Non-stationary Online Convex Optimization with Arbitrary Delays

Yuanyu Wan [1 2 3] **Chang Yao** [1 2 3] **Mingli Song** [1 3] **Lijun Zhang** [4 1]

Abstract

Online convex optimization (OCO) with arbitrary delays, in which gradients or other information of functions could be arbitrarily delayed, has received increasing attention recently. Different from previous studies that focus on stationary environments, this paper investigates the delayed OCO in non-stationary environments, and aims to minimize the dynamic regret with respect to any sequence of comparators. To this end, we first propose a simple algorithm, namely DOGD, which performs a gradient descent step for each delayed gradient according to their arrival order. Despite its simplicity, our novel analysis shows that the dynamic regret of DOGD can be automatically bounded by $O(\sqrt{\bar{d}T}(P_T+1))$ under mild assumptions, and $O(\sqrt{dT}(P_T+1))$ in the worst case, where $\bar{d}$ and d denote the average and maximum delay respectively, T is the time horizon, and P_T is the path-length of comparators. Furthermore, we develop an improved algorithm, which reduces those dynamic regret bounds achieved by DOGD to $O(\sqrt{\bar{d}T(P_T+1)})$ and $O(\sqrt{dT(P_T+1)})$, respectively. The key idea is to run multiple DOGD with different learning rates, and utilize a meta-algorithm to track the best one based on their delayed performance. Finally, we demonstrate that our improved algorithm is optimal in the worst case by deriving a matching lower bound.

1. Introduction

Online convex optimization (OCO) has become a popular paradigm for solving sequential decision-making problems (Shalev-Shwartz, 2011; Hazan, 2016; Orabona, 2019). In OCO, an online player acts as the decision maker, which chooses a decision $\mathbf{x}_t$ from a convex set $\mathcal{K} \subseteq \mathbb{R}^n$ at each round $t \in [T]$. After the decision $\mathbf{x}_t$ is committed, the player suffers a loss $f_t(\mathbf{x}_t)$, where $f_t(\mathbf{x}) : \mathcal{K} \mapsto \mathbb{R}$ is a convex function selected by an adversary. To improve the performance in subsequent rounds, the player needs to update the decision by exploiting information about loss functions in previous rounds. Plenty of algorithms and theories have been introduced to guide the player (Zinkevich, 2003; Shalev-Shwartz & Singer, 2007; Hazan et al., 2007).

However, most of existing studies assume that the information about each function $f_t(\mathbf{x})$ is revealed at the end of round t, which is not necessarily satisfied in many real applications. For example, in online advertisement (McMahan et al., 2013; He et al., 2014), each loss function depends on whether a user clicks an ad or not, which may not be decided even when the user has observed the ad for a long period of time. To tackle this issue, there has been a surge of research interest in OCO with arbitrary delays (Joulani et al., 2013; McMahan & Streeter, 2014; Quanrud & Khashabi, 2015; Joulani et al., 2016; Flaspohler et al., 2021; Wan et al., 2022a;b; 2023a), where the information about $f_t(\mathbf{x})$ is revealed at the end of round $t + d_t - 1$, and $d_t \geq 1$ denotes the delay. However, these studies focus on developing algorithms to minimize the static regret of the player, i.e., $R(T) = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x})$, which is only meaningful for stationary environments where at least one fixed decision can minimize the cumulative loss well, and thus cannot handle non-stationary environments where the best decision is drifting over time.

To address this limitation, we investigate the delayed OCO with a more suitable performance metric called dynamic regret (Zinkevich, 2003):

$$R(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)$$

which compares the player against any sequence of changing comparators $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{K}$. It is well-known that in the non-delayed setting, online gradient descent (OGD) can attain a dynamic regret bound of $O(\sqrt{T}(P_T+1))$ (Zinkevich, 2003), where $P_T = \sum_{t=2}^{T} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2$ is the path-length of comparators, and multiple OGD with different learning

[1]The State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, China [2]School of Software Technology, Zhejiang University, Ningbo, China [3]Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security, Hangzhou, China [4]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China. Correspondence to: Yuanyu Wan <wanyy@zju.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

rates can be combined to achieve an optimal dynamic regret bound of $O(\sqrt{T(P_T+1)})$ by using a mete-algorithm (Zhang et al., 2018a). Thus, it is natural to ask whether these algorithms and dynamic regret bounds can be generalized into the setting with arbitrary delays.

In this paper, we provide an affirmative answer to the above question. Specifically, we first propose delayed online gradient descent (DOGD), and provide a novel analysis on its dynamic regret. In the literature, Quanrud & Khashabi (2015) have developed a delayed variant of OGD for minimizing the static regret, which performs a gradient descent step by using the sum of gradients received in each round. Different from their algorithm, our DOGD performs a gradient descent step for each delayed gradient according to their arrival order, which allows us to exploit an In-Order property (i.e., delays do not change the arrival order of gradients) to reduce the dynamic regret. Let $\bar{d} = \sum_{t=1}^{T} d_t/T$ and $d = \max\{d_1, \ldots, d_T\}$ denote the average and maximum delay, respectively. Our analysis shows that the dynamic regret of DOGD can be automatically bounded by $O(\sqrt{\bar{d}T}(P_T+1))$ under mild assumptions such as the In-Order property, and $O(\sqrt{dT}(P_T+1))$ in the worst case.

Furthermore, inspired by Zhang et al. (2018a), we propose an improved algorithm based on DOGD, namely multiple delayed online gradient descent (Mild-OGD). The essential idea is to run multiple DOGD, each with a different learning rate that enjoys small dynamic regret for a specific path-length, and combine them with a meta-algorithm. Compared with Zhang et al. (2018a), the key challenge is that the performance of each DOGD is required by the meta-algorithm, but it is also arbitrarily delayed. To address this difficulty, our meta-algorithm is built upon the delayed Hedge—a technique for prediction with delayed expert advice (Korotin et al., 2020), which can track the best DOGD based on their delayed performance. We prove that the dynamic regret of Mild-OGD can be automatically bounded by $O(\sqrt{\bar{d}T(P_T+1)})$ under mild assumptions such as the In-Order property, and $O(\sqrt{dT(P_T+1)})$ in the worst case. In the special case without delay, both bounds reduce to the $O(\sqrt{T(P_T+1)})$ bound achieved by Zhang et al. (2018a). Finally, we demonstrate that our Mild-OGD is optimal in the worst case by deriving a matching lower bound.

2. Related Work

In this section, we briefly review related work on OCO with arbitrary delays and the dynamic regret.

2.1. OCO with Arbitrary Delays

To deal with arbitrary delays, Joulani et al. (2013) first propose a black-box technique, which can extend any non-delayed OCO algorithm into the delayed setting. The main idea is to pool multiple instances of the non-delayed algorithm, each of which runs over a subsequence of rounds that satisfies the non-delayed assumption. Moreover, Joulani et al. (2013) show that if the non-delayed algorithm has a static regret bound of $R(T)$, this technique can attain a static regret bound of $dR(T/d)$. Notice that in the non-delayed setting, there exist plenty of algorithms with an $O(\sqrt{T})$ static regret bound, such as OGD (Zinkevich, 2003). As a result, combining with OGD, this technique can achieve a static regret bound of $O(\sqrt{dT})$. However, despite the generality of this technique, it needs to run multiple instances of the non-delayed algorithm, which could be prohibitively resource-intensive (Quanrud & Khashabi, 2015; Joulani et al., 2016). For this reason, instead of adopting the technique of Joulani et al. (2013), subsequent studies extend many specific non-delayed OCO algorithms into the delayed setting by only running a single instance of them with the delayed information about all loss functions.

Specifically, Quanrud & Khashabi (2015) propose a delayed variant of OGD, and reduce the static regret to $O(\sqrt{\bar{d}T})$, which depends on the average delay $\bar{d}$, instead of the maximum delay d. By additionally assuming that the In-Order property holds, McMahan & Streeter (2014) develop a delayed variant of the adaptive gradient (AdaGrad) algorithm (McMahan & Streeter, 2010; Duchi et al., 2011), and establish a data-dependent static regret bound, which could be tighter than $O(\sqrt{dT})$ for sparse data. Later, Joulani et al. (2016) propose another delayed variant of AdaGrad, which can attain a data-dependent static regret bound without the In-Order property. Recently, Flaspohler et al. (2021) develop delayed variants of optimistic algorithms (Rakhlin & Sridharan, 2013; Joulani et al., 2017), which can make use of "hints" about expected future loss functions to improve the $O(\sqrt{dT})$ static regret. Wan et al. (2022a) extend the delayed variant of OGD (Quanrud & Khashabi, 2015) to further exploit the strong convexity of functions. Wan et al. (2022b; 2023a) develop a delayed variant of online Frank-Wolfe (Hazan & Kale, 2012), and obtain a static regret bound of $O(T^{3/4} + \bar{d}T^{1/4})$. Their algorithm is projection-free and can be efficiently implemented over complex constraints. We also notice that Korotin et al. (2020) consider the problem of prediction with expert advice—a special case of OCO with linear functions and simplex decision sets, and propose a delayed variant of Hedge (Freund & Schapire, 1997) to achieve the $O(\sqrt{\bar{d}T})$ static regret.

2.2. Dynamic Regret

Dynamic regret of OCO is first introduced by Zinkevich (2003), who demonstrates that OGD can attain a dynamic regret bound of $O(\sqrt{T}(P_T+1))$ by simply utilizing a constant learning rate. Later, Zhang et al. (2018a) establish a lower bound of $\Omega(\sqrt{T(P_T+1)})$ for the dynamic regret. Moreover, to improve the upper bound, Zhang et al. (2018a)

propose a novel algorithm that runs multiple instances of OGD with different learning rates in parallel, and tracks the best one via Hedge (Freund & Schapire, 1997). Although the strategy of maintaining multiple learning rates is originally proposed to adaptively minimize the static regret for multiple types of functions (van Erven & Koolen, 2016; van Erven et al., 2021), Zhang et al. (2018a) extend it to achieve an optimal dynamic regret bound of $O(\sqrt{T(P_T+1)})$. Subsequent studies achieve tighter dynamic regret bounds for special types of data (Cutkosky, 2020) and functions (Zhao et al., 2020; Baby & Wang, 2021; 2022; 2023), and reduce the computational complexity for handling complex constraints (Zhao et al., 2022; Wang et al., 2024). Besides, there also exist plenty of studies (Jadbabaie et al., 2015; Besbes et al., 2015; Yang et al., 2016; Mokhtari et al., 2016; Zhang et al., 2017; 2018b; Baby & Wang, 2019; Wan et al., 2021; 2023b; Zhao & Zhang, 2021; Wang et al., 2021; 2023) that focus on a restricted form of the dynamic regret, in which $\mathbf{u}_t = \mathbf{x}_t^* \in \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} f_t(\mathbf{x})$. However, as discussed by Zhang et al. (2018a), the restricted dynamic regret is too pessimistic and less flexible than the general one.

2.3. Discussions

Although both arbitrary delays and the dynamic regret have attracted much research interest, it is still unclear how arbitrary delays affect the dynamic regret. Recently, Wang et al. (2021; 2023) have demonstrated under a fixed and knowable delay d', simply performing OGD with a delayed gradient $\nabla f_{t-d'+1}(\mathbf{x}_{t-d'+1})$ is able to achieve a restricted dynamic regret bound of $O(\sqrt{d'T(P_T^*+1)})$ when $P_T^* = \sum_{t=2}^{T} \|\mathbf{x}_t^* - \mathbf{x}_{t-1}^*\|_2$ is also knowable.[1] However, their algorithm and theoretical results do not apply to the general dynamic regret under arbitrary delays. Moreover, one may try to extend existing algorithms with dynamic regret bounds into the delayed setting via the black-box technique of Joulani et al. (2013). However, we want to emphasize that they focus on the static regret, and their analysis cannot directly yield a dynamic regret bound. In addition, since their technique does not achieve the $O(\sqrt{\bar{d}T})$ static regret, it seems also unable to achieve the $O(\sqrt{\bar{d}T(P_T+1)})$ dynamic regret even under the In-Order assumption.

3. Main Results

In this section, we first introduce necessary assumptions, and then present our DOGD and Mild-OGD. Finally, we provide a matching lower bound to demonstrate the optimality of our Mild-OGD in the worst case.

[1]Note that Wang et al. (2021; 2023) aim to handle a special decision set with long-term constraints, and thus their algorithm is more complicated than OGD with the delayed gradient. Here, we omit other details of their algorithm because such a decision set is beyond the scope of this paper.

3.1. Assumptions

Assumption 3.1. The gradients of all functions are bounded by G, i.e., $\|\nabla f_t(\mathbf{x})\|_2 \leq G$ for any $\mathbf{x} \in \mathcal{K}$ and $t \in [T]$.

Assumption 3.2. The decision set $\mathcal{K}$ contains the origin $\mathbf{0}$, and its diameter is bounded by D, i.e., $\|\mathbf{x} - \mathbf{y}\|_2 \leq D$ for any $\mathbf{x}, \mathbf{y} \in \mathcal{K}$.

Assumption 3.3. Delays do not change the arrival order of gradients, i.e., the gradient $\nabla f_i(\mathbf{x}_i)$ is received before the gradient $\nabla f_j(\mathbf{x}_j)$, for any $1 \leq i < j \leq T$.

Remark: The first two assumptions have been commonly utilized in previous studies on OCO (Shalev-Shwartz, 2011; Hazan, 2016). To further justify the rationality of Assumption 3.3, we notice that parallel and distributed optimization (McMahan & Streeter, 2014; Zhou et al., 2018) is also a representative application of delayed OCO. For parallel optimization with many threads, the delay is mainly caused by the computing time of gradients. Thus, as in McMahan & Streeter (2014), it is reasonable to assume that these delays satisfy the In-Order assumption, because the gradient computed first is more likely to be obtained first. Even for general parallel and distributed optimization, polynomially growing delays, which imply $d_i \leq d_j$ for $i < j$ and thus satisfy the In-Order assumption, have received much attention in recent years (Zhou et al., 2018; Ren et al., 2020; Zhou et al., 2022). Moreover, we want to emphasize that Assumption 3.3 is only utilized to achieve the dynamic regret bound depending on the average delay $\bar{d}$, and the case without this assumption is also considered.

3.2. DOGD with Dynamic Regret

In the following, we first introduce detailed procedures of DOGD, and then present its theoretical guarantees.

3.2.1. Detailed Procedures

Recall that in the non-delayed setting, the classical OGD algorithm (Zinkevich, 2003) at each round t updates the decision as

$$\mathbf{x}_{t+1} = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{K}} \|\mathbf{x} - (\mathbf{x}_t - \eta\nabla f_t(\mathbf{x}_t))\|_2^2 \tag{1}$$

where η is a learning rate. To handle the setting with arbitrary delays, Quanrud & Khashabi (2015) have proposed a delayed variant of OGD by replacing $\nabla f_t(\mathbf{x}_t)$ with the sum of gradients received in round t. However, it ignores the arrival order of gradients, and thus cannot benefit from the In-Order property when minimizing the dynamic regret.

To address this limitation, we propose a new delayed variant of OGD, which performs a gradient descent step for each delayed gradient according to their arrival order. Specifically, our algorithm is named as delayed online gradient descent (DOGD) and outlined in Algorithm 1, where τ records the

Algorithm 1 DOGD

1: **Input:** a learning rate η
2: **Initialization:** set $\mathbf{y}_1 = \mathbf{0}$ and $\tau = 1$
3: **for** $t = 1, \ldots, T$ **do**
4: Play $\mathbf{x}_t = \mathbf{y}_\tau$ and query $\nabla f_t(\mathbf{x}_t)$
5: Receive $\{\nabla f_k(\mathbf{x}_k) | k \in \mathcal{F}_t\}$
6: **for** $k \in \mathcal{F}_t$ (in the ascending order) **do**
7: Compute $\mathbf{y}_{\tau+1}$ as in (2) and set $\tau = \tau + 1$
8: **end for**
9: **end for**

number of generated decisions and $\mathbf{y}_\tau$ denotes the τ-th generated decision. Initially, we set $\mathbf{y}_1 = \mathbf{0}$ and $\tau = 1$. At each round $t \in [T]$, we first play the latest decision $\mathbf{x}_t = \mathbf{y}_\tau$ and query the gradient $\nabla f_t(\mathbf{x}_t)$.

After that, due to the effect of arbitrary delays, we receive a set of delayed gradients $\{\nabla f_k(\mathbf{x}_k) | k \in \mathcal{F}_t\}$, where

$$\mathcal{F}_t = \{k \in [T] | k + d_k - 1 = t\}.$$

For each $k \in \mathcal{F}_t$, inspired by (1), we perform the following update

$$\mathbf{y}_{\tau+1} = \operatorname*{argmin}_{\mathbf{x} \in \mathcal{K}} \|\mathbf{x} - (\mathbf{y}_\tau - \eta \nabla f_k(\mathbf{x}_k))\|_2^2 \tag{2}$$

and then set $\tau = \tau + 1$. Moreover, to utilize the In-Order property, elements in the set $\mathcal{F}_t$ are sorted and traversed in the ascending order.

3.2.2. Theoretical Guarantees

We notice that due to the effect of delays, there could exist some gradients that arrive after round T. Although our DOGD does not need to utilize these gradients, they are useful to facilitate our analysis and discussion. Therefore, in the analysis of DOGD, we virtually set $\mathbf{x}_t = \mathbf{y}_\tau$ and perform steps 5 to 8 in Algorithm 1 at some additional rounds $t = T+1, \ldots, T+d-1$. In this way, all queried gradients are utilized to generate decisions

$$\mathbf{y}_1, \ldots, \mathbf{y}_{T+1}.$$

Moreover, we denote the time-stamp of the τ-th utilized gradient by c_τ. To help understanding, one can imagine that DOGD also sets $c_\tau = k$ at the beginning of its step 7.

Then, we establish the following theorem with only Assumptions 3.1 and 3.2.

Theorem 3.4. *Under Assumptions 3.1 and 3.2, for any comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{K}$, Algorithm 1 ensures*

$$\begin{aligned} & R(\mathbf{u}_1, \ldots, \mathbf{u}_T) \\ \leq & \frac{D^2 + DP_T}{\eta} + \eta G^2 \sum_{t=1}^{T} m_t + \sum_{t=1}^{T} G \|\mathbf{u}_t - \mathbf{u}_{c_t}\|_2 \end{aligned} \tag{3}$$

where $m_t = t - \sum_{i=1}^{t-1} |\mathcal{F}_i|$.

Remark: The value of $m_t - 1$ actually counts the number of gradients that have been queried, but still not received at the end of round $t-1$. Since the gradient $\nabla f_t(\mathbf{x}_t)$ will only be counted as an unreceived gradient in $d_t - 1$ rounds, it is easy to verify that

$$\sum_{t=1}^{T} m_t \leq \sum_{t=1}^{T} d_t = \bar{d}T. \tag{4}$$

Therefore, the first two terms in the right side of (3) are upper bounded by $(2D + P_T)G\sqrt{\bar{d}T}$ so long as

$$\eta = \frac{D}{G\sqrt{\sum_{t=1}^{T} m_t}}. \tag{5}$$

However, we still need to bound the last term in the right side of (3), which reflects the "comparator drift" caused by arbitrary delays, and has never appeared in previous studies on the delayed feedback and dynamic regret.

To this end, we establish the following lemma regarding the comparator drift.

Lemma 3.5. *Under Assumption 3.2, for any comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{K}$, Algorithm 1 ensures*

$$\sum_{t=1}^{T} \|\mathbf{u}_t - \mathbf{u}_{c_t}\|_2 \leq \min\{KD, 2dP_T\} \leq \sqrt{2dKDP_T}$$

where $K = \sum_{t=1}^{T} \mathbb{I}(t \neq c_t)$ and $\mathbb{I}(\cdot)$ denotes the indicator function.

Remark: Since Algorithm 1 utilizes the received gradients in the ascending order, the value of K counts the number of delays that are not in order. Therefore, Lemma 3.5 implies that the comparator drift can be upper bounded by $O(\sqrt{dTP_T})$ in the worst case because of $K \leq T$, and vanishes if the In-Order property holds, i.e., $K = 0$. To facilitate discussions, we mainly focus on these two extremes, though the comparator drift can be bounded by $O(\sqrt{\bar{d}TP_T})$ in an intermediate case with $K \leq O(T\bar{d}/d)$.

By further combining Theorem 3.4 with (4) and Lemma 3.5, we derive the following corollary.

Corollary 3.6. *Under Assumptions 3.1 and 3.2, by setting η as in (5), Algorithm 1 ensures*

$$R(\mathbf{u}_1, \ldots, \mathbf{u}_T) \leq (2D + P_T)G\sqrt{\bar{d}T} + C$$

for any comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{K}$, where

$$C = \begin{cases} 0, & \text{if Assumption 3.3 also holds;} \\ \min\{TGD, 2dGP_T\}, & \text{otherwise.} \end{cases} \tag{6}$$

Remark: From Corollary 3.6, our DOGD enjoys a dynamic regret bound of $O(\sqrt{\bar{d}T}(P_T + 1) + C)$, which is adaptive

to the upper bound of comparator drift. First, because of $\min\{TGD, 2dGP_T\} \leq G\sqrt{2dTDP_T}$ and $\bar{d} \leq d$, the dynamic regret of DOGD can be bounded by $O(\sqrt{dT}(P_T+1))$ in the worst case, which magnifies the $O(\sqrt{T}(P_T+1))$ dynamic regret of OGD (Zinkevich, 2003) in the non-delayed setting by a coefficient depending on the maximum delay d. Second, in case $C \leq O(\sqrt{\bar{d}T}P_T)$, the dynamic regret of DOGD automatically reduces to $O(\sqrt{\bar{d}T}(P_T+1))$, which depends on the average delay. According to (6), this condition can be simply satisfied for all possible P_T when the In-Order property holds or $d \leq \sqrt{\bar{d}T}$. Third, by substituting $\mathbf{u}_1 = \cdots = \mathbf{u}_T$ into Corollary 3.6, we find that DOGD can attain a static regret bound of $O(\sqrt{\bar{d}T})$ for arbitrary delays, which matches the best existing result (Quanrud & Khashabi, 2015).

Remark: At first glance, Corollary 3.6 needs to set the learning rate as in (5), which may become a limitation of DOGD, because the value of $\sum_{t=1}^{T} m_t$ is generally unknown in practice. However, we note that Quanrud & Khashabi (2015) also face this issue when minimizing the static regret of OCO with arbitrary delays, and have introduced a simple solution by utilizing the standard "doubling trick" (Cesa-Bianchi et al., 1997) to adaptively adjust the learning rate. The main insight behind this solution is that the value of $\sum_{t=1}^{T} m_t$ can be calculated on the fly. The details about DOGD with the doubling trick are provided in the appendix.

3.3. Mild-OGD with Improved Dynamic Regret

One unsatisfactory point of DOGD is that the dynamic regret linearly depends on the path-length. Notice that if only a specific path-length P_T is considered, from Theorem 3.4, we can tune the learning rate as

$$\eta_* = \frac{\sqrt{D(D+P_T)}}{G\sqrt{\sum_{t=1}^{T} m_t}}$$

and obtain the dynamic regret sublinear to P_T. However, our goal is to minimize the dynamic regret with respect to any possible path-length P_T. To address this dilemma, inspired by Zhang et al. (2018a), we develop an algorithm that runs multiple DOGD as experts, each with a different learning rate for a specific path-length, and combines them with a meta-algorithm.

It is worth noting that the meta-algorithm of Zhang et al. (2018a) is incompatible to the delayed setting studied here. To this end, we adopt the delayed Hedge (Korotin et al., 2020), an expert-tracking method under arbitrary delays, to design our meta-algorithm. Moreover, there exist two options for the meta-algorithm to maintain these expert-algorithms: running them over the original functions $\{f_t(\mathbf{x})\}_{t\in[T]}$ or the surrogate functions $\{\ell_t(\mathbf{x})\}_{t\in[T]}$, where

$$\ell_t(\mathbf{x}) = \langle \nabla f_t(\mathbf{x}_t), \mathbf{x} - \mathbf{x}_t \rangle \tag{7}$$

Algorithm 2 Mild-OGD: Meta-algorithm

Input: a parameter α and a set $\mathcal{H}$ containing learning rates for experts
Activate a set of experts $\{E^\eta | \eta \in \mathcal{H}\}$ by invoking the expert-algorithm for each learning rate $\eta \in \mathcal{H}$
Sort learning rates in the ascending order, i.e., $\eta_1 \leq \cdots \leq \eta_{|\mathcal{H}|}$, and set $w_1^{\eta_i} = \frac{|\mathcal{H}|+1}{i(i+1)|\mathcal{H}|}$
for $t = 1, \ldots, T$ **do**
  Receive $\mathbf{x}_t^\eta$ from each expert E^η
  Play the decision $\mathbf{x}_t = \sum_{\eta\in\mathcal{H}} w_t^\eta \mathbf{x}_t^\eta$
  Query $\nabla f_t(\mathbf{x}_t)$ and receive $\{\nabla f_k(\mathbf{x}_k) | k \in \mathcal{F}_t\}$
  Update the weight of each expert as in (8)
  Send $\{\nabla f_k(\mathbf{x}_k) | k \in \mathcal{F}_t\}$ to each expert E^η
end for

and $\mathbf{x}_t$ is the decision of the meta-algorithm. In this paper, we choose the second option, because the surrogate functions allow expert-algorithms to reuse the gradient of the meta-algorithm, and thus can avoid inconsistent delays between the meta-algorithm and expert-algorithms. Specifically, our algorithm is named as multiple delayed online gradient descent (Mild-OGD), and stated below.

Meta-algorithm Let $\mathcal{H}$ denote a set of learning rates for experts. We first activate a set of experts $\{E^\eta | \eta \in \mathcal{H}\}$ by invoking the expert-algorithm for each learning rate $\eta \in \mathcal{H}$. Let η_i be the i-th smallest learning rate in $\mathcal{H}$. Following Zhang et al. (2018a), the initial weight of each expert E^{η_i} is set as

$$w_1^{\eta_i} = \frac{|\mathcal{H}|+1}{i(i+1)|\mathcal{H}|}.$$

In each round $t \in [T]$, our meta-algorithm receives a decision $\mathbf{x}_t^\eta$ from each expert E^η, and then plays the weighted decision as

$$\mathbf{x}_t = \sum_{\eta\in\mathcal{H}} w_t^\eta \mathbf{x}_t^\eta.$$

After that, it queries the gradient $\nabla f_t(\mathbf{x}_t)$, but only receives $\{\nabla f_k(\mathbf{x}_k) | k \in \mathcal{F}_t\}$ due to the effect of arbitrary delays. Then, according to the delayed Hedge (Korotin et al., 2020), we update the weight of each expert as

$$w_{t+1}^\eta = \frac{w_t^\eta e^{-\alpha \sum_{k\in\mathcal{F}_t} \ell_k(\mathbf{x}_k^\eta)}}{\sum_{\mu\in\mathcal{H}} w_t^\mu e^{-\alpha \sum_{k\in\mathcal{F}_t} \ell_k(\mathbf{x}_k^\mu)}} \tag{8}$$

where α is a parameter and $\ell_k(\mathbf{x})$ is defined in (7). This is the critical difference between our meta-algorithm and that in Zhang et al. (2018a), which updates the weight according to the vanilla Hedge (Cesa-Bianchi et al., 1997).

Finally, we send gradients $\{\nabla f_k(\mathbf{x}_k) | k \in \mathcal{F}_t\}$ to each expert E^η so that they can update their own decisions without querying additional gradients. The detailed procedures of our meta-algorithm are summarized in Algorithm 2.

Algorithm 3 Mild-OGD: Expert-algorithm

Input: a learning rate η
Initialization: set $\mathbf{y}_1^\eta = \mathbf{0}$ and $\tau = 1$
for $t = 1, \ldots, T$ **do**
  Submit $\mathbf{x}_t^\eta = \mathbf{y}_\tau^\eta$ to the meta-algorithm
  Receive gradients $\{\nabla f_k(\mathbf{x}_k) | k \in \mathcal{F}_t\}$ from the meta-algorithm
  for $k \in \mathcal{F}_t$ (in the ascending order) **do**
    Compute $\mathbf{y}_{\tau+1}^\eta$ as in (9) and set $\tau = \tau + 1$
  end for
end for

Expert-algorithm The expert-algorithm is instantiated by running DOGD over the surrogate loss function defined in (7), instead of the real loss function. To emphasize this difference, we present its procedures in Algorithm 3. The input and initialization are the same as those in DOGD. At each round $t \in [T]$, the expert-algorithm first submits the decision $\mathbf{x}_t^\eta = \mathbf{y}_\tau^\eta$ to the meta-algorithm, and then receives gradients $\{\nabla f_k(\mathbf{x}_k) | k \in \mathcal{F}_t\}$ from the meta-algorithm. For each $k \in \mathcal{F}_t$, it updates the decision as

$$\mathbf{y}_{\tau+1}^\eta = \operatorname*{argmin}_{\mathbf{x} \in \mathcal{K}} \|\mathbf{x} - (\mathbf{y}_\tau^\eta - \eta \nabla f_k(\mathbf{x}_k))\|_2^2 \tag{9}$$

and sets $\tau = \tau + 1$.

We have the following theoretical guarantee for the dynamic regret of Mild-OGD.

Theorem 3.7. *Let $m_t = t - \sum_{i=1}^{t-1} |\mathcal{F}_i|$. Under Assumptions 3.1 and 3.2, by setting*

$$\mathcal{H} = \left\{ \eta_i = \frac{2^{i-1} D}{G\sqrt{\beta}} \middle| i = 1, \ldots, N \right\} \text{ and } \alpha = \frac{1}{GD\sqrt{\beta}}$$

where $N = \left\lceil \frac{1}{2} \log_2(T+1) \right\rceil + 1$ and $\beta = \sum_{t=1}^T m_t$, Algorithm 2 ensures

$$\begin{aligned} R(\mathbf{u}_1, \ldots, \mathbf{u}_T) \leq & (3\sqrt{D(D+P_T)} + D) G \sqrt{\bar{d}T} \\ & + C + 2GD\sqrt{\bar{d}T} \ln(k+1) \\ = & O\left(\sqrt{\bar{d}T(P_T+1)} + C\right) \end{aligned}$$

for any comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{K}$, where

$$k = \left\lfloor \log_2 \sqrt{(P_T + D)/D} \right\rfloor + 1$$

and C is defined in (6).

Remark: Theorem 3.7 shows that Mild-OGD can attain a dynamic regret bound of $O(\sqrt{\bar{d}T(P_T+1)} + C)$, which is also adaptive to the upper bound of comparator drift. It is easy to verify that this dynamic regret bound becomes $O(\sqrt{dT(P_T+1)})$ in the worst case. Moreover, it reduces to $O(\sqrt{\bar{d}T(P_T+1)})$ in case $C \leq O(\sqrt{\bar{d}TP_T})$, which can be satisfied for all possible P_T when the In-order property holds or for $P_T \leq \bar{d}T/d^2$. Compared with the dynamic regret of DOGD, Mild-OGD reduces the linear dependence on P_T to be sublinear. Moreover, compared with the optimal $O(\sqrt{T(P_T+1)})$ bound achieved in the non-delayed setting (Zhang et al., 2018a), Mild-OGD magnifies it by a coefficient depending on delays. We also notice that although Theorem 3.7 requires the value of $\sum_{t=1}^T m_t$ to tune parameters, as previously discussed, this requirement can be removed by utilizing the doubling trick. The details about Mild-OGD with the doubling trick are provided in the appendix.

3.4. Lower Bound

Finally, we show that our Mild-OGD is optimal in the worst case by establishing the following lower bound.

Theorem 3.8. *Let $L = \lceil TD/\max\{P, D\} \rceil$. Suppose $\mathcal{K} = [-D/(2\sqrt{n}), D/(2\sqrt{n})]^n$ which satisfies Assumption 3.2. For any OCO algorithm, any $P \in [0, TD]$, and any positive integer d, there exists a sequence of comparators $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{K}$ satisfying $P_T \leq P$, a sequence of functions $f_1(\mathbf{x}), \ldots, f_T(\mathbf{x})$ satisfying Assumption 3.1, and a sequence of delays $1 \leq d_1, \ldots, d_T \leq d$ such that*

$$R(\mathbf{u}_1, \ldots, \mathbf{u}_T) \geq \begin{cases} \dfrac{DGT}{2\sqrt{2}}, & \text{if } d > L; \\ \dfrac{G\sqrt{dD\max\{P, D\}T}}{4\sqrt{2}}, & \text{otherwise.} \end{cases}$$

Remark: From Theorem 3.8, if $d > L = \Omega(T/(P_T+1))$, there exists an $\Omega(T)$ lower bound on the dynamic regret, which can be trivially matched by any OCO algorithm including our Algorithm 2. As a result, we mainly focus on the case $d \leq L$, and notice that Theorem 3.8 essentially establishes an $\Omega(\sqrt{dT(P_T+1)})$ lower bound, which matches the $O(\sqrt{dT(P_T+1)})$ dynamic regret of our Mild-OGD in the worst case. To the best of our knowledge, this is the first lower bound for the dynamic regret of the delayed OCO.

4. Analysis

In this section, we prove Theorem 3.4, Lemma 3.5, Theorem 3.7, and Theorem 3.8 by introducing some lemmas. The omitted proofs can be found in the appendix.

4.1. Proof of Theorem 3.4

It is easy to verify that

$$R(\mathbf{u}_1, \ldots, \mathbf{u}_T) \leq \sum_{t=1}^T \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{u}_t \rangle$$

where the inequality is due to the convexity of functions.

Let $\tau_t = 1 + \sum_{i=1}^{t-1} |\mathcal{F}_i|$. Then, combining the above inequality with the fact that $c_1, \ldots, c_T$ is a permutation of $1, \ldots, T$, we have

$$\begin{aligned}
&R(\mathbf{u}_1, \ldots, \mathbf{u}_T) \\
\leq &\sum_{t=1}^{T} \langle \nabla f_{c_t}(\mathbf{x}_{c_t}), \mathbf{x}_{c_t} - \mathbf{u}_{c_t} \rangle \\
= &\sum_{t=1}^{T} \left\langle \nabla f_{c_t}(\mathbf{x}_{c_t}), \mathbf{y}_{\tau_{c_t}} - \mathbf{u}_{c_t} \right\rangle \\
= &\sum_{t=1}^{T} \left\langle \nabla f_{c_t}(\mathbf{x}_{c_t}), \mathbf{y}_t - \mathbf{u}_t + \mathbf{y}_{\tau_{c_t}} - \mathbf{y}_t \right\rangle \\
&+ \sum_{t=1}^{T} \langle \nabla f_{c_t}(\mathbf{x}_{c_t}), \mathbf{u}_t - \mathbf{u}_{c_t} \rangle \\
\leq &\sum_{t=1}^{T} \langle \nabla f_{c_t}(\mathbf{x}_{c_t}), \mathbf{y}_t - \mathbf{u}_t \rangle + \sum_{t=1}^{T} G \left\| \mathbf{y}_{\tau_{c_t}} - \mathbf{y}_t \right\|_2 \\
&+ \sum_{t=1}^{T} G \|\mathbf{u}_t - \mathbf{u}_{c_t}\|_2
\end{aligned} \tag{10}$$

where the first equality is due to $\mathbf{x}_t = \mathbf{y}_{\tau_t}$ in Algorithm 1, and the last inequality is due to Assumption 3.1.

Let $\mathbf{y}'_{t+1} = \mathbf{y}_t - \eta \nabla f_{c_t}(\mathbf{x}_{c_t})$. For the first term in the right side of (10), we have

$$\begin{aligned}
&\sum_{t=1}^{T} \langle \nabla f_{c_t}(\mathbf{x}_{c_t}), \mathbf{y}_t - \mathbf{u}_t \rangle = \sum_{t=1}^{T} \frac{\left\langle \mathbf{y}_t - \mathbf{y}'_{t+1}, \mathbf{y}_t - \mathbf{u}_t \right\rangle}{\eta} \\
= &\sum_{t=1}^{T} \frac{\left(\|\mathbf{y}_t - \mathbf{u}_t\|_2^2 - \|\mathbf{y}'_{t+1} - \mathbf{u}_t\|_2^2 + \|\mathbf{y}_t - \mathbf{y}'_{t+1}\|_2^2 \right)}{2\eta} \\
= &\sum_{t=1}^{T} \frac{\left(\|\mathbf{y}_t - \mathbf{u}_t\|_2^2 - \|\mathbf{y}'_{t+1} - \mathbf{u}_t\|_2^2 + \|\eta \nabla f_{c_t}(\mathbf{x}_{c_t})\|_2^2 \right)}{2\eta} \\
\leq &\sum_{t=1}^{T} \frac{1}{2\eta} \left(\|\mathbf{y}_t - \mathbf{u}_t\|_2^2 - \|\mathbf{y}_{t+1} - \mathbf{u}_t\|_2^2 + \eta^2 G^2 \right) \\
= &\sum_{t=1}^{T} \left(\frac{\left(\|\mathbf{y}_t\|_2^2 - \|\mathbf{y}_{t+1}\|_2^2 \right)}{2\eta} + \frac{\langle \mathbf{y}_{t+1} - \mathbf{y}_t, \mathbf{u}_t \rangle}{\eta} + \frac{\eta G^2}{2} \right) \\
\leq &\frac{1}{\eta} \langle \mathbf{y}_{T+1}, \mathbf{u}_T \rangle + \sum_{t=2}^{T} \frac{1}{\eta} \langle \mathbf{u}_{t-1} - \mathbf{u}_t, \mathbf{y}_t \rangle + \frac{\eta T G^2}{2} \\
\leq &\frac{1}{\eta} \|\mathbf{y}_{T+1}\|_2 \|\mathbf{u}_T\|_2 + \sum_{t=2}^{T} \frac{1}{\eta} \|\mathbf{u}_{t-1} - \mathbf{u}_t\|_2 \|\mathbf{y}_t\|_2 + \frac{\eta T G^2}{2} \\
\leq &\frac{D^2 + D P_T}{\eta} + \frac{\eta T G^2}{2}
\end{aligned} \tag{11}$$

where the first inequality is due to Assumption 3.1, the second inequality is due to $\mathbf{y}_1 = \mathbf{0}$ and $\|\mathbf{y}_{T+1}\|_2^2 \geq 0$, and the last inequality is due to Assumption 3.2.

Then, we proceed to bound the second term in the right side of (10). Note that before round c_t, Algorithm 1 has received $\tau_{c_t} - 1$ gradients, and thus has generated $\mathbf{y}_1, \ldots, \mathbf{y}_{\tau_{c_t}}$. Moreover, let $q = c_t + d_{c_t} - 1$. It is easy to verify that $q \geq c_t$, and thus Algorithm 1 has also generated $\mathbf{y}_1, \ldots, \mathbf{y}_{\tau_{c_t}}$ before round q. Since the gradient $\nabla f_{c_t}(\mathbf{x}_{c_t})$ is used to update $\mathbf{y}_t$ in round q, we have

$$\tau_{c_t} \leq t. \tag{12}$$

From (12), we have

$$\begin{aligned}
\sum_{t=1}^{T} \left\| \mathbf{y}_{\tau_{c_t}} - \mathbf{y}_t \right\|_2 \leq &\sum_{t=1}^{T} \sum_{k=\tau_{c_t}}^{t-1} \|\mathbf{y}_k - \mathbf{y}_{k+1}\|_2 \\
\leq &\sum_{t=1}^{T} \sum_{k=\tau_{c_t}}^{t-1} \left\| \mathbf{y}_k - \mathbf{y}'_{k+1} \right\|_2 \\
\leq &\sum_{t=1}^{T} \sum_{k=\tau_{c_t}}^{t-1} \|\eta \nabla f_{c_k}(\mathbf{x}_{c_k})\|_2 \\
\leq &\eta G \sum_{t=1}^{T} (t - \tau_{c_t})
\end{aligned} \tag{13}$$

where the last inequality is due to Assumption 3.1.

Moreover, because of the definitions of τ_t and m_t, we have

$$\begin{aligned}
\sum_{t=1}^{T} (t - \tau_{c_t}) = &\sum_{t=1}^{T} (t-1) - \sum_{t=1}^{T} \sum_{i=1}^{c_t - 1} |\mathcal{F}_i| \\
= &\sum_{t=1}^{T} (t-1) - \sum_{t=1}^{T} \sum_{i=1}^{t-1} |\mathcal{F}_i| \\
= &\sum_{t=1}^{T} \left(t - 1 - \sum_{i=1}^{t-1} |\mathcal{F}_i| \right) \\
= &\sum_{t=1}^{T} (m_t - 1)
\end{aligned} \tag{14}$$

where the second equality is due to the fact that $c_1, \ldots, c_T$ is a permutation of $1, \ldots, T$.

Then, combining (13) with (14), we have

$$\sum_{t=1}^{T} G \left\| \mathbf{y}_{\tau_{c_t}} - \mathbf{y}_t \right\|_2 \leq \eta G^2 \sum_{t=1}^{T} (m_t - 1). \tag{15}$$

Finally, combining (10) with (11) and (15), we have

$$\begin{aligned}
&\sum_{t=1}^{T} (f_t(\mathbf{x}_t) - f_t(\mathbf{u}_t)) \\
\leq &\frac{D^2 + D P_T}{\eta} + \eta G^2 \sum_{t=1}^{T} m_t + \sum_{t=1}^{T} G \|\mathbf{u}_t - \mathbf{u}_{c_t}\|_2
\end{aligned}$$

which completes this proof.

4.2. Proof of Lemma 3.5

Since $\nabla f_{c_t}(\mathbf{x}_{c_t})$ is the t-th used gradient and arrives at the end of round $c_t + d_{c_t} - 1$, it is not hard to verify that

$$t \leq c_t + d_{c_t} - 1 \leq c_t + d - 1 \tag{16}$$

for any $t \in [T]$, and there are at most $t-1$ arrived gradients before round $c_t + d_{c_t} - 1$. Notice that gradients queried at rounds $1, \ldots, t$ must have arrived at the end of round $t+d-1$. Therefore, we also have $c_t + d_{c_t} - 2 < t + d - 1$, which implies that

$$c_t \leq t + d - d_{c_t} \leq t + d - 1. \tag{17}$$

If $t \in [T]$ and $c_t \leq t$, according to (16), we have

$$\begin{aligned} \|\mathbf{u}_t - \mathbf{u}_{c_t}\|_2 &\leq \sum_{k=c_t}^{t-1} \|\mathbf{u}_{k+1} - \mathbf{u}_k\|_2 \\ &\leq \sum_{k=c_t}^{\min\{c_t+d-2, T-1\}} \|\mathbf{u}_{k+1} - \mathbf{u}_k\|_2. \end{aligned} \tag{18}$$

Otherwise, if $t \in [T]$ and $c_t > t$, according to (17), we have

$$\begin{aligned} \|\mathbf{u}_t - \mathbf{u}_{c_t}\|_2 &\leq \sum_{k=t}^{c_t-1} \|\mathbf{u}_{k+1} - \mathbf{u}_k\|_2 \\ &\leq \sum_{k=t}^{\min\{t+d-2, T-1\}} \|\mathbf{u}_{k+1} - \mathbf{u}_k\|_2. \end{aligned} \tag{19}$$

Therefore, combining (18) and (19), we have

$$\begin{aligned} &\sum_{t=1}^{T} \|\mathbf{u}_t - \mathbf{u}_{c_t}\|_2 \\ \leq &\sum_{t=1}^{T} \sum_{k=c_t}^{\min\{c_t+d-2, T-1\}} \|\mathbf{u}_{k+1} - \mathbf{u}_k\|_2 \\ &+ \sum_{t=1}^{T} \sum_{k=t}^{\min\{t+d-2, T-1\}} \|\mathbf{u}_{k+1} - \mathbf{u}_k\|_2 \\ =&2\sum_{t=1}^{T} \sum_{k=t}^{\min\{t+d-2, T-1\}} \|\mathbf{u}_{k+1} - \mathbf{u}_k\|_2 \\ \leq&2\sum_{k=1}^{d} \sum_{t=1}^{T-1} \|\mathbf{u}_{t+1} - \mathbf{u}_t\|_2 = 2dP_T \end{aligned}$$

where the equality is due to the fact that $c_1, \ldots, c_T$ is a permutation of $1, \ldots, T$.

Then, we complete this proof by further noticing that Assumption 3.2 and the definition of K can ensure

$$\begin{aligned} \sum_{t=1}^{T} \|\mathbf{u}_t - \mathbf{u}_{c_t}\|_2 &= \sum_{t=1}^{T} \mathbb{I}(t \neq c_t) \|\mathbf{u}_t - \mathbf{u}_{c_t}\|_2 \\ &\leq \sum_{t=1}^{T} \mathbb{I}(t \neq c_t) D = DK. \end{aligned}$$

4.3. Proof of Theorem 3.7

Let $\eta_* = \sqrt{D(D+P_T)/(\beta G^2)}$, where $\beta = \sum_{t=1}^{T} m_t$. From Assumption 3.2, we have

$$0 \leq P_T = \sum_{t=2}^{T} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2 \leq TD$$

which implies that

$$\eta_1 = \frac{D}{G\sqrt{\beta}} \leq \eta_* \leq \frac{D\sqrt{T+1}}{G\sqrt{\beta}} \leq \eta_{|\mathcal{H}|}.$$

Therefore, for any possible value of P_T, there must exist a learning rate $\eta_k \in \mathcal{H}$ such that

$$\eta_k \leq \eta_* \leq 2\eta_k \tag{20}$$

where $k = \lfloor \log_2 \sqrt{(P_T + D)/D} \rfloor + 1$.

Then, the dynamic regret can be upper bounded as follows

$$\begin{aligned} R(\mathbf{u}_1, \ldots, \mathbf{u}_T) \leq &\sum_{t=1}^{T} \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{u}_t \rangle \\ = &\sum_{t=1}^{T} \ell_t(\mathbf{x}_t) - \sum_{t=1}^{T} \ell_t(\mathbf{x}_t^{\eta_k}) \\ &+ \sum_{t=1}^{T} \ell_t(\mathbf{x}_t^{\eta_k}) - \sum_{t=1}^{T} \ell_t(\mathbf{u}_t). \end{aligned} \tag{21}$$

To bound the first term in the right side of (21), we introduce the following lemma.

Lemma 4.1. *Let $m_t = t - \sum_{i=1}^{t-1} |\mathcal{F}_i|$. Under Assumptions 3.1 and 3.2, for any $\eta \in \mathcal{H}$, Algorithm 2 has*

$$\sum_{t=1}^{T} \ell_t(\mathbf{x}_t) - \sum_{t=1}^{T} \ell_t(\mathbf{x}_t^{\eta}) \leq \frac{1}{\alpha} \ln \frac{1}{w_1^{\eta}} + \alpha G^2 D^2 \sum_{t=1}^{T} m_t.$$

Combining Lemma 4.1 with $(1/w_1^{\eta_k}) \leq (k+1)^2$ and $\alpha = \frac{1}{GD\sqrt{\sum_{t=1}^{T} m_t}}$, under Assumptions 3.1 and 3.2, we have

$$\begin{aligned} &\sum_{t=1}^{T} \ell_t(\mathbf{x}_t) - \sum_{t=1}^{T} \ell_t(\mathbf{x}_t^{\eta_k}) \\ \leq &2GD\sqrt{\sum_{t=1}^{T} m_t} \ln(k+1) + GD\sqrt{\sum_{t=1}^{T} m_t} \\ \leq &2GD\sqrt{\bar{d}T} \ln(k+1) + GD\sqrt{\bar{d}T} \end{aligned}$$

where the last inequality is due to (4).

Note that each expert E^{η} actually is equal to running Algorithm 1 with $\ell_1(\mathbf{x}), \ldots, \ell_T(\mathbf{x})$, where each gradient $\nabla \ell_t(\mathbf{x}_t^{\eta}) = \nabla f_t(\mathbf{x}_t)$ is delayed to the end of round $t + d_t - 1$.

Therefore, combining Theorem 3.4 with Lemma 3.5 and the definition of C in (6), under Assumptions 3.1 and 3.2, we have

$$\begin{aligned}&\sum_{t=1}^{T}\ell_t\left(\mathbf{x}_t^{\eta_k}\right)-\sum_{t=1}^{T}\ell_t\left(\mathbf{u}_t\right)\\ \leq&\frac{D^2+DP_T}{\eta_k}+\eta_k G^2\sum_{t=1}^{T}m_t+C\\ \leq&\frac{2(D^2+DP_T)}{\eta_*}+\eta_* G^2\sum_{t=1}^{T}m_t+C\\ \leq&3G\sqrt{D(D+P_T)}\sqrt{\bar{d}T}+C\end{aligned}$$

where the second inequality is due to (20), and the last inequality is due to the definition of η_* and (4).

Finally, we complete this proof by combining (21) with the above two inequalities.

4.4. Proof of Theorem 3.8

Inspired by the proof of the lower bound in the non-delayed setting (Zhang et al., 2018a), we first need to establish a lower bound of static regret in the delayed setting. Although the seminal work of Weinberger & Ordentlich (2002) has already provided such a lower bound, it only holds in the special case that d divides T. To address this limitation, we establish a lower bound of static regret for any d and T, which is presented in the following lemma.

Lemma 4.2. *Suppose $\mathcal{K}=[-D/(2\sqrt{n}),D/(2\sqrt{n})]^n$ which satisfies Assumption 3.2. For any OCO algorithm and any positive integer d, there exists a sequence of functions $f_1(\mathbf{x}),\ldots,f_T(\mathbf{x})$ satisfying Assumption 3.1 and a sequence of delays $1\leq d_1,\ldots,d_T\leq d$ such that*

$$R(T)\geq\frac{DGT}{2\sqrt{2\lceil T/d\rceil}}.$$

Let $Z=\lceil T/L\rceil$. We then divide the total T rounds into Z blocks, where the length of the first $Z-1$ blocks is L and that of the last block is $T-(Z-1)L$. In this way, we can define the set of rounds in the block z as

$$\mathcal{T}_z=\{(z-1)L+1,\ldots,\min\{zL,T\}\}.$$

Moreover, we define the feasible set of $\mathbf{u}_1,\ldots,\mathbf{u}_T$ as

$$\mathcal{C}(P)=\left\{\mathbf{u}_1,\ldots,\mathbf{u}_T\in\mathcal{K}\,\middle|\,\sum_{t=2}^{T}\|\mathbf{u}_t-\mathbf{u}_{t-1}\|_2\leq P\right\}$$

and construct a subset of $\mathcal{C}(P)$ as

$$\mathcal{C}'(P)=\{\mathbf{u}_1,\ldots,\mathbf{u}_T\in\mathcal{K}\,|\mathbf{u}_i=\mathbf{u}_j,\forall z\in[Z],i,j\in\mathcal{T}_z\}.$$

Notice that the connection $\mathcal{C}'(P)\subseteq\mathcal{C}(P)$ is derived by the fact that the comparator sequence in $\mathcal{C}'(P)$ only changes $Z-1\leq P/D$ times, and thus its path-length does not exceed P.

Then, because of $\mathcal{C}'(P)\subseteq\mathcal{C}(P)$ and Lemma 4.2, there exists a sequence of functions $f_1(\mathbf{x}),\ldots,f_T(\mathbf{x})$ satisfying Assumption 3.1 and a sequence of delays $1\leq d_1,\ldots,d_T\leq d$ such that

$$\begin{aligned}&\sum_{t=1}^{T}f_t(\mathbf{x}_t)-\min_{\mathbf{u}_1,\ldots,\mathbf{u}_T\in\mathcal{C}(P)}\sum_{t=1}^{T}f_t(\mathbf{u}_t)\\ \geq&\sum_{t=1}^{T}f_t(\mathbf{x}_t)-\min_{\mathbf{u}_1,\ldots,\mathbf{u}_T\in\mathcal{C}'(P)}\sum_{t=1}^{T}f_t(\mathbf{u}_t)\\ =&\sum_{z=1}^{Z}\left(\sum_{t\in\mathcal{T}_z}f_t(\mathbf{x}_t)-\min_{\mathbf{x}\in\mathcal{K}}\sum_{t\in\mathcal{T}_z}f_t(\mathbf{x})\right)\\ \geq&\sum_{z=1}^{Z}\frac{DG|\mathcal{T}_z|}{2\sqrt{2\lceil|\mathcal{T}_z|/d\rceil}}.\end{aligned}$$

Finally, we can complete this proof by further noticing that

$$\begin{aligned}\sum_{z=1}^{Z}\frac{DG|\mathcal{T}_z|}{2\sqrt{2\lceil|\mathcal{T}_z|/d\rceil}}\geq&\sum_{z=1}^{Z}\frac{DG|\mathcal{T}_z|}{2\sqrt{2\lceil L/d\rceil}}=\frac{DGT}{2\sqrt{2\lceil L/d\rceil}}\\ \geq&\begin{cases}\dfrac{DGT}{2\sqrt{2}},\ \text{if } d>L;\\ \dfrac{G\sqrt{dD\max\{P,D\}T}}{4\sqrt{2}},\ \text{otherwise;}\end{cases}\end{aligned}$$

where the first inequality is due to $|\mathcal{T}_z|\leq L$ for any $z\in[Z]$, and the last inequality is mainly due to

$$\begin{aligned}\lceil L/d\rceil\leq&2L/d=2\lceil TD/\max\{P,D\}\rceil/d\\ \leq&4TD/(\max\{P,D\}d)\end{aligned}$$

for $d\leq L$.

5. Conclusion and Future Work

In this paper, we study the dynamic regret of OCO with arbitrary delays. To this end, we first propose a simple algorithm called DOGD, the dynamic regret of which can be automatically bounded by $O(\sqrt{\bar{d}T}(P_T+1))$ under mild assumptions such as the In-Order property, and $O(\sqrt{dT}(P_T+1))$ in the worst case. Furthermore, based on DOGD, we develop an improved algorithm called Mild-OGD, which can automatically enjoy an $O(\sqrt{\bar{d}T(P_T+1)})$ dynamic regret bound under mild assumptions such as the In-Order property, and an $O(\sqrt{dT(P_T+1)})$ dynamic regret bound in the worst case. Finally, we provide a matching lower bound to show the optimality of our Mild-OGD in the worst case.

It is worth noting that there still are several directions for future research, which are discussed in the appendix due to the limitation of space.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (62306275, 62122037), the Zhejiang Province High-Level Talents Special Support Program "Leading Talent of Technological Innovation of Ten-Thousands Talents Program" (No. 2022R52046), the Key Research and Development Program of Zhejiang Province (No. 2023C03192), and the Open Research Fund of the State Key Laboratory of Blockchain and Data Security, Zhejiang University. The authors would also like to thank the anonymous reviewers for their helpful comments.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are some potential societal consequences of our work, but we feel that none of them must be specifically highlighted here.

References

Abernethy, J. D., Bartlett, P. L., Rakhlin, A., and Tewari, A. Optimal stragies and minimax lower bounds for online convex games. In *Proceedings of the 21st Annual Conference on Learning Theory*, pp. 415–424, 2008.

Baby, D. and Wang, Y.-X. Online forecasting of total-variation-bounded sequences. In *Advances in Neural Information Processing Systems 32*, pp. 11071–11081, 2019.

Baby, D. and Wang, Y.-X. Optimal dynamic regret in exp-concave online learning. In *Proceedings of 34th Annual Conference on Learning Theory*, pp. 359–409, 2021.

Baby, D. and Wang, Y.-X. 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*, pp. 1805–1845, 2022.

Baby, D. and Wang, Y.-X. Second order path variationals in non-stationary online learning. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pp. 9024–9075, 2023.

Besbes, O., Gur, Y., and Zeevi, A. Non-stationary stochastic optimization. *Operations Research*, 63(5):1227–1244, 2015.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D. P., Schapire, R. E., and Warmuth, M. K. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.

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

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

Flaspohler, G. E., Orabona, F., Cohen, J., Mouatadid, S., Oprescu, M., Orenstein, P., and Mackey, L. Online learning with optimism and delay. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 3363–3373, 2021.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

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

Hazan, E. and Kale, S. Projection-free online learning. In *Proceedings of the 29th International Conference on Machine Learning*, pp. 1843–1850, 2012.

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

He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., Shi, Y., Atallah, A., Herbrich, R., Bowers, S., and Candela, J. Q. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the 8th International Workshop on Data Mining for Online Advertising*, pp. 1–9, 2014.

Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Jadbabaie, A., Rakhlin, A., Shahrampour, S., and Sridharan, K. Online optimization: Competing with dynamic comparators. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, pp. 398–406, 2015.

Joulani, P., György, A., and Szepesvári, C. Online learning under delayed feedback. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 1453–1461, 2013.

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

Joulani, P., György, A., and Szepesvári, C. A modular analysis of adaptive (non-)convex optimization: Optimism, composite objectives, and variational bounds. In *Proceedings of the 28th International Conference on Algorithmic Learning Theory*, pp. 681–720, 2017.

Korotin, A., V'yugin, V., and Burnaev, E. Adaptive hedging under delayed feedback. *Neurocomputing*, 397:356–368, 2020.

McMahan, H. B. and Streeter, M. Adaptive bound optimization for online convex optimization. In *Proceedings of the 23rd Conference on Learning Theory*, pp. 244–256, 2010.

McMahan, H. B. and Streeter, M. Delay-tolerant algorithms for asynchronous distributed online learning. In *Advances in Neural Information Processing Systems 27*, pp. 2915–2923, 2014.

McMahan, H. B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., Nie, L., Phillips, T., Davydov, E., Golovin, D., Chikkerur, S., Liu, D., Wattenberg, M., Hrafnkelsson, A. M., Boulos, T., and Kubica, J. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1222–1230, 2013.

Mokhtari, A., Shahrampour, S., Jadbabaie, A., and Ribeiro, A. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *Proceedings of 55th Conference on Decision and Control*, pp. 7195–7201, 2016.

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

Quanrud, K. and Khashabi, D. Online learning with adversarial delays. In *Advances in Neural Information Processing Systems 28*, pp. 1270–1278, 2015.

Rakhlin, A. and Sridharan, K. Online learning with predictable sequences. In *Proceedings of the 26th Annual Conference on Learning Theory*, pp. 993–1019, 2013.

Ren, Z., Zhou, Z., Qiu, L., Deshpande, A., and Kalagnanam, J. Delay-adaptive distributed stochastic optimization. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 5503–5510, 2020.

Shalev-Shwartz, S. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

Shalev-Shwartz, S. and Singer, Y. A primal-dual perspective of online learning algorithm. *Machine Learning*, 69(2–3): 115–142, 2007.

van Erven, T. and Koolen, W. M. MetaGrad: Multiple learning rates in online learning. In *Advances in Neural Information Processing Systems 29*, pp. 3666–3674, 2016.

van Erven, T., Koolen, W. M., and van der Hoeven, D. Metagrad: Adaptation using multiple learning rates in online learning. *Journal of Machine Learning Research*, 22(161):1–61, 2021.

Wan, Y., Xue, B., and Zhang, L. Projection-free online learning in dynamic environments. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pp. 10067–10075, 2021.

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

Wan, Y., Tu, W.-W., and Zhang, L. Online Frank-Wolfe with arbitrary delays. In *Advances in Neural Information Processing Systems 35*, 2022b.

Wan, Y., Wang, Y., Yao, C., Tu, W.-W., and Zhang, L. Projection-free online learning with arbitrary delays. *arXiv:2204.04964v2*, 2023a.

Wan, Y., Zhang, L., and Song, M. Improved dynamic regret for online Frank-Wolfe. In *Proceedings of the 36th Annual Conference on Learning Theory*, 2023b.

Wang, J., Liang, B., Dong, M., Boudreau, G., and Abou-Zeid, H. Delay-tolerant constrained OCO with application to network resource allocation. In *Proceedings of the 2021 IEEE Conference on Computer Communications*, pp. 1–10, 2021.

Wang, J., Dong, M., Liang, B., Boudreau, G., and Abou-Zeid, H. Delay-tolerant OCO with long-term constraints: Algorithm and its application to network resource allocation. *IEEE/ACM Transactions on Networking*, 31(1): 147–163, 2023.

Wang, Y., Yang, W., Jiang, W., Lu, S., Wang, B., Tang, H., Wan, Y., and Zhang, L. Non-stationary projection-free online learning with dynamic and adaptive regret guarantees. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, pp. 15671–15679, 2024.

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

Yang, T., Zhang, L., Jin, R., and Yi, J. 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.

Zhang, L., Yang, T., Yi, J., Jin, R., and Zhou, Z.-H. Improved dynamic regret for non-degenerate functions. In *Advances in Neural Information Processing Systems 30*, pp. 732–741, 2017.

Zhang, L., Lu, S., and Zhou, Z.-H. Adaptive online learning in dynamic environments. In *Advances in Neural Information Processing Systems 31*, pp. 1323–1333, 2018a.

Zhang, L., Yang, T., Jin, R., and Zhou, Z.-H. Dynamic regret of strongly adaptive methods. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 5877–5886, 2018b.

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

Zhao, P., Zhang, Y.-J., Zhang, L., and Zhou, Z.-H. Dynamic regret of convex and smooth functions. In *Advances in Neural Information Processing Systems 33*, pp. 12510–12520, 2020.

Zhao, P., Xie, Y.-F., Zhang, L., and Zhou, Z.-H. Efficient methods for non-stationary online learning. In *Advances in Neural Information Processing Systems 35*, pp. 11573–11585, 2022.

Zhou, Z., Mertikopoulos, P., Bambos, N., Glynn, P., Ye, Y., Li, L.-J., and Fei-Fei, L. Distributed asynchronous optimization with unbounded delays: How slow can you go? In *Proceedings of the 35th International Conference on Machine Learning*, pp. 5970–5979, 2018.

Zhou, Z., Mertikopoulos, P., Bambos, N., Glynn, P. W., and Ye, Y. Distributed stochastic optimization with large delays. *Mathematics of Operations Research*, 47(3):2082–2111, 2022.

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

A. Detailed Discussions on Future Work

First, we notice that the $O(\sqrt{\bar{d}T})$ static regret bound can be achieved under arbitrary delays (Quanrud & Khashabi, 2015). Thus, it is natural to ask whether the $O(\sqrt{\bar{d}T(P_T+1)})$ dynamic regret bound can also be achieved without additional assumptions. However, from Theorem 3.4, compared with the static regret, it is more challenging to minimize the dynamic regret in the delayed setting, because delays will further cause a comparator drift, i.e., $\sum_{t=1}^{T}\|\mathbf{u}_t-\mathbf{u}_{c_t}\|_2$. It seems highly non-trivial to reduce the comparator drift without additional assumptions, and we leave this question as a future work.

Second, we have utilized the doubling trick to avoid tuning the learning rate with the unknown cumulative delay. One potential limitation of this technique is that it needs to repeatedly restart itself, while forgetting all the preceding information. For minimizing the static regret with arbitrary delays, Joulani et al. (2016) have addressed this limitation by continuously adjusting the learning rate according to the norm of received gradients. Thus, it is also appealing to extend this idea for minimizing the dynamic regret with arbitrary delays.

Third, our proposed algorithms require the time-stamp of delayed feedback. It is interesting to investigate how to minimize the dynamic regret with anonymous and arbitrary delays. A potential useful property is that under the In-Order assumption, the arrival order of the delayed gradients already ensures the ascending order of their time-stamps. Since our DOGD in Algorithm 1 only utilizes the time-stamp to sort the elements in $\mathcal{F}_t$, it actually can be implemented by simply performing the gradient descent step in (1) whenever a gradient arrives even without the time-stamp. However, in our Mild-OGD, the time-stamp is further utilized to compute the delayed surrogate loss of experts, i.e., $\ell_k(\mathbf{x}_k^\eta)$ in (8), which cannot be discarded.

B. Proof of Lemma 4.1

We first define

$$L_t^\eta=\sum_{i=1}^{t}\sum_{k\in\mathcal{F}_i}\ell_k(\mathbf{x}_k^\eta),\tilde{L}_t^\eta=\sum_{i=1}^{t}\ell_i(\mathbf{x}_i^\eta),\text{ and }\tilde{W}_t=\sum_{\eta\in\mathcal{H}}w_1^\eta e^{-\alpha\tilde{L}_t^\eta}.$$

Moreover, we define

$$\mathbf{c}_t=(L_t^\eta)_{\eta\in\mathcal{H}}\in\mathbb{R}^{|\mathcal{H}|},\tilde{\mathbf{c}}_t=(\tilde{L}_t^\eta)_{\eta\in\mathcal{H}}\in\mathbb{R}^{|\mathcal{H}|},\text{ and }\mathbf{w}_t=(w_t^\eta)_{\eta\in\mathcal{H}}\in\mathbb{R}^{|\mathcal{H}|}.$$

According to Algorithm 2, for any $t\geq 1$, it is easy to verify that

$$w_{t+1}^\eta=\frac{w_t^\eta e^{-\alpha\sum_{k\in\mathcal{F}_t}\ell_k(\mathbf{x}_k^\eta)}}{\sum_{\mu\in\mathcal{H}}w_t^\mu e^{-\alpha\sum_{k\in\mathcal{F}_t}\ell_k(\mathbf{x}_k^\mu)}}=\frac{w_1^\eta e^{-\alpha L_t^\eta}}{\sum_{\mu\in\mathcal{H}}w_1^\mu e^{-\alpha L_t^\mu}}.$$

Combining with the above definitions, we have

$$\mathbf{w}_{t+1}=\operatorname*{argmin}_{\mathbf{w}\in\Delta}\left\langle-\frac{1}{\alpha}\ln(\mathbf{w}_1)+\mathbf{c}_t,\mathbf{w}\right\rangle+\frac{1}{\alpha}\mathcal{R}(\mathbf{w})$$

where $\Delta=\{\mathbf{w}\succeq\mathbf{0}|\langle\mathbf{w},\mathbf{1}\rangle=1\}$ and $\mathcal{R}(\mathbf{w})=\sum_i w_i\ln w_i$.

Similarly, for any $t\geq 1$, we define

$$\tilde{\mathbf{w}}_{t+1}=\operatorname*{argmin}_{\mathbf{w}\in\Delta}\left\langle-\frac{1}{\alpha}\ln(\mathbf{w}_1)+\tilde{\mathbf{c}}_t,\mathbf{w}\right\rangle+\frac{1}{\alpha}\mathcal{R}(\mathbf{w})$$

In this way, for any $t\geq 1$, we also have $\tilde{\mathbf{w}}_{t+1}=(\tilde{w}_{t+1}^\eta)_{\eta\in\mathcal{H}}$, where

$$\tilde{w}_{t+1}^\eta=\frac{w_1^\eta e^{-\alpha\tilde{L}_t^\eta}}{\sum_{\mu\in\mathcal{H}}w_1^\mu e^{-\alpha\tilde{L}_t^\mu}}.$$

Moreover, we define $\tilde{\mathbf{w}}_1=\mathbf{w}_1$ and

$$\tilde{\mathbf{x}}_t=\sum_{\eta\in\mathcal{H}}\tilde{w}_t^\eta\mathbf{x}_t^\eta. \tag{22}$$

Then, we will bound the distance between $\tilde{\mathbf{x}}_t$ and $\mathbf{x}_t$ based on the following lemma.

Lemma B.1. *(Lemma 5 in Duchi et al. (2011)) Let* $\Pi_{\mathcal{K}}(\mathbf{u}, \alpha) = \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} \langle \mathbf{u}, \mathbf{x}\rangle + \frac{1}{\alpha}\mathcal{R}(\mathbf{x})$. *If* $\mathcal{R}(\mathbf{x})$ *is* 1*-strongly convex with respect to a norm* $\|\cdot\|$*, it holds that*

$$\|\Pi_{\mathcal{K}}(\mathbf{u}, \alpha) - \Pi_{\mathcal{K}}(\mathbf{v}, \alpha)\| \leq \alpha \|\mathbf{u} - \mathbf{v}\|_*$$

for any $\mathbf{u}$ *and* $\mathbf{v}$*, where* $\|\cdot\|_*$ *is the dual norm of* $\|\cdot\|$.

Since $\mathcal{R}(\mathbf{w}) = \sum_i w_i \ln w_i$ is 1-strongly convex with respect to $\|\cdot\|_1$, by applying Lemma B.1, for any $t > 1$, we have

$$\|\tilde{\mathbf{x}}_t - \mathbf{x}_t\|_2 = \left\| \sum_{\eta\in\mathcal{H}} (\tilde{w}_t^\eta - w_t^\eta)\mathbf{x}_t^\eta \right\|_2 \leq \sum_{\eta\in\mathcal{H}} |\tilde{w}_t^\eta - w_t^\eta| \, \|\mathbf{x}_t^\eta\|_2 \leq D\|\tilde{\mathbf{w}}_t - \mathbf{w}_t\|_1 \leq \alpha D \|\tilde{\mathbf{c}}_{t-1} - \mathbf{c}_{t-1}\|_\infty.$$

Let $\mathcal{U}_t = [t] \setminus \cup_{i\in[t]} \mathcal{F}_i$. Note that $\mathcal{U}_t$ actually records the time-stamp of gradients that are queried, but still not arrive at the end of round t. Then, for $t > 1$, it is not hard to verify that

$$\|\tilde{\mathbf{x}}_t - \mathbf{x}_t\|_2 \leq \alpha D \|\tilde{\mathbf{c}}_{t-1} - \mathbf{c}_{t-1}\|_\infty \leq \alpha D \max_{\eta\in\mathcal{H}} \left| \sum_{k\in\mathcal{U}_{t-1}} \ell_k(\mathbf{x}_k^\eta) \right| \leq \alpha \left(t - 1 - \sum_{i=1}^{t-1} |\mathcal{F}_i| \right) GD^2 = \alpha \left(m_t - 1 \right) GD^2 \tag{23}$$

where the last inequality is due to the definition of $\mathcal{U}_t$ and the fact that Assumptions 3.1 and 3.2 ensure

$$|\ell_k(\mathbf{x}_k^\eta)| = |\langle \nabla f_k(\mathbf{x}_k), \mathbf{x}_k^\eta - \mathbf{x}_k \rangle| \leq \|\nabla f_k(\mathbf{x}_k)\|_2 \|\mathbf{x}_k^\eta - \mathbf{x}_k\|_2 \leq GD \tag{24}$$

for any $k \in [T]$ and $\eta \in \mathcal{H}$.

The above inequality shows that $\tilde{\mathbf{x}}_t$ is close to $\mathbf{x}_t$. In the following, we first focus on the analysis of $\tilde{\mathbf{x}}_t$, and then combine with the distance between $\tilde{\mathbf{x}}_t$ and $\mathbf{x}_t$. To this end, we notice that

$$\ln \tilde{W}_T = \ln \left(\sum_{\eta\in\mathcal{H}} w_1^\eta e^{-\alpha \tilde{L}_T^\eta} \right) \geq \ln \left(\max_{\eta\in\mathcal{H}} w_1^\eta e^{-\alpha \tilde{L}_T^\eta} \right) = -\alpha \min_{\eta\in\mathcal{H}} \left(\tilde{L}_T^\eta + \frac{1}{\alpha} \ln \frac{1}{w_1^\eta} \right). \tag{25}$$

Next, for any $t \geq 2$, we have

$$\ln \left(\frac{\tilde{W}_t}{\tilde{W}_{t-1}} \right) = \ln \left(\frac{\sum_{\eta\in\mathcal{H}} w_1^\eta e^{-\alpha \tilde{L}_t^\eta}}{\sum_{\eta\in\mathcal{H}} w_1^\eta e^{-\alpha \tilde{L}_{t-1}^\eta}} \right) = \ln \left(\frac{\sum_{\eta\in\mathcal{H}} w_1^\eta e^{-\alpha \tilde{L}_{t-1}^\eta} e^{-\alpha \ell_t(\mathbf{x}_t^\eta)}}{\sum_{\eta\in\mathcal{H}} w_1^\eta e^{-\alpha \tilde{L}_{t-1}^\eta}} \right) = \ln \left(\sum_{\eta\in\mathcal{H}} \tilde{w}_t^\eta e^{-\alpha \ell_t(\mathbf{x}_t^\eta)} \right). \tag{26}$$

Combining (26) and $\tilde{w}_1^\eta = w_1^\eta$, we have

$$\ln \tilde{W}_T = \ln \tilde{W}_1 + \sum_{t=2}^{T} \ln \left(\frac{\tilde{W}_t}{\tilde{W}_{t-1}} \right) = \sum_{t=1}^{T} \ln \left(\sum_{\eta\in\mathcal{H}} \tilde{w}_t^\eta e^{-\alpha \ell_t(\mathbf{x}_t^\eta)} \right). \tag{27}$$

To proceed, we introduce Hoeffding's inequality (Hoeffding, 1963).

Lemma B.2. *Let* X *be a random variable with* $a \leq X \leq b$*. Then, for any* $s \in \mathbb{R}$*, it holds that*

$$\ln \mathbb{E}[e^{sX}] \leq s\mathbb{E}[X] + \frac{s^2(b-a)^2}{8}.$$

From (24) and Lemma B.2, we have

$$\ln \left(\sum_{\eta\in\mathcal{H}} \tilde{w}_t^\eta e^{-\alpha \ell_t(\mathbf{x}_t^\eta)} \right) \leq -\alpha \sum_{\eta\in\mathcal{H}} \tilde{w}_t^\eta \ell_t(\mathbf{x}_t^\eta) + \frac{\alpha^2 G^2 D^2}{2} \leq -\alpha \ell_t(\tilde{\mathbf{x}}_t) + \frac{\alpha^2 G^2 D^2}{2} \tag{28}$$

where the second inequality is due to Jensen's inequality and (22).

Combining (27) with (28), we have

$$\ln \tilde{W}_T \leq -\alpha \sum_{t=1}^{T} \ell_t\left(\tilde{\mathbf{x}}_t\right) + \frac{\alpha^2 G^2 D^2 T}{2}.$$

Then, by further combining with (25), we have

$$\sum_{t=1}^{T} \ell_t\left(\tilde{\mathbf{x}}_t\right) - \min_{\eta \in \mathcal{H}} \left(\sum_{t=1}^{T} \ell_t(\mathbf{x}_t^{\eta}) + \frac{1}{\alpha} \ln \frac{1}{w_1^{\eta}} \right) \leq \frac{\alpha G^2 D^2 T}{2}.$$

Finally, combining with (23), for any $\eta \in \mathcal{H}$, we have

$$\begin{aligned}
&\sum_{t=1}^{T} \ell_t\left(\mathbf{x}_t\right) - \left(\sum_{t=1}^{T} \ell_t(\mathbf{x}_t^{\eta}) + \frac{1}{\alpha} \ln \frac{1}{w_1^{\eta}} \right) \\
=&\sum_{t=1}^{T} \ell_t\left(\mathbf{x}_t\right) - \sum_{t=1}^{T} \ell_t\left(\tilde{\mathbf{x}}_t\right) + \sum_{t=1}^{T} \ell_t\left(\tilde{\mathbf{x}}_t\right) - \left(\sum_{t=1}^{T} \ell_t(\mathbf{x}_t^{\eta}) + \frac{1}{\alpha} \ln \frac{1}{w_1^{\eta}} \right) \\
\leq&\sum_{t=1}^{T} \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \tilde{\mathbf{x}}_t \rangle + \frac{\alpha G^2 D^2 T}{2} \leq \sum_{t=1}^{T} \|\nabla f_t(\mathbf{x}_t)\|_2 \|\mathbf{x}_t - \tilde{\mathbf{x}}_t\|_2 + \frac{\alpha G^2 D^2 T}{2} \\
\leq&\alpha G^2 D^2 \sum_{t=1}^{T} (m_t - 1) + \frac{\alpha G^2 D^2 T}{2} \leq \alpha G^2 D^2 \sum_{t=1}^{T} m_t
\end{aligned} \tag{29}$$

which completes this proof.

C. Proof of Lemma 4.2

Let $Z = \lceil T/d \rceil$. We first divide the total T rounds into Z blocks, where the length of the first $Z-1$ blocks is d and that of the last block is $T-(Z-1)d$. In this way, we can define the set of rounds in the block z as

$$\mathcal{T}_z = \{(z-1)d+1, \ldots, \min\{zd, T\}\}.$$

For any $z \in [Z]$ and $t \in \mathcal{T}_z$, we construct the delay as

$$d_t = \min\{zd, T\} - t + 1$$

which satisfies $1 \leq d_t \leq d$. These delays ensure that the information of any function in each block z is delayed to the end of the block, which is critical for us to construct loss functions that maximize the impact of delays on the static regret.

Note that to establish the lower bound of the static regret in the non-delayed setting, one can utilize a randomized strategy to select loss functions for each round (Abernethy et al., 2008). Here, to maximize the impact of delays, we only select one loss function $h_z(\mathbf{x})$ for all rounds in the same block z, i.e., $f_t(\mathbf{x}) = h_z(\mathbf{x})$ for any $t \in \mathcal{T}_z$. Specifically, we set

$$h_z(\mathbf{x}) = \frac{G}{\sqrt{n}} \langle \mathbf{w}_z, \mathbf{x} \rangle$$

where the i-th coordinate of $\mathbf{w}_z$ is ± 1 with probability $1/2$ for any $i \in [n]$ and will be denoted as $w_{z,i}$. It is not hard to verify that $h_z(\mathbf{x})$ satisfies Assumption 3.1.

From the above definitions, we have

$$\begin{aligned}
\mathbb{E}_{\mathbf{w}_1,\ldots,\mathbf{w}_Z}[R(T)] =&\mathbb{E}_{\mathbf{w}_1,\ldots,\mathbf{w}_Z} \left[\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}) \right] \\
=&\mathbb{E}_{\mathbf{w}_1,\ldots,\mathbf{w}_Z} \left[\sum_{z=1}^{Z} \sum_{t \in \mathcal{T}_z} \frac{G}{\sqrt{n}} \langle \mathbf{w}_z, \mathbf{x}_t \rangle - \min_{\mathbf{x} \in \mathcal{K}} \sum_{z=1}^{Z} \sum_{t \in \mathcal{T}_z} \frac{G}{\sqrt{n}} \langle \mathbf{w}_z, \mathbf{x} \rangle \right] \\
=&\mathbb{E}_{\mathbf{w}_1,\ldots,\mathbf{w}_Z} \left[- \min_{\mathbf{x} \in \mathcal{K}} \sum_{z=1}^{Z} \frac{G|\mathcal{T}_z|}{\sqrt{n}} \langle \mathbf{w}_z, \mathbf{x} \rangle \right]
\end{aligned}$$

where the third equality is due to $\mathbb{E}_{\mathbf{w}_1,\ldots,\mathbf{w}_Z}[\langle \mathbf{w}_z, \mathbf{x}_t\rangle] = 0$ for any $t \in \mathcal{T}_z$, which can be derived by the fact that any decision $\mathbf{x}_t$ in the block z is made before receiving the information of $\mathbf{w}_z$, and thus is independent with $\mathbf{w}_z$.

Since a linear function is minimized at the vertices of the cube, we further have

$$\begin{aligned}
\mathbb{E}_{\mathbf{w}_1,\ldots,\mathbf{w}_Z}[R(T)] =& -\mathbb{E}_{\mathbf{w}_1,\ldots,\mathbf{w}_Z}\left[\min_{\mathbf{x}\in\{-D/(2\sqrt{n}),D/(2\sqrt{n})\}^n} \sum_{z=1}^{Z} \frac{G|\mathcal{T}_z|}{\sqrt{n}} \langle \mathbf{w}_z, \mathbf{x}\rangle\right] \\
=& \mathbb{E}_{\mathbf{w}_1,\ldots,\mathbf{w}_Z}\left[\sum_{i=1}^{n} \frac{D}{2\sqrt{n}} \left|\sum_{z=1}^{Z} \frac{w_{z,i} G|\mathcal{T}_z|}{\sqrt{n}}\right|\right] \\
=& \frac{DG}{2}\mathbb{E}_{\mathbf{w}_1,\ldots,\mathbf{w}_Z}\left[\left|\sum_{z=1}^{Z} w_{z,1}|\mathcal{T}_z|\right|\right] \geq \frac{DG}{2\sqrt{2}}\sqrt{\sum_{z=1}^{Z}|\mathcal{T}_z|^2} \\
\geq& \frac{DG}{2\sqrt{2}}\sqrt{\frac{(\sum_{z=1}^{Z}|\mathcal{T}_z|)^2}{Z}} = \frac{DGT}{2\sqrt{2\lceil T/d\rceil}}
\end{aligned} \tag{30}$$

where the first inequality is due to the Khintchine inequality and the second inequality is due to the Cauchy-Schwarz inequality.

The expected lower bound in (30) implies that for any OCO algorithm and any positive integer d, there exists a particular choice of $\mathbf{w}_1, \ldots, \mathbf{w}_Z$ such that

$$R(T) \geq \frac{DGT}{2\sqrt{2\lceil T/d\rceil}}.$$

D. DOGD with the Doubling Trick

As discussed after Corollary 3.6, our DOGD needs a learning rate depending on the following value

$$\sum_{t=1}^{T} m_t = \sum_{t=1}^{T}\left(t - \sum_{i=1}^{t-1}|\mathcal{F}_i|\right).$$

However, it may be not available beforehand. Fortunately, the doubling trick (Cesa-Bianchi & Lugosi, 2006) provides a way to adaptively estimate this value. Specifically, it will divide the total T rounds into several epochs, and run a new instance of DOGD per epoch. Let s_v and $s_{v+1} - 1$ respectively denote the start round and the end round of the v-th epoch. In this way, to tune the learning rate for the v-th epoch, we only need to know the following value

$$\sum_{t=s_v}^{s_{v+1}-1}\left(t + 1 - s_v - \sum_{i=s_v}^{t-1}|\mathcal{F}_i^{s_v}|\right)$$

where $\mathcal{F}_i^{s_v} = \{k \in [s_v, i] | k + d_k - 1 = i\}$.

According to the doubling trick, we can estimate this value to be 2^v at the start round s_v of the v-th epoch. Then, for any $t > s_v$, we first judge whether the estimate is still valid, i.e.,

$$\sum_{j=s_v}^{t}\left(j + 1 - s_v - \sum_{i=s_v}^{j-1}|\mathcal{F}_i^{s_v}|\right) \leq 2^v$$

where the left side can be calculated at the beginning of round t. If the answer is positive, the round t is still assigned to the v-th epoch, and the instance of DOGD keeps running. Otherwise, the round t is set as the start round of the $(v+1)$-th epoch, and a new instance of DOGD is activated. Notice that in the start round of the $(v+1)$-th epoch, the new estimate must be valid, since $t = s_{v+1}$ and

$$\sum_{j=s_{v+1}}^{t}\left(j + 1 - s_{v+1} - \sum_{i=s_{v+1}}^{j-1}|\mathcal{F}_i^{s_{v+1}}|\right) = 1 \leq 2^{v+1}.$$

Algorithm 4 DOGD with the Doubling Trick

Initialization: set $\mathbf{y}_1 = \mathbf{0}$, $\tau = 1$, $v = 1$, and $s_v = 1$
for $t = 1, \ldots, T$ **do**
  if $\sum_{j=s_v}^{t} \left(j + 1 - s_v - \sum_{i=s_v}^{j-1} |\mathcal{F}_i^{s_v}| \right) > 2^v$ **then**
    Set $\mathbf{y}_1 = \mathbf{0}$, $\tau = 1$, $v = v + 1$, and $s_v = t$
  end if
  Play $\mathbf{x}_t = \mathbf{y}_\tau$ and query $\nabla f_t(\mathbf{x}_t)$
  Receive $\{\nabla f_k(\mathbf{x}_k) | k \in \mathcal{F}_t^{s_v}\}$, where $\mathcal{F}_t^{s_v} = \{k \in [s_v, t] | k + d_k - 1 = t\}$
  for $k \in \mathcal{F}_t^{s_v}$ (in the ascending order) **do**
    Compute $\mathbf{y}_{\tau+1} = \operatorname{argmin}_{\mathbf{x} \in \mathcal{K}} \|\mathbf{x} - (\mathbf{y}_\tau - \eta_v \nabla f_k(\mathbf{x}_k))\|_2^2$, where $\eta_v = \frac{D}{G 2^{v/2}}$
    Set $\tau = \tau + 1$
  end for
end for

Moreover, it is natural to set $s_1 = 1$. Then, the detailed procedures of DOGD with the doubling trick are summarized in Algorithm 4.

Remark: First, in Algorithm 4, the learning rate η_v is set by replacing $\sum_{t=1}^{T} m_t$ in the learning rate required by Corollary 3.6 with 2^v. Second, in each epoch v, we do not need to utilize gradients queried before this epoch. For this reason, in Algorithm 4, we only receive $\{\nabla f_k(\mathbf{x}_k) | k \in \mathcal{F}_t^{s_v}\}$, instead of $\{\nabla f_k(\mathbf{x}_k) | k \in \mathcal{F}_t\}$.

We have the following theorem, which can recover the dynamic regret bound in Corollary 3.6 up to a constant factor.

Theorem D.1. *Under Assumptions 3.1 and 3.2, for any comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{K}$, Algorithm 2 ensures*

$$R(\mathbf{u}_1, \ldots, \mathbf{u}_T) \leq \frac{2G\left(2D + P_T\right)\sqrt{\bar{d}T}}{\sqrt{2} - 1} + C$$

where C is defined in (6).

Proof. For any s_v and $j \geq s_v$, we first notice that the value of $j - s_v - \sum_{i=s_v}^{j-1} |\mathcal{F}_i^{s_v}|$ counts the number of gradients that have been queried over interval $[s_v, j-1]$, but still not arrive at the end of round $j - 1$. Moreover, the gradient $\nabla f_j(\mathbf{x}_j)$ will only be counted as an unreceived gradient in $d_j - 1$ rounds. Therefore, for any $s_v \leq t \leq T$, it is easy to verify that

$$\sum_{j=s_v}^{t} \left(j + 1 - s_v - \sum_{i=s_v}^{j-1} |\mathcal{F}_i^{s_v}| \right) \leq \sum_{j=s_v}^{t} d_j \leq \sum_{j=1}^{T} d_j = \bar{d}T.$$

For brevity, let V denote the final v of Algorithm 4, and let $S = \bar{d}T$. It is easy to verify that

$$V \leq 1 + \log_2 S. \tag{31}$$

Then, let $s_{V+1} = T + 1$. We notice that for $v \in [V]$, Algorithm 4 actually starts or restarts Algorithm 1 with the learning rate of η_v at round s_v, which ends at round $s_{v+1} - 1$. Therefore, combining Theorem 3.4 with Lemma 3.5, under Assumptions 3.1 and 3.2, we have

$$\begin{aligned} &\sum_{t=s_v}^{s_{v+1}-1} f_t(\mathbf{x}_t) - \sum_{t=s_v}^{s_{v+1}-1} f_t(\mathbf{u}_t) \\ \leq &\frac{D^2 + D\sum_{t=s_v+1}^{s_{v+1}-1} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2}{\eta_v} + \eta_v G^2 \sum_{j=s_v}^{s_{v+1}-1} \left(j + 1 - s_v - \sum_{i=s_v}^{j-1} |\mathcal{F}_i^{s_v}| \right) + C_v \end{aligned} \tag{32}$$

where

$$C_v = \begin{cases} 0, \text{ if Assumption 3.3 also holds;} \\ \min\left\{ (s_{v+1} - s_v)GD, 2dG \sum_{t=s_v+1}^{s_{v+1}-1} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2 \right\}, \text{ otherwise.} \end{cases} \tag{33}$$

Moreover, we notice that Algorithm 4 also ensures that

$$\sum_{j=s_v}^{s_{v+1}-1}\left(j+1-s_v-\sum_{i=s_v}^{j-1}|\mathcal{F}_i^{s_v}|\right)\leq 2^v. \tag{34}$$

By substituting the above inequality into (32), we have

$$\begin{aligned}\sum_{t=s_v}^{s_{v+1}-1} f_t(\mathbf{x}_t)-\sum_{t=s_v}^{s_{v+1}-1} f_t(\mathbf{u}_t)\leq&\frac{D^2+D\sum_{t=s_v+1}^{s_{v+1}-1}\|\mathbf{u}_t-\mathbf{u}_{t-1}\|_2}{\eta_v}+\eta_v G^2 2^v+C_v\\ =&G2^{v/2}\left(2D+\sum_{t=s_v+1}^{s_{v+1}-1}\|\mathbf{u}_t-\mathbf{u}_{t-1}\|_2\right)+C_v\\ \leq&G2^{v/2}\left(2D+P_T\right)+C_v.\end{aligned} \tag{35}$$

Then, because of (31), we have

$$\begin{aligned}R(\mathbf{u}_1,\ldots,\mathbf{u}_T)=&\sum_{v=1}^{V}\left(\sum_{t=s_v}^{s_{v+1}-1} f_t(\mathbf{x}_t)-\sum_{t=s_v}^{s_{v+1}-1} f_t(\mathbf{u}_t)\right)\leq\sum_{v=1}^{V}G2^{v/2}\left(2D+P_T\right)+\sum_{v=1}^{V}C_v\\ =&G\left(2D+P_T\right)\frac{\sqrt{2}(2^{V/2}-1)}{\sqrt{2}-1}+\sum_{v=1}^{V}C_v\leq\frac{2G\left(2D+P_T\right)\sqrt{S}}{\sqrt{2}-1}+\sum_{v=1}^{V}C_v.\end{aligned} \tag{36}$$

Moreover, it is not hard to verify that

$$\begin{aligned}&\sum_{v=1}^{V}\min\left\{(s_{v+1}-s_v)GD,2dG\sum_{t=s_v+1}^{s_{v+1}-1}\|\mathbf{u}_t-\mathbf{u}_{t-1}\|_2\right\}\\ \leq&\min\left\{\sum_{v=1}^{V}(s_{v+1}-s_v)GD,\sum_{v=1}^{V}2dG\sum_{t=s_v+1}^{s_{v+1}-1}\|\mathbf{u}_t-\mathbf{u}_{t-1}\|_2\right\}\\ \leq&\min\left\{TGD,2dGP_T\right\}\end{aligned}$$

which implies that

$$\sum_{v=1}^{V}C_v\leq C. \tag{37}$$

Finally, we complete this proof by substituting (37) and $S=\bar{d}T$ into (36). □

E. Mild-OGD with the Doubling Trick

Similar to DOGD, Mild-OGD requires the value of $\sum_{t=1}^{T}m_t$ for setting

$$\alpha=\frac{1}{GD\sqrt{\sum_{t=1}^{T}m_t}}\text{ and }\eta_i=\frac{2^{i-1}D}{G\sqrt{\sum_{t=1}^{T}m_t}} \tag{38}$$

where α is the learning rate for updating the weight, and η_i is the learning rate for the i-th expert. To address this limitation, we can utilize the doubling trick as described in the previous section. The only change is to replace DOGD with Mild-OGD. The detailed procedures of Mild-OGD with the doubling trick are outlined in Algorithms 5 and 6.

Remark: We would like to emphasize that since multiple instances of the expert-algorithm run over the surrogate losses defined by the meta-algorithm, these instances and the meta-algorithm will start a new epoch synchronously. Moreover, as shown in step 6 of Algorithm 5, in the start of each epoch, we need to reinitialize the weight of each expert E^η. As shown in step 11, in each epoch v, we update the weight by using the learning rate α_v, which replaces $\sum_{t=1}^{T}m_t$ in (38) with 2^v.

Algorithm 5 Mild-OGD with the Doubling Trick: Meta-algorithm

1: **Initialization:** set $v = 1$ and $s_v = 1$
2: Activate a set of experts $\{E^\eta | \eta \in \mathcal{H}\}$ by invoking the expert-algorithm for each constant $\eta \in \mathcal{X}$, where

$$\mathcal{H} = \left\{ \eta_i = \frac{D2^{i-1}}{G} \middle| i = 1, \ldots, \left\lceil \log_2 \sqrt{T+1} \right\rceil + 1 \right\}$$

3: Set $w_t^{\eta_i} = \frac{|\mathcal{H}|+1}{i(i+1)|\mathcal{H}|}$
4: **for** $t = 1, \ldots, T$ **do**
5: **if** $\sum_{j=s_v}^{t} \left(j + 1 - s_v - \sum_{i=s_v}^{j-1} |\mathcal{F}_i^{s_v}| \right) > 2^v$ **then**
6: Set $v = v + 1$, $s_v = t$, and $w_t^{\eta_i} = \frac{|\mathcal{H}|+1}{i(i+1)|\mathcal{H}|}$
7: **end if**
8: Receive $\mathbf{x}_t^\eta$ from each expert E^η
9: Play the decision $\mathbf{x}_t = \sum_{\eta \in \mathcal{H}} w_t^\eta \mathbf{x}_t^\eta$
10: Query $\nabla f_t(\mathbf{x}_t)$ and receive $\{\nabla f_k(\mathbf{x}_k) | k \in \mathcal{F}_t^{s_v}\}$, where $\mathcal{F}_t^{s_v} = \{k \in [s_v, t] | k + d_k - 1 = t\}$
11: Update the weight of each expert by

$$w_{t+1}^\eta = \frac{w_t^\eta e^{-\alpha_v \sum_{k \in \mathcal{F}_t^{s_v}} \ell_k(\mathbf{x}_k^\eta)}}{\sum_{\mu \in \mathcal{H}} w_t^\mu e^{-\alpha_v \sum_{k \in \mathcal{F}_t^{s_v}} \ell_k(\mathbf{x}_k^\mu)}}$$

where $\ell_k(\mathbf{x}) = \langle \nabla f_k(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle$ and $\alpha_v = \frac{1}{GD2^{v/2}}$
12: Send $\{\nabla f_k(\mathbf{x}_k) | k \in \mathcal{F}_t^{s_v}\}$ to each expert E^η
13: **end for**

Algorithm 6 Mild-OGD with the Doubling Trick: Expert-algorithm

1: **Input:** a constant η
2: **Initialization:** set $\mathbf{y}_1^\eta = \mathbf{0}$, $\tau = 1$, $v = 1$, and $s_v = 1$
3: **for** $t = 1, \ldots, T$ **do**
4: **if** $\sum_{j=s_v}^{t} \left(j + 1 - s_v - \sum_{i=s_v}^{j-1} |\mathcal{F}_i^{s_v}| \right) > 2^v$ **then**
5: Set $\mathbf{y}_1 = \mathbf{0}$, $\tau = 1$, $v = v + 1$, and $s_v = t$
6: **end if**
7: Submit $\mathbf{x}_t^\eta = \mathbf{y}_\tau^\eta$ to the meta-algorithm
8: Receive gradients $\{\nabla f_k(\mathbf{x}_k) | k \in \mathcal{F}_t^{s_v}\}$ from the meta-algorithm
9: **for** $k \in \mathcal{F}_t$ (in the ascending order) **do**
10: Compute $\mathbf{y}_{\tau+1}^\eta = \operatorname{argmin}_{\mathbf{x} \in \mathcal{K}} \left\| \mathbf{x} - \left(\mathbf{y}_\tau^\eta - \frac{\eta}{2^{v/2}} \nabla f_k(\mathbf{x}_k) \right) \right\|_2^2$
11: Set $\tau = \tau + 1$
12: **end for**
13: **end for**

Additionally, to facilitate presentation, in step 2 of Algorithm 5, each η_i in $\mathcal{H}$ only contains the constant part that does not depend on the value of $\sum_{t=1}^{T} m_t$. Meanwhile, according to steps 1 and 10 of Algorithm 6, the i-th expert will receive η_i from the meta-algorithm, and combine it with the estimation of $\sum_{t=1}^{T} m_t$ to compute the learning rate.

Furthermore, we have the following theorem, which can recover the dynamic regret bound in Theorem 3.7 up to a constant factor.

Theorem E.1. *Under Assumptions 3.1 and 3.2, for any comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{K}$, Algorithm 2 ensures*

$$R(\mathbf{u}_1, \ldots, \mathbf{u}_T) \leq \frac{2\left(\left(2\ln\left(\left\lfloor \log_2 \sqrt{(D+P_T)/D} \right\rfloor + 2\right) + 1\right) GD + 3G\sqrt{D^2 + DP_T}\right)\sqrt{\bar{d}T}}{\sqrt{2}-1} + C$$

where C is defined in (6).

Proof. Following the proof of Theorem D.1, we use V to denote the final v of Algorithms 5 and 6 and define $s_{V+1} = T+1$. Moreover, let $S = \bar{d}T$. It is easy to verify that (31) also holds.

Then, we consider the dynamic regret of Algorithm 5 over the interval $[s_v, s_{v+1}-1]$ for each $v \in [V]$. Let

$$\eta_*^v = \sqrt{\frac{D}{G^2}\left(D + \sum_{t=s_v+1}^{s_{v+1}-1} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2\right)}.$$

From Assumption 3.2, we have

$$0 \leq \sum_{t=s_v+1}^{s_{v+1}-1} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2 \leq (s_{v+1} - s_v - 1)D \leq TD$$

which implies that

$$\eta_1 = \frac{D}{G} \leq \eta_*^v \leq \frac{D\sqrt{T+1}}{G} \leq \eta_{|\mathcal{H}|}.$$

Therefore, for any possible value of $\sum_{t=s_v+1}^{s_{v+1}-1} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2$, there must exist a constant $\eta_{k_v} \in \mathcal{H}$ such that

$$\eta_{k_v} \leq \eta_*^v \leq 2\eta_{k_v} \tag{39}$$

where

$$k_v = \left\lfloor \log_2 \sqrt{\left(D + \sum_{t=s_v+1}^{s_{v+1}-1} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2\right)/D} \right\rfloor + 1 \leq \left\lfloor \log_2 \sqrt{(D+P_T)/D} \right\rfloor + 1.$$

Moreover, we notice that each expert E^η over the interval $[s_v, s_{v+1}-1]$ actually runs Algorithm 1 with the learning rate $\frac{\eta}{2^{v/2}}$ to handle the surrogate losses $\ell_{s_v}(\mathbf{x}), \ldots, \ell_{s_{v+1}-1}(\mathbf{x})$, where each gradient $\nabla \ell_t(\mathbf{x}_t^\eta) = \nabla f_t(\mathbf{x}_t)$ is delayed to the end of round $t + d_t - 1$ for $t \in [s_v, s_{v+1}-1]$.

Therefore, by combining Theorem 3.4 with Lemma 3.5, under Assumptions 3.1 and 3.2, we have

$$\begin{aligned}
&\sum_{t=s_v}^{s_{v+1}-1} \left(\ell_t(\mathbf{x}_t^{\eta_{k_v}}) - \ell_t(\mathbf{u}_t)\right) \\
\leq& \frac{2^{v/2}\left(D^2 + D\sum_{t=s_v+1}^{s_{v+1}-1} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2\right)}{\eta_{k_v}} + \frac{\eta_{k_v} G^2}{2^{v/2}} \sum_{j=s_v}^{s_{v+1}-1} \left(j + 1 - s_v - \sum_{i=s_v}^{j-1} |\mathcal{F}_i^{s_v}|\right) + C_v \\
\leq& \frac{2^{v/2}\left(D^2 + D\sum_{t=s_v+1}^{s_{v+1}-1} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2\right)}{\eta_{k_v}} + \eta_{k_v} G^2 2^{v/2} + C_v \\
\leq& 3G\sqrt{2^v\left(D^2 + D\sum_{t=s_v+1}^{s_{v+1}-1} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2\right)} + C_v \\
\leq& 3G\sqrt{2^v\left(D^2 + DP_T\right)} + C_v
\end{aligned} \tag{40}$$

where C_v is defined in (33), the second inequality is due to the fact that Algorithm 6 also ensures (34), and the third inequality is due to (39) and the definition of η_*^v.

Moreover, it is also easy to verify that Algorithm 5 actually starts or restarts Algorithm 2 with the learning rate of α_v at round s_v, which ends at round $s_{v+1}-1$. Then, by using Lemma 4.1 with $(1/w_{s_v}^{\eta_{k_v}}) \leq (k_v+1)^2$, under Assumptions 3.1 and 3.2, we have

$$\begin{aligned}
\sum_{t=s_v}^{s_{v+1}-1} \ell_t(\mathbf{x}_t) - \sum_{t=s_v}^{s_{v+1}-1} \ell_t(\mathbf{x}_t^{\eta_{k_v}}) \leq& \frac{2}{\alpha_v}\ln(k_v+1) + \alpha_v G^2 D^2 \sum_{j=s_v}^{s_{v+1}-1} \left(j + 1 - s_v - \sum_{i=s_v}^{j-1} |\mathcal{F}_i^{s_v}|\right) \\
\leq& \left(2\ln\left(\left\lfloor \log_2 \sqrt{(D+P_T)/D} \right\rfloor + 2\right) + 1\right) 2^{v/2} GD
\end{aligned} \tag{41}$$

where the second inequality is due to $\alpha_v = \frac{1}{GD2^{v/2}}$, the definition of k_v, and the fact that Algorithm 5 also ensures (34). By combining (40) and (41), it is not hard to verify that

$$\begin{aligned}
&\sum_{t=s_v}^{s_{v+1}-1} f_t(\mathbf{x}_t) - \sum_{t=s_v}^{s_{v+1}-1} f_t(\mathbf{u}_t) \\
\leq &\sum_{t=s_v}^{s_{v+1}-1} \ell_t(\mathbf{x}_t) - \sum_{t=s_v}^{s_{v+1}-1} \ell_t(\mathbf{u}_t) \\
= &\sum_{t=s_v}^{s_{v+1}-1} \ell_t(\mathbf{x}_t) - \sum_{t=s_v}^{s_{v+1}-1} \ell_t(\mathbf{x}_t^{\eta_{k_v}}) + \sum_{t=s_v}^{s_{v+1}-1} \ell_t(\mathbf{x}_t^{\eta_{k_v}}) - \sum_{t=s_v}^{s_{v+1}-1} \ell_t(\mathbf{u}_t) \\
\leq &\left(2\ln\left(\left\lfloor \log_2 \sqrt{(D+P_T)/D} \right\rfloor + 2\right) + 1\right) 2^{v/2} GD + 3G\sqrt{2^v \left(D^2 + DP_T\right)} + C_v
\end{aligned} \tag{42}$$

Then, we have

$$\begin{aligned}
&R(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_{v=1}^{V} \left(\sum_{t=s_v}^{s_{v+1}-1} f_t(\mathbf{x}_t) - \sum_{t=s_v}^{s_{v+1}-1} f_t(\mathbf{u}_t) \right) \\
\leq &\left(\left(2\ln\left(\left\lfloor \log_2 \sqrt{(D+P_T)/D} \right\rfloor + 2\right) + 1\right) GD + 3G\sqrt{D^2 + DP_T}\right) \sum_{v=1}^{V} 2^{v/2} + \sum_{v=1}^{V} C_v \\
= &\left(\left(2\ln\left(\left\lfloor \log_2 \sqrt{(D+P_T)/D} \right\rfloor + 2\right) + 1\right) GD + 3G\sqrt{D^2 + DP_T}\right) \frac{\sqrt{2}(2^{V/2} - 1)}{\sqrt{2} - 1} + \sum_{v=1}^{V} C_v \\
\leq &\frac{2\left(\left(2\ln\left(\left\lfloor \log_2 \sqrt{(D+P_T)/D} \right\rfloor + 2\right) + 1\right) GD + 3G\sqrt{D^2 + DP_T}\right)\sqrt{S}}{\sqrt{2} - 1} + \sum_{v=1}^{V} C_v
\end{aligned}$$

where the first inequality is due to (42), and the last inequality is due to (31).

Finally, by substituting (37) and $S = \bar{d}T$ into the above inequality, we complete this proof. □