries ∇f at that point. In particular, if $\mathcal{G}$ is Algorithm 2, then the selected point $\hat{\mathbf{y}}_m^i$ is $z * \mathbf{x}_m^i$ for some z sampled according to Line 2 of Algorithm 2.

Algorithm 5 Semi-Bandit Algorithm for Non-trivial Query

Input: decision set $\mathcal{K}$, horizon T, `DOCLO` block size K, step size η, error tolerance ϵ, number of agents N, weight matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}_+^{N\times N}$, Query Algorithm $\mathcal{G}$, transformation map $h(\cdot)$, `SFTT` block size $L > 1$
Set $\mathbf{x}_1^i = \tilde{\mathbf{y}}_1^i = \mathbf{c} \in \mathcal{K}$ for any $i = 1, \cdots, N$
for $m = 1, \cdots, \frac{T}{LK}$ **do**
  for each node $i = 1, \cdots, N$ in parallel **do**
    for q $= (m-1)K + 1, \cdots, mK$ **do**
      $\hat{\mathbf{x}}_q^i \leftarrow h(\mathbf{x}_m^i)$
      Let $\hat{\mathbf{y}}_q^i$ be the point selected by $\mathcal{G}(\mathbf{x}_m^i)$ (See above for what "selected by" means in this context)
      Sample t'_q uniformly from $\{(q-1)L + 1, \ldots, qL\}$
      for $t = (q-1)L + 1, \ldots, qL$ **do**
        if $t = t'_q$ **then**
          Play the action $\mathbf{z}_t^i = \hat{\mathbf{y}}_q^i$
          Query $\nabla f_{t,i}$ at $\hat{\mathbf{y}}_q^i$ and get $\mathbf{o}_q^i$
        else
          Play the action $\mathbf{z}_t^i = \hat{\mathbf{x}}_q^i$
        end if
      end for
    end for
    Communicate $\mathbf{x}_m^i$ and $\tilde{\mathbf{y}}_m^i$ with neighbours
    $\mathbf{y}_{m+1}^i \leftarrow \sum_{j\in\mathcal{N}_i} a_{ij}\tilde{\mathbf{y}}_m^j + \eta \sum_{q=(m-1)K+1}^{mK} \mathbf{o}_q^i$
    $(\mathbf{x}_{m+1}^i, \tilde{\mathbf{y}}_{m+1}^i) \leftarrow \mathcal{O}_{IP}(\mathcal{K}, \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}_m^j, \mathbf{y}_{m+1}^i, \epsilon)$
  end for
end for

Proof of Theorem 4. Let $\mathcal{A}'$ denote Algorithm 5 and $\mathcal{A}$ denote Algorithm 1. Following our description in Section 5.2, Algorithm 5 is equivalent to running Algorithm 1 on $\hat{f}_{q,i}(\mathbf{x}) = \frac{1}{L}\sum_{t=(q-1)L+1}^{qL} f_{t,i}(\mathbf{x})$, an average of $f_{t,i}$ over block q. In consistence with Algorithm 5 description, we let $\mathbf{z}_t^i$ denote the action taken by agent i at time-step t, whether it be $\hat{\mathbf{x}}_q$, point of action selected by `DOCLO`, or $\hat{\mathbf{y}}_q$, point of query selected by `DOCLO`. Thus, the regret of Algorithm 5 over horizon T is

$$
\begin{aligned}
\mathbb{E}\left[\mathcal{R}_{\alpha,T}^{i,\mathcal{A}'}\right] &= \frac{1}{N}\mathbb{E}\left[\alpha \max_{\mathbf{u}\in\mathcal{K}} \sum_{t=1}^{T}\sum_{j=1}^{N} f_{t,j}(\mathbf{u}) - \sum_{t=1}^{T}\sum_{j=1}^{N} f_{t,j}(\mathbf{z}_t^i)\right] \\
&= \frac{L}{N}\mathbb{E}\left[\alpha \max_{\mathbf{u}\in\mathcal{K}} \frac{1}{L}\sum_{t=1}^{T}\sum_{j=1}^{N} f_{t,j}(\mathbf{u}) - \frac{1}{L}\sum_{t=1}^{T}\sum_{j=1}^{N} f_{t,j}(\mathbf{z}_t^i)\right] \\
&= \frac{L}{N}\mathbb{E}\left[\alpha \max_{\mathbf{u}\in\mathcal{K}} \frac{1}{L}\sum_{j=1}^{N}\sum_{q=1}^{T/L}\sum_{t=(q-1)L+1}^{qL} f_{t,j}(\mathbf{u}) - \frac{1}{L}\sum_{j=1}^{N}\sum_{q=1}^{T/L}\sum_{t=(q-1)L+1}^{qL} f_{t,j}(\mathbf{z}_t^i)\right] \\
&= \frac{1}{N}\mathbb{E}\left[\sum_{j=1}^{N}\sum_{q=1}^{T/L}\sum_{t=(q-1)L+1}^{qL} \left(f_{t,j}(\hat{\mathbf{x}}_q^i) - f_{t,j}(\mathbf{z}_t^i)\right) + L\left(\alpha \max_{\mathbf{u}\in\mathcal{K}} \sum_{j=1}^{N}\sum_{q=1}^{T/L} \hat{f}_{q,j}(\mathbf{u}) - \sum_{j=1}^{N}\sum_{q=1}^{T/L} \hat{f}_{q,j}(\hat{\mathbf{x}}_q^i)\right)\right]
\end{aligned} \tag{35}
$$

Algorithm 5 ensures that in each block with a given q, there is only 1 iteration where $\mathbf{z}_t^i \neq \hat{\mathbf{x}}_q^i$, otherwise $\mathbf{z}_t^i = \hat{\mathbf{x}}_q^i$. Since $f_{t,j}$ are M_1-Lipschitz continuous, we have $|f_{t,j}(\hat{\mathbf{x}}_q^i) - f_{t,j}(\hat{\mathbf{y}}_t^i)| \leq M_1|\hat{\mathbf{x}}_q^i - \hat{\mathbf{y}}_t^i| \leq 2M_1R$ where the second equation comes from the restraint on $\mathcal{K}$. Since $\hat{\mathcal{K}}_\delta \subseteq \mathcal{K}$, we have $\max_{\mathbf{x}\in\hat{\mathcal{K}}_\delta}\|\mathbf{x}\| \leq \max_{\mathbf{x}\in\mathcal{K}}\|\mathbf{x}\| = R$.

Thus, we have

$$\sum_{j=1}^{N}\sum_{q=1}^{T/L}\sum_{t=(q-1)L+1}^{qL}\left|f_{t,j}(\hat{\mathbf{x}}_q^i)-f_{t,j}(\mathbf{x}_t^i)\right| \leq \sum_{j=1}^{N}\sum_{q=1}^{T/L}(0*(L-1)+2M_1R*1)=\frac{2NTM_1R}{L} \tag{36}$$

The second part of Equation 35 can be seen as the regret of running Algorithm 1 against $\left(\hat{f}_{q,i}\right)_{1\leq q\leq T/L, 1\leq i\leq N}$, over horizon T/L instead of T. We denote it with $\mathcal{R}_{\alpha,T/L}^{i,\mathcal{A}}$. Applying Theorem 1, we have

$$\mathbb{E}\left[R_{\alpha,T/L}^{i,\mathcal{A}}\right]=O\left(\frac{1}{\eta}+\frac{\eta TKG^2}{L}\right).$$

Putting together with Equation 35 and Equation 36, we have

$$\mathbb{E}\left[\mathcal{R}_{\alpha,T}^{i,\mathcal{A}'}\right]\leq\frac{2TM_1R}{L}+L\mathbb{E}\left[\mathcal{R}_{\alpha,T/L}^{i,\mathcal{A}}\right]$$

which means

$$\mathbb{E}\left[\mathcal{R}_{\alpha}^{i}\right]=O\left(\frac{L}{\eta}+\eta TKG^2+\frac{T}{L}\right).$$

Based on the implementation described in Algorithm 5, it queries oracle every L iterations, and communicate and make updates with infeasible projection operation every KL iteration. Thus, communication complexity for Algorithm 5 is $O(\frac{T}{KL})$, while LOO calls are $O(\frac{T}{\epsilon KL})$.

□

E Zeroth Order Full-Information Feedback for Non-Trivial Query

In this section, we describe and discuss the variation of `DOCLO` to handle zeroth-order full-information feedback for functions with non-trivial query oracle. The detailed implementation is given in the Algorithm 6 table, followed by proof of Theorem 5. Per request of `FOTZO`, Algorithm 6 requires additional input from the user: smoothing parameter $\delta\leq\alpha$, shrunk set $\hat{\mathcal{K}}_\delta$, linear space $\mathcal{L}_0$. In the case of non-trivial query oracle, $h(\cdot)$ is not necessarily an identity function.

Note that, in order to sample from $\mathcal{G}(\mathbf{x})$, in the two cases considered, $\mathcal{G}(\mathbf{x})$ "selects" a point and queries ∇f at that point. In particular, if $\mathcal{G}$ is Algorithm 2, then the selected point $\hat{\mathbf{y}}_m^i$ is $z*\mathbf{x}_m^i$ for some z sampled according to Line 2 of Algorithm 2.

In the following, we give proof of Theorem 5.

Proof of Theorem 5. Let $\mathcal{A}'$ denote Algorithm 6 and $\mathcal{A}$ denote Algorithm 1. Let $\hat{f}_{t,j}$ denote a δ-smoothed version of $f_{t,i}$. Let $\mathbf{x}^*\in\operatorname{argmax}_{\mathbf{x}\in\mathcal{K}}\sum_{t=1}^{T}\sum_{j=1}^{N}f_{t,j}(\mathbf{x})$ and $\hat{\mathbf{x}}^*\in\operatorname{argmax}_{\mathbf{x}\in\hat{\mathcal{K}}_\delta}\sum_{t=1}^{T}\sum_{j=1}^{N}\hat{f}_{t,j}(\mathbf{x})$. Following our description in Section 5.3, Algorithm 6 is equivalent to running Algorithm 1 on $\hat{f}_{t,i}$, a δ smoothed version of $f_{t,i}$ over a shrunk set $\hat{\mathcal{K}}_\delta$.

Algorithm 6 Zeroth Order Algorithm for Non-Trivial Query

1: **Input:** decision set $\mathcal{K}$, horizon T, block size K, step size η, error tolerance ϵ, number of agents N, weight matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}_+^{N\times N}$, Query Algorithm $\mathcal{G}$, transformation map $h(\cdot)$, smoothing parameter $\delta \le \alpha$, shrunk set $\hat{\mathcal{K}}_\delta$, linear space $\mathcal{L}_0$, value oracle
2: Let $k = \dim(\mathcal{L}_0)$
3: Set $\mathbf{x}_1^i = \tilde{\mathbf{y}}_1^i = \mathbf{c} \in \hat{\mathcal{K}}_\delta$ for any $i = 1, \cdots, N$
4: **for** $m = 1, \cdots, T/K$ **do**
5:   **for** each node $i = 1, \cdots, N$ in parallel **do**
6:     **for** $t = (m-1)K + 1, \ldots, mK$ **do**
7:       Play $h(\mathbf{x}_m^i)$
8:       Sample $\mathbf{v}_t^i \in \mathbb{S}^1 \cap \mathcal{L}_0$ uniformly
9:       Let $\mathbf{y}_t^i$ be the point selected by $\mathcal{G}(\mathbf{x}_m^i)$ (See above for what “selected by” means in this context)
10:       Query value oracle for $f_{t,i}$ at $\mathbf{y}_t^i + \delta \mathbf{v}_t^i$ and get o_t^i
11:       $\mathbf{o}_t^i \leftarrow \frac{k}{\delta} o_t^i \mathbf{v}_t^i$
12:     **end for**
13:     Communicate $\mathbf{x}_m^i$ and $\tilde{\mathbf{y}}_m^i$ with neighbours
14:     $\mathbf{y}_{m+1}^i \leftarrow \sum_{j\in\mathcal{N}_i} a_{ij}\tilde{\mathbf{y}}_m^j + \eta \sum_{t\in\mathcal{T}_m} \mathbf{o}_t^i$
15:     $(\mathbf{x}_{m+1}^i, \tilde{\mathbf{y}}_{m+1}^i) \leftarrow \mathcal{O}_{IP}(\hat{\mathcal{K}}_\delta, \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{x}_m^j, \mathbf{y}_{m+1}^i, \epsilon)$
16:   **end for**
17: **end for**

By the definition of regret, we have

$$\begin{aligned}
\mathbb{E}\left[\mathcal{R}_\alpha^{i,\mathcal{A}'}\right] - \mathbb{E}\left[\mathcal{R}_\alpha^{i,\mathcal{A}}\right] &= \frac{1}{N}\mathbb{E}\left[\alpha \sum_{t=1}^T \sum_{j=1}^N f_{t,j}(\mathbf{x}^*) - \sum_{t=1}^T \sum_{j=1}^N f_{t,j}(h(\mathbf{x}_t^i))\right] \\
&\quad - \frac{1}{N}\mathbb{E}\left[\alpha \sum_{t=1}^T \sum_{j=1}^N \hat{f}_{t,j}(\hat{\mathbf{x}}^*) - \sum_{t=1}^T \sum_{j=1}^N \hat{f}_{t,j}(h(\mathbf{x}_t^i))\right] \\
&= \frac{1}{N}\mathbb{E}\left[\left(\sum_{t=1}^T \sum_{j=1}^N \hat{f}_{t,j}(h(\mathbf{x}_t^i)) - \sum_{t=1}^T \sum_{j=1}^N f_{t,j}(h(\mathbf{x}_t^i))\right)\right. \\
&\quad \left. + \alpha\left(\sum_{t=1}^T \sum_{j=1}^N f_{t,j}(\mathbf{x}^*) - \sum_{t=1}^T \sum_{j=1}^N \hat{f}_{t,j}(\hat{\mathbf{x}}^*)\right)\right]. \qquad (37)
\end{aligned}$$

Based on Lemma 3 proved by Pedramfar et al. (2023), we have $|\hat{f}_{t,j}(h(\mathbf{x}_t^i)) - f_{t,j}(h(\mathbf{x}_t^i))| \le \delta M_1 < 2\delta M_1$.

It can be shown that the second part of Equation 37 follows the same upper bound as Equation 34.

Thus, putting it together, we have for Algorithm 4, the regret

$$\mathcal{R}_\alpha^{i,\mathcal{A}'} \le \mathcal{R}_\alpha^{i,\mathcal{A}} + \frac{1}{N}\left(2N\delta M_1 T + \left(1 + \frac{4R}{r}\right) N\delta M_1 T\right) = \mathcal{R}_\alpha^{i,\mathcal{A}} + \left(3 + \frac{4R}{r}\right)\delta M_1 T$$

Assuming the value oracle is bounded by B_0, from Line 10 of Algorithm 6 we see that the gradient sample that is being passed to $\mathcal{A}$ is bounded by $G = \frac{k}{\delta} B_0 = O(\delta^{-1})$. Substituting results from Theorem 1, we see that

$$\mathbb{E}\left[\mathcal{R}_\alpha^i\right] = O\left(\frac{1}{\eta} + \eta T K \delta^{-2} + \delta T\right).$$

Since we are doing same amount of infeasible projection operation and communication operation in Algorithm 6, LOO calls and communication complexity for Algorithm 6 remains the same as Algorithm 1. ☐

F Bandit Feedback for Nontrivial Query

In this section, we extend DOCLO over functions with nontrivial query oracle to handle bandit feedback, i.e., trivial query and zero-order feedback. We achieve this by applying FOTZO to handle zero-order full-information feedback, then applying SFTT to transform the algorithm into trivial queries.

Note that, in order to sample from $\mathcal{G}(\mathbf{x})$, in the two cases considered, $\mathcal{G}(\mathbf{x})$ "selects" a point and queries ∇f at that point. In particular, if $\mathcal{G}$ is Algorithm 2, then the selected point $\hat{\mathbf{y}}_m^i$ is $z * \mathbf{x}_m^i$ for some z sampled according to Line 2 of Algorithm 2.

Algorithm 7 Bandit Algorithm for Non-trivial Query

Input: decision set $\mathcal{K}$, horizon T, DOCLO block size K, step size η, error tolerance ϵ, number of agents N, weight matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}_+^{N\times N}$, Query Algorithm $\mathcal{G}$, transformation map $h(\cdot)$, SFTT block size $L > 1$, smoothing parameter $\delta \le \alpha$, shrunk set $\hat{\mathcal{K}}_\delta$, linear space $\mathcal{L}_0$, value oracle
Let $k = \dim(\mathcal{L}_0)$
Set $\mathbf{x}_1^i = \tilde{\mathbf{y}}_1^i = \mathbf{c} \in \hat{\mathcal{K}}_\delta$ for any $i = 1, \cdots, N$
for $m = 1, \cdots, T/LK$ **do**
  for each node $i = 1, \cdots, N$ in parallel **do**
    for q $= 1, 2, \ldots, K$ **do**
      Let $\hat{\mathbf{x}}_q^i = h(\mathbf{x}_m^i)$
      Sample $\mathbf{v}_q^i \in \mathbb{S}^1 \cap \mathcal{L}_0$ uniformly
      Let $\mathbf{y}_q^i$ be the point selected by $\mathcal{G}(\mathbf{x}_m^i)$ (See above for what "selected by" means in this context)
      Let $\hat{\mathbf{y}}_q^i = \mathbf{y}_q^i + \delta \mathbf{v}_q^i$
      Sample $t_q^{'}$ uniformly from $\{(m-1)KL + (q-1)L + 1, \ldots, (m-1)KL + qL\}$
      for $t = (m-1)KL + (q-1)L + 1, \ldots, (m-1)KL + qL$ **do**
        if $t = t_q^{'}$ **then**
          Play the action $\mathbf{z}_t = \hat{\mathbf{y}}_q^i$
          Query the value oracle for $f_{t,i}$ at $\hat{\mathbf{y}}_q$ and get o_q^i
          $\mathbf{o}_q^i \leftarrow \frac{k}{\delta} o_q^i \mathbf{v}_q^i$
        else
          Play the action $\mathbf{z}_t = \hat{\mathbf{x}}_q^i$
        end if
      end for
    end for
    Communicate $\mathbf{x}_m^i$ and $\tilde{\mathbf{y}}_m^i$ with neighbours
    $\mathbf{y}_{m+1}^i \leftarrow \sum_{j \in \mathcal{N}_i} a_{ij} \tilde{\mathbf{y}}_m^j + \eta \sum_{q=1}^{K} \mathbf{o}_q^i$
    $(\mathbf{x}_{m+1}^i, \tilde{\mathbf{y}}_{m+1}^i) \leftarrow \mathcal{O}_{IP}(\hat{\mathcal{K}}_\delta, \sum_{j \in \mathcal{N}_i} a_{ij} \mathbf{x}_m^j, \mathbf{y}_{m+1}^i, \epsilon)$
  end for
end for

In the following, we provide proof of Theorem 6.

Proof of Theorem 6. Let $\mathcal{A}'$ denote Algorithm 7 and $\mathcal{A}$ denote Algorithm 6. Following our description in Section 5.4, Algorithm 7 is equivalent to running Algorithm 6 on $\hat{f}_{q,i}(\mathbf{x}) = \frac{1}{L}\sum_{t=(q-1)L+1}^{qL} \hat{f}_{t,i}(\mathbf{x})$, an average of $\hat{f}_{t,i}$ over block q, where $\hat{f}_{t,i}$ is a δ-smoothed version of $f_{t,i}$. In consistence with Algorithm 7 description, we let $\mathbf{x}_t^i$ denote the action taken by agent i at iteration t, whether it be $\hat{\mathbf{x}}_q$, point of action selected by DOCLO, or $\hat{\mathbf{y}}_q$, point of query selected by DOCLO. Thus, the regret of Algorithm 7 over horizon T:

$$\begin{aligned}
\mathbb{E}\left[\mathcal{R}_\alpha^{i,\mathcal{A}'}\right] &= \frac{1}{N}\mathbb{E}\left[\alpha \max_{\mathbf{u}\in\mathcal{K}} \sum_{t=1}^{T}\sum_{j=1}^{N} \hat{f}_{t,j}(\mathbf{u}) - \sum_{t=1}^{T}\sum_{j=1}^{N} \hat{f}_{t,j}(\mathbf{z}_t^i)\right] \\
&= \frac{L}{N}\mathbb{E}\left[\alpha \max_{\mathbf{u}\in\mathcal{K}} \frac{1}{L}\sum_{t=1}^{T}\sum_{j=1}^{N} \hat{f}_{t,j}(\mathbf{u}) - \frac{1}{L}\sum_{t=1}^{T}\sum_{j=1}^{N} \hat{f}_{t,j}(\mathbf{z}_t^i)\right] \\
&= \frac{L}{N}\mathbb{E}\left[\alpha \max_{\mathbf{u}\in\mathcal{K}} \frac{1}{L}\sum_{j=1}^{N}\sum_{q=1}^{T/L}\sum_{t=(q-1)L+1}^{qL} \hat{f}_{t,j}(\mathbf{u}) - \frac{1}{L}\sum_{j=1}^{N}\sum_{q=1}^{T/L}\sum_{t=(q-1)L+1}^{qL} \hat{f}_{t,j}(\mathbf{z}_t^i)\right] \\
&= \frac{1}{N}\mathbb{E}\Bigg[\sum_{j=1}^{N}\sum_{q=1}^{T/L}\sum_{t=(q-1)L+1}^{qL} \left(\hat{f}_{t,j}(\hat{\mathbf{x}}_q^i) - \hat{f}_{t,j}(\mathbf{z}_t^i)\right) \\
&\qquad + L\left(\alpha \max_{\mathbf{u}\in\mathcal{K}} \sum_{j=1}^{N}\sum_{q=1}^{T/L} \hat{f}_{q,j}(\mathbf{u}) - \sum_{j=1}^{N}\sum_{q=1}^{T/L} \hat{f}_{q,j}(\hat{\mathbf{x}}_q^i)\right)\Bigg]
\end{aligned} \tag{38}$$

Algorithm 5 ensures that in each block q, there is only 1 iteration where $\mathbf{z}_t^i = \hat{\mathbf{y}}_t^i \neq \hat{\mathbf{x}}_t^i$, otherwise $\mathbf{x}_t^i = \hat{\mathbf{x}}_t^i$. Based on Lemma 3 proposed by Pedramfar et al. (2023), $\hat{f}_{t,j}$ is M_1-Lipscitz continuous if $f_{t,i}$ is M_1-Lipscitz continuous, i.e., $|f_{t,j}(\hat{\mathbf{x}}_q^i) - f_{t,j}(\hat{\mathbf{y}}_t^i)| \leq M_1|\hat{\mathbf{x}}_q^i - \hat{\mathbf{y}}_t^i| \leq 2M_1R$ where the second equation comes from the restraint on $\mathcal{K}$. Since $\hat{\mathcal{K}}_\delta \subseteq \mathcal{K}$, we have $\max_{\mathbf{x}\in\hat{\mathcal{K}}_\delta} \|\mathbf{x}\| \leq \max_{\mathbf{x}\in\mathcal{K}} \|\mathbf{x}\| = R$. Thus, we have

$$\sum_{j=1}^{N}\sum_{q=1}^{T/L}\sum_{t=(q-1)L+1}^{qL} \left(\hat{f}_{t,j}(\hat{\mathbf{x}}_q^i) - \hat{f}_{t,j}(\mathbf{x}_t^i)\right) = \sum_{j=1}^{N}\sum_{q=1}^{T/L} (0 * (L-1) + 2M_1R * 1) = \frac{2NTM_1R}{L} \tag{39}$$

The second part of Equation 38 can be seen as the regret of running Algorithm 6 against $\left(\hat{f}_{q,i}\right)_{1\leq q\leq T/L, 1\leq i\leq N}$, over horizon T/L instead of T. We denote it with $\mathcal{R}_{\alpha,T/L}^{i,\mathcal{A}}$. Applying Theorem 5, we have

$$\mathbb{E}\left[R_{\alpha,T/L}^{i,\mathcal{A}}\right] = O\left(\frac{1}{\eta} + \frac{\eta TK\delta^{-2}}{L} + \frac{\delta T}{L}\right).$$

Putting it together with Equation 35 and Equation 36, we have

$$\mathbb{E}\left[\mathcal{R}_\alpha^{i,\mathcal{A}'}\right] \leq \frac{2TM_1R}{L} + L\mathbb{E}\left[\mathcal{R}_{\alpha,T/L}^{i,\mathcal{A}}\right],$$

which means that

$$\mathbb{E}\left[\mathcal{R}_\alpha^i\right] = O\left(\frac{L}{\eta} + \eta TK\delta^{-2} + \delta T + \frac{T}{L}\right).$$

Based on the implementation described in Algorithm 5, it queries oracle every L iterations, and communicate and make updates with infeasible projection operation every KL iteration. Thus, communication complexity for Algorithm 5 is $O(\frac{T}{KL})$, while LOO calls are $O(\frac{T}{\epsilon KL})$. □