Bandit Convex Optimization in Non-stationary Environments

Peng Zhao ZHAOP@LAMDA.NJU.EDU.CN
Guanghui Wang WANGGH@LAMDA.NJU.EDU.CN
Lijun Zhang ZHANGLJ@LAMDA.NJU.EDU.CN
Zhi-Hua Zhou ZHOUZH@LAMDA.NJU.EDU.CN
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China

Editor: Laurent Orseau

Abstract

Bandit Convex Optimization (BCO) is a fundamental framework for modeling sequential decision-making with partial information, where the only feedback available to the player is the one-point or two-point function values. In this paper, we investigate BCO in non-stationary environments and choose the *dynamic regret* as the performance measure, which is defined as the difference between the cumulative loss incurred by the algorithm and that of any feasible comparator sequence. Let T be the time horizon and P_T be the path-length of the comparator sequence that reflects the non-stationarity of environments. We propose a novel algorithm that achieves $O(T^{3/4}(1+P_T)^{1/2})$ and $O(T^{1/2}(1+P_T)^{1/2})$ dynamic regret respectively for the one-point and two-point feedback models. The latter result is optimal, matching the $\Omega(T^{1/2}(1+P_T)^{1/2})$ lower bound established in this paper. Notably, our algorithm is adaptive to the non-stationary environments since it does not require prior knowledge of the path-length P_T ahead of time, which is generally unknown. We further extend the algorithm to an anytime version that does not require to know the time horizon T in advance. Moreover, we study the *adaptive regret*, another widely used performance measure for online learning in non-stationary environments, and design an algorithm that provably enjoys the adaptive regret guarantees for BCO problems. Finally, we present empirical studies to validate the effectiveness of the proposed approach.[1]

Keywords: Bandit Convex Optimization, Non-stationary Environments, Online Learning, Dynamic Regret, Adaptive Regret, Ensemble Methods

1. Introduction

Online Convex Optimization (OCO) is a powerful tool for modeling sequential decision-making problems, which can be regarded as an iterative game between the player and environments (Shalev-Shwartz, 2012; Hazan, 2016). At iteration t, the player commits a decision $\mathbf{x}_t$ from a convex feasible set $\mathcal{X} \subseteq \mathbb{R}^d$, simultaneously, a convex function $f_t : \mathcal{X} \mapsto \mathbb{R}$ is revealed by environments, which could be in an adversarial way. Then, the player will suffer an instantaneous loss $f_t(\mathbf{x}_t)$. The standard performance measure for online convex

[1] A preliminary version of this work appeared in *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020 (Zhao et al., 2020a).

©2021 Peng Zhao, Guanghui Wang, Lijun Zhang, and Zhi-Hua Zhou.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v22/20-763.html.

optimization is the *regret*, defined as

$$\text{S-Regret}_T = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}) \tag{1}$$

which is the difference between the cumulative loss of the player and that of the best *fixed* decision in hindsight. The measure is also called *static* regret to emphasize the fact that the comparator in (1) is fixed.

There are two setups for online convex optimization according to the information revealed by the environments (Hazan, 2016). In the *full-information* setup, the player has all the information of the function f_t, including the gradients of f_t over the feasible set $\mathcal{X}$. By contrast, in the *bandit* setup (or we call it the *partial-information* setup), the instantaneous loss is the only feedback available to the player. In this paper, we focus on the latter case, which is referred to as the problem of Bandit Convex Optimization (BCO).

BCO has attracted considerable attention because it successfully models many real-world scenarios where the feedback available to the decision maker is partial or incomplete (Hazan, 2016). The key challenge lies in the limited feedback, i.e., the player has no access to gradients of the function. In the standard *one-point feedback* model, the only feedback is the one-point function value, based on which Flaxman et al. (2005) constructed an unbiased estimator of the gradient and then appealed to the online gradient descent algorithm that developed in the full-information setting (Zinkevich, 2003) to establish an $O(T^{3/4})$ expected static regret. Another common variant is the *two-point feedback* model, where the player is allowed to query function values of two points at each iteration. Agarwal et al. (2010) demonstrated an optimal $O(\sqrt{T})$ static regret for convex functions under this feedback model. Algorithms and regret analysis were further developed in the later studies (Saha and Tewari, 2011; Hazan and Levy, 2014; Bubeck et al., 2015; Dekel et al., 2015; Yang and Mohri, 2016; Bubeck et al., 2017; Lattimore, 2020).

The static regret in (1) compares with a fixed benchmark, so it implicitly assumes that there is a reasonably good decision over all iterations. Unfortunately, this may not be true in non-stationary environments, where the underlying distribution of online functions changes (Sugiyama and Kawanabe, 2012; Gama et al., 2014; Zhao et al., 2021). To address this limitation, the notion of *dynamic regret* is introduced by Zinkevich (2003) and defined as the difference between the cumulative loss of the player and that of a comparator sequence $\mathbf{u}_1, \dots, \mathbf{u}_T \in \mathcal{X}$,

$$\text{D-Regret}_T(\mathbf{u}_1, \dots, \mathbf{u}_T) = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t). \tag{2}$$

In contrast to a fixed benchmark in the static regret, dynamic regret compares with a sequence of *changing* comparators and therefore is more suitable for measuring the performance of online algorithms in non-stationary environments. We remark that (2) is also called the *universal* dynamic regret, since the regret bound holds universally against *any* feasible comparator sequence $\mathbf{u}_1, \dots, \mathbf{u}_T$ in the feasible set.

In the literature, there is a variant named the *worst-case* dynamic regret (Besbes et al., 2015; Jadbabaie et al., 2015; Mokhtari et al., 2016; Zhang et al., 2017, 2018b; Baby and

Table 1: Comparisons of existing dynamic regret bounds for BCO problems. In the table, the column of "dynamic regret" summarizes the attained dynamic regret bound, where T is the time horizon, $P_T = P_T(\mathbf{u}_1, \ldots, \mathbf{u}_T)$ and $P_T^* = \max_{\{\mathbf{x}_t^* \in \mathcal{X}_t^*\}_{t=1}^T} P_T(\mathbf{x}_1^*, \cdots, \mathbf{x}_T^*)$ are the path-length that reflects the non-stationarity of the environments, with $\mathcal{X}_t^*$ being the set of all minimizers of the online function f_t to handle the potential non-uniqueness of the minimizers. The column of "type" specifies whether the dynamic regret bound holds for any comparator sequence or for the sequence of minimizers of online functions only. Besides, the column of "Parameter-free" indicates whether the algorithm requires to know the path-length in advance.

Feedback model	Dynamic regret	Type	Parameter-free	Reference
one-point BCO	$O\big(T^{\frac{3}{4}}(1+P_T^*)\big)$	worst-case	NO	Chen and Giannakis (2019)
one-point BCO	$O\big(T^{\frac{3}{4}}(1+P_T)^{\frac{1}{2}}\big)$	universal	YES	This work
two-point BCO	$O\big(\sqrt{T(1+P_T^*)}\big)$	worst-case	NO	Yang et al. (2016)
two-point BCO	$O\big(\sqrt{T}(1+P_T^*)\big)$	worst-case	NO	Chen and Giannakis (2019)
two-point BCO	$O\big(\sqrt{T(1+P_T)}\big)$	universal	YES	This work

Wang, 2019; Zhang et al., 2020b; Zhao and Zhang, 2021), defined as

$$\text{D-Regret}_T^* = \text{D-Regret}_T(\mathbf{x}_1^*, \ldots, \mathbf{x}_T^*) = \sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}_t^*), \tag{3}$$

where $\mathbf{x}_t^* \in \arg\min_{\mathbf{x}\in\mathcal{X}} f_t(\mathbf{x})$ is a minimizer of the online function f_t over the domain $\mathcal{X}$. We can see that the worst-case dynamic regret (3) can be regarded as a special case of the universal dynamic regret by specifying the comparator sequence of (2) to be the sequence of minimizers of online functions, namely, $\mathbf{u}_t = \mathbf{x}_t^* \in \arg\min_{\mathbf{x}\in\mathcal{X}} f_t(\mathbf{x})$. There are many works studying the worst-case dynamic regret, however, as pointed out by Zhang et al. (2018a), the worst-case dynamic regret is typically too pessimistic. By contrast, the universal dynamic regret is more general, and it can encompass the worst-case dynamic regret and static regret as special cases. Therefore, comparing to the static regret and the worst-case dynamic regret, the universal dynamic regret studied in this paper is more adaptive to the non-stationarity of environments and thus is more desired as the performance measure to guide the algorithm design of online learning in non-stationary environments.

Recently, there are some studies on designing algorithms to minimize the dynamic regret of BCO problems (Yang et al., 2016; Chen and Giannakis, 2019). However, they provide the worst-case dynamic regret only, and the algorithms require some quantities as the input which are generally unknown in advance. Therefore, it is desired to design algorithms that enjoy *universal* dynamic regret for BCO problems.

In this paper, we start with the bandit gradient descent (BGD) algorithm of Flaxman et al. (2005), and analyze its universal dynamic regret. We demonstrate that the optimal parameter configuration of vanilla BGD also requires prior information of the unknown path-length. To address this issue, we propose the Parameter-free Bandit Gradient Descent algorithm (PBGD), which is inspired by the strategy of maintaining multiple learning rates in adaptive online learning (van Erven and Koolen, 2016). Our approach is essentially an *online ensemble* method (Zhou, 2012), consisting of a meta-algorithm and several expert-algorithms. The basic idea is to maintain a pool of candidate parameters, and then invoke multiple instances of the expert-algorithm simultaneously, each of which is associated with a candidate parameter. Next, the meta-algorithm combines the predictions from expert-algorithms by an expert-tracking method (Cesa-Bianchi and Lugosi, 2006). However, it is prohibited to run multiple expert-algorithms with different parameters simultaneously in BCO problems, since the player is only allowed to query the function values of one/two points in the bandit setup. To overcome this difficulty, we carefully design a surrogate function, as the linearization of the smoothed version of the loss function in the sense of expectation, and make the strategy suitable for bandit convex optimization. Our algorithm and analysis accommodate one-point and two-point feedback models, and Table 1 summarizes existing dynamic regret bounds for BCO problems and our results. The main contributions of this work are listed as follows.

- We establish the first *universal* dynamic regret that supports to compare with any feasible comparator sequence for the bandit gradient descent algorithm, in a unified analysis framework.
- We propose a *parameter-free* algorithm, which does not require to know the upper bound of the path-length P_T ahead of time, and meanwhile enjoys the state-of-the-art dynamic regret guarantees.
- We establish the *first* minimax lower bound of the universal dynamic regret for bandit convex optimization problems.

Our algorithm enjoys universal dynamic regret guarantees and does not require prior knowledge of the path-length P_T ahead of time that is generally unknown. As a result, our proposed algorithm is more adaptive to the non-stationary environments than BCO algorithms designed for minimizing static regret or worse-case dynamic regret. Furthermore, we make several extensions. First, the proposed PBGD algorithm requires the time horizon T as an input, which is also an impractical demand in real implementations. We remove the undesired dependence and develop an *anytime* algorithm that does not need to know the time horizon T in advance. Second, we investigate another widely used performance measure for online learning in non-stationary environments—adaptive regret (Hazan and Seshadhri, 2009), which is defined as the maximum of "local" static regret in every time interval. We propose an algorithm called Minimizing Adaptive regret in Bandit Convex Optimization (MABCO) to minimize the measure, and analyze the adaptive regret bound for the proposed algorithm. To the best of our knowledge, we are the first to systematically study the two performance metrics (universal dynamic regret and adaptive regret) for bandit convex optimization in non-stationary environments. On the other hand, as far as we know, we are also the first to make the techniques originally developed for the full-information online learning (van Erven and Koolen, 2016) feasible for the bandit setting.

The rest of the paper is structured as follows. Section 2 briefly reviews related work. In Section 3, we introduce the bandit gradient descent algorithm for BCO problems and provide the dynamic regret analysis. Section 4 presents the parameter-free BGD algorithm, the main contribution of this paper, with dynamic regret analysis. We establish the lower bound of dynamic regret for BCO problems in Section 5. We further design online algorithms for adaptive regret minimization for BCO problems in Section 6. Section 7 reports the empirical studies to show the effectiveness of our proposed approach. Finally, we conclude the paper and discuss the future work in Section 8. We defer some preliminary knowledge to Appendix A and proofs of dynamic regret analysis for the BGD algorithm to Appendix B.

2. Related Work

In this section, we briefly introduce related work of bandit convex optimization, as well as the dynamic regret minimization for online learning in non-stationary environments.

2.1 Bandit Convex Optimization

In the bandit convex optimization setting, the player is only allowed to query function values of one point or two points, and the gradient information is not accessible as opposed to the full-information setting. In the following, we briefly discuss the progress of static regret minimization of the BCO problems.

For the one-point feedback model, the seminal work of Flaxman et al. (2005) constructed an unbiased gradient estimator and established an $O(T^{3/4})$ expected regret for convex and Lipschitz functions. A similar result was independently obtained by Kleinberg (2004). Later, an $O(T^{2/3})$ rate was shown to be attainable with either strong convexity (Agarwal et al., 2010) or smoothness (Saha and Tewari, 2011). When functions are both strongly convex and smooth, Hazan and Levy (2014) designed a novel algorithm that achieves a regret of $O(\sqrt{T \log T})$ based on the follow-the-regularized-leader framework with self-concordant barriers, matching the $\Omega(\sqrt{T})$ lower bound (Shamir, 2013) up to logarithmic factors. Furthermore, recent breakthroughs (Bubeck et al., 2015, 2017) showed that $O(\mathrm{ploy}(\log T)\sqrt{T})$ regret is attainable for the convex case, albeit with a high dependence on the dimension d. The dimension-dependence is recently improved from $d^{9.5}$ to $d^{2.5}$ by the information-theoretic argument (Lattimore, 2020).

BCO with two-point feedback was proposed and studied by Agarwal et al. (2010), and was also independently studied in the context of stochastic optimization (Nesterov, 2011). Agarwal et al. (2010) first established the expected regret of $O(d^2\sqrt{T})$ and $O(d^2 \log T)$ for convex Lipschitz and strongly convex Lipschitz functions, respectively. These bounds are proved to be minimax optimal in terms of the time horizon T (Agarwal et al., 2010), and the dependence on the dimension d is later improved to be optimal (Shamir, 2017).

Besides, bandit linear optimization is a special case of BCO where the feedback is assumed to be a linear function of the chosen decision, and has been studied extensively (Awerbuch and Kleinberg, 2004; McMahan and Blum, 2004; Dani et al., 2007; Abernethy et al., 2008; Bubeck et al., 2012, 2019).

2.2 Dynamic Regret

There are two types of dynamic regret as aforementioned. The universal dynamic regret holds against any feasible comparator sequence, while the worst-case one only compares with the sequence of minimizers of online functions.

For the universal dynamic regret, existing results are only limited to the full-information setting. Zinkevich (2003) showed that online gradient descent (OGD) achieves an $O(\sqrt{T}(1 + P_T))$ regret, where $P_T = P_T(\mathbf{u}_1, \ldots, \mathbf{u}_T)$ is the path-length of comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T$, defined as

$$P_T(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_{t=2}^{T} \|\mathbf{u}_{t-1} - \mathbf{u}_t\|_2.$$

Recently, Zhang et al. (2018a) demonstrated that this upper bound is not optimal by establishing an $\Omega(\sqrt{T(1+P_T)})$ lower bound, and further proposed an online algorithm that attains a minimax optimal dynamic regret bound of order $O(\sqrt{T(1+P_T)})$ for convex functions. Zhao et al. (2020b) designed more adaptive algorithms for convex and smooth functions, and they proved problem-dependent dynamic regret bounds that could be much smaller than the minimax rate and safeguard the worst case simultaneously. However, to the best of our knowledge, there is no algorithms designed for minimizing the universal dynamic regret in the bandit setting.

For the worst-case dynamic regret, there are many studies in the full-information setting (Besbes et al., 2015; Jadbabaie et al., 2015; Yang et al., 2016; Mokhtari et al., 2016; Zhang et al., 2017; Baby and Wang, 2019; Zhang et al., 2020b; Zhao and Zhang, 2021) as well as the bandit setting (Gur et al., 2014; Wei et al., 2016; Yang et al., 2016; Luo et al., 2018; Auer et al., 2019; Chen and Giannakis, 2019). We mainly focus on the bandit convex optimization setting. There are two works designed for minimizing the (worst-case) dynamic regret of bandit convex optimization (Yang et al., 2016; Chen and Giannakis, 2019). Specifically, suppose the value of path-length P_T^* is known, Yang et al. (2016) established an $O(\sqrt{T(1+P_T^*)})$ dynamic regret for BCO with two-point feedback, in which the path-length for the worst-case dynamic regret is defined as

$$P_T^* = \max_{\{\mathbf{x}_t^* \in \mathcal{X}_t^*\}_{t=1}^T} P_T(\mathbf{x}_1^*, \cdots, \mathbf{x}_T^*) = \max_{\{\mathbf{x}_t^* \in \mathcal{X}_t^*\}_{t=1}^T} \left\{ \sum_{t=2}^{T} \|\mathbf{x}_{t-1}^* - \mathbf{x}_t^*\|_2 \right\}$$

where $\mathcal{X}_t^*$ denotes the set of all minimizers of the online function f_t to handle the potential non-uniqueness of the minimizers. Later, Chen and Giannakis (2019) applied the BCO techniques to the dynamic Internet-of-Things management, showing $O(T^{3/4}(1+P_T^*))$ and $O(T^{1/2}(1+P_T^*))$ dynamic regret bounds for the one-point and two-point feedback models, respectively. We note that the above two algorithms only enjoy the worst-case dynamic regret (instead of the universal dynamic regret), and meanwhile their algorithms require the path-length P_T^* as an input parameter, which is unfortunately not available in advance.

Another closely related performance measure for online convex optimization in non-stationary environments is the *adaptive regret* (Hazan and Seshadhri, 2009), which is defined as the maximum of "local" static regret in *every* time interval $[q,s] \subseteq [T]$,

$$\text{A-Regret}_T = \max_{[q,s]\subseteq[T]} \sum_{t=q}^{s} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=q}^{s} f_t(\mathbf{x}).$$

Hazan and Seshadhri (2009) proposed an efficient algorithm that enjoys $O(\sqrt{T\log^3 T})$ and $O(d\log^2 T)$ regrets for convex and exponentially concave functions, respectively. The rate for convex functions was improved later (Daniely et al., 2015; Jun et al., 2017; Zhang et al., 2019). Moreover, Zhang et al. (2018b) studied the relation between the adaptive regret and the worst-case dynamic regret. Recently, novel online algorithms are proposed to minimize the adaptive regret and universal dynamic regret simultaneously for online convex optimization (Zhang et al., 2020a; Cutkosky, 2020).

3. Bandit Gradient Descent (BGD)

In this section, we first list assumptions used in the paper, and then present the bandit gradient descent (BGD) algorithm for BCO problems as well as its universal dynamic regret analysis. To the best of our knowledge, this is the first work that analyzes the universal dynamic regret of BGD.

3.1 Assumptions

We make the following common assumptions for the bandit convex optimization (Flaxman et al., 2005; Agarwal et al., 2010).

Assumption 1 (Bounded Region)**.** The feasible set $\mathcal{X}$ is closed and contains the ball of radius r centered at the origin and is contained in the ball of radius R, namely,

$$r\mathbb{B} \subseteq \mathcal{X} \subseteq R\mathbb{B} \tag{4}$$

where $\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_2 \leq 1\}$.

Assumption 2 (Bounded Function Value)**.** The absolute values of all the functions are bounded by C, namely,

$$\forall t \in [T], \quad \max_{\mathbf{x}\in\mathcal{X}} |f_t(\mathbf{x})| \leq C. \tag{5}$$

Assumption 3 (Lipschitz Continuity)**.** All the functions are L-Lipschitz continuous over the feasible set $\mathcal{X}$, that is, for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, we have

$$\forall t \in [T], \quad |f_t(\mathbf{x}) - f_t(\mathbf{y})| \leq L\|\mathbf{x} - \mathbf{y}\|_2. \tag{6}$$

Meanwhile, we consider loss functions and the comparators are chosen by an oblivious adversary, that is, the adversary chooses the loss functions and comparators at the start of the online game.

3.2 Algorithm and Regret Analysis

In this part, we present the bandit gradient descent (BGD) algorithm and analyze its dynamic regret.

We start from the online gradient descent (OGD) algorithm developed for the full-information setting (Zinkevich, 2003). OGD begins with any initial decision $\mathbf{x}_1 \in \mathcal{X}$ and performs the following update at each iteration:

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}[\mathbf{x}_t - \eta \nabla f_t(\mathbf{x}_t)], \tag{7}$$

where $\eta > 0$ is the step size and $\Pi_{\mathcal{X}}[\cdot]$ denotes the Euclidean projection onto the nearest point in the feasible set $\mathcal{X}$.

The key challenge of BCO problems is the lack of gradients. To address the issue, Flaxman et al. (2005) and Agarwal et al. (2010) proposed to approximate the gradient $\nabla f_t(\mathbf{x}_t)$ in (7) with a gradient estimator $\widetilde{\mathbf{g}}_t$, obtained by evaluating the function at one (in the one-point feedback model) or two (in the two-point feedback model) random points around $\mathbf{x}_t$. Details will be presented later. Based on the gradient estimator, the algorithm will then perform

Algorithm 1 Bandit Gradient Descent (BGD)

```
Input: time horizon $T$, perturbation parameter $\delta$, shrinkage parameter $\alpha$, step size $\eta$
 1: Let $\mathbf{y}_1 = \mathbf{0}$
 2: for $t = 1$ to $T$ do
 3:     Select a unit vector $\mathbf{s}_t$ uniformly at random
        {Case 1. One-Point Feedback Model}
 4:     Submit $\mathbf{x}_t = \mathbf{y}_t + \delta\mathbf{s}_t$
 5:     Receive $f_t(\mathbf{x}_t)$ as the feedback
 6:     Construct the gradient estimator by (8)
 7:     $\mathbf{y}_{t+1} = \Pi_{(1-\alpha)\mathcal{X}}[\mathbf{y}_t - \eta\widetilde{\mathbf{g}}_t]$
        {Case 2. Two-Point Feedback Model}
 8:     Submit $\mathbf{x}_t^{(1)} = \mathbf{y}_t + \delta\mathbf{s}_t$ and $\mathbf{x}_t^{(2)} = \mathbf{y}_t - \delta\mathbf{s}_t$
 9:     Receive $f_t(\mathbf{x}_t^{(1)})$ and $f_t(\mathbf{x}_t^{(2)})$ as the feedback
10:     Construct the gradient estimator by (9)
11:     $\mathbf{y}_{t+1} = \Pi_{(1-\alpha)\mathcal{X}}[\mathbf{y}_t - \eta\widetilde{\mathbf{g}}_t]$
12: end for
```

the online gradient descent as shown in the line 7 (one-point feedback model) and line 11 (two-point feedback model). We unify their algorithms in Algorithm 1, called the Bandit Gradient Descent (BGD). Notice that in lines 8 and 14 of the algorithm, the projection of $\mathbf{y}_{t+1}$ is on a slightly smaller set $(1-\alpha)\mathcal{X}$ instead of the original feasible set $\mathcal{X}$ to ensure that the final decision $\mathbf{x}_{t+1}$ lies in the feasible set $\mathcal{X}$. Note that the idea of clipping the feasible set for online learning is originated in the works of tracking the best expert (Herbster and Warmuth, 1998, 2001), but for completely different purposes (the clipping operation therein enforces the weight of each expert be away from 0 with a minimal level to keep track of changing best experts).

In the following, we will describe the gradient estimator and analyze the universal dynamic regret for BCO with one-point and two-point feedback models, respectively.

One-Point Feedback Model. In the seminal work of Flaxman et al. (2005), the authors proposed the following gradient estimator $\widetilde{\mathbf{g}}_t \in \mathbb{R}^d$,

$$\widetilde{\mathbf{g}}_t = \frac{d}{\delta} f_t(\mathbf{y}_t + \delta\mathbf{s}_t) \cdot \mathbf{s}_t \tag{8}$$

where $\mathbf{s}_t \in \mathbb{R}^d$ is a unit vector selected uniformly at random and $\delta > 0$ is the perturbation parameter. Then, the following lemma due to (Flaxman et al., 2005, Lemma 2.1), a consequence of Stoke's theorem, guarantees that the gradient estimator (8) is indeed an *unbiased* estimator of the gradient of the smoothed version of the loss function f_t.

Lemma 1. *For any convex (but not necessarily differentiable) function $f : \mathcal{X} \mapsto \mathbb{R}$, define its* smoothed *version $\widehat{f}(\mathbf{x}) = \mathbb{E}_{\mathbf{v}\in\mathbb{B}}[f(\mathbf{x} + \delta\mathbf{v})]$, where the expectation is taken over the random vectors $\mathbf{v} \in \mathbb{B}$ with $\mathbb{B}$ being the unit ball, i.e., $\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_2 \leq 1\}$. Then, for any $\delta > 0$, we have*

$$\mathbb{E}_{\mathbf{s}\in\mathbb{S}}\left[\frac{d}{\delta} f(\mathbf{x} + \delta\mathbf{s}) \cdot \mathbf{s}\right] = \nabla\widehat{f}(\mathbf{x}),$$

where the expectation is taken over the random unit vector $\mathbf{s} \in \mathbb{S}$ *with* $\mathbb{S}$ *being the unit sphere centered around the origin, namely,* $\mathbb{S} = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_2 = 1\}$*.*

Lemma 1 implies that the gradient estimator (8) satisfies $\mathbb{E}_{\mathbf{s}\in\mathbb{S}}[\widetilde{\mathbf{g}}_t] = \nabla \widehat{f}_t(\mathbf{y}_t)$, where $\widehat{f}_t(\mathbf{x}) = \mathbb{E}_{\mathbf{v}\in\mathbb{B}}[f_t(\mathbf{x}+\delta\mathbf{v})]$ is the smoothed version of original function f_t. So we have obtained an unbiased estimator for the gradient of smoothed function $\widehat{f}_t$ at the point $\mathbf{y}_t$. Meanwhile, we will set the decision $\mathbf{x}_t$ as $\mathbf{y}_t$ plus a small amount of perturbation, more specifically, $\mathbf{x}_t = \mathbf{y}_t + \delta\mathbf{s}_t$. As a result, the smoothed function $\widehat{f}_t$ approximates the original function f_t well, and the decision $\mathbf{x}_t$ is also very close to $\mathbf{y}_t$, both of which are controlled by the small perturbation parameter δ. Hence, we can minimize the regret over the sequence of smoothed functions $\{\widehat{f}_t\}_{t=1,\dots,T}$ using its expected gradients, which is a good surrogate for minimizing the regret over the sequence of original loss functions $\{f_t\}_{t=1,\dots,T}$. The idea essentially yields the bandit gradient descent (BGD) algorithm shown in Algorithm 1: the main update procedures of the one-point feedback model are summarized in Case 1 (lines 4-7), where we use the gradient estimator $\widetilde{\mathbf{g}}_t$ to perform the online gradient descent to obtain the intermediate decision $\mathbf{y}_{t+1}$ (line 7), and the final decision $\mathbf{x}_t$ at each iteration is attained by adding an extra perturbation $\mathbf{x}_t = \mathbf{y}_t + \delta\mathbf{s}_t$ (line 4). We have the following result regarding its universal dynamic regret.

Theorem 1. *Under Assumptions 1, 2, and 3, for any perturbation parameter* $\delta > 0$*, step size* $\eta > 0$*, and shrinkage parameter* $\alpha = \delta/r$*, the expected dynamic regret of* $\mathrm{BGD}(T, \delta, \alpha, \eta)$ *for the one-point feedback model satisfies*

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \leq \frac{7R^2 + RP_T}{4\eta} + \frac{\eta d^2 C^2 T}{2\delta^2} + \left(3L + \frac{LR}{r}\right)\delta T$$

for any *feasible comparator sequence* $\mathbf{u}_1, \dots, \mathbf{u}_T \in \mathcal{X}$*.*

The proof of Theorem 1 can be found in Appendix B.1.

Remark 1. For notational convenience, we denote by $\widetilde{L} = 3L + LR/r$ the *effective* Lipschitz constant. By setting $\eta = \eta^* = (dC\widetilde{L})^{-1/2}((7R^2 + RP_T)/T)^{3/4}$ and $\delta = \delta^* = (dC/\widetilde{L})^{1/2}((7R^2 + RP_T)/T)^{1/4}$, we can obtain an $O(T^{3/4}(1 + P_T)^{1/4})$ dynamic regret. However, such a configuration requires prior knowledge of the path-length P_T, which is generally unavailable. We will develop a parameter-free algorithm to eliminate the undesired dependence later in Section 4.

Two-Point Feedback Model. In this setup, the player is allowed to query two points. Specifically, we will commit two decisions symmetrically sampled around the point $\mathbf{y}_t$, namely, $\mathbf{x}_t^{(1)} = \mathbf{y}_t + \delta\mathbf{s}_t$ and $\mathbf{x}_t^{(2)} = \mathbf{y}_t - \delta\mathbf{s}_t$. Then, the function values $f_t(\mathbf{x}_t^{(1)})$ and $f_t(\mathbf{x}_t^{(2)})$ are revealed as the feedback. To leverage the benefit of the two-point feedback, we will use the following gradient estimator (Agarwal et al., 2010),

$$\widetilde{\mathbf{g}}_t = \frac{d}{2\delta}\Big(f_t(\mathbf{y}_t + \delta\mathbf{s}_t) - f_t(\mathbf{y}_t - \delta\mathbf{s}_t)\Big) \cdot \mathbf{s}_t. \tag{9}$$

Lemma 1 implies the above estimator also satisfies the unbiased property, by noticing that the distribution of perturbation $\mathbf{s}_t$ is symmetric.

We here explain the advantage of the two-point estimator over the one-point estimator. The major limitation of the one-point gradient estimator (8) is that it has a potentially large magnitude, proportional to the $1/\delta$ which is usually quite large, because the perturbation parameter δ is typically set small. The undesired feature is avoided in the two-point gradient estimator (9), whose magnitude is at most dL, independent of the perturbation parameter δ. The crucial advantage leads to the substantial improvement in the dynamic regret (also static regret) for the two-point BCO.

Theorem 2. *Under Assumptions 1, 2, and 3, for any perturbation parameter $\delta > 0$, step size $\eta > 0$, and shrinkage parameter $\alpha = \delta/r$, the expected dynamic regret of* $\mathrm{BGD}(T, \delta, \alpha, \eta)$ *for the two-point feedback model satisfies*

$$\mathbb{E}\left[\sum_{t=1}^{T} \frac{1}{2}\big(f_t(\mathbf{x}_t^{(1)}) + f_t(\mathbf{x}_t^{(2)})\big)\right] - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \leq \frac{7R^2 + RP_T}{4\eta} + \frac{\eta d^2 L^2}{2} T + \left(3L + \frac{LR}{r}\right) \delta T$$

for any *feasible comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$.*

The proof of Theorem 2 can be found in Appendix B.2.

Remark 2. By setting perturbation $\delta = \delta^* = dLR/(\widetilde{L}\sqrt{T})$ and step size $\eta = \eta^* = \sqrt{(7R^2 + RP_T)/(2d^2L^2T)}$, we can attain an $O(T^{1/2}(1 + P_T)^{1/2})$ dynamic regret. Here, $\widetilde{L} = 3L + LR/r$ is the effective Lipschitz constant, also defined in Remark 1. However, the above parameter configuration has an unpleasant dependence on the unknown quantity P_T, which will be removed in the next section. Furthermore, we would like to mention that we set the optimal perturbation parameter as $dLR/(\widetilde{L}\sqrt{T})$ for the sake of a more succinct and beautiful regret form, and one can alternatively choose other appropriate configurations like $dR/\sqrt{T}$ without affecting the regret order.

4. Parameter-free Bandit Gradient Descent (PBGD)

From Theorem 1 and Theorem 2, we observe that the optimal parameter configurations of BGD algorithm require to know the path-length P_T in advance, which is generally unknown. Thus, the parameter configurations are impractical for real implementations. In this section, we develop a parameter-free algorithm to address this limitation.

4.1 Algorithm

The fundamental obstacle in obtaining universal dynamic regret guarantees is that the path-length P_T is unknown in advance. Indeed, P_T *remains unknown even after all iterations*, because the comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T$ can be chosen arbitrarily from the feasible set. As a consequence, the well-known doubling trick (Cesa-Bianchi et al., 1997) cannot be employed to remove the dependence. Another possible technique to overcome this difficulty is based on the idea of *online ensemble*, more specifically, to grid search the optimal parameter by maintaining candidates in parallel and using expert-tracking algorithms to combine predictions and track the best parameter (van Erven and Koolen, 2016). However, it is infeasible to directly apply this ensemble method to bandit convex optimization because of the inherent difficulty of the bandit problems—it is only allowed to query the function value *once* at each iteration.

To address this issue, we need a closer investigation of the dynamic regret analysis of BCO problems. Taking the one-point feedback model as an example, we can decompose the expected dynamic regret into the following three terms,

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] - \sum_{t=1}^{T} f_t(\mathbf{u}_t)\\ = &\underbrace{\mathbb{E}\left[\sum_{t=1}^{T}\Big(\widehat{f}_t(\mathbf{y}_t) - \widehat{f}_t(\mathbf{v}_t)\Big)\right]}_{\texttt{term (a)}} + \underbrace{\mathbb{E}\left[\sum_{t=1}^{T}\Big(f_t(\mathbf{x}_t) - \widehat{f}_t(\mathbf{y}_t)\Big)\right]}_{\texttt{term (b)}} + \underbrace{\mathbb{E}\left[\sum_{t=1}^{T}\Big(\widehat{f}_t(\mathbf{v}_t) - f_t(\mathbf{u}_t)\Big)\right]}_{\texttt{term (c)}},\end{aligned} \tag{10}$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_T$ is the scaled comparator sequence, set as $\mathbf{v}_t = (1-\alpha)\mathbf{u}_t$ and α is the shrinkage parameter. Note that the shrinkage parameter is actually set as $\alpha = \delta/r$ and the perturbation parameter $\delta = O(1/T^{1/4})$, so we assume the time horizon T is large enough such that $\alpha \in (0,1)$. Among the three terms, term (b) and term (c) are essentially the approximation error, which capture the amount of error introduced by the perturbation of functions (f_t versus $\widehat{f}_t$) and decisions/comparators ($\mathbf{x}_t$ versus $\mathbf{y}_t$, and $\mathbf{u}_t$ versus $\mathbf{v}_t$), respectively. A rigorous analysis will show that term (b) and term (c) can be bounded by $2L\delta T$ and $(L\delta + L\alpha R)T$ respectively without involving the unknown path-length, and the argument can be found in (40) and (41) of Appendix B.1. On the other hand, term (a) essentially depicts the dynamic regret over the smoothed functions $\widehat{f}_1, \ldots, \widehat{f}_T$, comparing to the scale comparator sequence $\mathbf{v}_1, \ldots, \mathbf{v}_T$. The quantity will depend on the step size configuration of BGD algorithms, which relies on the knowledge of the unknown path-length. Hence, it suffices to design parameter-free algorithms to optimize the term (a).

However, it remains infeasible to maintain multiple learning rates for optimizing the dynamic regret of the sequence of smoothed functions $\widehat{f}_1, \ldots, \widehat{f}_T$. The reason is as follows. Suppose there are in total N experts where each expert is associated with a specific learning rate (step size), then at iteration t, expert-algorithms will require the information of $\nabla\widehat{f}_t(\mathbf{y}_t^1), \nabla\widehat{f}_t(\mathbf{y}_t^2), \ldots, \nabla\widehat{f}_t(\mathbf{y}_t^N)$ to perform the bandit gradient descent. This necessitates to query N function values of original loss f_t, which is *prohibited* in the bandit convex optimization setting. Fortunately, we discover that the expected dynamic regret of $\widehat{f}_t$ can be upper bounded by that of a linear function, as demonstrated in the following proposition.

Proposition 1. $\mathbb{E}[\widehat{f}_t(\mathbf{y}_t) - \widehat{f}_t(\mathbf{v}_t)] \le \mathbb{E}[\langle \widetilde{\mathbf{g}}_t, \mathbf{y}_t - \mathbf{v}_t\rangle]$.

Proof. First, from the convexity of the smoothed function $\widehat{f}_t$, we have

$$\widehat{f}_t(\mathbf{y}_t) - \widehat{f}_t(\mathbf{v}_t) \le \langle \nabla\widehat{f}_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{v}_t\rangle = \langle \nabla\widehat{f}_t(\mathbf{y}_t) - \widetilde{\mathbf{g}}_t, \mathbf{y}_t - \mathbf{v}_t\rangle + \langle \widetilde{\mathbf{g}}_t, \mathbf{y}_t - \mathbf{v}_t\rangle.$$

Besides, similar to the argument of Flaxman et al. (2005), let $\boldsymbol{\xi}_t = \nabla\widehat{f}_t(\mathbf{y}_t) - \widetilde{\mathbf{g}}_t$, then $\mathbb{E}[\boldsymbol{\xi}_t \mid \mathbf{x}_1, f_1, \ldots, \mathbf{x}_t, f_t] = \mathbf{0}$ due to Lemma 1. Thus, for any *fixed* $\mathbf{x} \in \mathcal{X}$, we have

$$\mathbb{E}[\boldsymbol{\xi}_t^{\mathrm{T}}\mathbf{x}] = \mathbb{E}[\mathbb{E}[\boldsymbol{\xi}_t^{\mathrm{T}}\mathbf{x} \mid \mathbf{x}_1, f_1, \ldots, \mathbf{x}_t, f_t]] = \mathbb{E}[\mathbb{E}[\boldsymbol{\xi}_t \mid \mathbf{x}_1, f_1, \ldots, \mathbf{x}_t, f_t]^{\mathrm{T}}\mathbf{x}] = 0,$$

which implies $\mathbb{E}[\langle \nabla\widehat{f}_t(\mathbf{y}_t) - \widetilde{\mathbf{g}}_t, \mathbf{y}_t - \mathbf{v}_t\rangle] = 0$, since the comparator sequence is assumed to be chosen by an oblivious adversary. This ends the proof. □

The feature brings us many benefits, and thereby we can resolve the aforementioned difficulty. Indeed, Proposition 1 motivates us to design the following *surrogate loss* function $\ell_t : (1-\alpha)\mathcal{X} \mapsto \mathbb{R}$,

$$\ell_t(\mathbf{y}) = \langle \widetilde{\mathbf{g}}_t, \mathbf{y} - \mathbf{y}_t \rangle, \tag{11}$$

which can be regarded as a linearization of the smoothed function $\widehat{f}_t$ at the point $\mathbf{y}_t$ in the expectation sense. Furthermore, the surrogate loss enjoys the following two properties.

Property 1. *For any* $\mathbf{y} \in (1-\alpha)\mathcal{X}$, $\nabla \ell_t(\mathbf{y}) = \widetilde{\mathbf{g}}_t$.

Property 2. *For any* $\mathbf{v} \in (1-\alpha)\mathcal{X}$, $\mathbb{E}[\widehat{f}_t(\mathbf{y}_t) - \widehat{f}_t(\mathbf{v})] \leq \mathbb{E}[\ell_t(\mathbf{y}_t) - \ell_t(\mathbf{v})]$.

Property 1 follows from the definition of surrogate loss, and Proposition 1 immediately implies Property 2. These two properties are simple yet quite useful, and they together make the online ensemble method feasible in bandit convex optimization. Concretely speaking,

- Property 1 implies that we can now initialize N experts to perform the bandit gradient descent *over the surrogate loss* where each expert is associated with a specific step size, since all the gradients $\nabla \ell_t(\mathbf{y}_t^1), \nabla \ell_t(\mathbf{y}_t^2), \ldots, \nabla \ell_t(\mathbf{y}_t^N)$ essentially equal $\widetilde{\mathbf{g}}_t$, which can be obtained by querying the function value of original loss function f_t only *once*.
- Property 2 guarantees the expected dynamic regret of smoothed functions $\widehat{f}_t$'s is upper bounded by that of the surrogate loss functions ℓ_t's.

Consequently, we propose to optimize the surrogate loss ℓ_t instead of the original loss f_t (or its smoothed version $\widehat{f}_t$). We note that the idea of constructing surrogate loss for maintaining multiple step sizes (learning rates) is originally due to van Erven and Koolen (2016) but for different purposes. They constructed a quadratic upper bound for the original loss f_t as the surrogate loss, with the aim to adapt to the potential curvature of online functions in the *full-information* online convex optimization. In this paper, we design the surrogate loss as the linearization of the smoothed function $\widehat{f}_t$ in terms of the expectation, to make the grid search of optimal parameter (or online ensemble) doable for bandit convex optimization. To the best of our knowledge, this paper is the first time to optimize the surrogate loss for maintaining multiple step sizes in the *bandit* setup.

In the following, we describe the design details of the parameter-free algorithm for BCO with one-point feedback, including the configurations of step size pool, expert-algorithm, and meta-algorithm. The algorithm for the two-point feedback model is similar and slightly simpler, whose specialized configurations and theoretical analysis will be presented in Section 4.2.2.

In the one-point feedback model, from the dynamic regret analysis of BGD (refer to Theorem 1 and Remark 1), we know that the optimal step size is $\eta^* = (dC\widetilde{L})^{-1/2}((7R^2 + RP_T)/T)^{3/4}$ and the optimal perturbation parameter is $\delta^* = (dC/\widetilde{L})^{1/2}((7R^2 + RP_T)/T)^{1/4}$, where $\widetilde{L} = 3L + LR/r$ is the effective Lipschitz constant for notational simplicity. The optimal tuning of both step size and perturbation parameter requires the prior knowledge of the unknown path-length P_T, and thus the tuning is infeasible in real implementations. As aforementioned, we will maintain multiple experts to grid search the optimal parameter tuning. There are two unknown parameters–the step size η and the perturbation parameter δ. Unfortunately, our online ensemble method can only support to grid search the step size, and it is not applicable for approximating the perturbation parameter. Otherwise, we have to query the function more than once at each iteration, since the perturbation is imposed in the final decision $\mathbf{x}_t = \mathbf{y}_t + \delta \mathbf{s}_t$. Therefore, we have to set the perturbation parameter δ a constant independent of the path-length P_T. More specifically, we will use the online ensemble method to grid search the following parameter configurations instead:

$$\delta^\dagger = \left(\frac{dCR}{\widetilde{L}}\right)^{\frac{1}{2}} T^{-\frac{1}{4}}, \text{ and } \eta^\dagger = \sqrt{\frac{R(7R^2 + RP_T)}{2dC\widetilde{L}}} T^{-\frac{3}{4}}. \tag{12}$$

Noticing that the perturbation parameter $\delta^\dagger$ now is independent of the path-length P_T. Substituting the parameters into the dynamic regret analysis of Theorem 1, we can attain an $O\big(T^{\frac{3}{4}}(1+P_T)^{\frac{1}{2}}\big)$ dynamic regret. We remark that the step size tuning of (12) still exhibits the dependence on the unknown path-length P_T. In the following, we will demonstrate how to maintain multiple experts to achieve the same dynamic regret bound without requiring the prior knowledge of path-length.

Step size pool. The best possible step size is $\eta^\dagger = \sqrt{R(7R^2 + RP_T)/(2dC\widetilde{L})} \cdot T^{-3/4}$, which is unavailable due to the unknown path-length P_T. Nevertheless, due to the non-negativity and boundedness of the path-length, namely, $0 \le P_T \le 2RT$, we assure that the best step size lies in the following range:

$$\sqrt{\frac{7R^3}{2dC\widetilde{L}}} \cdot T^{-\frac{3}{4}} \le \eta^\dagger \le \sqrt{\frac{(7+2T)R^3}{2dC\widetilde{L}}} \cdot T^{-\frac{3}{4}}.$$

Hence, we can construct the following pool of candidate step sizes denoted by $\mathcal{H}$ to discretize the above range,

$$\mathcal{H} = \left\{ \eta_i = 2^{i-1}\sqrt{\frac{7R^3}{2dC\widetilde{L}}} \cdot T^{-\frac{3}{4}} \mid i = 1, \ldots, N \right\}, \tag{13}$$

where N denotes the number of candidate step sizes, set as

$$N = \left\lceil \frac{1}{2} \log_2 \left(1 + \frac{2T}{7}\right) \right\rceil + 1.$$

Notice that there are only $N = O(\log T)$ candidate step sizes, thanks to the exponential grid in the construction of the step size pool. As a result, we do not have to pay too much computational efforts for the meta-expert aggregation.

The configuration of the step size pool (13) ensures that there exists an index $k \in \{1, \ldots, N-1\}$ such that

$$\eta_k \le \eta^\dagger \le \eta_{k+1} = 2\eta_k.$$

That is to say, there exists a step size in the pool $\mathcal{H}$ that is not optimal but sufficiently close to $\eta^\dagger$, even though we are unaware of the index of this particular expert. Next, we will instantiate N expert-algorithms, where the i-th expert is a BGD algorithm with parameters $\eta_i \in \mathcal{H}$ and $\delta = \delta^\dagger = (dCR/\widetilde{L})^{1/2} T^{-1/4}$. Then, an expert-tracking algorithm is adopted

Algorithm 2 PBGD: Meta-algorithm

```
Input: time horizon $T$, number of experts $N$, pool of candidate step sizes $\mathcal{H} = \{\eta_1, \ldots, \eta_N\}$,
    learning rate of the meta-algorithm $\varepsilon$
Run expert-algorithms (14) with different step sizes simultaneously
Initialize the weight of each expert $i \in [N]$ as $w_1^i = \frac{N+1}{N} \cdot \frac{1}{i(i+1)}$
for $t = 1$ to $T$ do
    Receive $\mathbf{y}_t^i$ from each expert $i \in [N]$
    Obtain $\mathbf{y}_t = \sum_{i \in [N]} w_t^i \mathbf{y}_t^i$
    Submit $\mathbf{x}_t = \mathbf{y}_t + \delta \mathbf{s}_t$ and incur loss $f_t(\mathbf{x}_t)$
    Compute gradient estimator $\widetilde{\mathbf{g}}_t$ by (8)
    Construct surrogate loss $\ell_t(\cdot)$ as (11)
    Update the weight of each expert $i \in [N]$ by
        $$w_{t+1}^i = \frac{w_t^i \exp(-\varepsilon \ell_t(\mathbf{y}_t^i))}{\sum_{i \in [N]} w_t^i \exp(-\varepsilon \ell_t(\mathbf{y}_t^i))}$$
    Send the gradient estimator $\widetilde{\mathbf{g}}_t$ to each expert
end for
```

as the meta-algorithm to combine predictions from all the experts to produce the final decision. Owing to nice theoretical guarantees of the meta-algorithm, the dynamic regret of final decisions is comparable to that of the best expert, i.e., the expert-algorithm with the near-optimal step size.

We now present the descriptions for expert-algorithm and meta-algorithm of PBGD (for BCO with one-point feedback).

Expert-algorithm. We initialize an expert for each candidate step size of the pool $\mathcal{H}$. So there are in total N experts, and the expert $i \in [N]$ runs the online gradient descent over the surrogate loss defined in (11). Specifically, at iteration t, the expert i performs

$$\mathbf{y}_{t+1}^i = \Pi_{(1-\alpha)\mathcal{X}} \left[\mathbf{y}_t^i - \eta_i \nabla \ell_t(\mathbf{y}_t^i)\right] = \Pi_{(1-\alpha)\mathcal{X}} \left[\mathbf{y}_t^i - \eta_i \widetilde{\mathbf{g}}_t\right], \tag{14}$$

where $\eta_i \in \mathcal{H}$ is the step size of the expert i, shown in (13).

The above update procedure once again demonstrates the necessity of constructing the surrogate loss. Due to the nice property of surrogate loss (Property 1), at each iteration, all the experts can perform the *exact* online gradient descent in the same direction $\widetilde{\mathbf{g}}_t$. By contrast, suppose each expert is conducted over the smoothed loss function $\widehat{f}_t$, then at each iteration it requires to query multiple gradients $\nabla \widehat{f}_t(\mathbf{y}_t^i)$, or equivalently, to query multiple function values $f_t(\mathbf{x}_t^i)$, which is unavailable in bandit convex optimization.

Meta-algorithm. To combine predictions returned from multiple experts, we adopt the exponentially weighted average forecaster algorithm (Cesa-Bianchi and Lugosi, 2006) with nonuniform initial weights as the meta-algorithm, whose input is the pool of candidate step sizes $\mathcal{H}$ in (13) and its own learning rate ε. The nonuniform initialization of weights aims to make the meta-regret smaller, which will be clear by checking the proofs in the next subsection. Algorithm 2 presents detailed procedures. Note that the meta-algorithm itself does not require any prior information of the unknown path-length P_T.

The meta-algorithm in Algorithm 2, together with the expert-algorithm (14), yields our proposed PBGD (short for Parameter-free Bandit Gradient Descent) for minimizing the dynamic regret of bandit convex optimization in non-stationary environments.

4.2 Dynamic Regret Analysis

The following theorem states the dynamic regret of the proposed PBGD algorithm.

Theorem 3. *Under Assumptions 1, 2, and 3, with a proper setting of the pool of candidate step sizes $\mathcal{H}$ and the learning rate ε for the meta-algorithm, our* PBGD *algorithm enjoys the following expected dynamic regret guarantees.*

- *For the one-point feedback model,* PBGD *algorithm satisfies that*

$$\begin{aligned}&\mathbb{E}[\text{D-Regret}_T(\mathbf{u}_1,\ldots,\mathbf{u}_T)] = \mathbb{E}\left[\sum_{t=1}^T f_t(\mathbf{x}_t)\right] - \sum_{t=1}^T f_t(\mathbf{u}_t)\\ &\le \sqrt{2dCR(3L+LR/r)}T^{\frac{3}{4}}\big(2+2\ln(1+\lceil\log_2(1+P_T/(7R))\rceil)+\sqrt{7+(P_T/R)}\big)\\ &= O\big(T^{\frac{3}{4}}(1+P_T)^{\frac{1}{2}}\big).\end{aligned}$$

- *For the two-point feedback model,* PBGD *algorithm satisfies that*

$$\begin{aligned}&\mathbb{E}[\text{D-Regret}_T(\mathbf{u}_1,\ldots,\mathbf{u}_T)] = \mathbb{E}\left[\sum_{t=1}^T \frac{1}{2}\big(f_t(\mathbf{x}_t^{(1)})+f_t(\mathbf{x}_t^{(2)})\big)\right] - \sum_{t=1}^T f_t(\mathbf{u}_t)\\ &\le dLR\sqrt{2T}\big(3+2\ln(1+\lceil\log_2(1+P_T/(7R))\rceil)+\sqrt{7+(P_T/R)}\big)\\ &= O\big(T^{\frac{1}{2}}(1+P_T)^{\frac{1}{2}}\big).\end{aligned}$$

The above results hold universally against any *feasible comparator sequence* $\mathbf{u}_1,\ldots,\mathbf{u}_T \in \mathcal{X}$*.*

Remark 3. Theorem 3 shows that the dynamic regret can be improved from $O\big(T^{\frac{3}{4}}(1+P_T)^{\frac{1}{2}}\big)$ to $O\big(T^{\frac{1}{2}}(1+P_T)^{\frac{1}{2}}\big)$ when it is allowed to query two points at each iteration. The attained dynamic regret (though in expectation) of BCO with two-point feedback, surprisingly, is in the same order with that of the full-information setting (Zhang et al., 2018a). This extends the claim argued by Agarwal et al. (2010) *knowing the value of each loss function at two points is almost as useful as knowing the value of each function everywhere* to dynamic regret analysis. Furthermore, we will show in Theorem 5 in the next section that the obtained dynamic regret for the two-point feedback model is actually minimax optimal.

4.2.1 Proof of Theorem 3 (One-Point Feedback Model)

Proof. As shown in (10), the expected dynamic regret can be decomposed into three terms, consisting of term (a), term (b) and term (c). Concretely, from the analysis of BGD in (40) and (41), we know that the term (b) and term (c) are at most $2L\delta T$ and $(L\delta + L\alpha R)T$ respectively. Hence, it suffices to bound term (a). Since term (a) is over the original loss functions while the algorithm performs over the surrogate loss function, we need to establish their relationship. Indeed, Proposition 1 implies that the term (a) can be upper bounded by

$$\texttt{term (a)} \leq \mathbb{E}\left[\sum_{t=1}^{T} \big(\ell_t(\mathbf{y}_t) - \ell_t(\mathbf{v}_t)\big)\right]. \tag{15}$$

Notably, the quantity inside the expectation is essentially the dynamic regret over the surrogate loss and can be decomposed as

$$\sum_{t=1}^{T} \big(\ell_t(\mathbf{y}_t) - \ell_t(\mathbf{v}_t)\big) = \underbrace{\sum_{t=1}^{T} \ell_t(\mathbf{y}_t) - \sum_{t=1}^{T} \ell_t(\mathbf{y}_t^k)}_{\texttt{meta-regret}} + \underbrace{\sum_{t=1}^{T} \ell_t(\mathbf{y}_t^k) - \sum_{t=1}^{T} \ell_t(\mathbf{v}_t)}_{\texttt{expert-regret}}, \tag{16}$$

where $\mathbf{y}_1^k, \ldots, \mathbf{y}_T^k$ is the prediction sequence returned by the expert k. Note that the above decomposition holds for any expert $k \in [N]$. In the following, we will bound the expert-regret and meta-regret respectively.

First, we examine the expert-regret. The regret decomposition (16) holds for any expert $k \in [N]$, we therefore choose the best expert to obtain a sharpest possible bound. Specifically, due to the boundedness of path-length P_T and the setting of the desired step size $\eta^\dagger$, we can verify that there exists an index $k^* \in \{1, \ldots, N-1\}$ such that $\eta_{k^*} \leq \eta^\dagger \leq \eta_{k^*+1} = 2\eta_{k^*}$ with

$$k^* \leq \left\lceil \frac{1}{2} \log_2 \left(1 + \frac{P_T}{7R}\right) \right\rceil + 1. \tag{17}$$

In other words, the expert k^* is the best expert in the pool in the sense that it has a near-optimal step size η_{k^*} to approximate the unknown step size $\eta^\dagger$. Since each expert performs the deterministic online gradient descent over the surrogate loss, we can apply the existing dynamic regret guarantee of OGD (shown in Theorem 8 of Appendix A) and obtain

$$\begin{aligned}
\texttt{expert-regret} &= \sum_{t=1}^{T} \ell_t(\mathbf{y}_t^{k^*}) - \sum_{t=1}^{T} \ell_t(\mathbf{v}_t) \\
&\leq \frac{7R^2 + RP_T}{4\eta_{k^*}} + \frac{\eta_{k^*} \widetilde{G}^2 T}{2} \\
&\leq \frac{7R^2 + RP_T}{2\eta^\dagger} + \frac{\eta^\dagger d^2 C^2 T}{2\delta^2} \\
&= \sqrt{\frac{(7R^2 + RP_T) dC\widetilde{L}}{2R}} \cdot T^{\frac{3}{4}} + \sqrt{\frac{(7R^2 + RP_T) dC\widetilde{L}}{4R}} \cdot T^{\frac{3}{4}} \\
&= \sqrt{2dC\widetilde{L}(7R + P_T)} \cdot T^{\frac{3}{4}},
\end{aligned} \tag{18}$$

where the first inequality follows from the dynamic regret guarantee of OGD, and $\widetilde{G}$ is the upper bound of gradient norm. Actually, $\widetilde{G} = \sup_{t \in [T]} \|\widetilde{\mathbf{g}}_t\|_2 = dC/\delta$ holds for the one-point gradient estimator (8) by noticing that for all $t \in [T]$,

$$\|\widetilde{\mathbf{g}}_t\|_2 \leq \left\| \frac{d}{\delta} f_t(\mathbf{y}_t + \delta \mathbf{s}_t) \mathbf{s}_t \right\|_2 \overset{(5)}{\leq} \frac{dC}{\delta}. \tag{19}$$

Moreover, the second inequality of (18) holds due to $\eta_{k^*} \leq \eta^\dagger \leq 2\eta_{k^*}$, and the last equality holds by substituting the specific setting of the best possible step size $\eta^\dagger = \sqrt{R(7R^2 + RP_T)/(2dC\widetilde{L})} \cdot T^{-3/4}$ and the perturbation parameter $\delta = \delta^\dagger = (dCR/\widetilde{L})^{1/2}T^{-1/4}$.

Next, we bound the meta-regret. Note that the meta-algorithm is essentially the exponentially weighted average forecaster with nonuniform initial weights, and the magnitude of surrogate loss ℓ_t satisfies

$$|\ell_t(\mathbf{y})| = |\langle \widetilde{\mathbf{g}}_t, \mathbf{y} - \mathbf{y}_t \rangle| \leq \|\widetilde{\mathbf{g}}_t\|_2 \|\mathbf{y} - \mathbf{y}_t\|_2 \overset{(4)}{\leq} 2\widetilde{G}R$$

for any $\mathbf{y} \in (1-\alpha)\mathcal{X}$ and $t \in [T]$. We can thus apply the standard regret guarantee of exponentially weighted average forecaster with nonuniform initial weights (Cesa-Bianchi and Lugosi, 2006, Excercise 2.5) and obtain the following meta-regret bound, whose proof can be found in Appendix A.4.

Lemma 2. *For any step size $\varepsilon > 0$, we have*

$$\sum_{t=1}^T \ell_t(\mathbf{y}_t) - \min_{i \in [N]} \left(\sum_{t=1}^T \ell_t(\mathbf{y}_t^i) + \frac{1}{\varepsilon} \ln \frac{1}{w_1^i} \right) \leq 2\varepsilon T \widetilde{G}^2 R^2.$$

Therefore, by setting $\varepsilon = \sqrt{1/(2T\widetilde{G}^2R^2)}$ to minimize the above upper bound, we obtain that

$$\sum_{t=1}^T \ell_t(\mathbf{y}_t) - \sum_{t=1}^T \ell_t(\mathbf{y}_t^i) \leq \widetilde{G}R\sqrt{2T}\left(1 + \ln \frac{1}{w_1^i}\right)$$

holds for any index $i \in [N]$, where $\widetilde{G}$ is the magnitude of the gradient estimator.

Notice that Lemma 2 holds for any expert $i \in [N]$. In particular, the lemma holds for expert k^* (see its definition in (17)), so we have

$$\begin{aligned} \texttt{meta-regret} &= \sum_{t=1}^T \ell_t(\mathbf{y}_t) - \sum_{t=1}^T \ell_t(\mathbf{y}_t^{k^*}) \\ &\leq \widetilde{G}R\sqrt{2T}\left(1 + \ln(1/w_1^{k^*})\right) \\ &\leq \frac{dCR}{\delta}\sqrt{2T}\big(1 + 2\ln(k^*+1)\big) \\ &= \sqrt{2dCR\widetilde{L}}T^{3/4}\big(1 + 2\ln(k^*+1)\big) \end{aligned} \tag{20}$$

By combining upper bounds of expert-regret (18) and meta-regret (20), we conclude that the term (a) of (10) is at most

$$\begin{aligned} \texttt{term (a)} &= \mathbb{E}\left[\sum_{t=1}^T \widehat{f}_t(\mathbf{y}_t) - \sum_{t=1}^T \widehat{f}_t(\mathbf{v}_t)\right] \\ &\overset{(15)}{\leq} \mathbb{E}\left[\sum_{t=1}^T \ell_t(\mathbf{y}_t) - \sum_{t=1}^T \ell_t(\mathbf{v}_t)\right] \end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\mathbf{y}_t) - \sum_{t=1}^{T} \ell_t(\mathbf{y}_t^{k^*})\right] + \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\mathbf{y}_t^{k^*}) - \sum_{t=1}^{T} \ell_t(\mathbf{v}_t)\right] \\
&\overset{(18),\,(20)}{\leq} \sqrt{2dCR\widetilde{L}}T^{3/4}\big(1 + 2\ln(k^*+1) + \sqrt{7+(P_T/R)}\big)
\end{aligned}
$$

which in conjunction with upper bounds of term (b) and term (c) in (40) and (41) finally yields the expected dynamic regret bound as follows,

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \\
&\overset{(10)}{=} \texttt{term (a)} + \texttt{term (b)} + \texttt{term (c)} \\
&\leq \texttt{term (a)} + 2L\delta T + (L\delta + L\alpha R)T \\
&\leq \sqrt{2dCR\widetilde{L}}T^{3/4}\big(1 + 2\ln(k^*+1) + \sqrt{7+(P_T/R)}\big) + (3L + LR/r)\sqrt{dCR/\widetilde{L}}T^{3/4} \\
&\leq \sqrt{2dCR\widetilde{L}}T^{3/4}\big(2 + 2\ln(k^*+1) + \sqrt{7+(P_T/R)}\big) \\
&\leq \sqrt{2dCR(3L+LR/r)}T^{\frac{3}{4}}\big(2 + 2\ln(1 + \lceil\log_2(1+P_T/(7R))\rceil) + \sqrt{7+(P_T/R)}\big) \\
&= O\big(T^{3/4}(1+P_T)^{1/2}\big),
\end{aligned}
$$

where the last inequality plugs in the setting of perturbation parameter $\delta = (dCR/\widetilde{L})^{1/2}T^{-1/4}$ and makes use of the upper bound of index k^* in (17). □

4.2.2 Proof of Theorem 3 (Two-Point Feedback Model)

In this part, we first present the configuration of the step size pool $\mathcal{H}$ for the two-point feedback model, and then provide the proof of dynamic regret.

In the two-point feedback model, the optimal step size is $\eta^* = \sqrt{\frac{7R^2+RP_T}{2d^2L^2T}}$, and we know

$$
\sqrt{\frac{7R^2}{2d^2L^2T}} \leq \eta^* \leq \sqrt{\frac{7R^2+2R^2T}{2d^2L^2T}}
$$

always holds due to $0 \leq P_T \leq 2RT$ (by the boundedness assumption of the feasible set, shown in Assumption 1). Hence, we will construct the following pool of candidate step sizes,

$$
\mathcal{H} = \left\{\eta_i = 2^{i-1}\sqrt{\frac{7R^2}{2d^2L^2T}} \mid i = 1, \ldots, N\right\},
$$

where N denotes the number of candidate step sizes, set as

$$
N = \left\lceil \frac{1}{2}\log_2\left(1 + \frac{2T}{7}\right)\right\rceil + 1.
$$

Based on the configurations, we proceed to present the proof of Theorem 3 for the two-point feedback model.

Proof. The proof is analogous to that of the one-point feedback model, where the main differences lie in two quantities: the index of optimal expert k^*, and the magnitude of the gradient estimator $\widetilde{G}$. In the two-point feedback model, we can ensure that the index of best expert k^* is at most

$$k^* \leq \left\lceil \frac{1}{2} \log_2 \left(1 + \frac{P_T}{7R}\right) \right\rceil + 1 \tag{21}$$

and the associated step size satisfies $\eta_{k^*} \leq \eta^* \leq \eta_{k+1}$. Besides, the magnitude of the gradient estimator $\widetilde{G} = \sup_{t\in[T]} \|\widetilde{\mathbf{g}}_t\|_2 = dL$ holds for two-point gradient estimator (9) by noticing that for all $t \in [T]$,

$$\begin{aligned} \|\widetilde{\mathbf{g}}_t\|_2 &= \frac{d}{2\delta} \big\| \big(f_t(\mathbf{y}_t + \delta\mathbf{s}_t) - f_t(\mathbf{y}_t - \delta\mathbf{s}_t)\big)\mathbf{s}_t \big\|_2 \\ &= \frac{d}{2\delta} |f_t(\mathbf{y}_t + \delta\mathbf{s}_t) - f_t(\mathbf{y}_t - \delta\mathbf{s}_t)| \\ &\overset{(6)}{\leq} \frac{dL}{2\delta} \|2\delta\mathbf{s}_t\|_2 = dL. \end{aligned} \tag{22}$$

In above, to obtain the last inequality we use the Lipschitz property by Assumption 3. We remark that in stark contrast to that in the one-point feedback model as shown in (19), the upper bound of gradient norm $\widetilde{G}$ here is *independent* of the $1/\delta$, which leads to substantially improved bounds for both the static regret analysis (Agarwal et al., 2010) and the dynamic regret analysis presented in this paper.

For the two-point feedback model, its expected dynamic regret can be decomposed as

$$\begin{aligned} & \mathbb{E}\left[\sum_{t=1}^T \frac{1}{2}\big(f_t(\mathbf{x}_t^{(1)}) + f_t(\mathbf{x}_t^{(2)})\big)\right] - \sum_{t=1}^T f_t(\mathbf{u}_t) \\ = {} & \mathbb{E}\left[\sum_{t=1}^T \frac{1}{2}\big(f_t(\mathbf{y}_t + \delta\mathbf{s}_t) + f_t(\mathbf{y}_t - \delta\mathbf{s}_t)\big)\right] - \sum_{t=1}^T f_t(\mathbf{u}_t) \\ \overset{(44)}{\leq} {} & \mathbb{E}\left[\sum_{t=1}^T f_t(\mathbf{y}_t)\right] + L\delta T - \sum_{t=1}^T f_t(\mathbf{u}_t) \\ = {} & \underbrace{\mathbb{E}\left[\sum_{t=1}^T \widehat{f}_t(\mathbf{y}_t) - \sum_{t=1}^T \widehat{f}_t(\mathbf{v}_t)\right]}_{\texttt{term (a)}} + \delta LT + \underbrace{\mathbb{E}\left[\sum_{t=1}^T \Big(f_t(\mathbf{y}_t) - \widehat{f}_t(\mathbf{y}_t)\Big)\right]}_{\texttt{term (b)}} + \underbrace{\left[\sum_{t=1}^T \Big(\widehat{f}_t(\mathbf{v}_t) - f_t(\mathbf{u}_t)\Big)\right]}_{\texttt{term (c)}}, \end{aligned} \tag{23}$$

where the first inequality exploits the Lipschitz property of the loss function. So we will bound the three terms individually.

First, the expert-regret is upper bounded by

$$\begin{aligned}\texttt{expert-regret} &= \sum_{t=1}^{T} \ell_t(\mathbf{y}_t^{k^*}) - \sum_{t=1}^{T} \ell_t(\mathbf{v}_t)\\ &\leq \frac{7R^2 + RP_T}{4\eta_{k^*}} + \frac{\eta_{k^*}\widetilde{G}^2 T}{2}\\ &\leq \frac{7R^2 + RP_T}{2\eta^*} + \frac{\eta^* d^2 L^2 T}{2}\\ &= \frac{3\sqrt{2}}{4} dL\sqrt{T(7R^2 + RP_T)},\end{aligned}$$

where the last equation is obtained by plugging the parameter setting of $\eta^* = \sqrt{\frac{7R^2+RP_T}{2d^2L^2T}}$.

Besides, the meta-regret is bounded by

$$\begin{aligned}\texttt{meta-regret} &= \sum_{t=1}^{T} \ell_t(\mathbf{y}_t) - \sum_{t=1}^{T} \ell_t(\mathbf{y}_t^{k^*})\\ &\leq \widetilde{G}R\sqrt{2T}\left(1 + \ln(1/w_1^{k^*})\right)\\ &\leq dLR\sqrt{2T}\big(1 + 2\ln(k^* + 1)\big).\end{aligned}$$

Therefore, by combining upper bounds of meta-regret and expert-regret, we have

$$\begin{aligned}\texttt{term (a)} &= \mathbb{E}\left[\sum_{t=1}^{T} \widehat{f}_t(\mathbf{y}_t) - \sum_{t=1}^{T} \widehat{f}_t(\mathbf{v}_t)\right]\\ &\overset{(15)}{\leq} \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\mathbf{y}_t) - \sum_{t=1}^{T} \ell_t(\mathbf{v}_t)\right]\\ &= \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\mathbf{y}_t) - \sum_{t=1}^{T} \ell_t(\mathbf{y}_t^{k^*})\right] + \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\mathbf{y}_t^{k^*}) - \sum_{t=1}^{T} \ell_t(\mathbf{v}_t)\right]\\ &\leq \frac{3\sqrt{2}}{4} dL\sqrt{T(7R^2 + RP_T)} + dLR\sqrt{2T}\big(1 + 2\ln(k^* + 1)\big)\\ &\leq dLR\sqrt{2T}\left(1 + 2\ln(k^* + 1) + \sqrt{7 + (P_T/R)}\right)\end{aligned}$$

which in conjunction with upper bounds of term (b) and term (c) in (40) and (41) finally yields the expected dynamic regret bound as follows,

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{T} \frac{1}{2}\big(f_t(\mathbf{x}_t^{(1)}) + f_t(\mathbf{x}_t^{(2)})\big)\right] - \sum_{t=1}^{T} f_t(\mathbf{u}_t)\\ &\overset{(23)}{\leq} \texttt{term (a)} + \texttt{term (b)} + \texttt{term (c)} + L\delta T\\ &\leq \texttt{term (a)} + L\delta T + (L\delta + L\alpha R)T + L\delta T\\ &\leq dLR\sqrt{2T}\big(1 + 2\ln(k^* + 1)\big) + \frac{3\sqrt{2}}{4} dL\sqrt{T(7R^2 + RP_T)} + dLR\sqrt{T}\end{aligned}$$

$$
\begin{aligned}
&\le dLR\sqrt{2T}\big(2 + 2\ln(1 + \lceil \log_2(1 + P_T/(7R))\rceil) + \sqrt{7 + (P_T/R)}\big)\\
&= O\big(T^{1/2}(1 + P_T)^{1/2}\big).
\end{aligned}
$$

where the last inequality plugs in the setting of perturbation parameter $\delta = dLR/(\widetilde{L}\sqrt{T})$ with $\widetilde{L} = 3L + LR/r$ called the effective Lipschitz constant, and the last equation makes use of the upper bound of index k^* in (21). □

4.3 Extension to Anytime Algorithm

Notice that the proposed PBGD algorithm requires the time horizon T as an input, which is not available in advance especially for the streaming scenarios and thus makes the algorithm hard to deploy. In this part, we present a simple approach to remove the undesired dependence and develop an *anytime* variant, that is, an online algorithm without requiring the time horizon in advance.

Our method is essentially a standard implementation of the doubling trick (Cesa-Bianchi et al., 1997). The idea is to make a guess on the time horizon T. Once the actual number of iterations exceeds the guess, double the guess and restart the algorithm. The initial guess of the time horizon is set as 2. So there will be $K = \lfloor \log_2 T \rfloor + 1$ epochs and the i-th epoch contains 2^i iterations. We have following regret guarantees for the above anytime algorithm.

Theorem 4. *Under the same conditions with Theorem 3, the anytime version of* PBGD *(via the above strategy of doubling trick) enjoys the following expected dynamic regret,*

- *One-Point Feedback Model:* $O\big(T^{\frac{3}{4}}(\log T + P_T)^{\frac{1}{2}}\big)$;
- *Two-Point Feedback Model:* $O\big(T^{\frac{1}{2}}(\log T + P_T)^{\frac{1}{2}}\big)$.

The above results hold universally against any *feasible comparator sequence* $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$.

Proof. We take BCO with one-point feedback model as an example and provide a brief analysis as follows. Actually, by the strategy of doubling trick, we can bound the dynamic regret of the anytime algorithm by

$$
\begin{aligned}
\sum_{i=1}^{K} T_i^{\frac{3}{4}}(1 + P_i)^{\frac{1}{2}} &\le \sqrt{\sum_{i=1}^{K} T_i^{\frac{3}{2}}}\sqrt{\sum_{i=1}^{K}(1 + P_i)}\\
&= \sqrt{\sum_{i=1}^{K} 2^{\frac{3i}{2}}}\sqrt{\log T + P_T}\\
&= O\big(T^{\frac{3}{4}}(\log T + P_T)^{\frac{1}{2}}\big).
\end{aligned}
$$

The result for two-point BCO can be similarly proved. □

Compared with the $O(T^{3/4}(1 + P_T)^{1/2})$ rate of the original "non-anytime" PBGD algorithm, we observe that an extra $\log T$ term is suffered due to the anytime demand.

5. Lower Bound

In this section, we establish the lower bound of the expected universal dynamic regret for bandit convex optimization problems.

Theorem 5. *For any algorithm designed for the one-point feedback BCO and any real value $\tau \in [0, 2T]$, there exists a sequence of loss functions $f_1, \ldots, f_T$ that satisfy Assumptions 1 and 3 with feasible set $\mathcal{X} \subseteq \mathbb{B}$ and Lipschitzness $L = 4$, and a sequence of comparators $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$ whose path-length P_T is less than τ, such that the outputs of the algorithm denoted by $\mathbf{x}_1, \ldots, \mathbf{x}_T$ satisfies that*

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)\right] \geq C_1 d\sqrt{\tau T}, \tag{24}$$

where C_1 is a positive constant independent of time horizon T and dimension d, and the expectation is taken over the randomness of both the algorithm and the loss functions.

Similarly, for any algorithm designed for the two-point feedback BCO and any real value $\tau \in [0, 2T]$, there exists a sequence of loss functions $f_1, \ldots, f_T$ that satisfy Assumptions 1 and 3 with feasible set $\mathcal{X} \subseteq \mathbb{B}$ and Lipschitzness $L = 1$, and a sequence of comparators $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$ whose path-length P_T is less than τ, such that the outputs of the algorithm denoted by $(\mathbf{x}_1^{(1)}, \mathbf{x}_1^{(2)}), \ldots, (\mathbf{x}_T^{(1)}, \mathbf{x}_T^{(2)})$ satisfies that

$$\mathbb{E}\left[\sum_{t=1}^{T} \frac{1}{2}\big(f_t(\mathbf{x}_t^{(1)}) + f_t(\mathbf{x}_t^{(2)})\big) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)\right] \geq C_2\sqrt{d\tau T}, \tag{25}$$

where C_2 is a positive constant independent of time horizon T and dimension d, and the expectation is taken over the randomness of both the algorithm and the loss functions.

Proof. For a given $\tau \in [0, 2T]$, we design a piecewise-stationary comparator sequence, whose path-length is constructed to be smaller than τ. Then, we can split the whole time horizon into several pieces, where comparators are fixed in each piece. So we can appeal to the established minimax lower bound of BCO in terms of static regret (Shamir, 2013; Duchi et al., 2015) in each piece, and finally sum over all pieces to obtain the lower bound of dynamic regret.

Specifically, for the one-point feedback model, based on Theorem 7 of Shamir (2013), for any algorithm, we can always find a sequence of loss functions $f_1, \ldots, f_T$ that satisfy Assumptions 1 and 3 with feasible set $\mathcal{X} \subseteq \mathbb{B}$ and Lipschitzness $L = 4$, such that the static regret satisfies that

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x})\right] \geq C_3 d\sqrt{T},$$

where C_3 is a constant independent of T. Then, for the case $\tau \leq 2$, we can always find a comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$, such that

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)\right] &\geq \mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x})\right] \\ &\geq C_3 d\sqrt{T} \geq C_1 d\sqrt{\tau T},\end{aligned}$$

where $C_1 = C_3/\sqrt{2}$ is a constant. Next, we consider the case $\tau \geq 2$. Without loss of generality, we assume $\lceil \tau \rceil$ divides T, and let $K = T/\lceil \tau \rceil$ be the length of each piece. To proceed, we construct the following piecewise-stationary comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T$: for any $i \in [1, \lceil \tau \rceil]$, denote by $\mathcal{I}_i = [(i-1)K+1, iK]$ the i-th piece for $i = 1, \ldots, \lceil \tau \rceil$, the comparators within the interval are set as

$$\mathbf{u}_{(i-1)K+1} = \mathbf{u}_{(i-1)K+2} = \cdots = \mathbf{u}_{iK} = \operatorname*{arg\,min}_{\mathbf{u} \in \mathcal{X}} \sum\nolimits_{t \in \mathcal{I}_i} f_t(\mathbf{u}). \tag{26}$$

Note that the path-length of this sequence does not exceeds τ. Thus, the dynamic regret competing with the comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T$ can be evaluated as,

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{u}_t)\right] &= \mathbb{E}\left[\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{i=1}^{\lceil \tau \rceil} \min_{\mathbf{x} \in \mathcal{X}} \sum\nolimits_{t \in \mathcal{I}_i} f_t(\mathbf{x})\right] \\ &\geq \lceil \tau \rceil C_3 d\sqrt{K} = C_3 d\sqrt{T\lceil \tau \rceil} \geq C_1 d\sqrt{T\tau}.\end{aligned}$$

For the two-point feedback model, based on Proposition 2 of Duchi et al. (2015), for any algorithm, we can always find a sequence of loss functions $f_1, \ldots, f_T$ that satisfy Assumptions 1 and 3 with feasible set $\mathcal{X} \subseteq \mathbb{B}$ and Lipschitzness $L = 1$, such that

$$\mathbb{E}\left[\sum_{t=1}^T \frac{1}{2}\big(f_t(\mathbf{x}_t^{(1)}) + f_t(\mathbf{x}_t^{(2)})\big) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x})\right] \geq C_4\sqrt{dT},$$

where C_4 is a constant independent of T. For the case $\tau \leq 2$, we can similarly obtain

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^T \frac{1}{2}\big(f_t(\mathbf{x}_t^{(1)}) + f_t(\mathbf{x}_t^{(2)})\big) - \sum_{t=1}^T f_t(\mathbf{u}_t)\right] \\ \geq\, &\mathbb{E}\left[\sum_{t=1}^T \frac{1}{2}\big(f_t(\mathbf{x}_t^{(1)}) + f_t(\mathbf{x}_t^{(2)})\big) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x})\right] \\ \geq\, &C_4\sqrt{dT} \geq C_2\sqrt{d\tau T}\end{aligned}$$

where $C_2 = C_4/\sqrt{2}$ is a constant. For the case $\tau \geq 2$, we can also construct the comparators similar to (26) and ensure that

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^T \frac{1}{2}\big(f_t(\mathbf{x}_t^{(1)}) + f_t(\mathbf{x}_t^{(2)})\big) - \sum_{t=1}^T f_t(\mathbf{u}_t)\right] \\ =\, &\mathbb{E}\left[\sum_{t=1}^T \frac{1}{2}\big(f_t(\mathbf{x}_t^{(1)}) + f_t(\mathbf{x}_t^{(2)})\big) - \sum_{i=1}^{\lceil \tau \rceil} \min_{\mathbf{x} \in \mathcal{X}} \sum\nolimits_{t \in \mathcal{I}_i} f_t(\mathbf{x})\right] \\ \geq\, &\lceil \tau \rceil C_4\sqrt{dK} = C_4\sqrt{dT\lceil \tau \rceil} \geq C_2\sqrt{dT\tau}.\end{aligned}$$

Hence, we complete proofs of lower bounds of both one-point and two-point BCO models. □

Remark 4. From the above lower bounds and the upper bounds in Theorem 3, we know that our dynamic regret for the two-point feedback model is minimax optimal in terms of the dependence on time horizon T and path-length P_T; while the rate for one-point feedback model remains sub-optimal. Notice that our lower bound $\Omega(\sqrt{TP_T})$ implies that no algorithm can achieve sub-linear regret unless the path-length P_T is asymptotically smaller than T. Consequently, we are interested in the instances with path-length $P_T = o(T)$, for example, $P_T = \sqrt{T}$. We note that under such circumstances the desired upper bound $O(T^{3/4}(1 + P_T)^{1/4})$ (as demonstrated in Remark 1) does not contradict with the minimax lower bound, since $O(T^{3/4}(1 + P_T)^{1/4}) = O(T^{1/2}T^{1/4}(1 + P_T)^{1/4})$ is larger than the $\Omega(T^{1/2}(1 + P_T)^{1/2})$ lower bound. Furthermore, we point out the challenge in deriving a tighter bound for one-point BCO problems. Our attained $O(T^{3/4}(1 + P_T)^{1/2})$ dynamic regret bound exhibits a square-root dependence on the path-length, and it becomes vacuous when $P_T \geq \sqrt{T}$, though the path-length is arguably small (otherwise, the problem would be too hard to learn as indicated by the lower bound). The challenge is that our online ensemble technique cannot support to grid search and approximate the optimal perturbation δ^*, which also depends on the unknown path-length P_T. Otherwise, we have to query the function value more than once per iteration, which is prohibited in the bandit setting.

Remark 5. The above remark mainly focuses on the regret dependence on the time horizon T and the path-length P_T, and here we further discuss the dependence on the dimension d. We first examine the two-point BCO: our approach enjoys an $O(d\sqrt{TP_T})$ dynamic regret bound as shown in Theorem 3, while the lower bound is $O(\sqrt{dTP_T})$. It is interesting to investigate how to eliminate the gap in terms of dimension d, and the techniques of (Shamir, 2017) might be useful. Next, we inspect the one-point BCO: our approach achieves an $O(d^{1/2}T^{3/4}P_T^{1/2})$ dynamic regret bound as shown in Theorem 3, while the lower bound is $O(d\sqrt{TP_T})$. At first glance this is kind of be strange that the upper bound exhibits a square-root dependence on dimension, even smaller than the lower bound. We argue that this is reasonable as our approach employs the framework of online gradient descent with the unbiased gradient estimation (Flaxman et al., 2005), which usually yields a low dimension-dependence at the cost of a sub-optimal dependence on time horizon T. Indeed, the same framework also delivers an $O(d^{1/2}T^{3/4})$ static regret (Flaxman et al., 2005, Theorem 3.3). On the other hand, existing literatures for minimizing static regret of BCO problems show that the dimension-dependence would typically become larger when one pursues a milder dependence on T based on more sophisticated techniques. For example, the seminal work of Bubeck et al. (2017) used kernel-based gradient estimators to design an BCO algorithm with $O(d^{9.5}\sqrt{T}(\log T)^{7.5})$ static regret, and recent breakthrough (Lattimore, 2020) improved the result to $O(d^{2.5}\sqrt{T}\log T)$ by an information-theoretic argument.

Remark 6. Note that an $\Omega(\sqrt{T(1 + P_T)})$ lower bound of *deterministic* universal dynamic regret for the full-information OCO was established in the previous work (Zhang et al., 2018a, Theorem 2), which naturally implies the same lower bound of deterministic dynamic regret for BCO problems since BCO is evidently harder than the full-information OCO. However, the aforementioned result does not imply a lower bound in terms of the *expected* dynamic regret, because the expected dynamic regret is a weaker performance metric than the deterministic one. Our result holds for the expected dynamic regret and demonstrates that even for this easier measure, its minimax rate for BCO problems is still no less than the order of $\Omega(\sqrt{T(1+P_T)})$, which implies the hardness of learning with bandit feedback. Moreover, for the one-point BCO problems, the lower bound (24) holds even when all the online functions are strongly convex and smooth, because the minimax static regret of BCO with one-point feedback can neither benefit from strongly convexity nor smoothness (Shamir, 2013). This is to be contrasted to the full-information setting (Hazan et al., 2007), which also implies the hardness of the bandit online learning.

6. Adaptive Regret

Aside from dynamic regret, *adaptive regret* is another performance measure used to guide the algorithm designs for online learning in non-stationary environments. In this section, we develop online algorithms to minimize the adaptive regret for bandit convex optimization problems. Following the seminal work of Hazan and Seshadhri (2009), we define the expected adaptive regret for BCO as

$$\mathbb{E}[\text{A-Regret}_T] = \max_{[q,s]\subseteq[T]} \left(\mathbb{E}\left[\sum_{t=q}^{s} f_t(\mathbf{x}_t)\right] - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=q}^{s} f_t(\mathbf{x}) \right).$$

We note that, in the full-information setting, a stronger version of adaptive regret named strongly adaptive regret is introduced by Daniely et al. (2015), which aims to strengthen the adaptive regret guarantee for small intervals. Nevertheless, the authors have proved that it is impossible to achieve meaningful strongly adaptive regret in the *bandit* setting (Daniely et al., 2015, Section 4), so we focus on the notion defined by Hazan and Seshadhri (2009).

6.1 Algorithm

In this part, we present our algorithm for minimizing the adaptive regret in bandit convex optimization. The proposed method is based on the Coin-Betting for Changing Environment (CBCE) algorithm (Jun et al., 2017), which is primely developed for minimizing the adaptive regret in the full-information setting. We first give a brief introduction of the CBCE algorithm, and then describe how to adapt it to the BCO setting.

The CBCE algorithm mainly consists of three parts:

- An expert-algorithm, which can minimize the static regret of a given interval;
- A set of intervals, each of which is associated with an expert-algorithm that minimizes the static regret of that interval. At iteration t, an expert will be activated only when its interval includes t. For an interval I, we denote by E_I the expert assigned to I, and by q_{E_I} and s_{E_I} its starting and ending points;
- A meta-algorithm, which combines the predictions of active experts at each iteration.

For the expert-algorithm, CBCE chooses the standard online gradient decent (OGD) algorithm. In particular, for a given interval $I = [q, s] \subseteq [T]$, OGD with step size proportional to $O(1/\sqrt{t-q+1})$ can achieve the optimal $O(\sqrt{|I|})$ static regret.

For the set of intervals, CBCE constructs a set of geometric covering (GC) intervals (Daniely et al., 2015), defined as

$$\mathcal{I} = \bigcup_{k\in\mathbb{N}\cup\{0\}} \mathcal{I}_k \tag{27}$$

where each interval is defined as $\mathcal{I}_k = \{[i\cdot 2^k, (i+1)\cdot 2^k - 1] : i \in \mathbb{N}\}$ for any $k \in \mathbb{N}\cup\{0\}$. We denote the set of active intervals at iteration t as $\mathcal{C}_t = \{I \mid t \in I, I \in \mathcal{I}\}$, and the set of active experts $\mathcal{A}_t = \{E_I \mid I \in \mathcal{C}_t\}$. It can be shown that $|\mathcal{A}_t| = O(\log t)$.

For the meta-algorithm, CBCE uses the Sleeping Coin-Betting (SCB) algorithm. Specifically, at iteration t, each active expert E_I is assigned with a coin flip $\widetilde{g}_{t,E_I}$, a wealth w_{t,E_I}, and a weight $\widehat{p}_{t,E_I}$. At the beginning of each iteration, SCB algorithm first configures w_{t,E_I} for each active expert as

$$w_{t,E_I} = \frac{\sum_{\tau=q_{E_I}}^{t-1} \widetilde{g}_{\tau,E_I}}{t - q_{E_I} + 1}$$

and sets $\widehat{p}_{t,E_I} = \pi_{E_I}\max\{w_{t,E_I}, 0\}$, where π_{E_I} is the prior weight of E_I. After that, SCB algorithm computes

$$p_{t,E_I} = \begin{cases} \frac{\widehat{p}_{t,E_I}}{\sum_{E_I\in\mathcal{A}_t}\widehat{p}_{t,E_I}}, & \sum_{E_I\in\mathcal{A}_t}\widehat{p}_{t,E_I} > 0 \\ \pi_{E_I}, & \text{otherwise.} \end{cases}$$

Next, the algorithm receives the decision of each expert $\mathbf{x}_{t,E_I}$, and picks the final decision $\mathbf{x}_t$ by $\mathbf{x}_t = \sum_{E_I\in\mathcal{A}_t}\mathbf{x}_{t,E_I}p_{t,E_I}$. Finally, after observing the loss function $f_t(\cdot)$, $\widetilde{g}_{t,E_I}$ is updated by

$$\widetilde{g}_{t,E_I} = \mathbb{1}_{w_{t,E_I}>0}(f_t(\mathbf{x}_t) - f_t(\mathbf{x}_{t,E_I})) + \mathbb{1}_{w_{t,E_I}\leq 0}\max\{f_t(\mathbf{x}_t) - f_t(\mathbf{x}_{t,E_I})\}.$$

It can be shown that CBCE can achieve an $O(\sqrt{T\log T})$ adaptive regret bound in the full-information setting. However, it is prohibited to directly apply this algorithm to the bandit scenario by simply making use of the estimated gradients and BGD, because the CBCE algorithm requires to query the loss function $O(\log t)$ times at each iteration t, which is not allowed in the bandit setup. To address this issue, we follow the same idea of the development of dynamic regret for BCO problems, presented in Section 4. Concretely, we introduce the surrogate loss function ℓ_t (defined in (28) and (29) for different feedback models), whose function values as well as gradients can be computed by only using $f_t(\mathbf{x}_t)$ (or $f_t(\mathbf{x}_t^{(1)})$ and $f_t(\mathbf{x}_t^{(2)})$ for the two-point feedback model), without further queries of the loss function. We then deploy the exact CBCE with OGD on the sequence of surrogate loss functions $\ell_1, \ldots, \ell_T$, which now can be considered as a full-information problem. The algorithmic details are summarized in Algorithm 3 (the meta-algorithm, i.e., CBCE) and Algorithm 4 (the expert-algorithm, i.e., OGD). Based on the relationships between the surrogate loss ℓ_t and the original loss f_t, our proposed algorithm finally minimizes the expected adaptive regret on the sequence of original loss functions $f_1, \ldots, f_T$.

6.2 Adaptive Regret Analysis

We provide theoretical guarantees in Theorem 6 (one-point BCO) and Theorem 7 (two-point BCO) as follows, whose proof are presented in Section 6.2.1 and Section 6.2.1, respectively.

Algorithm 3 MABCO: Meta-algorithm

```
Input: time horizon $T$, perturbation parameter $\delta$, shrinkage parameter $\alpha$
Initialize GC interval $\mathcal{I}$ as (27), active expert set $\mathcal{A}_1$, and prior weight of each expert $E_I$:
    $p_{1,E_I} = \pi_{E_I} = \frac{6}{\pi^2}\left(\frac{1}{q_{E_I}(1+\lfloor 1+\log_2 q_{E_I}\rfloor)}\right).$
for $t = 1$ to $T$ do
    Receive $\mathbf{y}_{t,E_I}$ from each expert $E_I \in \mathcal{A}_t$
    Obtain $\mathbf{y}_t = \sum_{E_I \in \mathcal{A}_t} \mathbf{y}_{t,E_I} p_{t,E_I}$
    Select a unit vector $\mathbf{s}_t$ uniformly at random
    {Case 1. One-Point Feedback Model}
    Submit $\mathbf{x}_t = \mathbf{y}_t + \delta\mathbf{s}_t$
    Receive $f_t(\mathbf{x}_t)$ as the feedback
    Compute gradient estimator $\widetilde{\mathbf{g}}_t$ by (8), namely, $\widetilde{\mathbf{g}}_t = \frac{d}{\delta} f_t(\mathbf{x}_t) \cdot \mathbf{s}_t$
    Construct surrogate loss $\ell_t(\cdot)$ as (28)
    {Case 2. Two-Point Feedback Model}
    Submit $\mathbf{x}_t^{(1)} = \mathbf{y}_t + \delta\mathbf{s}_t$ and $\mathbf{x}_t^{(2)} = \mathbf{y}_t - \delta\mathbf{s}_t$
    Receive $f_t(\mathbf{x}_t^{(1)})$ and $f_t(\mathbf{x}_t^{(2)})$ as the feedback
    Compute gradient estimator $\widetilde{\mathbf{g}}_t$ by (9), namely, $\widetilde{\mathbf{g}}_t = \frac{d}{2\delta}(f_t(\mathbf{x}_t^{(1)}) - f_t(\mathbf{x}_t^{(2)})) \cdot \mathbf{s}_t$
    Construct surrogate loss $\ell_t(\cdot)$ as (29)
    [either one-point or two-point model] Update the weight for each expert $E_I \in \mathcal{A}_{t+1}$ by
        $p_{t+1,E_I} = \begin{cases} \frac{\widehat{p}_{t+1,E_I}}{\sum_{E_I \in \mathcal{A}_{t+1}} \widehat{p}_{t+1,E_I}}, & \sum_{E_I \in \mathcal{A}_{t+1}} \widehat{p}_{t+1,E_I} > 0 \\ \pi_{E_I}, & \text{otherwise} \end{cases}$
    where $\widehat{p}_{t+1,E_I} = \pi_{E_I} \max\{w_{t+1,E_I}, 0\}$, $w_{t+1,E_I} = \frac{\sum_{\tau=q_{E_I}}^{t} \widetilde{g}_{\tau,E_I}}{t-q_{E_I}}$, and
        $\widetilde{g}_{t,E_I} = \mathbb{1}_{w_{t,E_I}>0}(\ell_t(\mathbf{y}_t) - \ell_t(\mathbf{y}_{t,E_I})) + \mathbb{1}_{w_{t,E_I}\le 0} \max\{\ell_t(\mathbf{y}_t) - \ell_t(\mathbf{y}_{t,E_I})\}$
    Send the gradient estimator $\widetilde{\mathbf{g}}_t$ to each expert
end for
```

Algorithm 4 MABCO: Expert-algorithm

```
Let $\widehat{G} = \max_{\mathbf{y} \in (1-\alpha)\mathcal{X}, t \in [T]} \|\nabla \ell_t(\mathbf{y})\|_2$.
if $q_{E_I} = t$ then
    $\mathbf{y}_{t,E_I} = 0$
else
    $\mathbf{y}_{t,E_I} = \Pi_{(1-\alpha)\mathcal{X}}\left[\mathbf{y}_{t-1,E_I} - \frac{R}{\widehat{G}\sqrt{t-q_{E_I}+1}} \nabla \ell_{t-1}(\mathbf{y}_{t-1,E_I})\right]$
end if
```

Theorem 6 (one-point feedback model)**.** *Under Assumptions 1, 2, and 3, define the surrogate loss function $\ell_t : (1-\alpha)\mathcal{X} \mapsto \mathbb{R}$ as*

$$\ell_t(\mathbf{y}) = \frac{1}{2G^{one}R}\langle \widetilde{\mathbf{g}}_t, \mathbf{y} - \mathbf{y}_t\rangle + \frac{1}{2} \tag{28}$$

where $G^{one} = dC/\delta$ and $\widetilde{\mathbf{g}}_t$ is the gradient estimator defined in (8)*. Let Algorithm 3 be the meta-algorithm, which is fed with $\ell_1, \ldots, \ell_T$ as loss functions, and Algorithm 4 be the expert-algorithm. Set the perturbation parameter δ as in* (32) *and shrinkage parameter $\alpha = \delta/r$. Then the expected adaptive regret satisfies*

$$\mathbb{E}[\text{A-Regret}_T] \leq \sqrt{Cd\big(15R\sqrt{T} + 16R\sqrt{7\log T + 5}\sqrt{T}\big)\big(3LT + \frac{LR}{r}T\big)} = O\big(T^{\frac{3}{4}}(\log T)^{\frac{1}{4}}\big).$$

Theorem 7 (two-point feedback model)**.** *Under Assumptions 1, 2, and 3, define the surrogate loss function $\ell_t : (1-\alpha)\mathcal{X} \mapsto \mathbb{R}$ as*

$$\ell_t(\mathbf{y}) = \frac{1}{2G^{two}R}\langle \widetilde{\mathbf{g}}_t, \mathbf{y} - \mathbf{y}_t\rangle + \frac{1}{2} \tag{29}$$

where $G^{two} = dL$ and $\widetilde{\mathbf{g}}_t$ is the gradient estimator defined in (9)*. Let Algorithm 3 be the meta-algorithm, which is fed with $\ell_1, \ldots, \ell_T$ as loss functions, and Algorithm 4 be the expert-algorithm. Set the perturbation parameter $\delta = dR/\sqrt{T}$ and shrinkage parameter $\alpha = \delta/r$. Then the expected adaptive regret satisfies*

$$\mathbb{E}[\text{A-Regret}_T] \leq dLR\left(19\sqrt{T} + 16\sqrt{7\log T + 5}\sqrt{T}\right) + \frac{dLR^2}{r}\sqrt{T} = O\big(T^{\frac{1}{2}}(\log T)^{\frac{1}{2}}\big).$$

Note that we cannot hope for an adaptive regret that is better than the static regret. The adaptive regret bounds in Theorem 6 and Theorem 7 match the $O(T^{3/4})$ and $O(T^{1/2})$ static regret bounds for the BGD methods of one-point (Flaxman et al., 2005) and two-point (Agarwal et al., 2010) feedback models, up to logarithmic factors.

6.2.1 Proof of Theorem 6

Proof. Similar to the analysis of dynamic regret, for any time interval $I = [q, s] \subseteq [T]$, the adaptive regret can be decomposed into three terms as follows.

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=q}^{s} f_t(\mathbf{x}_t)\right] - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=q}^{s} f_t(\mathbf{x}) \\
={}& \underbrace{\mathbb{E}\left[\sum_{t=q}^{s} \widehat{f}_t(\mathbf{y}_t)\right] - \min_{\mathbf{y}\in(1-\alpha)\mathcal{X}} \sum_{t=q}^{s} \widehat{f}_t(\mathbf{y})}_{\texttt{term (a)}} \\
&\qquad + \underbrace{\mathbb{E}\left[\sum_{t=q}^{s} f_t(\mathbf{x}_t) - \widehat{f}_t(\mathbf{y}_t)\right]}_{\texttt{term (b)}} + \underbrace{\min_{\mathbf{y}\in(1-\alpha)\mathcal{X}} \sum_{t=q}^{s} \widehat{f}_t(\mathbf{y}) - \min_{\mathbf{x}\in\mathcal{X}} f_t(\mathbf{x})]}_{\texttt{term (c)}}
\end{aligned} \tag{30}$$

Among the three terms, term (b) and term (c) capture the amount of approximation error introduced by the perturbation of functions (f_t versus $\widehat{f}_t$) and minimizer ($\mathbf{y}^* \in \arg\min_{\mathbf{y}\in(1-\alpha)\mathcal{X}} \sum_{t=q}^{s} \widehat{f}_t(\mathbf{y})$ versus $\mathbf{x}^* \in \arg\min_{\mathbf{x}\in\mathcal{X}} f_t(\mathbf{x})$).

Following the dynamic regret analysis of (40) and (41), term (b) and term (c) can be bounded as follows.

$$\texttt{term (b)} \leq 2L\delta T,$$

and

$$\texttt{term (c)} \leq L\delta T + \frac{LR}{r}\delta T.$$

Moreover, since $\mathbf{y}_t$ is the weighted combination of $\mathbf{y}_{i,t}$, it still satisfies $\mathbf{y}_t \in (1-\alpha)\mathcal{X}$.

Now, it remains to bound term (a). Define the function $h_t : (1-\alpha)\mathcal{X} \mapsto \mathbb{R}$ by $h_t(\mathbf{y}) = \widehat{f}_t(\mathbf{y}) + \langle \mathbf{y}, \boldsymbol{\xi}_t \rangle$, where $\boldsymbol{\xi}_t = \widetilde{\mathbf{g}}_t - \nabla \widehat{f}_t(\mathbf{y}_t)$ with

$$\widetilde{\mathbf{g}}_t = \frac{d}{\delta} f_t(\mathbf{y}_t + \delta \mathbf{s}_t) \cdot \mathbf{s}_t.$$

By the analysis of dynamic regret (see (37)), we know that $\mathbb{E}[h_t(\mathbf{y})] = \mathbb{E}[\widehat{f}_t(\mathbf{y})]$ for any fixed $\mathbf{y} \in (1-\alpha)\mathcal{X}$. Besides, since $\nabla h_t(\mathbf{y}_t) = \nabla \widehat{f}_t(\mathbf{y}_t) + \boldsymbol{\xi}_t = \widetilde{\mathbf{g}}_t$, the following holds for any $\mathbf{y} \in (1-\alpha)\mathcal{X}$,

$$h_t(\mathbf{y}_t) - h_t(\mathbf{y}) \leq \nabla h_t(\mathbf{y}_t)^{\mathrm{T}}(\mathbf{y}_t - \mathbf{y}) \overset{(28)}{=} -2G^{one}R\ell_t(\mathbf{y}) + G^{one}R.$$

Note that since $\ell_t(\mathbf{y}_t) = \frac{1}{2}$, we know that for any $\mathbf{y} \in (1-\alpha)\mathcal{X}$,

$$\mathbb{E}\left[\widehat{f}_t(\mathbf{y}_t) - \widehat{f}_t(\mathbf{y})\right] = \mathbb{E}\left[h_t(\mathbf{y}_t) - h_t(\mathbf{y})\right] \leq 2G^{one}R \cdot \mathbb{E}\left[\ell_t(\mathbf{y}_t) - \ell_t(\mathbf{y})\right]. \tag{31}$$

On the other hand, Algorithm 3 is actually a standard CBCE algorithm deploying on a full-information online learning problem where the loss function sequence is $\ell_1, \ldots, \ell_T$. Hence, Theorem 10 implies

$$\max_{[q,s]\subseteq[T]} \left(\sum_{t=q}^{s} \ell_t(\mathbf{y}_t) - \min_{\mathbf{y}\in(1-\alpha)\mathcal{X}} \sum_{t=q}^{s} \ell_t(\mathbf{y}) \right) \leq 15R\widehat{G}\sqrt{T} + 8\sqrt{7\log T + 5}\sqrt{T}$$

where $\widehat{G} = \sup_{\mathbf{y}\in(1-\alpha)\mathcal{X}, t\in[T]} \|\nabla \ell_t(\mathbf{y})\|_2 \leq \frac{1}{2R}$. This in conjunction with (31) yields the following adaptive regret guarantees over the smoothed functions:

$$\max_{[q,s]\subseteq[T]} \left(\mathbb{E}\left[\sum_{t=q}^{s} \widehat{f}_t(\mathbf{y}_t)\right] - \min_{\mathbf{y}\in(1-\alpha)\mathcal{X}} \sum_{t=q}^{s} \widehat{f}_t(\mathbf{y}) \right) \leq 15G^{one}R\sqrt{T} + 16G^{one}R\sqrt{7\log T + 5}\sqrt{T}.$$

Plugging the above inequality into (30), we get

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=q}^{s} f_t(\mathbf{x}_t)\right] - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=q}^{s} f_t(\mathbf{x}) \\
&\leq 15G^{one}R\sqrt{T} + 16G^{one}R\sqrt{7\log T + 5}\sqrt{T} + 3L\delta T + \frac{LR}{r}\delta T
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{Cd}{\delta}\left(15R\sqrt{T} + 16R\sqrt{7\log T + 5}\sqrt{T}\right) + \left(3L + \frac{LR}{r}\right)\delta T \\
&= \sqrt{Cd\left(15R\sqrt{T} + 16R\sqrt{7\log T + 5}\sqrt{T}\right)\left(3L + \frac{LR}{r}\right)T} \\
&= O\big(T^{\frac{3}{4}}(\log T)^{\frac{1}{4}}\big)
\end{aligned}
$$

where we set the perturbation parameter δ optimally as

$$
\delta = \sqrt{\frac{Cd(15R\sqrt{T} + 16R(\sqrt{7\log T + 5}\sqrt{T}))}{(3L + LR/r)T}}. \tag{32}
$$

We thus complete the proof. □

6.2.2 Proof of Theorem 7

Proof. The proof is similar to that in Section 6.2.1. Define the function $h_t : (1-\alpha)\mathcal{X} \mapsto \mathbb{R}$ by

$$
h_t(\mathbf{y}) = \widehat{f}_t(\mathbf{y}) + \mathbf{y}^{\mathrm{T}}\boldsymbol{\xi}_t,
$$

where $\boldsymbol{\xi}_t = \widetilde{\mathbf{g}}_t - \nabla\widehat{f}_t(\mathbf{y}_t)$ with

$$
\widetilde{\mathbf{g}}_t = \frac{d}{2\delta}\big(f_t(\mathbf{y}_t + \delta\mathbf{s}_t) - f_t(\mathbf{y}_t - \delta\mathbf{s}_t)\big)\cdot\mathbf{s}_t.
$$

Similarly, $\mathbb{E}[h_t(\mathbf{y})] = \mathbb{E}[\widehat{f}_t(\mathbf{y})]$ holds for any fixed $\mathbf{y} \in (1-\alpha)\mathcal{X}$. Besides, since $\nabla h_t(\mathbf{y}_t) = \nabla\widehat{f}_t(\mathbf{y}_t) + \boldsymbol{\xi}_t = \widetilde{\mathbf{g}}_t$, we know that for any $\mathbf{y} \in (1-\alpha)\mathcal{X}$,

$$
h_t(\mathbf{y}_t) - h_t(\mathbf{y}) \leq -2G^{two}R\ell_t(\mathbf{y}) + G^{two}R.
$$

Note that since $\ell_t(\mathbf{y}_t) = \frac{1}{2}$, we have that

$$
\mathbb{E}\left[\widehat{f}_t(\mathbf{y}_t) - \widehat{f}_t(\mathbf{y})\right] = \mathbb{E}\left[h_t(\mathbf{y}_t) - h_t(\mathbf{y})\right] \leq 2G^{two}R\cdot\mathbb{E}\left[\ell_t(\mathbf{y}_t) - \ell_t(\mathbf{y})\right]
$$

holds for any $\mathbf{y} \in (1-\alpha)\mathcal{X}$.

Hence, by deploying the standard CBCE algorithm on the loss function sequence $\ell_1, \ldots, \ell_T$ (Algorithm 3), and based on Theorem 10 in Appendix A.3, we have

$$
\max_{[q,s]\subseteq[T]}\left(\sum_{t=q}^{s}\ell_t(\mathbf{y}_t) - \min_{\mathbf{y}\in(1-\alpha)\mathcal{X}}\sum_{t=q}^{s}\ell_t(\mathbf{y})\right) \leq 15R\widehat{G}\sqrt{T} + 8\sqrt{7\log T + 5}\sqrt{T} \tag{33}
$$

where $\widehat{G} = \max_{\mathbf{y}\in(1-\alpha)\mathcal{X}, t\in[T]}\|\nabla\ell_t(\mathbf{y})\|_2 \leq \frac{1}{2R}$. Thus,

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{t=q}^{s}\frac{1}{2}\Big(f_t(\mathbf{x}_t^{(1)}) + f_t(\mathbf{x}_t^{(2)})\Big)\right] - \min_{\mathbf{x}\in\mathcal{X}}\sum_{t=q}^{s}f_t(\mathbf{x}) \\
&\leq 15G^{two}R\sqrt{T} + 16G^{two}R\sqrt{7\log T + 5}\sqrt{T} + 4L\delta T + \frac{LR}{r}\delta T
\end{aligned}
$$

$$
\begin{aligned}
&\leq dL\left(15R\sqrt{T} + 16R\sqrt{7\log T + 5}\sqrt{T}\right) + \left(4L + \frac{LR}{r}\right)\delta T\\
&= dLR\left(15\sqrt{T} + 16\sqrt{7\log T + 5}\sqrt{T}\right) + \left(4L + \frac{LR}{r}\right)dR\sqrt{T}\\
&= dLR\left(19\sqrt{T} + 16\sqrt{7\log T + 5}\sqrt{T}\right) + \frac{dLR^2}{r}\sqrt{T}\\
&= O\big(T^{\frac{1}{2}}(\log T)^{1/2}\big),
\end{aligned}
$$

where we set the perturbation parameter as $\delta = dR/\sqrt{T}$. Hence, we complete the proof. □

7. Experiments

Despite that the focus of this paper is mainly theoretical, we conduct empirical evaluations on several datasets to validate the effectiveness of our proposed approach.

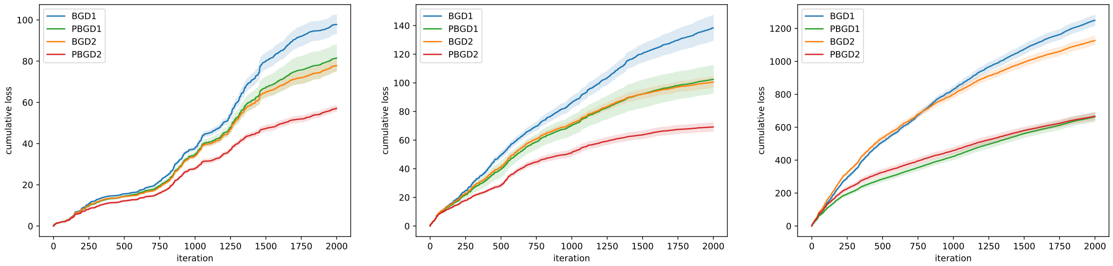
Settings. To simulate the online learning scenario, the player will receive the instance information and make the prediction *sequentially*. We focus on the regression setting. Specifically, at iteration t, an instance $(\mathbf{x}_t, y_t)$ arrives with $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^d$ being the feature and $y_t \in \mathbb{R}$ being the output, and the player can only observe the feature $\mathbf{x}_t$. The player will then make the prediction $\widehat{y}_t \in \mathbb{R}$, probably in the form of $\widehat{y}_t = g(\mathbf{w}_t, \mathbf{x}_t)$, where $g : \mathcal{W} \times \mathcal{X} \mapsto \widehat{Y}$ is some parametric model with $\mathbf{w} \in \mathcal{W}$ as the parameter. After that, the environments will reveal the loss value of $\ell(y_t, \widehat{y}_t)$ to the player as the feedback, where $\ell : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ is the loss function that is *unknown* to the player. In our experimental setting, the player will adopt a simple linear model, denoted by $\widehat{y}_t = g(\mathbf{w}_t, \mathbf{x}_t) = \mathbf{w}_t^{\mathrm{T}}\mathbf{x}_t$. Besides, the loss function is chosen as the squared loss $\ell(y, \widehat{y}) = (y - \widehat{y})^2$. Therefore, the online function at each iteration $f_t : \mathcal{W} \mapsto \mathbb{R}$ is essentially a couple of loss function ℓ and the instance $(\mathbf{x}_t, y_t)$, namely,

$$
f_t(\mathbf{w}) = \ell(\mathbf{w}^{\mathrm{T}}\mathbf{x}_t, y_t) = (\mathbf{w}^{\mathrm{T}}\mathbf{x}_t - y_t)^2.
$$

Moreover, the features of the datasets are normalized to the range of $[0, 1]$, which implies the diameter of the feasible set $R = 1$ (cf. Assumption 1 for the definition of diameter R).

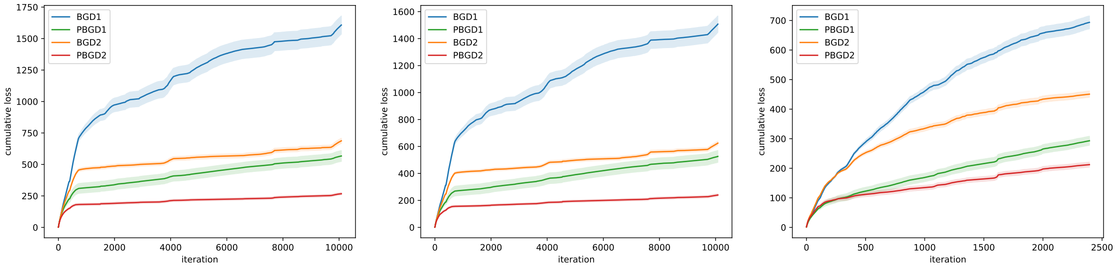
We would like to emphasize again that in the bandit convex optimization setting the only feedback to the player is the function value of the online function, i.e., $f_t(\mathbf{w}_t)$; while the player is not allowed to access the function gradients even the function itself. The player requires to perform bandit convex optimization algorithms for update to obtain the new model $\mathbf{w}_{t+1}$. In the experiments, we choose the *cumulative loss* as the performance measure of different learning algorithms.

Contender. We take the BGD algorithm as the contender, whose step size tuning is set according to the time horizon T without involving the unknown path-length P_T (Yang et al., 2016; Chen and Giannakis, 2019), and more precisely, the step size of BGD algorithm is set to $R^2/(\sqrt{d}T^{3/4})$ and $R/(dL\sqrt{T})$ for one-point and two-point BCO respectively, as suggested by their theory. Meanwhile, in the experiments we consider both settings of BCO with one-point feedback and BCO with two-point feedback, and we will use the suffix of "1" and "2" after the algorithm's name to denote the two versions, respectively. For example, for



(a) synGradual (b) synAbrupt (c) hyperplane

Figure 1: Cumulative loss of BGD and PBGD on three synthetic datasets: synGradual, synAbrupt, and hyperplane. BGD1/PBGD1 denotes the methods for BCO with one-point feedback, and BGD2/PBGD2 denotes the methods for BCO with two-point feedback.



(a) SRU-1 (b) SRU-2 (c) debutanizer

Figure 2: Cumulative loss of BGD and PBGD on three real-world datasets: SRU-1, SRU-2, and debutanizer. BGD1/PBGD1 denotes the methods for BCO with one-point feedback, and BGD2/PBGD2 denotes the methods for BCO with two-point feedback.

our approach PBGD, we call its version for one-point BCO as PBGD1 and the version for two-point BCO as PBGD2. Similarly, the contender BGD algorithm also has two versions—BGD1 and BGD2—for two models respectively.

Data sets. We evaluate above methods on both synthetic and real-world datasets. First, we conduct experiments over three synthetic datasets. We generate two datasets called *synGradual* and *synAbrupt*, where the synGradual dataset simulates the scenario of gradual change and the synAbrupt dataset synthesizes the abrupt change. For both datasets, the feature space is set as the 3-dimensional Euclidean ball, and we simulate 2000 data items (i.e., the time horizon $T = 2000$). At each iteration, the ground-truth output variable is generated by $y_t = {\mathbf{w}_t^*}^{\mathrm{T}} \mathbf{x}_t$ for some unknown regression vector $\mathbf{w}_t^* \in \mathbb{R}^d$. For the synGradual dataset, the ground-truth regression vector changes with a random drift, namely, $\mathbf{w}_{t+1}^* = \mathbf{w}_t^* + \Delta_t$ with the drifting term Δ_t being a random vector sampled from d-dimensional standard normal distribution scaled by 0.1. For the synAbrupt dataset, the whole time horizon is split to four stages, each with 500 consecutive data items sharing the same regression vector, and the regression vectors vary over different stages. Another used synthetic dataset is the *hyperplane* dataset (Kolter and Maloof, 2005), which is widely used in the literature of streaming data with changing distributions.[2] We further include the following three real-world datasets frequently used in the problems of non-stationary streaming regression.[3]

- *Sulfur recovery unit* (Fortuna et al., 2007) is a real-world dataset with records of gas diffusion. Feature consists of 5 different chemical and physical indexes, with in total 10,081 data samples. There are two outputs in original dataset which represent the concentration of SO_2 and H_2S, and we split it into SRU-1 and SRU-2, respectively.

- *Debutanizer column* (Fortuna et al., 2007) is a real-world dataset with records of chemical reactions, with in total 2,394 data samples consisting of 7 different features. The output represents C4 content in the debutanizer bottoms.

We repeat the experiments for 10 times and report the average and standard deviation to avoid the effect of the randomness caused by the gradient estimation in bandit convex optimization (cf. descriptions in Section 3.2).

Results. Figure 1 plots the curves of *cumulative loss* of BGD (contender) and PBGD (our approach) on three synthetic datasets: synGradual, synAbrupt, and hyperplane. Besides, Figure 2 shows the results for three real-world datasets: SRU-1, SRU-2, and debutanizer. The lower the cumulative loss is, the better the performance is. From the results, we can observe that our PBGD algorithm is always better than the BGD algorithm, no matter in one-point or two-point feedback models. Thus, the results validate the effectiveness of our approach. Actually, the crucial advantage of our approach relies on the online ensemble mechanism, which combine multiple expert-algorithms with different step sizes to hedge the unknown non-stationarity of the changing environments.

On the other hand, we would like to mention the power of two-point feedback model. Let us see the comparison between BGD1 and BGD2, as well as PBGD1 and PBGD2. Obviously, the performance is substantially improved when the player is allowed to query two-point feedback at each iteration. Notably, the advantage lies in both mean and standard deviation. The phenomenon accords to our theoretical understandings well, that is, the two-point feedback model can substantially improve the regret bound (both dynamic regret and static regret) since the gradient estimator is now in a better quality whose variance is significantly reduced and is independent of the perturbation parameter δ.

8. Conclusion and Future Work

In this paper, we study the bandit convex optimization (BCO) problems in non-stationary environments. We propose the Parameter-free Bandit Gradient Descent (PBGD) algorithm that achieves the state-of-the-art $O(T^{3/4}(1+P_T)^{1/2})$ and $O(T^{1/2}(1+P_T)^{1/2})$ dynamic regret for one-point and two-point feedback models respectively. The regret bounds hold universally against any feasible comparator sequence. Meanwhile, the algorithm does not need to know prior information of the path length, which is unknown but required in previous studies. Furthermore, we demonstrate the regret bound for the two-point feedback model is minimax optimal by establishing the first lower bound for the universal dynamic regret in the bandit convex optimization setup. We extend the algorithm to an anytime version. Besides, we

[2] The dataset can be downloaded from `https://www.win.tue.nl/~mpechen/data/DriftSets/`.

[3] The datasets can be downloaded from `https://home.isr.uc.pt/~fasouza/datasets.html`.

also present algorithms for BCO problems to optimize the adaptive regret, another measure for the non-stationary online learning. Finally, we conduct experiments on both synthetic and real-world data to further validate the effectiveness of the proposed approach.

There are many interesting issues worthy exploring for the future research. It remains open on how to achieve optimal or sharper dynamic regret bounds for BCO with one-point feedback, where some techniques of variance reduction might be useful. Moreover, we will consider incorporating other properties, like strong convexity and smoothness, to further enhance the dynamic regret. Another future work is to develop algorithms with problem-dependent dynamic regret for BCO. Note that such results are provably attainable in the full-information setting (Zhao et al., 2020b) and recent works (Bubeck et al., 2019; Lee et al., 2020) have demonstrated data-dependent static regret guarantees for adversarial linear bandits, which warrant special attention when pursuing problem-dependent dynamic regret for general bandit convex optimization.

Acknowledgments

This research was supported by the National Science Foundation of China (61921006), JiangsuSF (BK20200064), and the Collaborative Innovation Center of Novel Software Technology and Industrialization. We are also grateful for the anonymous reviewers and the editor for their helpful comments.

A. Preliminaries

In this section, we introduce some preliminaries for analyzing dynamic regret and adaptive regret of algorithms for BCO.

A.1 Projection Issues

Notice that we run the algorithm on a slightly smaller set $(1-\alpha)\mathcal{X}$ rather than the original feasible set $\mathcal{X}$, where the shrinkage parameter $\alpha > 0$ needs to be sufficiently large so that the decision $\mathbf{y}_t + \delta \mathbf{s}_t$ (and $\mathbf{y}_t - \delta \mathbf{s}_t$) can be guaranteed to locate in $\mathcal{X}$. Consequently, there are some additional terms involved due to the projection over a shrunk set. In the following we provide some lemmas justifying the relationships between the original feasible set and the shrunk set. Note that most of these results can be found in the seminal paper of Flaxman et al. (2005), we provide the proofs for self-containedness.

Lemma 3. *For any feasible point $\mathbf{x} \in (1-\alpha)\mathcal{X}$, the ball of radius αr centered at $\mathbf{x}$ belongs to the feasible set $\mathcal{X}$.*

Proof. The result is originally proved in Observation 3.2 of Flaxman et al. (2005). The proof is based on the simple observation that

$$(1-\alpha)\mathcal{X} + \alpha r \mathbb{B} \subseteq (1-\alpha)\mathcal{X} + \alpha \mathcal{X} = \mathcal{X}$$

holds since $r\mathbb{B} \subseteq \mathcal{X}$ and $\mathcal{X}$ is convex. □

Indeed, since the perturbation parameter is set as $\delta = \alpha r$ (cf. Theorem 1 and Theorem 2), the final decision is $\mathbf{x}_t = \mathbf{y}_t + \delta \mathbf{s}_t = \mathbf{y}_t + \alpha \cdot r \cdot \mathbf{s}_t$. Therefore, Lemma 3 ensures that $\mathbf{x}_t \in \mathcal{X}$ by noticing $\mathbf{y}_t \in (1-\alpha)\mathcal{X}$ and $\mathbf{s}_t$ being a random unit vector.

The following lemma, originally raised in Observation 3.3 of Flaxman et al. (2005), establishes a bound on the maximum that the function can change in $(1-\alpha)\mathcal{X}$, which essentially acts as an effective Lipschitz condition.

Lemma 4. *For any $\mathbf{x} \in (1-\alpha)\mathcal{X}$, under Assumption 3, we have*

$$|\widehat{f}_t(\mathbf{x}) - f_t(\mathbf{x})| \leq L\delta. \tag{34}$$

Proof. Since the smoothed function $\widehat{f}_t$ is an average over inputs within δ of $\mathbf{x}$, the Lipschitz continuity of the function f_t yields the result. □

A.2 Dynamic Regret

We have following dynamic regret bounds for the online gradient descent (OGD) algorithm (Zinkevich, 2003).

Theorem 8 (Dynamic Regret of OGD)**.** *Consider the online gradient descent (OGD), which starts with any $\mathbf{x}_1 \in \mathcal{X}$ and performs the following update*

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}[\mathbf{x}_t - \eta \nabla f_t(\mathbf{x}_t)].$$

Suppose the feasible domain $\mathcal{X}$ is bounded, i.e., $\|\mathbf{x} - \mathbf{y}\|_2 \leq D$ for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$; meanwhile, the online functions have bounded gradient magnitude, i.e., $\|\nabla f_t(\mathbf{x})\|_2 \leq G$ for any $\mathbf{x} \in \mathcal{X}$ and $t \in [T]$. Then, the dynamic regret of OGD is upper bounded by

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \leq \frac{7D^2 + DP_T}{4\eta} + \frac{\eta G^2 T}{2},$$

for any *comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$. In above, P_T is the path-length defined as $P_T = \sum_{t=2}^{T} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2$.*

In the bandit convex optimization setting, we cannot access the true gradient but the unbiased gradient estimation instead. Therefore, we extend Theorem 8 to the randomized version for the loss function chosen from adaptive environments as follows.

Theorem 9 (Expected Dynamic Regret of Randomized OGD)**.** *Consider the following randomized version online gradient descent. The randomized OGD begins with any $\mathbf{x}_1 \in \mathcal{X}$ and performs*

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}[\mathbf{x}_t - \eta \mathbf{g}_t],$$

where $\mathbb{E}[\mathbf{g}_t | \mathbf{x}_1, f_1, \ldots, \mathbf{x}_t, f_t] = \nabla f_t(\mathbf{x}_t)$ and $\|\mathbf{g}_t\|_2 \leq \widetilde{G}$ for some $\widetilde{G} > 0$. Then, the expected dynamic regret of OGD is upper bounded by

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \leq \frac{7D^2 + DP_T}{4\eta} + \frac{\eta \widetilde{G}^2 T}{2}, \tag{35}$$

for any *fixed comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$.*

Proof. Define the function $h_t : \mathcal{X} \to \mathbb{R}$ by

$$h_t(\mathbf{x}) = f_t(\mathbf{x}) + \langle \mathbf{x}, \boldsymbol{\xi}_t \rangle, \quad \text{where } \boldsymbol{\xi}_t = \mathbf{g}_t - \nabla f_t(\mathbf{x}_t).$$

Clearly, $\nabla h_t(\mathbf{x}_t) = \nabla f_t(\mathbf{x}_t) + \boldsymbol{\xi}_t = \mathbf{g}_t$. So we can leverage the result of deterministic version OGD in Theorem 8 on the function h_t and obtain the following dynamic regret over the sequence of functions $h_1, \ldots, h_T$:

$$\sum_{t=1}^{T} h_t(\mathbf{x}_t) - \sum_{t=1}^{T} h_t(\mathbf{u}_t) \leq \frac{7D^2 + DP_T}{4\eta} + \frac{\eta \widetilde{G}^2 T}{2}. \tag{36}$$

Note that for any *fixed* $\mathbf{x} \in \mathcal{X}$, we have

$$\begin{aligned} \mathbb{E}[h_t(\mathbf{x})] &= \mathbb{E}[f_t(\mathbf{x})] + \mathbb{E}[\boldsymbol{\xi}_t^{\mathrm{T}} \mathbf{x}] \\ &= \mathbb{E}[f_t(\mathbf{x})] + \mathbb{E}[\mathbb{E}[\boldsymbol{\xi}_t^{\mathrm{T}} \mathbf{x} | \mathbf{x}_1, f_1, \ldots, \mathbf{x}_t, f_t]] \\ &= \mathbb{E}[f_t(\mathbf{x})] + \mathbb{E}[\mathbb{E}[\boldsymbol{\xi}_t | \mathbf{x}_1, f_1, \ldots, \mathbf{x}_t, f_t]^{\mathrm{T}} \mathbf{x}] \\ &= \mathbb{E}[f_t(\mathbf{x})]. \end{aligned} \tag{37}$$

Therefore, when both the function sequence and comparator sequence are chosen by an oblivious adversary (as specified in Section 3.1), we can take expectations over both sides of (36) and obtain the desired result. □

A.3 Adaptive Regret

In the full-information setting, we have the following adaptive regret bound for the Coin Betting for Changing Environment (CBCE) algorithm proposed by Jun et al. (2017) .

Theorem 10 (Adaptive Regret of CBCE (Jun et al., 2017, Theorem 1))**.** *Consider an OCO problem where the player iteratively selects a decision $\mathbf{x}_t \in \mathcal{X}$ and observes a loss function $h_t : \mathcal{X} \mapsto \mathbb{R}$. Assume the gradient of all the loss functions are bounded by G, the diameter of $\mathcal{X}$ is bounded by D, and the function value of h_t lies in $[0, 1]$, for any $st \in [T]$. Then, the CBCE algorithm with the standard OGD algorithm as its expert-algorithm and $h_1, \ldots, h_T$ as the input loss functions achieves the following adaptive regret,*

$$\max_{[q,s] \subseteq [T]} \left(\sum_{t=q}^{s} h_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=q}^{s} h_t(\mathbf{x}) \right) \leq 15DG\sqrt{T} + 8\sqrt{7 \log T + 5}\sqrt{T}.$$

The algorithm above is inefficient in the sense that it requires to query the gradient of the loss function $O(\log t)$ times at iteration t. To address this limitation, Wang et al. (2018) introduced a surrogate loss function $\ell_t : \mathcal{X} \mapsto [0, 1]$,

$$\ell_t(\mathbf{x}) = \frac{1}{2DG} \nabla h_t(\mathbf{x}_t)^{\top} (\mathbf{x} - \mathbf{x}_t) + \frac{1}{2}$$

for which we have for any $\mathbf{x} \in \mathcal{X}$,

$$h_t(\mathbf{x}_t) - h_t(\mathbf{x}) \leq -2DG\ell_t(\mathbf{x}) + DG = 2DG(\ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x})). \tag{38}$$

Notice that the inequality (38) implies that, to solve the original problem where the loss functions are $h_1(\cdot), \ldots, h_T(\cdot)$, we can deploy CBCE on a new problem where the loss functions are $\ell_1(\cdot), \ldots, \ell_T(\cdot)$. The benefits here is that in this way we only need to query the gradient of h_t once at each iteration and the order of the regret bound remains the same. To be more specific, we have the following regret bound.

Theorem 11. *Consider the same learning setting as in Theorem 10. Then, the CBCE algorithm with the standard OGD algorithm as its expert-algorithm and $\ell_1, \ldots, \ell_T$ as the input loss functions achieves the following adaptive regret,*

$$\max_{[q,s]\subseteq[T]} \left(\sum_{t=q}^{s} h_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=q}^{s} h_t(\mathbf{x}) \right) \le 15DG\sqrt{T} + 8DG\sqrt{7\log T + 5}\sqrt{T}.$$

A.4 Proof of Lemma 2

Proof. Lemma 2 is essentially the regret guarantee for the exponentially weighted average forecaster with nonuniform initial weights (Cesa-Bianchi and Lugosi, 2006, Excercise 2.5). We will prove the lemma by the standard potential argument (Cesa-Bianchi and Lugosi, 2006, Chapter 2).

Let $L_t(i) = \sum_{s=1}^{t} \ell_s(\mathbf{y}_s^i)$ denote the cumulative loss, and

$$\Phi_t = \sum_{i\in[N]} w_1^i \exp(-\varepsilon L_t^i) = \sum_{i\in[N]} w_1^i \exp\left(-\varepsilon \sum_{s=1}^{t} \ell_s(\mathbf{y}_s^i)\right)$$

denote the potential function. Then, from the update procedure of line 9 in (2), we have

$$w_{t+1}^i = \frac{w_t^i \exp(-\varepsilon \ell_t(\mathbf{y}_t^i))}{\sum_{i\in[N]} w_t^i \exp(-\varepsilon \ell_t(\mathbf{y}_t^i))} = \frac{w_1^i \exp(-\varepsilon L_{t-1}^i)}{\sum_{i\in[N]} w_1^i \exp(-\varepsilon L_{t-1}^i)}.$$

On one hand, from the non-negativity, we know that

$$\ln \Phi_T = \ln\left(\sum_{i\in[N]} w_1^i \exp(-\varepsilon L_T^i)\right) \ge \ln\left(\max_{i\in[N]} w_1^i \exp(-\varepsilon L_T^i)\right) = -\varepsilon \min_{i\in[N]} \left(L_T^i + \frac{1}{\varepsilon}\ln\frac{1}{w_1^i}\right).$$

On the other hand, we have

$$\ln\left(\frac{\Phi_t}{\Phi_{t-1}}\right) = \ln\left(\frac{\sum_{i\in[N]} w_1^i \exp(-\varepsilon L_t^i)}{\sum_{i\in[N]} w_1^i \exp(-\varepsilon L_{t-1}^i)}\right) = \ln\left(\sum_{i\in[N]} w_t^i \exp(-\varepsilon \ell_t^i(\mathbf{y}_t^i))\right),$$

and meanwhile

$$\ln \Phi_1 = \ln\left(\sum_{i\in[N]} w_1^i \exp(-\varepsilon \ell_1(\mathbf{y}_1^i))\right).$$

Thus,

$$\ln \Phi_T = \ln \Phi_1 + \sum_{t=2}^{T} \ln\left(\frac{\Phi_t}{\Phi_{t-1}}\right) = \sum_{t=1}^{T} \ln\left(\sum_{i\in[N]} w_t^i \exp(-\varepsilon \ell_t(\mathbf{y}_t^i))\right).$$

Furthermore, denote by $c = 4\widetilde{G}R$ the maximal range of the loss function, then by Hoeffding's inequality, we have

$$\begin{aligned}\ln\left(\sum_{i\in[N]} w_t^i \exp(-\varepsilon\ell_t(\mathbf{y}_t^i))\right) \leq{}& -\varepsilon\sum_{i\in[N]} w_t^i\ell_t(\mathbf{y}_t^i) + \frac{\varepsilon^2 c^2}{8}\\ \leq{}& -\varepsilon\ell_t\left(\sum_{i\in[N]} w_t^i\mathbf{y}_t^i\right) + \frac{\varepsilon^2 c^2}{8}\\ ={}& -\varepsilon\ell_t\left(\mathbf{y}_t\right) + \frac{\varepsilon^2 c^2}{8}\end{aligned}$$

where the last inequality holds due to the convexity of the loss function and Jensen's inequality. Combining the inequalities of both directions, we have

$$-\varepsilon\min_{i\in[N]}\left(L_T^i + \frac{1}{\varepsilon}\ln\frac{1}{w_1^i}\right) \leq \ln\Phi_T \leq \sum_{t=1}^T -\varepsilon\ell_t\left(\mathbf{y}_t\right) + \frac{\varepsilon^2 c^2}{8}.$$

Thus, we get

$$\sum_{t=1}^T \ell_t(\mathbf{y}_t) - \min_{i\in[N]}\left(\sum_{t=1}^T \ell_t(\mathbf{y}_t^i) + \frac{1}{\varepsilon}\ln\frac{1}{w_1^i}\right) \leq \frac{\varepsilon^2 c^2}{8}.$$

We complete the proof by plugging the value $c = 4\widetilde{G}R$ into the above inequality. □

B. Analysis of BGD Algorithm

In this section, we provide the proofs of theoretical guarantees for the BGD algorithm including Theorem 1 (one-point feedback model) and Theorem 2 (two-point feedback model).

Before presenting rigorous proofs, we first highlight the main idea and procedures of the argument as follows.

(1) We first guarantee that for any $t \in [T]$, $\mathbf{x}_t$ is a feasible point in $\mathcal{X}$, which holds due to the fact that the projection in Algorithm 1 is over $\mathbf{y}_t$ instead of $\mathbf{x}_t$.

(2) We then analyze the dynamic regret of the smoothed functions $\widehat{f}_1, \ldots, \widehat{f}_T$ in terms of the scaled comparator sequence.

(3) We finally check the gap between the dynamic regret of the smoothed functions $\widehat{f}_1, \ldots, \widehat{f}_T$ and that of the original functions $f_1, \ldots, f_T$.

B.1 Proof of Theorem 1

Proof. Notice that the projection in Algorithm 1 only guarantees that $\mathbf{y}_t$ is in a slightly smaller set $(1-\alpha)\mathcal{X}$, so we first need to prove that $\forall t \in [T]$, $\mathbf{x}_t$ is a feasible point in $\mathcal{X}$. This is convinced by Lemma 3, since we know that $\delta \leq \alpha r$ from the parameter setting ($\alpha = \delta/r$).

Next, as demonstrated in (10), the expected dynamic regret can be decomposed as

$$\mathbb{E}\left[\sum_{t=1}^T f_t(\mathbf{x}_t)\right] - \sum_{t=1}^T f_t(\mathbf{u}_t)$$

$$= \underbrace{\mathbb{E}\left[\sum_{t=1}^{T}\left(\widehat{f}_t(\mathbf{y}_t) - \widehat{f}_t(\mathbf{v}_t)\right)\right]}_{\texttt{term (a)}} + \underbrace{\mathbb{E}\left[\sum_{t=1}^{T}\left(f_t(\mathbf{x}_t) - \widehat{f}_t(\mathbf{y}_t)\right)\right]}_{\texttt{term (b)}} + \underbrace{\mathbb{E}\left[\sum_{t=1}^{T}\left(\widehat{f}_t(\mathbf{v}_t) - f_t(\mathbf{u}_t)\right)\right]}_{\texttt{term (c)}},$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_T$ is the scaled comparator sequence, set as $\mathbf{v}_t = (1-\alpha)\mathbf{u}_t$ and $\alpha \in (0,1)$ is the shrinkage parameter. So we will bound the three terms separately.

The term (a) is essentially the dynamic regret of the smoothed functions. In the one-point feedback model, the gradient estimator is set according to (8), and we know that $\mathbb{E}[\widetilde{\mathbf{g}}_t] = \nabla \widehat{f}_t(\mathbf{y}_t)$ due to Lemma 1. Therefore, the procedure of $\mathbf{y}_{t+1} = \Pi_{(1-\alpha)\mathcal{X}}[\mathbf{y}_t - \eta\widetilde{\mathbf{g}}_t]$ is actually the randomized online gradient descent over the smoothed function $\widehat{f}_t$. So term (a) can be upper bound by using Theorem 9.

$$\texttt{term(a)} \overset{(35)}{\leq} \frac{7\widetilde{D}^2 + \widetilde{D}\widetilde{P_T}}{4\eta} + \frac{\eta \widetilde{G}^2 T}{2} \leq \frac{7R^2 + RP_T}{4\eta} + \frac{\eta d^2 C^2 T}{2\delta^2}, \tag{39}$$

where $\widetilde{P}_T = \sum_{t=2}^{T} \|\mathbf{v}_{t-1} - \mathbf{v}_t\|_2 = (1-\alpha)P_T$, $\widetilde{D} = (1-\alpha)R \leq R$ and $\widetilde{G} = dC/\delta$ as shown in (19). Next, by Assumption 3 and Lemma 4, we have

$$\texttt{term (b)} = \mathbb{E}\left[\sum_{t=1}^{T}\left(f_t(\mathbf{x}_t) - f_t(\mathbf{y}_t) + f_t(\mathbf{y}_t) - \widehat{f}_t(\mathbf{y}_t)\right)\right] \leq 2L\delta T. \tag{40}$$

Moreover, term (c) can be bounded by

$$\begin{aligned} \texttt{term (c)} &\leq \mathbb{E}\left[\sum_{t=1}^{T}\left(|\widehat{f}_t(\mathbf{v}_t) - f_t(\mathbf{v}_t)| + |f_t(\mathbf{v}_t) - f_t(\mathbf{u}_t)|\right)\right] \\ &\leq \mathbb{E}\left[\sum_{t=1}^{T}(L\delta + L\|\mathbf{v}_t - \mathbf{u}_t\|_2)\right] \\ &\leq \mathbb{E}\left[\sum_{t=1}^{T}(L\delta + L\alpha R)\right] \\ &= \left(L + \frac{LR}{r}\right)\delta T \end{aligned} \tag{41}$$

where the second inequality holds due to Lemma 4 and Assumption 3.

Combining upper bounds of three terms in (39), (40) and (41), we obtain the dynamic regret of the original function f_t over the comparator sequence of $\mathbf{u}_1, \ldots, \mathbf{u}_T$,

$$\begin{aligned} \mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] - \sum_{t=1}^{T} f_t(\mathbf{u}_t) &= \texttt{term (a)} + \texttt{term (b)} + \texttt{term (c)} \\ &\leq \frac{7R^2 + RP_T}{4\eta} + \frac{\eta d^2 C^2 T}{2\delta^2} + 2L\delta T + (L\delta + L\alpha R)T \\ &\leq \frac{7R^2 + RP_T}{4\eta} + \frac{\eta d^2 C^2 T}{2\delta^2} + \left(3L + \frac{LR}{r}\right)\delta T \end{aligned} \tag{42}$$

$$
\begin{aligned}
&= \sqrt[4]{32(7R^2 + RP_T)d^2C^2L^2T^3} && (43)\\
&\leq 2(dCL)^{\frac{1}{2}}(7R^2 + RP_T)^{\frac{1}{4}}T^{\frac{3}{4}}\\
&= O\Big((1 + P_T)^{\frac{1}{4}}T^{\frac{3}{4}}\Big),
\end{aligned}
$$

where (42) follows from the setting of $\alpha = \delta/r$. The equation (43) is obtained by the AM-GM inequality via optimizing values of η and δ, where the optimal parameter configuration is

$$
\eta^* = (dC\widetilde{L})^{-\frac{1}{2}}\left(\frac{7R^2 + RP_T}{T}\right)^{\frac{3}{4}}, \quad \delta^* = \left(\frac{dC}{\widetilde{L}}\right)^{\frac{1}{2}}\left(\frac{7R^2 + RP_T}{T}\right)^{\frac{1}{4}},
$$

and $\widetilde{L} = 3L + LR/r$ is the effective Lipschitz constant. After simplifying the upper bound, we complete the proof of Theorem 1. □

B.2 Proof of Theorem 2

Proof. In the two-point feedback model, the gradient estimator is constructed according to (9), whose norm can be upper bounded by dL, as shown in (22) when analyzing the regret of the PBGD algorithm for two-point BCO. We emphasize again that in contrast to the one in the one-point feedback model as shown in (19), the upper bound of gradient norm $\widetilde{G}$ here is *independent* of the $1/\delta$, which leads to a substantially improved regret bound.

Meanwhile, by exploiting the Lipschitz property of the online functions, we have

$$
f_t(\mathbf{y}_t + \delta\mathbf{s}_t) \leq f_t(\mathbf{y}_t) + L\|\delta\mathbf{s}_t\|_2 = f_t(\mathbf{y}_t) + \delta L, \tag{44}
$$

and similar result holds for $f_t(\mathbf{y}_t - \delta\mathbf{s}_t)$. We can thus bound the expected regret as follows,

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\frac{1}{2}\big(f_t(\mathbf{y}_t + \delta\mathbf{s}_t) + f_t(\mathbf{y}_t - \delta\mathbf{s}_t)\big)\right] - \sum_{t=1}^{T} f_t(\mathbf{u}_t)\\
&\overset{(44)}{\leq} \mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{y}_t)\right] + \delta LT - \sum_{t=1}^{T} f_t(\mathbf{u}_t)\\
&= \mathbb{E}\left[\sum_{t=1}^{T}\widehat{f}_t(\mathbf{y}_t) - \sum_{t=1}^{T}\widehat{f}_t(\mathbf{v}_t)\right] + \delta LT\\
&\qquad\qquad + \mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{y}_t) - \sum_{t=1}^{T}\widehat{f}_t(\mathbf{y}_t)\right] + \left[\sum_{t=1}^{T}\widehat{f}_t(\mathbf{v}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)\right]\\
&\leq \frac{7R^2 + RP_T}{4\eta} + \frac{\eta d^2L^2}{2}T + \left(3L + \frac{LR}{r}\right)\delta T && (45)\\
&= dL\sqrt{T}\sqrt{(7R^2 + RP_T)/2} + dRL\sqrt{T} && (46)\\
&\leq dL\sqrt{T}\sqrt{8R^2 + RP_T} && (47)\\
&= O\Big((1 + P_T)^{\frac{1}{2}}T^{\frac{1}{2}}\Big).
\end{aligned}
$$

We remark that the core characteristic of analysis of the two-point feedback model lies in the second term of (45), which is independent of $1/\delta$, and thus is much smaller than that of the one-point feedback model shown in (42). The critical advantage owes to the benefit of the gradient estimator evaluated by two points at each iteration, which reduces the variance substantially. Notice that (46) can be obtained by setting the step size η and perturbation parameter δ optimally as

$$\eta^* = \sqrt{\frac{7R^2 + RP_T}{2d^2L^2T}}, \quad \delta^* = \frac{dLR}{\widetilde{L}\sqrt{T}},$$

where $\widetilde{L} = (3L + LR/r)$ is the effective Lipschitz constant as aforementioned. Note that we set the optimal perturbation parameter as $dLR/(\widetilde{L}\sqrt{T})$ for the sake of a more succinct and beautiful regret form, and one may also choose other appropriate configurations like $dR/\sqrt{T}$ without affecting the regret order. The inequality (47) makes use of the fact that $\sqrt{a} + \sqrt{b} \leq \sqrt{2(a+b)}$ holds for any $a, b > 0$. Hence we compete the proof of Theorem 2. □

References

Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, pages 263–274, 2008.

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

Peter Auer, Yifang Chen, Pratik Gajane, Chung-Wei Lee, Haipeng Luo, Ronald Ortner, and Chen-Yu Wei. Achieving optimal dynamic regret for non-stationary bandits without prior information. In *Proceedings of the 32nd Conference on Learning Theory (COLT)*, pages 159–163, 2019.

Baruch Awerbuch and Robert D. Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 45–53, 2004.

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

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

Sébastien Bubeck, Nicolò Cesa-Bianchi, and Sham M. Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, 2012.

Sébastien Bubeck, Yin Tat Lee, and Ronen Eldan. Kernel-based methods for bandit convex optimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 72–85, 2017.

Sébastien Bubeck, Yuanzhi Li, Haipeng Luo, and Chen-Yu Wei. Improved path-length regret bounds for bandits. In *Proceedings of the 32nd Conference on Learning Theory (COLT)*, pages 508–528, 2019.

Sébastien Bubeck, Ofer Dekel, Tomer Koren, and Yuval Peres. Bandit convex optimization: $\sqrt{T}$ regret in one dimension. In *Proceedings of the 28th Conference on Learning Theory (COLT)*, volume 40, pages 266–278, 2015.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006.

Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.

Tianyi Chen and Georgios B. Giannakis. Bandit convex optimization for scalable and dynamic IoT management. *IEEE Internet of Things Journal*, 6(1):1276–1286, 2019.

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

Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 345–352, 2007.

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

Ofer Dekel, Ronen Eldan, and Tomer Koren. Bandit smooth convex optimization: Improving the bias-variance tradeoff. In *Advances in Neural Information Processing Systems 28 (NIPS)*, pages 2926–2934, 2015.

John C. Duchi, Michael I. Jordan, Martin J. Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

Abraham Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 385–394, 2005.

Luigi Fortuna, Salvatore Graziani, Alessandro Rizzo, and Maria Gabriella Xibilia. *Soft sensors for monitoring and control of industrial processes.* Springer Science & Business Media, 2007.

João Gama, Indre Zliobaite, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44:1–44:37, 2014.

Yonatan Gur, Assaf J. Zeevi, and Omar Besbes. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 199–207, 2014.

Elad Hazan. Introduction to Online Convex Optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.

Elad Hazan and Kfir Y. Levy. Bandit convex optimization: Towards tight bounds. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 784–792, 2014.

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

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

Mark Herbster and Manfred K. Warmuth. Tracking the best expert. *Machine Learning*, 32 (2):151–178, 1998.

Mark Herbster and Manfred K. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1:281–309, 2001.

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 (AISTATS)*, pages 398–406, 2015.

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 (AISTATS)*, pages 943–951, 2017.

Robert D. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems 17 (NIPS)*, pages 697–704, 2004.

Jeremy Z. Kolter and Marcus A. Maloof. Using additive expert ensembles to cope with concept drift. In *Proceedings of the 22rd International Conference on Machine Learning (ICML)*, pages 449–456, 2005.

Tor Lattimore. Improved regret for zeroth-order adversarial bandit convex optimisation. *ArXiv preprint*, arXiv:2006.00475, 2020.

Chung-Wei Lee, Haipeng Luo, Chen-Yu Wei, and Mengxiao Zhang. Bias no more: high-probability data-dependent regret bounds for adversarial bandits and MDPs. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pages 15522–15533, 2020.

Haipeng Luo, Chen-Yu Wei, Alekh Agarwal, and John Langford. Efficient contextual bandits in non-stationary worlds. In *Proceedings of the 31st Conference On Learning Theory (COLT)*, pages 1739–1776, 2018.

H. Brendan McMahan and Avrim Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *Proceedings of the 17th Annual Conference on Learning Theory (COLT)*, pages 109–123, 2004.

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 (CDC)*, pages 7195–7201, 2016.

Yurii Nesterov. Random gradient-free minimization of convex functions. Technical report, Université catholique de Louvain, Center for Operations Research and Econometrics (ECORE), 2011.

Ankan Saha and Ambuj Tewari. Improved regret guarantees for online smooth convex optimization with bandit feedback. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 636–642, 2011.

Shai Shalev-Shwartz. Online Learning and Online Convex Optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

Ohad Shamir. On the complexity of bandit and derivative-free stochastic convex optimization. In *Proceedings of the 26th Annual Conference on Learning Theory (COLT)*, pages 3–24, 2013.

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

Masashi Sugiyama and Motoaki Kawanabe. *Machine Learning in Non-stationary Environments: Introduction to Covariate Shift Adaptation*. The MIT Press, 2012.

Tim van Erven and Wouter M. Koolen. Metagrad: Multiple learning rates in online learning. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 3666–3674, 2016.

Guanghui Wang, Dakuan Zhao, and Lijun Zhang. Minimizing adaptive regret with one gradient per iteration. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2762–2768, 2018.

Chen-Yu Wei, Yi-Te Hong, and Chi-Jen Lu. Tracking the best expert in non-stationary stochastic environments. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 3972–3980, 2016.

Scott Yang and Mehryar Mohri. Optimistic bandit convex optimization. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 2289–2297, 2016.

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 (ICML)*, pages 449–457, 2016.

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

Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive online learning in dynamic environments. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 1330–1340, 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 (ICML)*, 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 (ICML)*, pages 7414–7423, 2019.

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 (AISTATS)*, pages 309–319, 2020a.

Yu-Jie Zhang, Peng Zhao, and Zhi-Hua Zhou. A simple online algorithm for competing with dynamic comparators. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 390–399, 2020b.

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 (L4DC)*, 2021.

Peng Zhao, Guanghui Wang, Lijun Zhang, and Zhi-Hua Zhou. Bandit convex optimization in non-stationary environments. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1508–1518, 2020a.

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 (NeurIPS)*, pages 12510–12520, 2020b.

Peng Zhao, Xinqiang Wang, Siyu Xie, Lei Guo, and Zhi-Hua Zhou. Distribution-free one-pass learning. *IEEE Transaction on Knowledge and Data Engineering*, 33:951–963, 2021.

Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms.* Chapman & Hall/CRC Press, 2012.

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