Proceedings of Machine Learning Research vol 291:1–27, 2025 38th Annual Conference on Learning Theory

Black-Box Reductions for Decentralized Online Convex Optimization in Changing Environments

Yuanyu Wan WANYY@ZJU.EDU.CN
School of Software Technology, Zhejiang University, Ningbo, China
State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, China

Editors: Nika Haghtalab and Ankur Moitra

Abstract

We investigate decentralized online convex optimization (D-OCO) in changing environments, and choose adaptive regret and dynamic regret as the performance metric. Specifically, these two metrics compare each local learner against the optimal comparator over every interval, and any sequence of comparators over all rounds, respectively. It is well-known that in the centralized setting, plenty of algorithms with (nearly) optimal bounds on these two metrics have been proposed. However, none of them has been extended into D-OCO, possibly due to the difficulty in handling their commonly used two-level structure. To fill the gap, in this paper, we propose black-box reductions from minimizing these two metrics of D-OCO to minimizing them in the centralized setting. Let n, ρ, and T denote the number of local learners, the spectral gap of the communication matrix, and the time horizon, respectively. For adaptive regret, our reduction can achieve an $\tilde{O}(n\rho^{-1/4}\sqrt{\tau}\log T)$ bound over any interval of length τ in general, and an improved one of $\tilde{O}(n\rho^{-1/2}(\log T)^3)$ when facing strongly convex functions. These two bounds match existing lower bounds up to polylogarithmic factors. For dynamic regret, our reduction can achieve an $\tilde{O}(n\rho^{-1/4}\sqrt{T(1+P_T)\log T})$ bound in general, where P_T is the path-length of comparators. We also provide the first lower bound for dynamic regret of D-OCO to demonstrate that our dynamic regret is nearly optimal.

Keywords: Online Convex Optimization, Decentralized Optimization, Adaptive Regret, Dynamic Regret, Black-Box Reductions

1. Introduction

Decentralized online convex optimization (D-OCO) has become a popular learning framework for modeling various real-time distributed applications (Li et al., 2023). Specifically, it can be formulated as a repeated game between an adversary and n local learners in a network defined by an undirected graph $\mathcal{G} = ([n], E)$ with the edge set $E \subseteq [n] \times [n]$. In each round $t \in [T]$, each local learner $i \in [n]$ must first select a decision $\mathbf{x}_i(t)$ from a convex set $\mathcal{K} \subseteq \mathbb{R}^d$, and then receives a convex loss function $f_{t,i}(\mathbf{x}) : \mathcal{K} \to \mathbb{R}$ selected by the adversary. Let $f_t(\mathbf{x}) = \sum_{j=1}^{n} f_{t,j}(\mathbf{x})$ denote the global function in each round t. To minimize the global loss, in each round, these local learners are allowed to communicate with their immediate neighbors once via a single gossip step (Xiao and Boyd, 2004; Boyd et al., 2006), i.e., computing a weighted average of some local variables based on a weight matrix $P \in \mathbb{R}^{n\times n}$ given beforehand. The standard performance measure of D-OCO is regret of each local learner i:

$$\mathrm{R}(T, i) = \sum_{t=1}^{T} f_t(\mathbf{x}_i(t)) - \min_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}) \tag{1}$$

which is the cumulative global loss of the learner minus that of a fixed optimal decision.

Over the past decade, the regret of D-OCO has been extensively studied, yielding various algorithms and theories (Yan et al., 2013; Hosseini et al., 2013; Zhang et al., 2017b; Wan et al., 2020, 2022c; Wang et al., 2023; Wan et al., 2024a,b). Most notably, Wan et al. (2024a,b) establish nearly optimal regret bounds of $\tilde{O}(n\rho^{-1/4}\sqrt{T})$ and $\tilde{O}(n\rho^{-1/2}\log T)$ for convex and strongly convex functions respectively,[1] where $\rho < 1$ is the spectral gap of P. However, since only a fixed comparator is introduced in (1), the traditional regret actually cannot reflect the hardness of problems with changing environments, where the best decision could drift over time. To address this limitation, we investigate D-OCO with two more suitable metrics including adaptive regret (Hazan and Seshadhri, 2007, 2009; Daniely et al., 2015) and dynamic regret (Zinkevich, 2003). Specifically, the adaptive regret evaluates the performance of local learners on every interval of length τ, and is defined as

$$\mathrm{SAR}(T,\tau,i) = \max_{[s,s+\tau-1]\subseteq[T]} \left(\sum_{t=s}^{s+\tau-1} f_t(\mathbf{x}_i(t)) - \min_{\mathbf{x}\in\mathcal{K}} \sum_{t=s}^{s+\tau-1} f_t(\mathbf{x}) \right) \tag{2}$$

where comparators in different intervals could be changing. The dynamic regret is defined as

$$\mathrm{DR}(\mathbf{u}(1),\ldots,\mathbf{u}(T),i) = \sum_{t=1}^{T} f_t(\mathbf{x}_i(t)) - \sum_{t=1}^{T} f_t(\mathbf{u}(t))$$

which compares local learners against any sequence of comparators $\mathbf{u}(1),\ldots,\mathbf{u}(T)\in\mathcal{K}$.

Note that in the centralized setting with $n=1$, D-OCO reduces to the standard online convex optimization (OCO) problem (Hazan, 2016), and there exist plenty of algorithms with (nearly) optimal bounds on adaptive regret (Hazan and Seshadhri, 2007, 2009; Daniely et al., 2015; Jun et al., 2017; Zhang et al., 2018b, 2019a, 2021; Yang et al., 2024) and dynamic regret (Zhang et al., 2018a; Zhao et al., 2020, 2024; Baby and Wang, 2022). However, to the best of our knowledge, none of these algorithms has been extended into the general D-OCO problem. It is possibly due to the difficulty in handling their commonly used two-level framework—running multiple expert-algorithms in parallel and combining their decisions with a meta-algorithm. In contrast, there has been a surge of interest in developing decentralized variants of the classical online gradient descent (OGD) algorithm (Zinkevich, 2003) for minimizing the dynamic regret (Shahrampour and Jadbabaie, 2018; Dixit et al., 2019; Zhang et al., 2019b; Lu et al., 2020; Li et al., 2022; Eshraghi and Liang, 2022). Unfortunately, these studies focus on the restricted dynamic regret with $\mathbf{u}(t)=\mathbf{x}^*(t)=\mathrm{argmin}_{\mathbf{x}\in\mathcal{K}} f_t(\mathbf{x})$, which is too pessimistic (Zhang et al., 2018a). Moreover, even for the restricted dynamic regret, the best existing algorithm can only achieve a suboptimal bound of $O(n^{5/4}\rho^{-1/2}\sqrt{T}(1+P_T^*))$ in general, where $P_T^* = \sum_{t=2}^{T}\|\mathbf{x}^*(t)-\mathbf{x}^*(t-1)\|_2$ denotes the path-length of the restricted comparators (Shahrampour and Jadbabaie, 2018).

In this paper, we propose black-box reductions that can minimize adaptive regret and dynamic regret of D-OCO by simply utilizing existing (nearly) optimal algorithms in OCO. Specifically, our black-box reductions are established in a two-stage way. First, we show novel reductions from minimizing these two metrics of D-OCO to minimizing them of OCO with delayed feedback. Second, we adopt an existing black-box reduction from the delayed OCO to the standard OCO (Joulani et al., 2013). It is worth noting that Joulani et al. (2013) only focus on the traditional regret in (1). For the first time, we prove that this black-box technique can also convert non-delayed algorithms for adaptive regret and dynamic regret into the delayed setting, which may be of independent interest.

1. The $\tilde{O}(\cdot)$ notation hides constant factors as well as polylogarithmic factors in n, instead of T.

Table 1: Summary of upper and lower bounds on adaptive regret and dynamic regret of D-OCO. Abbreviations: convex → cvx, strongly convex → scvx.

Metric	$f_{t,i}(\mathbf{x})$	Upper Bound	Lower Bound
Adaptive Regret	cvx	$O(n\rho^{-1/4}\sqrt{\tau(\log T)\log(Tn)})$ Theorem 5	$\Omega(n\rho^{-1/4}\sqrt{\tau})$ Wan et al. (2024b)
	scvx	$O(n\rho^{-1/2}(\log T)^2\log(Tn))$ Theorem 6	$\Omega(n\rho^{-1/2}\log\tau)$ Wan et al. (2024b)
Dynamic Regret	cvx	$O(n\rho^{-1/4}\sqrt{T(1+P_T)\log(Tn)})$ Theorem 7	$\Omega(n\rho^{-1/4}\sqrt{T(1+P_T)})$ Theorem 8

Then, combining with the algorithms in Jun et al. (2017) and Hazan and Seshadhri (2007, 2009), we can achieve $\tilde{O}(n\rho^{-1/4}\sqrt{\tau}\log T)$ and $\tilde{O}(n\rho^{-1/2}(\log T)^3)$ adaptive regret bounds for convex and strongly convex functions, respectively. These two bounds nearly match existing lower bounds over any fixed interval of length τ, i.e., $\Omega(n\rho^{-1/4}\sqrt{\tau})$ for convex functions and $\Omega(n\rho^{-1/2}\log\tau)$ for strongly convex functions (Wan et al., 2024b). Moreover, combining with the algorithm in Zhang et al. (2018a), we can achieve an $\tilde{O}(n\rho^{-1/4}\sqrt{T(1+P_T)\log T})$ dynamic regret bound for convex functions, where $P_T=\sum_{t=2}^{T}\|\mathbf{u}(t)-\mathbf{u}(t-1)\|_2$ denotes the path-length of any sequence of comparators. Finally, we also demonstrate that this bound is nearly optimal by deriving the first lower bound for dynamic regret of D-OCO, i.e, $\Omega(n\rho^{-1/4}\sqrt{T(1+P_T)})$. A detailed comparison between these upper and lower bounds is presented in Table 1.

2. Related Work

In this section, we briefly review related work on OCO and D-OCO, especially those about adaptive regret and dynamic regret.

2.1. Online Convex Optimization (OCO)

Over the past two decades, there are extensive studies on regret, adaptive regret, and dynamic regret of OCO. Specifically, OGD is sufficient to achieve the optimal $O(\sqrt{T})$ and $O(\log T)$ regret bounds for convex and strongly convex functions, respectively (Zinkevich, 2003; Hazan et al., 2007; Abernethy et al., 2008; Hazan and Kale, 2014). Moreover, Hazan et al. (2007) consider the case with exponentially concave (abbr. exp-concave) functions that are more general than strongly convex functions, and propose online Newton step (ONS) to achieve an $O(d\log T)$ regret bound.

The adaptive regret is first introduced by Hazan and Seshadhri (2007, 2009) as in the following weaker form (omitting the subscript of the learner 1 for brevity):

$$\mathrm{AR}(T)=\max_{[s,q]\subseteq[T]}\left(\sum_{t=s}^{q}f_t(\mathbf{x}(t))-\min_{\mathbf{x}\in\mathcal{K}}\sum_{t=s}^{q}f_t(\mathbf{x})\right)$$

which is the maximum regret over any interval. They first propose a meta-algorithm called follow-the-leading-history (FLH), which activates a new instance of a low-regret algorithm per round as an expert and combines these experts via an expert-tracking technique. By using OGD and ONS as the expert-algorithm, the weakly adaptive regret of FLH can be bounded by $O(\sqrt{T \log T})$ and $O(d \log T)$ for convex and exp-concave functions, respectively. However, FLH needs to maintain $O(t)$ experts per round, which could be time-consuming. Therefore, Hazan and Seshadhri (2007, 2009) have also developed an efficient variant of FLH (EFLH) with only $O(\log t)$ experts per round, while weakening the previous two bounds to $O(\sqrt{T}(\log T)^{3/2})$ and $O(d(\log T)^2)$. Nonetheless, one limitation of their results is that the bound for convex functions is meaningless for intervals of length $O(\sqrt{T})$. To address this issue, Daniely et al. (2015) propose the strengthened adaptive regret defined in (2), which emphasizes the dependency on the interval length, and develop a new meta-algorithm to achieve an $O(\sqrt{\tau} \log T)$ adaptive regret bound for convex functions. Later, Jun et al. (2017) develop an improved meta-algorithm called coin betting for changing environment (CBCE) to obtain an $O(\sqrt{\tau \log T})$ adaptive regret bound for convex functions. Both meta-algorithms of Daniely et al. (2015) and Jun et al. (2017) also only maintain $O(\log t)$ experts per round. Moreover, Zhang et al. (2018b) show that by using OGD as the expert-algorithm, EFLH can exploit the strong convexity of functions to achieve an $O((\log T)^2)$ adaptive regret bound. The smoothness of functions has also been exploited to improve the adaptive regret when the comparator has a small loss (Zhang et al., 2019a). Recently, novel meta-algorithms (Zhang et al., 2021; Yang et al., 2024) have been proposed to universally achieve the best adaptive regret for different types of functions.

The study of dynamic regret dates back to the pioneering work of Zinkevich (2003), who shows that OGD can achieve an $O(\sqrt{T}(1 + P_T))$ dynamic regret bound. This result remains unchanged until the work of Zhang et al. (2018a), who propose a novel algorithm called Ader to achieve an $O(\sqrt{T(1 + P_T)})$ dynamic regret bound. The key idea of Ader is to run $O(\log T)$ instances of OGD with different learning rates and track the best one via a meta-algorithm. They also establish an $\Omega(\sqrt{T(1 + P_T)})$ lower bound for dynamic regret of OCO, which implies that Ader is optimal for convex functions. Later, Zhao et al. (2020, 2024) propose new algorithms for smooth functions to replace the $\sqrt{T}$ part in the dynamic regret of Zhang et al. (2018a) with some data-dependent terms. When considering an improper learning setting, Baby and Wang (2021) demonstrate that FLH can be utilized to achieve $O(d^{7/2}T^{1/3}C_T^{2/3} \operatorname{poly}(\log T))$ and $O(d^2 T^{1/3}C_T^{2/3} \operatorname{poly}(\log T))$ dynamic regret bounds for exp-concave and strongly convex functions respectively, where $C_T = \sum_{t=2}^{T} \|\mathbf{u}(t) - \mathbf{u}(t-1)\|_1$. The improper assumption is later removed by Baby and Wang (2022), who further improve the previous two bounds to $O(d^3 T^{1/3}C_T^{2/3} \operatorname{poly}(\log T))$ and $O(d^{1/3}T^{1/3}C_T^{2/3} \operatorname{poly}(\log T))$. Additionally, the restricted dynamic regret with $\mathbf{u}(t) = \mathbf{x}^*(t)$ has also attracted much attention (Jadbabaie et al., 2015; Besbes et al., 2015; Mokhtari et al., 2016; Yang et al., 2016; Zhang et al., 2017a, 2018b; Baby and Wang, 2019; Zhao and Zhang, 2021; Wan et al., 2021, 2023). However, as discussed in Zhang et al. (2018a), the restricted one is too pessimistic and fails to recover the traditional regret even for stationary environments. Moreover, different from the general dynamic regret that is commonly minimized by a two-level framework, the restricted one can even be simply minimized by OGD for smooth and strongly convex functions (Mokhtari et al., 2016).

We also notice that several algorithms (Zhang et al., 2020; Cutkosky, 2020; Wang et al., 2024) have been proposed to minimize adaptive regret and dynamic regret simultaneously.

2.2. Decentralized Online Convex Optimization (D-OCO)

Compared with OCO, the main challenge of D-OCO is that each local learner only has direct access to the local function, instead of the global function. To address this challenge, the pioneering work of Yan et al. (2013) proposes a decentralized variant of OGD (D-OGD). Their key idea is to first compute a weighted average decision among local learners via a standard gossip step (Xiao and Boyd, 2004) and then perform a gradient descent step according to the local function. Note that the use of the standard gossip step not only leads to average consensus, i.e., all local decisions can asymptotically converge to their average, but also ensures that the average decision is virtually updated by applying OGD with the average gradient. Based on these properties, they show that D-OGD can achieve $O(n^{5/4}\rho^{-1/2}\sqrt{T})$ and $O(n^{3/2}\rho^{-1}\log T)$ for convex and strongly convex functions, respectively.

After that, decentralized variants of many other OCO algorithms have been proposed to minimize the regret under different scenarios (Hosseini et al., 2013; Zhang et al., 2017b; Wan et al., 2020, 2022c; Wang et al., 2023; Wan et al., 2024a,b). However, most of them are still based on the standard gossip step, and cannot improve the regret of D-OGD. The only exception is the work of Wan et al. (2024a,b) that exploits an accelerated gossip strategy (Liu and Morse, 2011; Ye et al., 2023) and a blocking update mechanism to develop novel D-OCO algorithms with $\tilde{O}(n\rho^{-1/4}\sqrt{T})$ and $\tilde{O}(n\rho^{-1/2}\log T)$ regret bounds for convex and strongly convex functions, respectively. They also provide $\Omega(n\rho^{-1/4}\sqrt{T})$ and $\Omega(n\rho^{-1/2}\log T)$ lower regret bounds for convex and strongly convex functions respectively, which indicate the near optimality of their algorithms.

Besides the regret, previous studies have also considered the restricted dynamic regret of D-OCO. Specifically, Shahrampour and Jadbabaie (2018) propose a non-Euclidean variant of D-OGD, and establish an $O(n^{5/4}\rho^{-1/2}\sqrt{T}(1+P_T^*))$ restricted dynamic regret bound for convex functions. Compared with existing results on the regret of D-OCO and the dynamic regret of OCO, this bound is suboptimal in terms of n, ρ, and P_T^*. Although many subsequent studies have been devoted to improving this bound, they typically require additional assumptions, including the smoothness and/or strong convexity of functions (Dixit et al., 2019; Zhang et al., 2019b; Lu et al., 2020; Li et al., 2022; Eshraghi and Liang, 2022). Moreover, these improved algorithms are still developed by extending OGD, which owes to the previously mentioned power of OGD in minimizing the restricted dynamic regret for smooth and strongly convex functions.

3. Main Results

In this section, we first present a novel reduction from D-OCO to the delayed OCO, and then revisit an existing reduction for the delayed OCO. Finally, we provide theoretical guarantees on adaptive regret and dynamic regret of D-OCO. All proofs can be found in the appendix.

3.1. Assumptions

Following previous studies on D-OCO (Wan et al., 2024a,b), we also require several common assumptions.

Assumption 1 *At each round $t \in [T]$, the loss function $f_{t,i}(\mathbf{x})$ of each learner $i \in [n]$ is G-Lipschitz over $\mathcal{K}$, i.e., it holds that $|f_{t,i}(\mathbf{x}) - f_{t,i}(\mathbf{y})| \leq G\|\mathbf{x}-\mathbf{y}\|_2,\ \forall \mathbf{x},\mathbf{y} \in \mathcal{K}$.*

Assumption 2 *The set $\mathcal{K}$ contains the origin, i.e., $\mathbf{0} \in \mathcal{K}$, and its diameter is bounded by D, i.e., it holds that $\|\mathbf{x}-\mathbf{y}\|_2 \leq D,\ \forall \mathbf{x},\mathbf{y} \in \mathcal{K}$.*

Assumption 3 *At each round $t \in [T]$, the loss function $f_{t,i}(\mathbf{x})$ of each learner $i \in [n]$ is α-strongly convex over $\mathcal{K}$, i.e., it holds that $f_{t,i}(\mathbf{y}) \geq f_{t,i}(\mathbf{x}) + \langle \nabla f_{t,i}(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{\alpha}{2}\|\mathbf{y}-\mathbf{x}\|_2^2,\ \forall \mathbf{x},\mathbf{y} \in \mathcal{K}$.*

Assumption 4 *The communication matrix $P \in \mathbb{R}^{n\times n}$ is supported on the graph $\mathcal{G} = ([n], E)$, symmetric, and doubly stochastic, which satisfies:*
- $P_{ij} > 0$ *only if* $(i, j) \in E$ *or* $i = j$*;*
- $\sum_{j=1}^{n} P_{ij} = \sum_{j\in N_i} P_{ij} = 1, \forall i \in [n]$*;*
- $\sum_{i=1}^{n} P_{ij} = \sum_{i\in N_j} P_{ij} = 1, \forall j \in [n]$*;*

where $N_i = \{j \in [n] | (i, j) \in E\} \cup \{i\}$ *for any* $i \in [n]$*. Moreover,* P *is positive semidefinite, and its second largest singular value denoted by* $\sigma_2(P)$ *is strictly smaller than* 1*.*

Note that Assumption 3 with $\alpha = 0$ reduces to the general convex case, and we have $\rho = 1 - \sigma_2(P)$ from Assumption 4.

3.2. Our Reduction from D-OCO to Delayed OCO

Before introducing our techniques, we notice that the analysis of most previous D-OCO algorithms (Yan et al., 2013; Hosseini et al., 2013) relies on the following relaxation:

$$f_t(\mathbf{x}_i(t)) - f_t(\mathbf{x}) = O\left(n\left\langle \bar{\mathbf{g}}(t), \bar{\mathbf{x}}(t) - \mathbf{x}\right\rangle + n \max_{i\in[n]} \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\|_2\right) \tag{3}$$

for any $i \in [n]$ and $\mathbf{x}, \bar{\mathbf{x}}(t) \in \mathcal{K}$, where $\bar{\mathbf{g}}(t) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_{t,i}(\mathbf{x}_i(t))$ denotes the average gradient at round t. For these algorithms, there exactly exists a suitable $\bar{\mathbf{x}}(t)$ that is close to each local decision $\mathbf{x}_i(t)$ and has a small value of $\langle \bar{\mathbf{g}}(t), \bar{\mathbf{x}}(t)\rangle$, e.g., $\bar{\mathbf{x}}(t) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i(t)$ in D-OGD (Yan et al., 2013). However, it is difficult to apply this methodology for extending those two-level algorithms with adaptive regret and dynamic regret bounds (Jun et al., 2017; Hazan and Seshadhri, 2007, 2009; Zhang et al., 2018a) into D-OCO. To be precise, a natural idea is to maintain multiple experts and a meta-algorithm for each local learner i, and set the local decision $\mathbf{x}_i(t)$ as a weighted average of decisions $\mathbf{x}_i^1(t), \ldots, \mathbf{x}_i^M(t)$ generated by these experts, where M denotes the number of experts. Moreover, the expert-algorithm should be an existing D-OCO algorithm, e.g., D-OGD, such that the decision of every expert j for each local learner i, i.e., $\mathbf{x}_i^j(t)$, can converge to some $\bar{\mathbf{x}}^j(t)$. By further assuming that the meta-algorithm of each local learner maintains the same weight $w_1, \ldots, w_M$, we have $\mathbf{x}_i(t) = \sum_{j=1}^{M} w_j \mathbf{x}_i^j(t)$ and it is intuitive to set $\bar{\mathbf{x}}(t) = \sum_{j=1}^{M} w_j \bar{\mathbf{x}}^j(t)$. Unfortunately, even under this ideal assumption, the distance between $\mathbf{x}_i(t)$ and $\bar{\mathbf{x}}(t)$ could be unsatisfactory because the heterogeneity of experts, e.g., different learning rates, leads to different convergence rates of their decisions $\mathbf{x}_i^1(t), \ldots, \mathbf{x}_i^M(t)$.

To address the above challenge, we first still consider an ideal case, in which arbitrary communication steps are allowed per round. Our main idea is to treat the two-level structure that needs to be maintained by each local learner i as a black box, and only focus on the black-box output at each round t, which is now denoted as a preparatory decision $\hat{\mathbf{x}}_i(t)$. By defining $\bar{\mathbf{x}}(t) = \frac{1}{n}\sum_{i=1}^{n} \hat{\mathbf{x}}_i(t)$, one can apply multiple gossip steps (Xiao and Boyd, 2004; Liu and Morse, 2011) over these preparatory decisions to generate a decision $\mathbf{x}_i(t)$ that approximates $\bar{\mathbf{x}}(t)$ well for each local learner i. Thus, the remaining challenge is to control the term $\langle \bar{\mathbf{g}}(t), \bar{\mathbf{x}}(t)\rangle$ in (3) by generating appropriate preparatory decisions. Note that in the analysis of those previous algorithms (Yan et al., 2013; Hosseini et al., 2013), this term is generally bounded due to a global property of $\bar{\mathbf{x}}(t)$, i.e., it can be simply computed based on historical average gradients. However, it is highly non-trivial to establish this global property for our $\bar{\mathbf{x}}(t)$ due to both the two-level structure and the black-box way. To this end, we further apply multiple gossip steps over those local gradients to generate a local approximation

$\mathbf{g}_i(t)$ for the average gradient $\bar{\mathbf{g}}(t)$, and notice that

$$n\langle\bar{\mathbf{g}}(t), \bar{\mathbf{x}}(t)\rangle = \sum_{i=1}^{n} \langle\bar{\mathbf{g}}(t), \hat{\mathbf{x}}_i(t)\rangle = \sum_{i=1}^{n} \langle\mathbf{g}_i(t), \hat{\mathbf{x}}_i(t)\rangle + O\left(n \max_{i\in[n]} \|\bar{\mathbf{g}}(t) - \mathbf{g}_i(t)\|_2\right). \quad (4)$$

From (4), we only need to update $\hat{\mathbf{x}}_i(t)$ according to the local approximation $\mathbf{g}_i(t)$, which can be implemented by invoking an existing two-level algorithm locally.

Now, we proceed to extend the above idea into the practical case, in which only one communication step is allowed per round. Inspired by Wan et al. (2024a,b), we adopt the blocking updating mechanism and the accelerated gossip strategy (Liu and Morse, 2011). The former technique allows us to utilize multiple communication steps over each block, and the latter one is critical for controlling the effect of n and ρ on our theoretical results. Specifically, according to the blocking updating mechanism, we divide total T rounds into $Z = T/(2L)$ blocks, where L will be specified later, and assume that Z is an integer without loss of generality. For each local learner i at all rounds in any block z, we will maintain a fixed preparatory decision as well as a fixed decision, and denote them as $\hat{\mathbf{x}}_i(z)$ and $\mathbf{x}_i(z)$, respectively. To make $\mathbf{x}_i(z)$ close to the average preparatory decision, i.e., $\bar{\mathbf{x}}(z) = \frac{1}{n}\sum_{i=1}^{n} \hat{\mathbf{x}}_i(z)$, we simply set $\mathbf{x}_i(1) = \hat{\mathbf{x}}_i(1) = \mathbf{0}$, and generate $\mathbf{x}_i(z+1)$ via the accelerated gossip strategy, i.e, iteratively performing the following update for $k_x = 0, 1, \ldots, L-1$:

$$\mathbf{x}_i^{k_x+1}(z+1) = (1+\theta)\sum_{j\in N_i} P_{ij}\mathbf{x}_j^{k_x}(z+1) - \theta\mathbf{x}_i^{k_x-1}(z+1)$$

during the first L rounds of each block z, and setting $\mathbf{x}_i(z+1) = \mathbf{x}_i^{L}(z+1)$ at the end of this block, where $\mathbf{x}_i^{0}(z+1) = \mathbf{x}_i^{-1}(z+1) = \hat{\mathbf{x}}_i(z+1)$ and θ will be specified later.

Then, we notice that the preparatory decision $\hat{\mathbf{x}}_i(z+1)$ is required at the beginning of block z, but has not been determined. From our previous discussions about (4), to generate an appropriate $\hat{\mathbf{x}}_i(z+1)$, we originally require a local approximation $\mathbf{g}_i(z)$ for the cumulative average gradient over each block z, i.e., $\bar{\mathbf{g}}(z) = \frac{1}{n}\sum_{i=1}^{n}\sum_{t\in\mathcal{T}_z} \nabla f_{t,i}(\mathbf{x}_i(z))$, where $\mathcal{T}_z = \{2(z-1)L+1, \ldots, 2zL\}$. However, such an approximation is not convenient for exploiting the strong convexity of functions in the black-box way. To tackle this issue, inspired by the strongly convex algorithm of Wan et al. (2024a,b), we will maintain $\mathbf{g}_i(z)$ to approximate $\bar{\mathbf{g}}(z) = \frac{1}{n}\sum_{i=1}^{n}\sum_{t\in\mathcal{T}_z}(\nabla f_{t,i}(\mathbf{x}_i(z)) - \alpha\mathbf{x}_i(z))$, where α is given by Assumption 3. Correspondingly, we further pursue a small value of $\ell_{z,i}(\hat{\mathbf{x}}_i(z))$, where $\ell_{z,i}(\mathbf{x}) = \langle\mathbf{g}_i(z), \mathbf{x}\rangle + \alpha L\|\mathbf{x}\|_2^2$ generalizes the linear loss in (4) by adding a quadratic term. Moreover, during each block $z \geq 2$, we actually can only approximate $\bar{\mathbf{g}}(z-1)$ instead of $\bar{\mathbf{g}}(z)$, because the cumulative gradient of this block has not been collected at its beginning. Therefore, we set $\mathbf{g}_i^{0}(z-1) = \mathbf{g}_i^{-1}(z-1) = \sum_{t\in\mathcal{T}_{z-1}}(\nabla f_{t,i}(\mathbf{x}_i(z-1)) - \alpha\mathbf{x}_i(z-1))$, and iteratively perform the following update for $k_g = 0, 1, \ldots, L-1$:

$$\mathbf{g}_i^{k_g+1}(z-1) = (1+\theta)\sum_{j\in N_i} P_{ij}\mathbf{g}_j^{k_g}(z-1) - \theta\mathbf{g}_i^{k_g-1}(z-1)$$

during the remaining L rounds of each block $z \geq 2$. In this way, we can finally set $\mathbf{g}_i(z) = \mathbf{g}_i^{L}(z)$ for any $z \in [Z]$, but it is worth noting that only $\mathbf{g}_i(1), \ldots, \mathbf{g}_i(z-2)$ are available at the beginning of block z. This implies that $\hat{\mathbf{x}}_i(z+1)$ should be computed with a suitable delayed OCO algorithm $\mathcal{D}$, instead of directly using an existing two-level algorithm as in the ideal case.

Based on these discussions, we are ready to reduce D-OCO to the delayed OCO problem, and the detailed procedure is summarized in Algorithm 1. Here, to simplify the initialization of each

Algorithm 1 Black-Box Reduction from D-OCO to Delayed OCO

1: **Input:** L, θ, α, and a delayed OCO algorithm $\mathcal{D}$ with the initial decision $\mathbf{0}$
2: Create an instance $\mathcal{D}_i$ of $\mathcal{D}$, get $\hat{\mathbf{x}}_i(1)=\mathbf{0}$ from $\mathcal{D}_i$, and set $\mathbf{x}_i(1)=\hat{\mathbf{x}}_i(1),\forall i\in[n]$
3: **for** $z=1,\dots,T/(2L)$ **do**
4: Define $\ell_{z,i}(\mathbf{x})=\langle \mathbf{g}_i(z),\mathbf{x}\rangle+\alpha L\|\mathbf{x}\|_2^2$ and send it to $\mathcal{D}_i$ once $\mathbf{g}_i(z)$ is available, $\forall i\in[n]$
5: Get $\hat{\mathbf{x}}_i(z+1)$ from $\mathcal{D}_i$, and set $\mathbf{x}_i^0(z+1)=\mathbf{x}_i^{-1}(z+1)=\hat{\mathbf{x}}_i(z+1),\forall i\in[n]$
6: **for** $t=2(z-1)L+1,\dots,2zL$ **do**
7: **for** each local learner $i\in[n]$ **do**
8: Play $\mathbf{x}_i(z)$, query $\nabla f_{t,i}(\mathbf{x}_i(z))$, and set $k_x=t-2(z-1)L-1$, $k_g=t-(2z-1)L-1$
9: **if** $t\leq(2z-1)L$ **then**
10: Update $\mathbf{x}_i^{k_x+1}(z+1)=(1+\theta)\sum_{j\in N_i}P_{ij}\mathbf{x}_j^{k_x}(z+1)-\theta\mathbf{x}_i^{k_x-1}(z+1)$
11: **else if** $2\leq z$ **then**
12: Update $\mathbf{g}_i^{k_g+1}(z-1)=(1+\theta)\sum_{j\in N_i}P_{ij}\mathbf{g}_j^{k_g}(z-1)-\theta\mathbf{g}_i^{k_g-1}(z-1)$
13: **end if**
14: **end for**
15: **end for**
16: Set $\mathbf{x}_i(z+1)=\mathbf{x}_i^L(z+1),\forall i\in[n]$, and if $2\leq z$, set $\mathbf{g}_i(z-1)=\mathbf{g}_i^L(z-1),\forall i\in[n]$
17: Set $\mathbf{g}_i^0(z)=\mathbf{g}_i^{-1}(z)=\sum_{t=2(z-1)L+1}^{2zL}(\nabla f_{t,i}(\mathbf{x}_i(z))-\alpha\mathbf{x}_i(z)),\forall i\in[n]$
18: **end for**

local decision, we further assume that the delayed OCO algorithm $\mathcal{D}$ is initialized with $\mathbf{0}$. Moreover, according to lines 2, 4, 5, and 16 of Algorithm 1, each instance $\mathcal{D}_i$ of $\mathcal{D}$ needs to generate the preparatory decision $\hat{\mathbf{x}}_i(z)$ with only $\ell_{1,i}(\mathbf{x}),\dots,\ell_{z-3,i}(\mathbf{x})$. In the following, we establish theoretical guarantees on adaptive regret and dynamic regret of Algorithm 1, respectively.

Theorem 1 *Let $o_t=\lceil t/(2L)\rceil$ denote the block index of any round t, and*

$$\theta=\frac{1}{1+\sqrt{1-\sigma_2^2(P)}},\ L=\left\lceil\frac{\sqrt{2}\ln(T\sqrt{14n})}{(\sqrt{2}-1)\sqrt{1-\sigma_2(P)}}\right\rceil. \tag{5}$$

Under Assumptions 1, 2, 3, and 4, Algorithm 1 with θ, L defined in (5) ensures

$$\sum_{t=s}^{q}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_i(o_t))-f_{t,j}(\mathbf{x}))\leq\sum_{j=1}^{n}\sum_{z=o_s+1}^{o_q-1}(\ell_{z,j}(\hat{\mathbf{x}}_j(z))-\ell_{z,j}(\mathbf{x}))+4nGD(2+L)+2n\alpha D^2$$

for any $[s,q]\subseteq[T]$, $\mathbf{x}\in\mathcal{K}$, and $i\in[n]$.

Theorem 2 *Let $m_z=2(z-1)L+1$ and recall that $Z=T/(2L)$. Under Assumptions 1, 2, 3, and 4, Algorithm 1 with θ, L defined in (5) ensures*

$$\begin{aligned}&\sum_{z=1}^{Z}\sum_{t=2(z-1)L+1}^{2zL}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_i(z))-f_{t,j}(\mathbf{u}(t)))\\ \leq&\sum_{j=1}^{n}\sum_{z=1}^{Z}(\ell_{z,j}(\hat{\mathbf{x}}_j(z))-\ell_{z,j}(\mathbf{u}(m_z)))+\min\{nDGT,2nLGP_T\}+8nGD+2n\alpha D^2\end{aligned} \tag{6}$$

for any sequence of comparators $\mathbf{u}(1),\dots,\mathbf{u}(T)\in\mathcal{K}$ and $i\in[n]$.

From Theorem 1, the regret of Algorithm 1 over any interval $[s,q] \subseteq [T]$ can be bounded by the total regret of $\mathcal{D}_1,\ldots,\mathcal{D}_n$ over the interval $[o_s+1, o_q-1]$. Therefore, it actually provides a reduction from minimizing the adaptive regret of D-OCO to minimizing the adaptive regret of the delayed OCO with loss functions $\ell_{1,i}(\mathbf{x}),\ldots,\ell_{Z,i}(\mathbf{x})$ for all $i \in [n]$. Similarly, Theorem 2 provides a reduction from minimizing the dynamic regret of D-OCO to minimizing the dynamic regret of the delayed OCO with these loss functions.

Remark 1 One possible concern about the reduction for dynamic regret is that the comparators at two sides of (6) are different. Intriguingly, from the definitions of m_z and Z in Theorem 2, we have

$$\sum_{z=2}^{Z}\|\mathbf{u}(m_z)-\mathbf{u}(m_{z-1})\|_2 \leq \sum_{z=2}^{Z}\sum_{t=2(z-2)L+2}^{2(z-1)L+1}\|\mathbf{u}(t)-\mathbf{u}(t-1)\|_2 \leq \sum_{t=2}^{T}\|\mathbf{u}(t)-\mathbf{u}(t-1)\|_2 \quad (7)$$

which implies that the difference of comparators will not affect the final dependence on the path-length P_T. Moreover, it is worth noting that the function $\ell_{z,i}(\mathbf{x})$ is $(2\alpha L)$-strongly convex for any $z \in [Z]$ and $i \in [n]$, which allows us to exploit the strong convexity during the reductions provided by both Theorems 1 and 2.

3.3. Revisiting An Existing Reduction for Delayed OCO

Note that although many efforts have been devoted to the delayed OCO (Joulani et al., 2013; Quanrud and Khashabi, 2015; Joulani et al., 2016; Flaspohler et al., 2021; Wan et al., 2022a,b, 2024c,d), most of them focus on the traditional regret, rather than the two metrics for changing environments. The only exception is the work of Wan et al. (2024d), which develops a variant of Ader (Zhang et al., 2018a) for minimizing the dynamic regret under arbitrary delays. Moreover, they have also established a lower bound to show the optimality of the delayed Ader in the worst case. Thus, one can simply combine our Theorem 2 with the delayed Ader to minimize the dynamic regret of D-OCO. However, for the generality of our reductions, we revisit an existing black-box technique of Joulani et al. (2013) for delayed OCO, though they only establish guarantees on the traditional regret. The detailed procedure of this technique is outlined in Algorithm 2, which can convert any non-delayed OCO algorithm $\mathcal{B}$ to handle arbitrary delays.

Specifically, here we consider a sequence of convex loss functions $\ell_1(\mathbf{x}),\ldots,\ell_Z(\mathbf{x})$, and assume that each function $\ell_z(\mathbf{x})$ can only be received at the end of round $z+c_z-1$ for any $z \in [Z]$, where $c_z \geq 1$ denotes the delay. The main idea of Algorithm 2 is to pool multiple instances of the base algorithm $\mathcal{B}$ as needed. In the pool, if one instance is ready to make the next decision, it is marked as free. In each round $z \in [Z]$, a free instance will be selected from the pool to make the decision $\mathbf{x}(z)$, or created newly when there does not exist any free instance. Then, this instance will be halted for waiting the function $\ell_z(\mathbf{x})$. Due to effect of arbitrary delays, a set of feedback $\{\ell_k(\mathbf{x})|k \in \mathcal{F}_z\}$ could be received at the end of each round z, where $\mathcal{F}_z = \{k \in [Z]|k+c_k-1=z\}$. Thus, we can feed each function in this set to update the corresponding instance, which is then marked as free. Moreover, since the base algorithm $\mathcal{B}$ may have an additional assumption on the range of loss value, in line 7 of Algorithm 2, a transformed function $\hat{\ell}_k(\mathbf{x}) = \gamma_1\ell_k(\mathbf{x})+\gamma_2$ is actually utilized in the update, where γ_1 is a scaling factor and γ_2 is a shifting factor.

Let $c_{\max} = \max\{c_1,\ldots,c_Z\}$ denote the maximum delay. A critical property of this technique is that it maintains at most $c_{\max}$ instances, which has been utilized to convert the $\mathrm{R}(Z)$ regret of the base algorithm to the $c_{\max}\,\mathrm{R}(Z/c_{\max})$ regret in the delayed case (Joulani et al., 2013). In this paper, we utilize this property to derive the following reductions for adaptive regret and dynamic regret.

Algorithm 2 Black-Box Reduction for Delayed OCO (Joulani et al., 2013)

Input: a base OCO algorithm $\mathcal{B}$, a scaling factor γ_1, and a shifting factor γ_2
Set $v = 1$, and create an instance of $\mathcal{B}$ denoted as $\mathcal{B}_1$ with the initial decision $\mathbf{0}$
for $z = 1, \dots, Z$ **do**
  Pick a free instance of $\mathcal{B}$, i.e., $\mathcal{B}_{i_z}$ with $i_z \in [v]$ has already received the required feedback
  Use $\mathcal{B}_{i_z}$ to make the decision $\mathbf{x}(z)$, and then halt it for waiting the function $\ell_z(\mathbf{x})$
  Receive delayed functions $\{\ell_k(\mathbf{x}) | k \in \mathcal{F}_z\}$, where $\mathcal{F}_z = \{k \in [Z] | k + c_k - 1 = z\}$
  For any $k \in \mathcal{F}_z$, feed $\hat{\ell}_k(\mathbf{x}) = \gamma_1 \ell_k(\mathbf{x}) + \gamma_2$ to update the instance $\mathcal{B}_{i_k}$, and mark it as free
  If there does not exist any free instance, set $v = v + 1$ and create a new instance $\mathcal{B}_v$
end for

Theorem 3 *Let $\tau = q - s + 1$. Assume that the non-delayed $\mathcal{B}$ can enjoy*

$$\sum_{z=s}^{q} \left(\hat{\ell}_z(\mathbf{x}(z)) - \hat{\ell}_z(\mathbf{x})\right) \leq C_1 \tau^{\beta_1} \left(1 + C_2 (\log_2(Z+1))^{\beta_2}\right) \tag{8}$$

for any interval $[s, q] \subseteq [Z]$ and $\mathbf{x} \in \mathcal{K}$, where $C_1 > 0$, $C_2 > 0$, $1 > \beta_1 \geq 0$, and $\beta_2 > 0$ are some constants. Then, for any interval $[s, q] \subseteq [Z]$ and $\mathbf{x} \in \mathcal{K}$, Algorithm 2 ensures

$$\sum_{z=s}^{q} (\ell_z(\mathbf{x}(z)) - \ell_z(\mathbf{x})) \leq \frac{C_1}{\gamma_1} \tau^{\beta_1} c_{\max}^{1-\beta_1} \left(1 + C_2 (\log_2(Z+1))^{\beta_2}\right).$$

Theorem 4 *Let $P_Z = \sum_{z=2}^{Z} \|\mathbf{u}(z) - \mathbf{u}(z-1)\|_2$. Assume that the non-delayed $\mathcal{B}$ can enjoy*

$$\sum_{z=1}^{Z} \left(\hat{\ell}_z(\mathbf{x}(z)) - \hat{\ell}_z(\mathbf{u}(z))\right) \leq C_1 Z^{\beta} (1 + C_2 P_Z)^{1-\beta} \tag{9}$$

for any sequence of comparators $\mathbf{u}(1), \dots, \mathbf{u}(Z) \in \mathcal{K}$, where C_1, C_2, and $\beta \in (0,1)$ are some constants. Then, for any sequence of comparators $\mathbf{u}(1), \dots, \mathbf{u}(Z) \in \mathcal{K}$, Algorithm 2 ensures

$$\sum_{z=1}^{Z} (\ell_z(\mathbf{x}(z)) - \ell_z(\mathbf{u}(z))) \leq \frac{C_1}{\gamma_1} Z^{\beta} c_{\max}^{1-\beta} (1 + C_2 P_Z)^{1-\beta}.$$

Recall that the assumptions in Theorems 3 and 4 can be respectively satisfied by many existing algorithms for adaptive regret and dynamic regret. For example, by using OGD (Zinkevich, 2003; Hazan et al., 2007) as the expert, CBCE (Jun et al., 2017) and EFLH (Hazan and Seshadhri, 2007, 2009) can satisfy (8) with $\beta_1 = \beta_2 = 1/2$ and $\beta_1 = 0, \beta_2 = 2$ for convex and strongly convex functions, respectively. For the dynamic rerget of convex functions, Ader (Zhang et al., 2018a) can satisfy (9) with $\beta = 1/2$. Note that for these results, the two factors γ_1 and γ_2 only need to be some constants that will be specified later. Therefore, Theorems 3 and 4 can extend these existing guarantees into the delayed case by mainly introducing a multiplicative factor of the maximum delay $c_{\max}$. Specifically, we can derive $O(\sqrt{c_{\max}\tau \log Z})$ and $O(c_{\max}(\log Z)^2)$ adaptive regret bounds for convex and strongly convex functions respectively, as well as an $O(\sqrt{c_{\max} Z(1 + P_Z)})$ dynamic regret bound for convex functions. These two adaptive regret bounds can nearly match existing worst-case lower bounds for the regret of delayed OCO over any fixed interval of length τ (Weinberger and Ordentlich, 2002). Moreover, this dynamic regret bound also matches the existing worst-case lower bound for the dynamic regret of delayed OCO (Wan et al., 2024d).

Remark 2 First, one may notice that Algorithm 2 could be resource-intensive for maintaining too many instances of the base algorithm $\mathcal{B}$ when $c_{\max}$ is very large. However, in our Algorithm 1, we only need to utilize Algorithm 2 to handle a constant delay of $c_{\max} = c_z = 3$, which is almost as efficient as that of $\mathcal{B}$. Moreover, although the effect of the constant delay on theoretical results actually can be ignored, Algorithm 2 is very significant as it provides a convenient way to exploit existing guarantees. Second, besides the above results, we can combine a slight modification of Theorem 4 with Baby and Wang (2022) to derive an improved $O(d^{1/3}c_{\max}^{2/3}Z^{1/3}C_Z^{2/3}\,\mathrm{poly}(\log Z))$ dynamic regret bound for strongly convex functions in the delayed case, where $C_Z = \sum_{z=2}^{Z}\|\mathbf{u}(z)-\mathbf{u}(z-1)\|_1$. It is also possible to bound the dynamic regret of D-OCO with strongly convex functions by combining Theorem 2 with this result. However, the corresponding dependence on $\log T$, $\log n$, and ρ would not be determined, because the dependence of their original bound on the strong convexity and Lipschitz constant is unclear. For this reason, we leave this problem as a future work. Finally, since Algorithm 2 actually supports the exploitation of exp-concave functions, one may also want to extend our reductions for D-OCO into exp-concave functions. However, there still lacks a theoretical guarantee on the traditional regret of D-OCO with exp-concave functions. Thus, this extension seems highly non-trivial and will also be investigated in the future.

3.4. Theoretical Guarantees on Adaptive Regret and Dynamic Regret of D-OCO

Now, we demonstrate the power of our reductions by establishing the following guarantees on adaptive regret and dynamic regret of D-OCO.

Theorem 5 *Let $\mathcal{B}$ denote the combination of CBCE (Jun et al., 2017) and OGD for convex functions (Zinkevich, 2003). Under Assumptions 1, 2, 3 with $\alpha = 0$, and 4, by setting θ, L as in (5), and $\mathcal{D}$ to be Algorithm 2 with $\mathcal{B}$, $\gamma_1 = 1/(8LGD)$, and $\gamma_2 = 1/2$, Algorithm 1 ensures*

$$\sum_{t=s}^{q}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_i(o_t)) - f_{t,j}(\mathbf{x})) \leq 32nGD\sqrt{6L(q-s)}\left(4+\sqrt{7\log_2(T+1)}\right) + 4nGD(2+L)$$

for any $[s,q] \subseteq [T]$, $\mathbf{x} \in \mathcal{K}$, and $i \in [n]$, where $o_t = \lceil t/(2L) \rceil$ denotes the block index of round t.

Combining Theorem 5 with the value of L in (5), we can obtain an $O(n\rho^{-1/4}\sqrt{\tau(\log T)\log(Tn)})$ adaptive regret bound for convex functions, where τ denotes any interval length. It matches the existing $\Omega(n\rho^{-1/4}\sqrt{\tau})$ lower bound (Wan et al., 2024b) up to polylogarithmic factors on n and T. Moreover, the values of γ_1 and γ_2 in Theorem 5 are chosen to ensure that the loss value suffered by $\mathcal{B}$ belongs to $[0, 1]$, which is required by CBCE (Jun et al., 2017). In the following, we simply set $\gamma_1 = 1$ and $\gamma_2 = 0$.

Theorem 6 *Let $\mathcal{B}$ denote the combination of EFLH (Hazan and Seshadhri, 2007, 2009) and OGD for strongly convex functions (Hazan et al., 2007). Under Assumptions 1, 2, 3 with $\alpha > 0$, and 4, by setting θ, L as in (5), and $\mathcal{D}$ to be Algorithm 2 with $\mathcal{B}$, $\gamma_1 = 1$, and $\gamma_2 = 0$, Algorithm 1 ensures*

$$\begin{aligned}&\sum_{t=s}^{q}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_i(o_t)) - f_{t,j}(\mathbf{x}))\\ \leq &\frac{3(4G+6\alpha D)^2 nL}{2\alpha}\left(1+7(\log_2(T+1))^2\right) + 4nGD(2+L) + 2n\alpha D^2\end{aligned}$$

for any $[s,q] \subseteq [T]$, $\mathbf{x} \in \mathcal{K}$, and $i \in [n]$, where $o_t = \lceil t/(2L) \rceil$ denotes the block index of round t.

Combining Theorem 6 with the value of L in (5), we can further derive an adaptive regret bound of $O(n\rho^{-1/2}(\log T)^2\log(Tn))$ for strongly convex functions, which removes the sublinear dependence of the previous bound on the interval length, and thus is much better over those long intervals. Moreover, this bound can match the existing $\Omega(n\rho^{-1/2}\log\tau)$ lower bound (Wan et al., 2024b) up to polylogarithmic factors on n and T. If ignoring the computational efficiency, we can also replace EFLH in Theorem 6 with FLH (Hazan and Seshadhri, 2007, 2009) to achieve a better adaptive regret bound of $O(n\rho^{-1/2}(\log T)\log(Tn))$ for strongly convex functions.

Theorem 7 *Let $\mathcal{B}$ denote Ader (Zhang et al., 2018a). Under Assumptions 1, 2, 3 with $\alpha = 0$, and 4, by setting θ, L as in (5), and $\mathcal{D}$ to be Algorithm 2 with $\mathcal{B}$, $\gamma_1 = 1$, and $\gamma_2 = 0$, Algorithm 1 ensures*

$$\begin{aligned}&\sum_{z=1}^{Z}\sum_{t=2(z-1)L+1}^{2zL}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_i(z)) - f_{t,j}(\mathbf{u}(t)))\\ \leq& n\left(3GD\sqrt{7}+10GD\right)\sqrt{TL\left(3+\frac{12P_T}{7D}\right)} + nG\sqrt{2DLTP_T} + 8nGD\end{aligned}$$

for any sequence of comparators $\mathbf{u}(1),\ldots,\mathbf{u}(T)\in\mathcal{K}$ and $i\in[n]$, where $Z = T/(2L)$.

Combining Theorem 7 with the value of L in (5), we can obtain an $O(n\rho^{-1/4}\sqrt{T(1+P_T)\log(Tn)})$ dynamic regret bound for convex functions, and notice that this bound holds for any sequence of comparators. Even if considering the restricted case with $\mathbf{u}(t) = \mathbf{x}^*(t)$, our bound is much better than the existing $O(n^{5/4}\rho^{-1/2}\sqrt{T}(1+P_T^*))$ bound for convex functions (Shahrampour and Jadbabaie, 2018). This improvement benefits from the use of both the accelerated gossip strategy (Liu and Morse, 2011) and Ader (Zhang et al., 2018a). Additionally, we also show that our dynamic regret is nearly optimal for convex functions by establishing the following lower bound.

Theorem 8 *Let $L = \lceil TD/\max\{C, D\}\rceil$ and $\mathcal{K} = [-D/(2\sqrt{d}), D/(2\sqrt{d})]^d$ which satisfies Assumption 2. Suppose L divides T, and $n = 2(m+1)\leq 8L+8$ for some positive integer m. For any D-OCO algorithm and $C\in[0, TD]$, there exists a sequence of comparators $\mathbf{u}(1),\ldots,\mathbf{u}(T)$ satisfying $P_T\leq C$, a sequence of loss functions satisfying Assumption 1, a graph $\mathcal{G} = ([n], E)$, and a matrix P satisfying Assumption 4 such that*

$$\sum_{t=1}^{T}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_1(t)) - f_{t,j}(\mathbf{u}(t))) \geq \frac{n\sqrt{\pi}G\sqrt{D\max\{C,D\}T}}{32\sqrt{2}(1-\sigma_2(P))^{1/4}}.$$

If $n > 8L+8$, it is easy to derive an $\Omega(nT)$ lower bound (see (39) in the appendix for details), which can be trivially matched. Thus, we only need to consider the case with $n\leq 8L+8$, and Theorem 8 essentially establishes an $\Omega(n\rho^{-1/4}\sqrt{T(1+P_T)})$ lower bound. It can match the upper bound derived from Theorem 7 up to polylogarithmic factors on n and T.

4. Conclusion

This paper proposes black-box reductions that allow us to minimize the adaptive regret and dynamic regret of D-OCO by simply exploiting existing OCO algorithms for these two metrics. Based on these reductions, we have established nearly optimal adaptive regret bounds for D-OCO with convex and strongly convex functions, as well as a nearly optimal dynamic regret bound for D-OCO with convex functions.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China (62306275), Zhejiang Provincial Natural Science Foundation of China (LMS25F030002), Yongjiang Talent Introduction Programme (2023A-193-G), and Ningbo Natural Science Foundation (2024J206). The authors would also like to thank the anonymous reviewers for their helpful comments.

References

Jacob Abernethy, Peter L. Bartlett, Alexander Rakhlin, and Ambuj Tewari. Optimal strategies and minimax lower bounds for online convex games. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 415–423, 2008.

Dheeraj Baby and Yu-Xiang Wang. Online forecasting of total-variation-bounded sequences. In *Advances in Neural Information Processing Systems 32*, pages 11071–11081, 2019.

Dheeraj Baby and Yu-Xiang Wang. Optimal dynamic regret in exp-concave online learning. In *Proceedings of the 34th Conference on Learning Theory*, pages 359–409, 2021.

Dheeraj Baby and Yu-Xiang Wang. Optimal dynamic regret in proper online learning with strongly convex losses and beyond. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, pages 1805–1845, 2022.

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

Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, 2006.

Ashok Cutkosky. Parameter-free, dynamic, and strongly-adaptive online learning. In *Proceedings of the 37th International Conference on Machine Learning*, pages 2250–2259, 2020.

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

Rishabh Dixit, Amrit Singh Bedi, Ketan Rajawat, and Alec Koppel. Distributed online learning over time-varying graphs via proximal gradient descent. In *Proceedings of the 58th IEEE Conference on Decision and Control*, pages 2745–2751, 2019.

Nima Eshraghi and Ben Liang. Improving dynamic regret in distributed online mirror descent using primal and dual information. In *Proceedings of the 4th Conference on Learning for Dynamics and Control*, pages 1–13, 2022.

Genevieve E Flaspohler, Francesco Orabona, Judah Cohen, Soukayna Mouatadid, Miruna Oprescu, Paulo Orenstein, and Lester Mackey. Online learning with optimism and delay. In *Proceedings of the 38th International Conference on Machine Learning*, pages 3363–3373, 2021.

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

Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: Optimal algorithms for stochastic strongly-convex optimization. *Journal of Machine Learning Research*, 15(71):2489–2512, 2014.

Elad Hazan and C. Seshadhri. Adaptive algorithms for online decision problems. *Electronic Colloquium on Computational Complexity*, 14(88), 2007.

Elad Hazan and C. Seshadhri. Efficient learning algorithms for changing environments. In *Proceedings of the 26th International Conference on Machine Learning*, pages 393–400, 2009.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2):169–192, 2007.

Saghar Hosseini, Airlie Chapman, and Mehran Mesbahi. Online distributed optimization via dual averaging. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 1484–1489, 2013.

Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization: Competing with dynamic comparators. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, pages 398–406, 2015.

Pooria Joulani, Andras Gyorgy, and Csaba Szepesvari. Online learning under delayed feedback. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1453–1461, 2013.

Pooria Joulani, András György, and Csaba Szepesvári. Delay-tolerant online convex optimization: Unified analysis and adaptive-gradient algorithms. *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 1744–1750, 2016.

Kwang-Sung Jun, Francesco Orabona, Stephen Wright, and Rebecca Willett. Improved strongly adaptive online learning using coin betting. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 943–951, 2017.

Xiuxian Li, Xinlei Yi, and Lihua Xie. Distributed online convex optimization with an aggregative variable. *IEEE Transactions on Control of Network Systems*, 9(1):438–449, 2022.

Xiuxian Li, Lihua Xie, and Na Li. A survey on distributed online optimization and online games. *Annual Reviews in Control*, 56(100904):1–24, 2023.

Ji Liu and A. Stephen Morse. Accelerated linear iterations for distributed averaging. *Annual Reviews in Control*, 35(2):160–165, 2011.

Kaihong Lu, Gangshan Jing, and Long Wang. Online distributed optimization with strongly pseudoconvex-sum cost functions. *IEEE Transactions on Automatic Control*, 65(1):426–433, 2020.

Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *Proceedings of the 55th IEEE Conference on Decision and Control*, pages 7195–7201, 2016.

Kent Quanrud and Daniel Khashabi. Online learning with adversarial delays. In *Advances in Neural Information Processing Systems 28*, pages 1270–1278, 2015.

Shahin Shahrampour and Ali Jadbabaie. Distributed online optimization in dynamic environments using mirror descent. *IEEE Transactions on Automatic Control*, 63(3):714–725, 2018.

Yuanyu Wan, Wei-Wei Tu, and Lijun Zhang. Projection-free distributed online convex optimization with $O(\sqrt{T})$ communication complexity. In *Proceedings of the 37th International Conference on Machine Learning*, pages 9818–9828, 2020.

Yuanyu Wan, Bo Xue, and Lijun Zhang. Projection-free online learning in dynamic environments. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 10067–10075, 2021.

Yuanyu Wan, Wei-Wei Tu, and Lijun Zhang. Online strongly convex optimization with unknown delays. *Machine Learning*, 111(3):871–893, 2022a.

Yuanyu Wan, Wei-Wei Tu, and Lijun Zhang. Online Frank-Wolfe with arbitrary delays. In *Advances in Neural Information Processing Systems 35*, pages 19703–19715, 2022b.

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, 2022c.

Yuanyu Wan, Lijun Zhang, and Mingli Song. Improved dynamic regret for online Frank-Wolfe. In *Proceedings of the 36th Annual Conference on Learning Theory*, pages 3304–3327, 2023.

Yuanyu Wan, Tong Wei, Mingli Song, and Lijun Zhang. Nearly optimal regret for decentralized online convex optimization. In *Proceedings of the 37th Annual Conference on Learning Theory*, pages 4862–4888, 2024a.

Yuanyu Wan, Tong Wei, Bo Xue, Mingli Song, and Lijun Zhang. Optimal and efficient algorithms for decentralized online convex optimization. *arXiv:2402.09173v3*, 2024b.

Yuanyu Wan, Chang Yao, Mingli Song, and Lijun Zhang. Improved regret for bandit convex optimization with delayed feedback. In *Advances in Neural Information Processing Systems 37*, pages 169–196, 2024c.

Yuanyu Wan, Chang Yao, Mingli Song, and Lijun Zhang. Non-stationary online convex optimization with arbitrary delays. In *Proceedings of the 41st International Conference on Machine Learning*, pages 49991–50011, 2024d.

Yibo Wang, Yuanyu Wan, Shimao Zhang, and Lijun Zhang. Distributed projection-free online learning for smooth and convex losses. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, pages 10226–10234, 2023.

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

Marcelo J. Weinberger and Erik Ordentlich. On delayed prediction of individual sequences. *IEEE Transactions on Information Theory*, 48(7):1959–1976, 2002.

Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53(1):65–78, 2004.

Feng Yan, Shreyas Sundaram, S.V.N. 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, 2013.

Tianbao Yang, Lijun Zhang, Rong Jin, and Jinfeng Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

Wenhao Yang, Wei Jiang, Yibo Wang, Ping Yang, Yao Hu, and Lijun Zhang. Small-loss adaptive regret for online convex optimization. In *Proceedings of the 41st International Conference on Machine Learning*, pages 56156–56195, 2024.

Haishan Ye, Luo Luo, Ziang Zhou, and Tong Zhang. Multi-consensus decentralized accelerated gradient descent. *Journal of Machine Learning Research*, 24(306):1–50, 2023.

Lijun Zhang, Tianbao Yang, Jinfeng Yi, Rong Jin, and Zhi-Hua Zhou. Improved dynamic regret for non-degenerate functions. In *Advances in Neural Information Processing Systems 30*, pages 732–741, 2017a.

Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive online learning in dynamic environments. In *Advances in Neural Information Processing Systems 31*, pages 1323–1333, 2018a.

Lijun Zhang, Tianbao Yang, Rong Jin, and Zhi-Hua Zhou. Dynamic regret of strongly adaptive methods. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5877–5886, 2018b.

Lijun Zhang, Tie-Yan Liu, and Zhi-Hua Zhou. Adaptive regret of convex and smooth functions. In *Proceedings of the 36th International Conference on Machine Learning*, pages 7414–7423, 2019a.

Lijun Zhang, Shiyin Lu, and Tianbao Yang. Minimizing dynamic regret and adaptive regret simultaneously. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, pages 309–319, 2020.

Lijun Zhang, Guanghui Wang, Wei-Wei Tu, Wei Jiang, and Zhi-Hua Zhou. Dual adaptivity: A universal algorithm for minimizing the adaptive regret of convex functions. In *Advances in Neural Information Processing Systems 34*, pages 24968–24980, 2021.

Wenpeng Zhang, Peilin Zhao, Wenwu Zhu, Steven C. H. Hoi, and Tong Zhang. Projection-free distributed online learning in networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 4054–4062, 2017b.

Yan Zhang, Robert J. Ravier, Michael M. Zavlanos, and Vahid Tarokh. A distributed online convex optimization algorithm with improved dynamic regret. In *Proceedings of the 58th IEEE Conference on Decision and Control*, pages 2449–2454, 2019b.

Peng Zhao and Lijun Zhang. Improved analysis for dynamic regret of strongly convex and smooth functions. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, pages 48–59, 2021.

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

Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Adaptivity and non-stationarity: Problem-dependent dynamic regret for online convex optimization. *Journal of Machine Learning Research*, 25(98):1–52, 2024.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, pages 928–936, 2003.

Appendix A. Proof Theorem 1

Recall that $\mathcal{T}_z = \{2(z-1)L+1, \ldots, 2zL\}$, and $o_t = \lceil t/(2L) \rceil$ denotes the block index of any round t. It is not hard to verify that

$$\begin{aligned}
&\sum_{t=s}^{q}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_i(o_t)) - f_{t,j}(\mathbf{x})) \\
=&\sum_{t=s}^{2Lo_s}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_i(o_t)) - f_{t,j}(\mathbf{x})) + \sum_{z=o_s+1}^{o_q-1}\sum_{t\in\mathcal{T}_z}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_i(z)) - f_{t,j}(\mathbf{x})) \\
&+ \sum_{t=2L(o_q-1)+1}^{q}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_i(o_t)) - f_{t,j}(\mathbf{x})) \\
\leq&\sum_{z=o_s+1}^{o_q-1}\sum_{t\in\mathcal{T}_z}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_i(z)) - f_{t,j}(\mathbf{x})) + 4nGDL
\end{aligned} \tag{10}$$

where the inequality is due to $2Lo_s - s + 1 \leq 2L$, $q - 2L(o_q - 1) \leq 2L$, and $f_{t,j}(\mathbf{x}_i(o_t)) - f_{t,j}(\mathbf{x}) \leq G\|\mathbf{x}_i(o_t) - \mathbf{x}\|_2 \leq GD$ under Assumptions 1 and 2.

To bound the first term in the right side of (10), we first introduce the following lemma, where $\bar{\mathbf{g}}(z) = \frac{1}{n}\sum_{i=1}^{n}\sum_{t\in\mathcal{T}_z}(\nabla f_{t,i}(\mathbf{x}_i(z)) - \alpha\mathbf{x}_i(z))$ and $\bar{\mathbf{x}}(z) = \frac{1}{n}\sum_{i=1}^{n}\hat{\mathbf{x}}_i(z)$.

Lemma 1 *Under Assumptions 1, 2, and 4, Algorithm 1 with θ and L defined in (5) ensures*

$$\|\mathbf{g}_i(z) - \bar{\mathbf{g}}(z)\|_2 \leq \frac{4L(G+\alpha D)}{T} \text{ and } \|\mathbf{x}_i(z) - \bar{\mathbf{x}}(z)\|_2 \leq \frac{2D}{T}$$

for any $i \in [n]$ and $z \in [T/(2L)]$.[2]

Combining with Assumptions 1 and 3, for any $z \in [T/(2L)]$, $t \in \mathcal{T}_z$, $j \in [n]$, and $\mathbf{x} \in \mathcal{K}$, we have

$$\begin{aligned}
&f_{t,j}(\mathbf{x}_i(z)) - f_{t,j}(\mathbf{x}) \\
\leq& f_{t,j}(\mathbf{x}_j(z)) - f_{t,j}(\mathbf{x}) + G\|\mathbf{x}_j(z) - \mathbf{x}_i(z)\|_2 \\
\leq&\langle\nabla f_{t,j}(\mathbf{x}_j(z)), \mathbf{x}_j(z) - \mathbf{x}\rangle - \frac{\alpha}{2}\|\mathbf{x}_j(z) - \mathbf{x}\|_2^2 + G\|\mathbf{x}_j(z) - \bar{\mathbf{x}}(z) + \bar{\mathbf{x}}(z) - \mathbf{x}_i(z)\|_2 \\
\leq&\langle\nabla f_{t,j}(\mathbf{x}_j(z)), \bar{\mathbf{x}}(z) - \mathbf{x}\rangle - \frac{\alpha}{2}\|\mathbf{x}_j(z) - \mathbf{x}\|_2^2 + \langle\nabla f_{t,j}(\mathbf{x}_j(z)), \mathbf{x}_j(z) - \bar{\mathbf{x}}(z)\rangle + \frac{4GD}{T} \\
\leq&\langle\nabla f_{t,j}(\mathbf{x}_j(z)), \bar{\mathbf{x}}(z) - \mathbf{x}\rangle - \frac{\alpha}{2}\|\mathbf{x}_j(z) - \mathbf{x}\|_2^2 + G\|\mathbf{x}_j(z) - \bar{\mathbf{x}}(z)\|_2 + \frac{4GD}{T} \\
\leq&\langle\nabla f_{t,j}(\mathbf{x}_j(z)), \bar{\mathbf{x}}(z) - \mathbf{x}\rangle - \frac{\alpha}{2}\|\mathbf{x}_j(z) - \mathbf{x}\|_2^2 + \frac{6GD}{T}
\end{aligned} \tag{11}$$

where the third and last inequalities are due to Lemma 1.

Moreover, for any $\mathbf{x}$, it is easy to verify that

$$\begin{aligned}
\|\mathbf{x}_j(z) - \mathbf{x}\|_2^2 =&\|\mathbf{x}_j(z) - \bar{\mathbf{x}}(z)\|_2^2 + 2\langle\mathbf{x}_j(z), \bar{\mathbf{x}}(z) - \mathbf{x}\rangle + \|\mathbf{x}\|_2^2 - \|\bar{\mathbf{x}}(z)\|_2^2 \\
\geq&2\langle\mathbf{x}_j(z), \bar{\mathbf{x}}(z) - \mathbf{x}\rangle + \|\mathbf{x}\|_2^2 - \|\bar{\mathbf{x}}(z)\|_2^2.
\end{aligned} \tag{12}$$

2. In the analysis, we virtually compute $\mathbf{g}_i(z)$ for $z = T/(2L)$, though it is not utilized by our algorithm.

Combining (11) with (12) and $\ell_{z,j}(\mathbf{x}) = \langle \mathbf{g}_j(z), \mathbf{x}\rangle + \alpha L\|\mathbf{x}\|_2^2$, we have

$$
\begin{aligned}
&\sum_{z=o_s+1}^{o_q-1}\sum_{t\in\mathcal{T}_z}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_i(z)) - f_{t,j}(\mathbf{x}))\\
\leq &\sum_{z=o_s+1}^{o_q-1}\sum_{t\in\mathcal{T}_z}\sum_{j=1}^{n}\left(\langle\nabla f_{t,j}(\mathbf{x}_j(z)) - \alpha\mathbf{x}_j(z), \bar{\mathbf{x}}(z)-\mathbf{x}\rangle + \frac{\alpha}{2}\left(\|\bar{\mathbf{x}}(z)\|_2^2 - \|\mathbf{x}\|_2^2\right)\right) + 6nGD\\
= &\sum_{z=o_s+1}^{o_q-1}\left(n\langle\bar{\mathbf{g}}(z), \bar{\mathbf{x}}(z)-\mathbf{x}\rangle + \alpha nL\left(\|\bar{\mathbf{x}}(z)\|_2^2 - \|\mathbf{x}\|_2^2\right)\right) + 6nGD\\
\leq &\sum_{z=o_s+1}^{o_q-1}\left(\left\langle\bar{\mathbf{g}}(z), \sum_{j=1}^{n}(\hat{\mathbf{x}}_j(z)-\mathbf{x})\right\rangle + \alpha L\sum_{j=1}^{n}\left(\|\hat{\mathbf{x}}_j(z)\|_2^2 - \|\mathbf{x}\|_2^2\right)\right) + 6nGD\\
= &\sum_{j=1}^{n}\sum_{z=o_s+1}^{o_q-1}\left(\langle\bar{\mathbf{g}}(z), \hat{\mathbf{x}}_j(z)-\mathbf{x}\rangle + \alpha L\left(\|\hat{\mathbf{x}}_j(z)\|_2^2 - \|\mathbf{x}\|_2^2\right)\right) + 6nGD\\
\leq &\sum_{j=1}^{n}\sum_{z=o_s+1}^{o_q-1}\left(\langle\mathbf{g}_j(z), \hat{\mathbf{x}}_j(z)-\mathbf{x}\rangle + \alpha L\left(\|\hat{\mathbf{x}}_j(z)\|_2^2 - \|\mathbf{x}\|_2^2\right)\right) + 6nGD\\
&+\sum_{j=1}^{n}\sum_{z=o_s+1}^{o_q-1}\|\bar{\mathbf{g}}(z)-\mathbf{g}_j(z)\|_2\|\hat{\mathbf{x}}_j(z)-\mathbf{x}\|_2\\
\leq &\sum_{j=1}^{n}\sum_{z=o_s+1}^{o_q-1}\left(\langle\mathbf{g}_j(z), \hat{\mathbf{x}}_j(z)-\mathbf{x}\rangle + \alpha L\left(\|\hat{\mathbf{x}}_j(z)\|_2^2 - \|\mathbf{x}\|_2^2\right)\right) + 8nGD + 2n\alpha D^2\\
= &\sum_{j=1}^{n}\sum_{z=o_s+1}^{o_q-1}(\ell_{z,j}(\hat{\mathbf{x}}_j(z)) - \ell_{z,j}(\mathbf{x})) + 8nGD + 2n\alpha D^2
\end{aligned}
$$

where the second inequality is due to the definition of $\bar{\mathbf{x}}(z)$ and Jensen's inequality, and the last inequality is due to Assumption 2 and Lemma 1. By substituting the above inequality into (10), we complete this proof.

Appendix B. Proof of Lemma 1

Let $X^k = [\mathbf{g}_1^k(z)^\top; \mathbf{g}_2^k(z)^\top; \ldots; \mathbf{g}_n^k(z)^\top] \in \mathbb{R}^{n\times d}$ for any integer $k \geq -1$. According to Algorithm 1, it is not hard to verify that the sequence of $X^1, \ldots, X^L$ satisfies

$$X^{k+1} = (1+\theta)PX^k - \theta X^{k-1}. \tag{13}$$

Let $\bar{X} = \left[\bar{\mathbf{g}}(z)^\top; \ldots; \bar{\mathbf{g}}(z)^\top\right]$, where $\bar{\mathbf{g}}(z) = \frac{1}{n}\sum_{i=1}^{n}\sum_{t\in\mathcal{T}_z}(\nabla f_{t,i}(\mathbf{x}_i(z)) - \alpha\mathbf{x}_i(z))$. The update rule in (13) enjoys the following property.

Lemma 2 *(Proposition 1 in Ye et al. (2023)) Under Assumption 4, for $L \geq 1$, the iterations of (13) with θ defined in (5) ensure*

$$\left\|X^L - \bar{X}\right\|_F \leq \sqrt{14}\left(1 - \left(1 - \frac{1}{\sqrt{2}}\right)\sqrt{1-\sigma_2(P)}\right)^L \left\|X^0 - \bar{X}\right\|_F.$$

For brevity, let $c = 1 - (1/\sqrt{2})$. Because of Lemma 2, for any $i \in [n]$ and $z \in [T/(2L)]$, we have

$$\|\mathbf{g}_i(z) - \bar{\mathbf{g}}(z)\|_2 \leq \left\|X^L - \bar{X}\right\|_F \leq \sqrt{14}\left(1 - c\sqrt{1-\sigma_2(P)}\right)^L \|X^0 - \bar{X}\|_F. \tag{14}$$

Then, according to the value of L defined in (5), we have

$$\begin{aligned}
&\sqrt{14}\left(1 - c\sqrt{1-\sigma_2(P)}\right)^L \leq \sqrt{14}\left(1 - c\sqrt{1-\sigma_2(P)}\right)^{\frac{\ln(T\sqrt{14n})}{c\sqrt{1-\sigma_2(P)}}} \\
\leq&\sqrt{14}\left(1 - c\sqrt{1-\sigma_2(P)}\right)^{\frac{\ln(T\sqrt{14n})}{\ln\left(1-c\sqrt{1-\sigma_2(P)}\right)^{-1}}} = \frac{1}{T\sqrt{n}}
\end{aligned} \tag{15}$$

where the last inequality is due to $\ln x^{-1} \geq 1 - x$ for any $x > 0$.

By substituting (15) into (14), for any $i \in [n]$ and $z \in [T/(2L)]$, we have

$$\begin{aligned}
\|\mathbf{g}_i(z) - \bar{\mathbf{g}}(z)\|_2 \leq& \frac{\|X^0 - \bar{X}\|_F}{T\sqrt{n}} \leq \frac{\|X^0\|_F + \|\bar{X}\|_F}{T\sqrt{n}} \\
\leq&\frac{2\sqrt{\sum_{i=1}^n \left\|\sum_{t\in\mathcal{T}_z} (\nabla f_{t,i}(\mathbf{x}_i(z)) - \alpha\mathbf{x}_i(z))\right\|_2^2}}{T\sqrt{n}} \\
\leq&\frac{4L(G+\alpha D)}{T}
\end{aligned} \tag{16}$$

where the last inequality is due to the fact that Assumptions 1 and 2 ensure

$$\left\|\sum_{t\in\mathcal{T}_z} (\nabla f_{t,i}(\mathbf{x}_i(z)) - \alpha\mathbf{x}_i(z))\right\|_2 \leq \sum_{t\in\mathcal{T}_z} \|\nabla f_{t,i}(\mathbf{x}_i(z)) - \alpha\mathbf{x}_i(z)\|_2 \leq 2L(G+\alpha D). \tag{17}$$

By repeating the above processes but considering the gap between $\mathbf{x}_i(z)$ and $\bar{\mathbf{x}}(z) = \frac{1}{n}\sum_{i=1}^n \hat{\mathbf{x}}_i(z)$, it is easy to verify that

$$\|\mathbf{x}_i(z) - \bar{\mathbf{x}}(z)\|_2 \leq \frac{2\sqrt{\sum_{i=1}^n \|\hat{\mathbf{x}}_i(z)\|_2^2}}{T\sqrt{n}} \leq \frac{2D}{T}$$

for any $i \in [n]$ and $z = 2, \dots, T/(2L)$. Moreover, for $z = 1$, we have $\mathbf{x}_i(1) = \hat{\mathbf{x}}_i(1) = \mathbf{0}$, and thus

$$\|\mathbf{x}_i(1) - \bar{\mathbf{x}}(1)\|_2 = 0 \leq \frac{2D}{T}.$$

Finally, we complete this proof by combining the above two inequalities with (16).

Appendix C. Proof of Theorem 2

Recall that $\mathcal{T}_z = \{2(z-1)L+1, \dots, 2zL\}$. We first decompose the dynamic regret of Algorithm 1 as follows:

$$\begin{aligned}
&\sum_{z=1}^{Z}\sum_{t=2(z-1)L+1}^{2zL}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_i(z)) - f_{t,j}(\mathbf{u}(t))) \\
\leq&\sum_{z=1}^{Z}\sum_{t\in\mathcal{T}_z}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_i(z)) - f_{t,j}(\mathbf{u}(m_z))) + \sum_{z=1}^{Z}\sum_{t\in\mathcal{T}_z}\sum_{j=1}^{n}(f_{t,j}(\mathbf{u}(m_z)) - f_{t,j}(\mathbf{u}(t)))\,.
\end{aligned} \tag{18}$$

Let $\bar{\mathbf{g}}(z) = \frac{1}{n}\sum_{i=1}^{n}\sum_{t\in\mathcal{T}_z}(\nabla f_{t,i}(\mathbf{x}_i(z)) - \alpha\mathbf{x}_i(z))$ and $\bar{\mathbf{x}}(z) = \frac{1}{n}\sum_{i=1}^{n}\hat{\mathbf{x}}_i(z)$. Note that (11) and (12) in the proof of Theorem 1 also hold here. Then, combining (11) with (12) and setting $\mathbf{x} = \mathbf{u}(m_z)$, we have

$$
\begin{aligned}
&\sum_{z=1}^{Z}\sum_{t\in\mathcal{T}_z}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_i(z)) - f_{t,j}(\mathbf{u}(m_z)))\\
\leq&\sum_{z=1}^{Z}\sum_{t\in\mathcal{T}_z}\sum_{j=1}^{n}\left(\langle\nabla f_{t,j}(\mathbf{x}_j(z)) - \alpha\mathbf{x}_j(z), \bar{\mathbf{x}}(z) - \mathbf{u}(m_z)\rangle + \frac{\alpha}{2}\left(\|\bar{\mathbf{x}}(z)\|_2^2 - \|\mathbf{u}(m_z)\|_2^2\right)\right)\\
&+ 6nGD\\
=&\sum_{z=1}^{Z}\left(n\langle\bar{\mathbf{g}}(z), \bar{\mathbf{x}}(z) - \mathbf{u}(m_z)\rangle + \alpha nL\left(\|\bar{\mathbf{x}}(z)\|_2^2 - \|\mathbf{u}(m_z)\|_2^2\right)\right) + 6nGD\\
\leq&\sum_{z=1}^{Z}\left(\left\langle\bar{\mathbf{g}}(z), \sum_{j=1}^{n}(\hat{\mathbf{x}}_j(z) - \mathbf{u}(m_z))\right\rangle + \alpha L\sum_{j=1}^{n}\left(\|\hat{\mathbf{x}}_j(z)\|_2^2 - \|\mathbf{u}(m_z)\|_2^2\right)\right) + 6nGD\\
=&\sum_{j=1}^{n}\sum_{z=1}^{Z}\left(\langle\bar{\mathbf{g}}(z), \hat{\mathbf{x}}_j(z) - \mathbf{u}(m_z)\rangle + \alpha L\left(\|\hat{\mathbf{x}}_j(z)\|_2^2 - \|\mathbf{u}(m_z)\|_2^2\right)\right) + 6nGD\\
\leq&\sum_{j=1}^{n}\sum_{z=1}^{Z}\left(\langle\mathbf{g}_j(z), \hat{\mathbf{x}}_j(z) - \mathbf{u}(m_z)\rangle + \alpha L\left(\|\hat{\mathbf{x}}_j(z)\|_2^2 - \|\mathbf{u}(m_z)\|_2^2\right)\right) + 6nGD\\
&+ \sum_{j=1}^{n}\sum_{z=1}^{Z}\|\bar{\mathbf{g}}(z) - \mathbf{g}_j(z)\|_2\|\hat{\mathbf{x}}_j(z) - \mathbf{u}(m_z)\|_2\\
\leq&\sum_{j=1}^{n}\sum_{z=1}^{Z}\left(\langle\mathbf{g}_j(z), \hat{\mathbf{x}}_j(z) - \mathbf{u}(m_z)\rangle + \alpha L\left(\|\hat{\mathbf{x}}_j(z)\|_2^2 - \|\mathbf{u}(m_z)\|_2^2\right)\right) + 8nGD + 2n\alpha D^2
\end{aligned}
$$

where the second inequality is due to the definition of $\bar{\mathbf{x}}(z)$ and Jensen's inequality, and the last inequality is due to Assumption 2 and Lemma 1.

Combining the above inequality with $\ell_{z,j}(\mathbf{x}) = \langle\mathbf{g}_j(z), \mathbf{x}\rangle + \alpha L\|\mathbf{x}\|_2^2$, we have

$$
\begin{aligned}
\sum_{z=1}^{Z}\sum_{t\in\mathcal{T}_z}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_i(z)) - f_{t,j}(\mathbf{u}(m_z))) \leq&\sum_{j=1}^{n}\sum_{z=1}^{Z}(\ell_{z,j}(\hat{\mathbf{x}}_j(z)) - \ell_{z,j}(\mathbf{u}(m_z)))\\
&+ 8nGD + 2n\alpha D^2.
\end{aligned} \tag{19}
$$

Next, for the second term in the right side of (18), due to Assumption 1, we have

$$
\begin{aligned}
\sum_{z=1}^{Z}\sum_{t\in\mathcal{T}_z}\sum_{j=1}^{n}(f_{t,j}(\mathbf{u}(m_z)) - f_{t,j}(\mathbf{u}(t))) \leq&\sum_{z=1}^{Z}\sum_{t\in\mathcal{T}_z}\sum_{j=1}^{n}G\|\mathbf{u}(m_z) - \mathbf{u}(t)\|_2\\
=&\sum_{z=1}^{Z}\sum_{t\in\mathcal{T}_z}nG\|\mathbf{u}(m_z) - \mathbf{u}(t)\|_2.
\end{aligned} \tag{20}
$$

Due to Assumption 2, it is easy to verify that

$$\sum_{z=1}^{Z}\sum_{t\in\mathcal{T}_z} nG\left\|\mathbf{u}(m_z)-\mathbf{u}(t)\right\|_2 \leq nDGT. \tag{21}$$

Moreover, because of $m_z = 2(z-1)L+1$, we also have

$$\begin{aligned}\sum_{z=1}^{Z}\sum_{t\in\mathcal{T}_z} nG\left\|\mathbf{u}(m_z)-\mathbf{u}(t)\right\|_2 \leq & nG\sum_{z=1}^{Z}\sum_{t\in\mathcal{T}_z}\sum_{i=2(z-1)L+1}^{t-1}\left\|\mathbf{u}(i)-\mathbf{u}(i+1)\right\|_2 \\ \leq & 2nLG\sum_{z=1}^{Z}\sum_{i=2(z-1)L+1}^{2zL-1}\left\|\mathbf{u}(i)-\mathbf{u}(i+1)\right\|_2 \\ \leq & 2nLG\sum_{t=2}^{T}\left\|\mathbf{u}(t)-\mathbf{u}(t-1)\right\|_2.\end{aligned} \tag{22}$$

Combining (20) with (21) and (22), we have

$$\sum_{z=1}^{Z}\sum_{t\in\mathcal{T}_z}\sum_{j=1}^{n}\left(f_{t,j}(\mathbf{u}(m_z))-f_{t,j}(\mathbf{u}(t))\right) \leq \min\{nDGT, 2nLGP_T\}. \tag{23}$$

By substituting (19) and (23) into (18), we complete this proof.

Appendix D. Proof of Theorem 3

Let M denote the total number of instances created by Algorithm 2. It is easy to verify that the total Z rounds are divided into M subsets, each handled by an instance of $\mathcal{B}$. For any $i \in [M]$, we assume that the subset handled by $\mathcal{B}_i$ has Z_i rounds, and denote it as $\mathcal{S}_i = \left\{s_1^i, \dots, s_{Z_i}^i\right\}$. Moreover, let $I = [s, q]$, and $\tau_i = |\mathcal{S}_i \cap I|$ for any $i \in [M]$. Then, we have

$$\begin{aligned}\sum_{z=s}^{q}\left(\ell_z(\mathbf{x}(z))-\ell_z(\mathbf{x})\right) = & \sum_{i=1}^{M}\sum_{z\in\mathcal{S}_i\cap I}\frac{1}{\gamma_1}\left(\hat{\ell}_z(\mathbf{x}(z))-\hat{\ell}_z(\mathbf{x})\right) \\ \leq & \sum_{i=1}^{M}\frac{C_1}{\gamma_1}\tau_i^{\beta_1}\left(1+C_2(\log_2(Z_i+1))^{\beta_2}\right) \\ \leq & \frac{C_1}{\gamma_1}\left(1+C_2(\log_2(Z+1))^{\beta_2}\right)\sum_{i=1}^{M}\tau_i^{\beta_1} \\ \leq & \frac{C_1}{\gamma_1}\left(1+C_2(\log_2(Z+1))^{\beta_2}\right)M^{1-\beta_1}\left(\sum_{i=1}^{M}\tau_i\right)^{\beta_1}\end{aligned} \tag{24}$$

where the first inequality is due to the power of $\mathcal{B}$ assumed in (8), and the last inequality is due to Hölder's inequality.

Next, we only need to provide an upper bound for M. Let v_z denote the number of instances maintained in the beginning of round z. According to Algorithm 2, we only create an instance of

$\mathcal{B}$ when there does not exist any free instance. In other words, we have $v_{z+1} = v_z + 1$ only if $z - \sum_{i=1}^{z} |\mathcal{F}_i| = v_z$, and $v_{z+1} = v_z$ otherwise, which implies that

$$M = \max_{z\in[Z]} \left\{ z - \sum_{i=1}^{z} |\mathcal{F}_i| \right\} + 1 \leq c_{\max} \tag{25}$$

where the last inequality is due to the fact that all gradients queried before round $z - c_{\max} + 1$ must be received before the end of round z.

Finally, combining (24) with (25) and $\sum_{i=1}^{M} \tau_i = |I| = \tau$, we have

$$\sum_{z=s}^{q} (\ell_z(\mathbf{x}(z)) - \ell_z(\mathbf{x})) \leq \frac{C_1}{\gamma_1} \tau^{\beta_1} c_{\max}^{1-\beta_1} \left(1 + C_2(\log_2(Z+1))^{\beta_2}\right).$$

Appendix E. Proof of Theorem 4

We adopt the same definition of M, Z_i, and $\mathcal{S}_i$ as in the proof of Theorem 3. Then, we have

$$\begin{aligned}
\sum_{z=1}^{Z} (\ell_z(\mathbf{x}(z)) - \ell_z(\mathbf{u}(z))) &= \sum_{i=1}^{M} \sum_{z\in S_i} \frac{1}{\gamma_1} \left(\hat{\ell}_z(\mathbf{x}(z)) - \hat{\ell}_z(\mathbf{u}(z))\right) \\
&\leq \sum_{i=1}^{M} \frac{C_1}{\gamma_1} Z_i^{\beta} \left(1 + C_2 \sum_{k=2}^{Z_i} \left\|\mathbf{u}(s_k^i) - \mathbf{u}(s_{k-1}^i)\right\|_2\right)^{1-\beta} \\
&\leq \frac{C_1}{\gamma_1} \left(\sum_{i=1}^{M} Z_i\right)^{\beta} \left(M + C_2 \sum_{i=1}^{M} \sum_{k=2}^{Z_i} \left\|\mathbf{u}(s_k^i) - \mathbf{u}(s_{k-1}^i)\right\|_2\right)^{1-\beta} \\
&= \frac{C_1}{\gamma_1} Z^{\beta} \left(M + C_2 \sum_{i=1}^{M} \sum_{k=2}^{Z_i} \left\|\mathbf{u}(s_k^i) - \mathbf{u}(s_{k-1}^i)\right\|_2\right)^{1-\beta}.
\end{aligned} \tag{26}$$

where the first inequality is due to the power of $\mathcal{B}$ assumed in (9), and the last inequality is due to Hölder's inequality.

Moreover, it is not hard to verify that

$$\begin{aligned}
\sum_{i=1}^{M} \sum_{k=2}^{Z_i} \left\|\mathbf{u}(s_k^i) - \mathbf{u}(s_{k-1}^i)\right\|_2 &\leq \sum_{i=1}^{M} \sum_{k=2}^{Z_i} \sum_{j=s_{k-1}^i+1}^{s_k^i} \|\mathbf{u}(j) - \mathbf{u}(j-1)\|_2 \\
&= \sum_{i=1}^{M} \sum_{j=s_1^i+1}^{s_{Z_i}^i} \|\mathbf{u}(j) - \mathbf{u}(j-1)\|_2 \\
&\leq M \sum_{z=2}^{Z} \|\mathbf{u}(z) - \mathbf{u}(z-1)\|_2.
\end{aligned} \tag{27}$$

Note that (25) in the proof of Theorem 3 also holds here. Then, combining (26) with (25) and (27), we have

$$\sum_{z=1}^{Z} (\ell_z(\mathbf{x}(z)) - \ell_z(\mathbf{u}(z))) \leq \frac{C_1}{\gamma_1} Z^{\beta} c_{\max}^{1-\beta} \left(1 + C_2 \sum_{z=2}^{Z} \|\mathbf{u}(z) - \mathbf{u}(z-1)\|_2\right)^{1-\beta}. \tag{28}$$

Appendix F. Proof of Theorem 5

According to Theorem 1, we only need to bound $\sum_{z=o_s+1}^{o_q-1}(\ell_{z,i}(\hat{\mathbf{x}}_i(z))-\ell_{z,i}(\mathbf{x}))$ for any $i\in[n]$, where $o_q=\lceil q/(2L)\rceil$ and $o_s=\lceil s/(2L)\rceil$. Moreover, we notice that $\hat{\mathbf{x}}_i(1),\ldots,\hat{\mathbf{x}}_i(Z)$ are generated by running an instance of Algorithm 2 over $\ell_{1,i}(\mathbf{x}),\ldots,\ell_{Z,i}(\mathbf{x})$, where $Z=T/(2L)$. Therefore, such a bound can be established by utilizing Theorem 3 about the adaptive regret of Algorithm 2. To this end, we define $\hat{\ell}_{z,i}(\mathbf{x})=\gamma_1\ell_{z,i}(\mathbf{x})+\gamma_2$ with $\gamma_1=1/(8LGD)$ and $\gamma_2=1/2$, and notice that $\ell_{z,i}(\mathbf{x})=\langle\mathbf{g}_i(z),\mathbf{x}\rangle$ for $\alpha=0$. Let $\bar{\mathbf{g}}(z)=\frac{1}{n}\sum_{i=1}^n\sum_{t\in\mathcal{T}_z}(\nabla f_{t,i}(\mathbf{x}_i(z))-\alpha\mathbf{x}_i(z))$, where $\mathcal{T}_z=\{2(z-1)L+1,\ldots,2zL\}$. For any $i\in[n]$, combining Lemma 1 and (17) used in the proof of Lemma 1, we have

$$\begin{aligned}\max_{z\in[Z]}\|\mathbf{g}_i(z)\|_2\leq&\max_{z\in[Z]}\{\|\mathbf{g}_i(z)-\bar{\mathbf{g}}(z)\|_2+\|\bar{\mathbf{g}}(z)\|_2\}\\ \leq&\max_{z\in[Z]}\left\{\frac{4L(G+\alpha D)}{T}+2L(G+\alpha D)\right\}\leq 4L(G+\alpha D)\end{aligned}\tag{29}$$

where the last inequality is due to $T\geq 2L\geq 2$. Due to (29) and $\alpha=0$, we have

$$\left\|\nabla\hat{\ell}_{z,i}(\mathbf{x})\right\|_2=\left\|\frac{\mathbf{g}_i(z)}{8LGD}\right\|_2\leq\frac{1}{2D}\tag{30}$$

for any $\mathbf{x}\in\mathcal{K}$. Similarly, it is easy to verify that

$$0\leq-\frac{\|\mathbf{g}_i(z)\|_2\|\mathbf{x}\|_2}{8LGD}+\frac{1}{2}\leq\hat{\ell}_{z,i}(\mathbf{x})\leq\frac{\|\mathbf{g}_i(z)\|_2\|\mathbf{x}\|_2}{8LGD}+\frac{1}{2}\leq 1\tag{31}$$

where the first and last inequalities are due to (29) with $\alpha=0$ and Assumption 2.

Note that the base algorithm $\mathcal{B}$ in Algorithm 2 has been set to be the combination of CBCE and OGD for convex functions. Combining Theorem 2 of Jun et al. (2017) and Theorem 3.1 of Hazan (2016) with (30) and (31), if Algorithm 2 is run without delays, $\mathcal{B}$ can ensure

$$\sum_{z=s}^{q}\left(\hat{\ell}_{z,i}(\hat{\mathbf{x}}_i(z))-\hat{\ell}_{z,i}(\mathbf{x})\right)\leq\frac{3\sqrt{\tau}}{\sqrt{2}-1}+8\sqrt{\tau(7\ln(Z)+5)}\leq 32\sqrt{\tau}\left(1+\frac{\sqrt{7\log_2(Z+1)}}{4}\right)$$

for any interval $[s,q]\subseteq[Z]$ and $\mathbf{x}\in\mathcal{K}$, where $\tau=q-s+1$. By further applying Theorem 3 with the above inequality, $c_{\max}=3$, and $\gamma_1=1/(8LGD)$, Algorithm 2 can enjoy

$$\sum_{z=s}^{q}(\ell_{z,i}(\hat{\mathbf{x}}_i(z))-\ell_{z,i}(\mathbf{x}))\leq 256LGD\sqrt{3\tau}\left(1+\frac{\sqrt{7\log_2(Z+1)}}{4}\right)\tag{32}$$

for any interval $[s,q]\subseteq[Z]$ and $\mathbf{x}\in\mathcal{K}$.

Then, combining (32) and $Z=T/(2L)$, if $o_q-o_s-1\geq 1$, for any $i\in[n]$, we have

$$\begin{aligned}\sum_{z=o_s+1}^{o_q-1}(\ell_{z,i}(\hat{\mathbf{x}}_i(z))-\ell_{z,i}(\mathbf{x}))\leq&256LGD\sqrt{3(o_q-o_s-1)}\left(1+\frac{\sqrt{7\log_2(Z+1)}}{4}\right)\\ \leq&32GD\sqrt{6L(q-s)}\left(4+\sqrt{7\log_2(T+1)}\right).\end{aligned}\tag{33}$$

Finally, combining Theorem 1 with $\alpha=0$ and (33), we have

$$\sum_{t=s}^{q}\sum_{j=1}^{n}(f_{t,j}(\mathbf{x}_i(o_t))-f_{t,j}(\mathbf{x}))\leq 32nGD\sqrt{6L(q-s)}\left(4+\sqrt{7\log_2(T+1)}\right)+4nGD(2+L)$$

for any $[s,q]\subseteq[T]$, $\mathbf{x}\in\mathcal{K}$, and $i\in[n]$.

Appendix G. Proof of Theorem 6

This proof is similar to the proof of Theorem 5, and we only need to make some modifications to exploit the strong convexity of functions. Specifically, we first notice that $\hat{\ell}_{z,i}(\mathbf{x}) = \gamma_1 \ell_{z,i}(\mathbf{x}) + \gamma_2 = \ell_{z,i}(\mathbf{x}) = \langle \mathbf{g}_i(z), \mathbf{x}\rangle + \alpha L\|\mathbf{x}\|_2^2$ due to $\gamma_1 = 1$, $\gamma_2 = 0$, and $\alpha > 0$, and the base algorithm $\mathcal{B}$ in Algorithm 2 is now set to be the combination of EFLH and OGD for strongly convex functions. From Theorem 2 of Zhang et al. (2018b), if Algorithm 2 is run without delays, $\mathcal{B}$ can ensure

$$\begin{aligned}\sum_{z=s}^{q}\left(\hat{\ell}_{z,i}(\hat{\mathbf{x}}_i(z)) - \hat{\ell}_{z,i}(\mathbf{x})\right) \leq & \frac{(G')^2}{4\alpha L}\left(\log_2(Z) + 2 + (3\log_2(Z) + 10)\ln Z\right) \\ \leq & \frac{(G')^2}{2\alpha L}\left(1 + 7(\log_2(Z+1))^2\right)\end{aligned}$$

for any interval $[s, q] \subseteq [Z]$ and $\mathbf{x} \in \mathcal{K}$, where $G' = \max_{z\in[Z],\mathbf{x}\in\mathcal{K}} \|\mathbf{g}_i(z) + 2\alpha L\mathbf{x}\|_2$.

Then, by applying Theorem 3 with the above inequality, $c_{\max} = 3$, and $\gamma_1 = 1$, Algorithm 2 can enjoy

$$\sum_{z=s}^{q}\left(\ell_{z,i}(\hat{\mathbf{x}}_i(z)) - \ell_{z,i}(\mathbf{x})\right) \leq \frac{3(G')^2}{2\alpha L}\left(1 + 7(\log_2(Z+1))^2\right). \tag{34}$$

for any interval $[s, q] \subseteq [Z]$ and $\mathbf{x} \in \mathcal{K}$. Moreover, due to (29) in the proof of Theorem 5 and Assumption 2, it is easy to verify that

$$G' \leq 4L(G + \alpha D) + 2\alpha L\|\mathbf{x}\|_2 \leq 4LG + 6\alpha LD. \tag{35}$$

Combining (34) with (35), and $Z = T/(2L)$, for any $i \in [n]$, if $o_q - o_s - 1 \geq 1$, we have

$$\sum_{z=o_s+1}^{o_q-1}\left(\ell_{z,i}(\hat{\mathbf{x}}_i(z)) - \ell_{z,i}(\mathbf{x})\right) \leq \frac{3(4G + 6\alpha D)^2 L}{2\alpha}\left(1 + 7(\log_2(T+1))^2\right).$$

Finally, combining Theorem 1 with the above inequality, we have

$$\begin{aligned}&\sum_{t=s}^{q}\sum_{j=1}^{n}\left(f_{t,j}(\mathbf{x}_i(o_t)) - f_{t,j}(\mathbf{x})\right) \\ \leq & \frac{3(4G + 6\alpha D)^2 nL}{2\alpha}\left(1 + 7(\log_2(T+1))^2\right) + 4nGD(2 + L) + 2n\alpha D^2\end{aligned}$$

for any $[s, q] \subseteq [T]$, $\mathbf{x} \in \mathcal{K}$, and $i \in [n]$.

Appendix H. Proof of Theorem 7

From Theorem 2, we only need to bound $\sum_{z=1}^{Z}\left(\ell_{z,i}(\hat{\mathbf{x}}_i(z)) - \ell_{z,i}(\mathbf{u}(m_z))\right)$ for any $i \in [n]$. Since $\hat{\mathbf{x}}_i(1), \ldots, \hat{\mathbf{x}}_i(Z)$ are generated by running an instance of Algorithm 2 over $\ell_{1,i}(\mathbf{x}), \ldots, \ell_{Z,i}(\mathbf{x})$, such a bound can be established by utilizing Theorem 4 about the dynamic regret of Algorithm 2. To this end, we first define $\hat{\ell}_{z,i}(\mathbf{x}) = \gamma_1 \ell_{z,i}(\mathbf{x}) + \gamma_2$ and notice that $\hat{\ell}_{z,i}(\mathbf{x}) = \ell_{z,i}(\mathbf{x}) = \langle \mathbf{g}_i(z), \mathbf{x}\rangle$ due to $\gamma_1 = 1$, $\gamma_2 = 0$, and $\alpha = 0$. From (35) in the proof of Theorem 6, it is easy to verify that

$$\left\|\hat{\ell}_{z,i}(\mathbf{x})\right\|_2 = \|\mathbf{g}_i(z)\|_2 \leq 4LG. \tag{36}$$

Moreover, we have set the base algorithm $\mathcal{B}$ in Algorithm 2 to be Ader.

Then, combining Theorem 4 of Zhang et al. (2018a) with $P_Z = \sum_{z=2}^{Z} \|\mathbf{u}(z) - \mathbf{u}(z-1)\|_2$ and (36), if Algorithm 2 is run without delays, $\mathcal{B}$ can ensure

$$\begin{aligned}
&\sum_{z=1}^{Z} \left(\hat{\ell}_z(\hat{\mathbf{x}}_i(z)) - \hat{\ell}_z(\mathbf{u}(z))\right) \\
\leq& 3LG\sqrt{2Z(7D^2 + 4DP_Z)} + 2LGD\sqrt{2Z}\left(1 + 2\ln(k+1)\right) \\
\leq& 3LG\sqrt{2Z(7D^2 + 4DP_Z)} + 2LGD\sqrt{2Z}\left(3 + 2\sqrt{1 + \frac{4P_Z}{7D}}\right) \\
\leq& 3LGD\sqrt{14Z\left(1 + \frac{4P_Z}{7D}\right)} + 10LGD\sqrt{2Z\left(1 + \frac{4P_Z}{7D}\right)} \\
=& \left(3LGD\sqrt{14} + 10LGD\sqrt{2}\right)\sqrt{Z\left(1 + \frac{4P_Z}{7D}\right)}
\end{aligned}$$

for any $\mathbf{u}(1), \ldots, \mathbf{u}(Z) \in \mathcal{K}$, where $k = 1 + \lfloor \log_2 \sqrt{1 + 4P_Z/(7D)} \rfloor$.

By further applying Theorem 4 with the above inequality, $c_{\max} = 3$, and $\gamma_1 = 1$, Algorithm 2 can enjoy

$$\sum_{z=1}^{Z} \left(\ell_z(\hat{\mathbf{x}}_i(z)) - \ell_z(\mathbf{u}(z))\right) \leq \left(3LGD\sqrt{14} + 10LGD\sqrt{2}\right)\sqrt{Z\left(3 + \frac{12P_Z}{7D}\right)} \tag{37}$$

for any $\mathbf{u}(1), \ldots, \mathbf{u}(Z) \in \mathcal{K}$. Combining (37) with (7) and $Z = T/(2L)$, for any $i \in [n]$, we have

$$\sum_{z=1}^{Z} \left(\ell_{z,i}(\hat{\mathbf{x}}_i(z)) - \ell_{z,i}(\mathbf{u}(m_z))\right) \leq \left(3GD\sqrt{7} + 10GD\right)\sqrt{TL\left(3 + \frac{12P_T}{7D}\right)}.$$

Finally, combining Theorem 2 with $\alpha = 0$ and the above inequality, we have

$$\begin{aligned}
&\sum_{z=1}^{Z} \sum_{t=2(z-1)L+1}^{2zL} \sum_{j=1}^{n} \left(f_{t,j}(\mathbf{x}_i(z)) - f_{t,j}(\mathbf{u}(t))\right) \\
\leq& n\left(3GD\sqrt{7} + 10GD\right)\sqrt{TL\left(3 + \frac{12P_T}{7D}\right)} + \min\left\{nDGT, 2nLGP_T\right\} + 8nGD \\
\leq& n\left(3GD\sqrt{7} + 10GD\right)\sqrt{TL\left(3 + \frac{12P_T}{7D}\right)} + nG\sqrt{2DLTP_T} + 8nGD
\end{aligned}$$

for any $\mathbf{u}(1), \ldots, \mathbf{u}(T) \in \mathcal{K}$ and $i \in [n]$.

Appendix I. Proof of Theorem 8

Inspired by the existing analysis of the lower dynamic regret bound for OCO (Zhang et al., 2018a), we first divide the total T rounds into $Z' = T/L$ blocks, where each block contains L rounds. In this way, we can define the set of rounds in the block z as $\mathcal{T}'_z = \{(z-1)L+1, \ldots, zL\}$. Moreover, we define the feasible set of $\mathbf{u}(1), \ldots, \mathbf{u}(T)$ as

$$\mathcal{C}(C) = \left\{ \mathbf{u}(1), \ldots, \mathbf{u}(T) \in \mathcal{K} \left| \sum_{t=2}^{T} \|\mathbf{u}(t) - \mathbf{u}(t-1)\|_2 \leq C \right. \right\}$$

and construct a subset as $\mathcal{C}'(C) = \{\mathbf{u}(1), \ldots, \mathbf{u}(T) \in \mathcal{K} | \mathbf{u}(i) = \mathbf{u}(j), \forall z \in [Z'], i, j \in \mathcal{T}'_z\}$. Note that the connection $\mathcal{C}'(C) \subseteq \mathcal{C}(C)$ is due to the fact that the comparator sequence in $\mathcal{C}'(C)$ only changes

$$Z' - 1 = \frac{T}{L} - 1 \leq \frac{\max\{C, D\}}{D} - 1 \leq \frac{C}{D}$$

times, and thus its path-length does not exceed C.

Then, due to $\mathcal{C}'(C) \subseteq \mathcal{C}(C)$, it is easy to verify that

$$\begin{aligned}
&\sum_{t=1}^{T} \sum_{j=1}^{n} f_{t,j}(\mathbf{x}_1(t)) - \min_{\mathbf{u}(1),\ldots,\mathbf{u}(T) \in \mathcal{C}(P)} \sum_{t=1}^{T} \sum_{j=1}^{n} f_{t,j}(\mathbf{u}(t)) \\
\geq &\sum_{t=1}^{T} \sum_{j=1}^{n} f_{t,j}(\mathbf{x}_1(t)) - \min_{\mathbf{u}(1),\ldots,\mathbf{u}(T) \in \mathcal{C}'(P)} \sum_{t=1}^{T} \sum_{j=1}^{n} f_{t,j}(\mathbf{u}(t)) \\
= &\sum_{z=1}^{Z'} \left(\sum_{t \in \mathcal{T}'_z} \sum_{j=1}^{n} f_{t,j}(\mathbf{x}_1(t)) - \min_{\mathbf{x} \in \mathcal{K}} \sum_{t \in \mathcal{T}'_z} \sum_{j=1}^{n} f_{t,j}(\mathbf{x}) \right).
\end{aligned} \tag{38}$$

Next, we introduce an existing lower bound on the regret of any D-OCO algorithm.

Lemma 3 *(Theorem 3 of Wan et al. (2024a)) Suppose $\mathcal{K} = [-D/(2\sqrt{d}), D/(2\sqrt{d})]^d$, and $n = 2(m+1)$ for some positive integer m. For any D-OCO algorithm, there exists a sequence of loss functions satisfying Assumption 1, a graph $\mathcal{G} = ([n], E)$, and a matrix P satisfying Assumption 4 such that*

$$\text{if } n \leq 8T+8, \mathrm{R}(T,1) \geq \frac{n\sqrt{\pi}DG\sqrt{T}}{32(1-\sigma_2(P))^{1/4}}, \text{ and otherwise, } \mathrm{R}(T,1) \geq \frac{nDGT}{8}.$$

Combining (38) with Lemma 3, there exists a sequence of loss functions satisfying Assumption 1, a graph $\mathcal{G} = ([n], E)$, and a matrix P satisfying Assumption 4 such that

$$\begin{aligned}
&\sum_{t=1}^{T} \sum_{j=1}^{n} f_{t,j}(\mathbf{x}_1(t)) - \min_{\mathbf{u}(1),\ldots,\mathbf{u}(T) \in \mathcal{C}(P)} \sum_{t=1}^{T} \sum_{j=1}^{n} f_{t,j}(\mathbf{u}(t)) \\
\geq &\sum_{z=1}^{Z'} \frac{n\sqrt{\pi}DG\sqrt{L}}{32(1-\sigma_2(P))^{1/4}} = \frac{n\sqrt{\pi}DGT}{32(1-\sigma_2(P))^{1/4}\sqrt{L}} \geq \frac{n\sqrt{\pi}G\sqrt{D\max\{C,D\}T}}{32\sqrt{2}(1-\sigma_2(P))^{1/4}}
\end{aligned}$$

when $n \leq 8L+8$, where the last inequality is due to $L = \lceil TD/\max\{C,D\} \rceil \leq 2TD/\max\{C,D\}$.

In the same way, we can prove that if $n > 8L+8$, there exists a sequence of loss functions satisfying Assumption 1, a graph $\mathcal{G} = ([n], E)$, and a matrix P satisfying Assumption 4 such that

$$\sum_{t=1}^{T} \sum_{j=1}^{n} f_{t,j}(\mathbf{x}_1(t)) - \min_{\mathbf{u}(1),\ldots,\mathbf{u}(T) \in \mathcal{C}(P)} \sum_{t=1}^{T} \sum_{j=1}^{n} f_{t,j}(\mathbf{u}(t)) \geq \sum_{z=1}^{Z'} \frac{nDGL}{8} = \frac{nDGT}{8}. \tag{39}$$