Journal of Machine Learning Research 24 (2023) 1-50 Submitted 12/21; Revised 3/23; Published 8/23

Online Non-stochastic Control with Partial Feedback

Yu-Hu Yan YANYH@LAMDA.NJU.EDU.CN
Peng Zhao ZHAOP@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: Ambuj Tewari

Abstract

Online control with non-stochastic disturbances and adversarially chosen convex cost functions, referred to as *online non-stochastic control*, has recently attracted increasing attention. We study online non-stochastic control with *partial feedback*, where learners can only access partially observed states and partially informed (bandit) costs. The problem setting arises naturally in real-world decision-making applications and strictly generalizes exceptional cases studied disparately by previous works. We propose the first online algorithm for this problem, with an $\widetilde{\mathcal{O}}(T^{3/4})$ regret competing with the best policy in hindsight, where T denotes the time horizon and the $\widetilde{\mathcal{O}}(\cdot)$-notation omits the poly-logarithmic factors in T. To further enhance the algorithms' robustness to changing environments, we then design a novel method with a two-layer structure to optimize the *dynamic regret*, a more challenging measure that competes with time-varying policies. Our method is based on the online ensemble framework by treating the controller above as the base learner. On top of that, we design two different meta-combiners to simultaneously handle the unknown variation of environments and the memory issue arising from the online control. We prove that the two resulting algorithms enjoy $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/2})$ and $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/4}+T^{5/6})$ dynamic regret respectively, where P_T measures the environmental non-stationarity. Our results are further extended to unknown transition matrices. Finally, empirical studies in both synthetic linear and simulated nonlinear tasks validate our method's effectiveness, thus supporting the theoretical findings.

Keywords: online non-stochastic control, partial feedback, dynamic regret, online ensemble, online learning with memory, bandit convex optimization

1 Introduction

In online decision making, an agent interacts with unknown environments and receives specific feedback for making decisions to maximize the cumulative rewards (or equivalently, minimizing the cumulative costs). The problem is of great significance because it, on the one hand, draws tight connections with different disciplines such as reinforcement learning (Sutton and Barto, 2018), control theory (Kirk, 2004), online learning (Shalev-Shwartz, 2012), and operations research (Taha, 2003) while, on the other hand, has many applications in real-world tasks ranging from game playing (Marden and Shamma, 2018) to the dialog system (Bubeck et al., 2023).

A large body of literature has recently been devoted to leveraging modern machine learning techniques to design online decision-making methods with provable non-asymptotic

guarantees. In particular, *online non-stochastic control* (Agarwal et al., 2019) has attracted increasing attention, which studies the problem of controlling linear dynamical systems with non-stochastic disturbances and adversarially chosen convex cost functions. Specifically, the learner aims to control the following linear dynamical system (LDS),

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \boldsymbol{\xi}_t, \tag{1.1}$$

where $\mathbf{x}_t \in \mathcal{X}, \mathbf{u}_t \in \mathcal{U}$ denote the state and action, A, B are the system transition matrices, and $\boldsymbol{\xi}_t$ is a non-stochastic disturbance, that is, without any distributional assumptions. Ahead of time, environments adversarially choose convex cost functions $c_1, \ldots, c_T : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^+$ unknown to the learner, where T is the time horizon. In each round $t \in [T]$, the learner submits an action $\mathbf{u}_t$, suffers a cost $c_t(\mathbf{x}_t, \mathbf{u}_t)$ and updates the policy. This problem is of great importance. On the one hand, it relaxes the requirements of traditional control theory, leading to more robust and practical algorithms for real-world decision-making applications. On the other hand, it establishes profound connections between modern online learning and decision-making problems, promoting more methods with sound theoretical guarantees.

The goal of online non-stochastic control is to minimize the game-theoretic *regret* (Zinkevich, 2003), defined as the difference between the cumulative cost of the learner's policy and that of the best one in hindsight. Formally,

$$\text{S-Reg}_T \triangleq \sum_{t=1}^{T} c_t(\mathbf{x}_t, \mathbf{u}_t) - \min_{\pi \in \Pi} \sum_{t=1}^{T} c_t(\mathbf{x}_t^\pi, \mathbf{u}_t^\pi), \tag{1.2}$$

where $\mathbf{x}_t^\pi, \mathbf{u}_t^\pi$ are produced by the compared policy π from a certain policy class Π. The measure (1.2) is also called *static regret* to emphasize that the compared policy is fixed over the horizon. Agarwal et al. (2019) initiated the study of online non-stochastic control and proposed a reduction to online convex optimization with memory (Anava et al., 2015) via a novel policy parameterization, with an $\widetilde{\mathcal{O}}(\sqrt{T})$ regret. The $\widetilde{\mathcal{O}}(\cdot)$-notation omits the poly-logarithmic factors in T. Note that the result holds for the standard *full feedback*, where states are fully observed and the learner has full information of the cost functions.

Nevertheless, in real-world decision-making tasks, usually the learner can only receive *partial feedback*. For example, in a data center, multiple servers operating near-by might produce a considerable amount of heat, so monitoring and cooling are essential to avoid equipment damage due to the high temperature (Lazic et al., 2018). Recently, data-driven control method has been explored as a means of this problem to have a minimal electricity cost (Gao, 2014). The data center cooling problem is naturally with partial feedback. First, the state observation is partial since we may only receive a small part of the descriptive statistics. Moreover, the cost is partially informed because it is hard to know exactly how the electricity is consumed. Targeting such problems, we aim to design effective methods with sound theoretical guarantees for online non-stochastic control with partial feedback.

Some recent efforts are devoted to relaxing the requirements of feedback quality. Among them, Gradu et al. (2020); Cassel and Koren (2020) considered the bandit costs, and Simchowitz et al. (2020) studied the partially observed states. The two kinds of partial feedback reflect certain aspects of the problem and are likely to exist together in real applications, as mentioned in the data center cooling example above. However, previous works do not consider them simultaneously. Besides, all aforementioned results choose the static regret (1.2)

as the performance measure. Recently, *open-environment machine learning* (Zhou, 2022a) has received much attention, where the machine learning process has to handle unknown changes that have never occurred in training data, e.g., emerging new classes (Mu et al., 2017; Cai et al., 2019), feature change (Hou et al., 2017; Hou and Zhou, 2018), and data distribution change (Sugiyama and Kawanabe, 2012; Zhao et al., 2021b). Under such situations, the performance of the best fixed policy could be poor, which necessitates the learner to compete with a sequence of changing policies. Thus in this work, we further consider the *dynamic regret*, defined as

$$\text{D-Reg}(\pi_{1:T}) \triangleq \sum_{t=1}^{T} c_t(\mathbf{x}_t, \mathbf{u}_t) - \sum_{t=1}^{T} c_t(\mathbf{x}_t^{\pi_t}, \mathbf{u}_t^{\pi_t}), \tag{1.3}$$

where $\pi_1, \ldots, \pi_T \in \Pi$ represent a sequence of time-varying policies. This metric is general since the compared policies are allowed to change. In particular, the static regret (1.2) is a special case by taking all the compared policies as the best fixed one in hindsight, namely, $\pi_t = \pi^\star \in \arg\min_{\pi\in\Pi} \sum_{t=1}^{T} c_t(\mathbf{x}_t^\pi, \mathbf{u}_t^\pi)$ for any $t \in [T]$. It is well-known that a sublinear dynamic regret is impossible in the worst case without additional restrictions on the comparators (Besbes et al., 2015; Zhao and Zhang, 2021). To this end, we introduce the cumulative variation of compared policies, referred to as path length $P_T \triangleq \sum_{t=2}^{T} d(\pi_{t-1}, \pi_t)$, where $d(\cdot, \cdot)$ quantifies the difference between two policies. Notice that the path length essentially measures the environmental non-stationarity, and thus an ideal dynamic regret upper bound should be a function of this metric. Indeed, the fundamental obstacle in dynamic regret optimization lies in automatically adapting to the unknown non-stationarity (Zhao et al., 2020, 2021a,c), which in this context refers to the unknown path length P_T.

In this work, we first adopt a policy parameterization (Agarwal et al., 2019; Simchowitz et al., 2020) called Disturbance-Response Policy (see Definition 2 for details) to reduce online non-stochastic control with partial feedback to bandit convex optimization with memory and inexact feedback, and then design an online method with $\mathcal{O}(T^{3/4})$ static regret, extending the previous work of Cassel and Koren (2020) to partially observed state information. We further demonstrate that our proposed method achieves an $\mathcal{O}(T^{3/4}(1 + P_T)^{1/4})$ dynamic regret bound whenever the environmental non-stationarity measure P_T is known apriori to the learner and can be used as the algorithm input.

Furthermore, since the prior knowledge of environmental changes is typically unavailable in practice, inspired by recent advancements in non-stationary online learning (Zhao et al., 2021a, 2022b), we propose a two-layer method using an online ensemble framework (Zhou, 2012; Zhao, 2021) to tackle this uncertainty. This method consists of multiple base learners for exploration and a meta learner employing expert-tracking techniques to adaptively track the best base learner. Additionally, the limited feedback information and the memory effect inherent in online control make the problem considerably more challenging. Consequently, we introduce two distinct base learner scheduling schemes based on *weighted combination* and *optional selection* strategies, respectively, to handle environmental non-stationarity under limited feedback. We also employ carefully designed regularized surrogate costs to address the memory effect. Accordingly, we propose a new method for online non-stochastic control with partial feedback, called PaRtial feedback Online Non-stochastic Control (abbreviated as Pronc), with two implementations: Pronc-Combine

Table 1: Summary of our results in both known and unknown systems. Overall we propose two online ensemble algorithms, called Pronc-Combine and Pronc-Select, based on the weighted combination and optional selection strategy, respectively. The two methods are comparable under different degrees of the environmental non-stationarity.

	Known System		**Unknown System**	
	regret	**preferred**	**regret**	**preferred**
Pronc-Combine [Section 4.3]	$\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/2})$ [Theorem 2 and 4]	$P_T \leq \mathcal{O}(T^{1/6})$	$\widetilde{\mathcal{O}}(T^{4/5}(1+P_T)^{1/2})$ [Theorem 5]	$P_T \leq \mathcal{O}(T^{1/15})$
Pronc-Select [Section 4.4]	$\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/4}+T^{5/6})$ [Theorem 3 and 4]	$P_T \geq \Omega(T^{1/6})$	$\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/4}+T^{5/6})$ [Theorem 5]	$P_T \geq \Omega(T^{1/15})$

(weighted combination) and Pronc-Select (optional selection). We prove that the two algorithms enjoy $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/2})$ and $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/4}+T^{5/6})$ dynamic regret guarantees, respectively. Note that the two algorithms are favored in different situations. Please refer to the 'Known System' column in Table 1 for details. By doing so, we establish the first dynamic regret guarantee for online non-stochastic control with partial feedback.

Furthermore, we extend the results to unknown systems where transition matrices in (1.1) are unavailable to the learner. We tackle this issue by adopting a plug-in system estimation operation (Simchowitz et al., 2020), achieving $\widetilde{\mathcal{O}}(T^{4/5}(1+P_T)^{1/2})$ and $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/4}+T^{5/6})$ dynamic regret bounds respectively. Please refer to the 'Unknown System' column in Table 1 for the comparison of the two bounds in different cases. We finally conduct experiments in synthetic linear and simulated nonlinear environments to validate the effectiveness and efficiency of the proposed method.

The main contributions of this paper are summarized as follows.

- We introduce the problem of online non-stochastic control with *partial feedback* that considers not only non-stochastic disturbances and adversarial cost functions but also the ubiquitous partial feedback in real-world online decision-making tasks.
- We present the *first* dynamic regret for this problem by designing novel online methods, which admit the online ensemble framework with different scheduling schemes (weighted combination or optional selection). The two realized algorithms enjoy $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/2})$ and $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/4}+T^{5/6})$ dynamic regret bounds respectively.
- As a byproduct, we give an $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/4}+T^{5/6})$ dynamic regret bound for standard bandit convex optimization problem, improving upon the best known result of $\mathcal{O}(T^{3/4}(1+P_T)^{1/2})$ (Zhao et al., 2021a) when the environmental non-stationarity is large, more concretely, $P_T \geq \Omega(T^{1/6})$. The result may be of independent interest.

We organize the rest of the paper as follows. Section 2 presents brief reviews of the related work. Section 3 introduces the problem setup and lists the assumptions used throughout this paper. In Section 4, the main contribution of this paper, we propose our method in detail and offer the corresponding theoretical guarantees. Section 5 presents the empirical performance of our proposed method, and Section 6 concludes the paper. We defer

some additional notations and preliminaries to Appendix A, proofs of all the theorems to Appendix B, and supporting lemmas to Appendix C.

2 Related Work

This section briefly reviews related works of online non-stochastic control and dynamic regret of online convex optimization (with memory).

Online Non-stochastic Control. Online control is a field with a rich and extensive history. In the past few decades, considerable efforts have been taken in the expansive field of classic online control (Ljung and Söderström, 1983; Guo and Ljung, 1995; Harold et al., 1997; Fiechter, 1997; Abbasi-Yadkori and Szepesvári, 2011; Cohen et al., 2018; Guo, 2020). This list is far from exhaustive, and interested readers can delve into the references therein for recent developments in online control.

In recent years, there has been a surge in efforts to develop robust control algorithms using data-driven techniques from statistics and machine learning. A notable line of research focuses on the problem of *online non-stochastic control* (Hazan, 2020), in which both the cost functions and disturbances can exhibit adversarial behavior. In the pioneering work of Agarwal et al. (2019), the authors reduced the problem to online convex optimization with memory (Anava et al., 2015) at an acceptable cost by the disturbance-action policy parameterization as well as a truncation operation and obtained an optimal $\widetilde{\mathcal{O}}(\sqrt{T})$ regret against the best linear controller. As a side note, the optimal rate is also attainable in unknown systems with stochastic disturbances (Cassel et al., 2022b) or stochastic convex costs (Cassel et al., 2022a). The subsequent works considered more challenging tasks. Hazan et al. (2020) studied unknown system transition and achieved an $\widetilde{\mathcal{O}}(T^{2/3})$ regret via system identification. Simchowitz et al. (2020) investigated partially observed states via a novel perspective called "Nature's y" and established theoretical guarantees in various cases. Specifically, in known systems, they attained $\widetilde{\mathcal{O}}(\sqrt{T})$ regret for convex losses with adversarial disturbances and $\mathcal{O}(\mathrm{poly}(\log T))$ regret for strongly convex as well as smooth losses with semi-non-stochastic disturbances. In unknown systems, they obtained $\widetilde{\mathcal{O}}(T^{2/3})$ and $\widetilde{\mathcal{O}}(\sqrt{T})$ regret in the two aforementioned cases. Gradu et al. (2020); Cassel and Koren (2020) both assumed bandit (partially informed) cost functions and proved an $\widetilde{\mathcal{O}}(T^{3/4})$ regret. Simchowitz (2020) considered strongly convex cost functions and presented $\mathcal{O}(\log T)$ and $\widetilde{\mathcal{O}}(\sqrt{T})$ regret in known and unknown systems respectively based on a novel variant of the online Newton step algorithm (Hazan et al., 2006).

While static regret minimization yields fruitful results, in many real-world applications the environments can be non-stationary such that the best fixed comparator may also perform poorly over the whole time horizon. To this end, Zhao et al. (2022b) introduced dynamic (policy) regret for online non-stochastic control and achieved an optimal $\widetilde{\mathcal{O}}(\sqrt{T(1+P_T)})$ bound, where P_T reflects the variation of compared policies.[1] Note that they only considered the full feedback, i.e., fully informed costs and fully observed states. Our paper substantially extends theirs by considering the more challenging partial feedback setting. Detailed comparisons of problem setups can be found in Table 2. As for methods,

1. The extended journal version (Zhao et al., 2023) provides further improvements by achieving the optimal dependence on the memory length and extending the results to unknown systems.

briefly, both works employ the online ensemble framework to optimize the dynamic regret, but nevertheless the bandit feedback considered in our work brings unique challenges. To handle the lack of feedback information, we propose two novel online ensemble methods based on parallel and serial updates, which can fully exploit the feedback information to update the base learner(s) properly. We defer a more detailed discussion of the technical differences from Zhao et al. (2022b) to Remark 4 in Section 4.3. Apart from convex costs, the work by Baby and Wang (2022) achieved an enhanced dynamic regret for online non-stochastic control with quadratic cost functions.

Moreover, Gradu et al. (2023); Zhang et al. (2022b) adopted adaptive regret to ensure low-regret guarantees in arbitrary intervals, that is, for any $I \subseteq [1, T]$,

$$\text{A-Reg}_I \triangleq \sum_{t \in I} c_t(\mathbf{x}_t, \mathbf{u}_t) - \min_{\pi \in \Pi} \sum_{t \in I} c_t(\mathbf{x}_t^\pi, \mathbf{u}_t^\pi).$$

Specifically, Gradu et al. (2023) studied the weakly adaptive regret and obtained the expected regret bound as $\max_{I \subseteq [1,T]} \text{A-Reg}_I \leq \widetilde{\mathcal{O}}(\sqrt{T})$. Zhang et al. (2022b) achieved an $\widetilde{\mathcal{O}}(\sqrt{|I|})$ deterministic strongly adaptive regret for any interval $I \subseteq [1, T]$. Although a black-box reduction from dynamic regret to adaptive regret has been proved in the online linear optimization over simplex problem (Luo and Schapire, 2015), in the general setup of online convex optimization, the relationship between dynamic regret and adaptive regret is generally unclear (Zhang, 2020, Section 5). Besides, the techniques to optimize these two measures are significantly different. Concretely, to optimize adaptive regret, static algorithms need to run on some carefully designed interval covering (Daniely et al., 2015; Zhang et al., 2019), while for dynamic regret, base algorithms run on the whole horizon but have to deal with a certain degree of non-stationarity (Zhang et al., 2018; Zhao et al., 2021c).

Apart from regret minimization studied by aforementioned works, optimizing the competitive ratio (i.e., the worst-case ratio of the total cost incurred by the online learner and the offline optimal cost) is also an important and rising topic in online control (Shi et al., 2020; Zhang et al., 2021; Goel et al., 2023; Goel and Hassibi, 2023).

Dynamic Regret of Online Convex Optimization (with Memory). Regret minimization serves as a cornerstone in online learning (Hazan, 2016). A well-studied performance measure is static regret, which depicts the learner's excess loss compared to the best fixed comparator in hindsight. However, such a measure is not favorable enough in changing environments, because even the best comparator may also perform poorly under such conditions. Recognizing this limitation, in the past few decades, there have been efforts devoted to other performance measures considering more non-stationarity. In the field of prediction with expert advice, a measure called tracking regret (also known as shifting/switching regret), has received much attention (Herbster and Warmuth, 1998; Cesa-Bianchi et al., 2012; György and Szepesvári, 2016; Wei et al., 2016; Luo et al., 2022). This measure allows for a more flexible comparator that can adapt to changes multiple times.

In online convex optimization, Zinkevich (2003) proposed the general notion of dynamic regret, which generalizes the standard static regret by competing with an arbitrary sequence of changing comparators inside the feasible domain, namely,

$$\text{D-Reg}(\mathbf{v}_{1:T}) \triangleq \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{v}_t). \tag{2.1}$$

It recovers the static regret by taking all comparators as the best one in hindsight, i.e., $\mathbf{v}_t = \mathbf{v}^\star \in \arg\min_{\mathbf{v}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{v})$ for all $t \in [T]$, where $\mathcal{K}$ is the feasible set. With the prior knowledge of the path length P_T, online gradient descent with an optimal step size is able to obtain an $\mathcal{O}(\sqrt{T(1+P_T)})$ regret bound (Zinkevich, 2003). While without such prior knowledge, this result degenerates to $\mathcal{O}(\sqrt{T}(1+P_T))$. Later Zhang et al. (2018) proposed a tighter $\mathcal{O}(\sqrt{T(1+P_T)})$ result even without knowing P_T in advance and established a matching lower bound to prove its optimality. Subsequently, problem-dependent dynamic regret guarantees in terms of the gradient variation and small loss were developed by Zhao et al. (2020) for smooth functions. Moreover, a more efficient version with the same bounds and only one gradient per iteration was achieved by the collaborated online ensemble framework proposed by Zhao et al. (2021c), even though $\mathcal{O}(\log T)$ base learners are simultaneously performed at each round. There are many recent advancements and applications of dynamic regret minimization (Cutkosky, 2020; Yuan and Lamperski, 2020; Zhang et al., 2021; Baby and Wang, 2021; Jacobsen and Cutkosky, 2022; Zhang et al., 2022a; Zhao et al., 2022c; Bai et al., 2022; Yan et al., 2023; Zhang et al., 2023).

Besides works that only consider the effect of the current decision, others study the impact of the decisions in the near past, called online learning with memory. The memory problem is first studied in the expert setting (Merhav et al., 2002; Geulen et al., 2010; György and Neu, 2014). For general convex sets, Anava et al. (2015) initiated the study of OCO with memory, benchmarked by the following policy regret (Dekel et al., 2012):

$$\text{S-Reg}_T \triangleq \sum_{t=H+1}^{T} f_t(\mathbf{w}_{t-H:t}) - \min_{\mathbf{v}\in\mathcal{K}} \sum_{t=H+1}^{T} f_t(\mathbf{v},\ldots,\mathbf{v}), \tag{2.2}$$

where $f_t : \mathcal{K}^{H+1} \mapsto \mathbb{R}$ is a function of the past $H+1$ decisions $\mathbf{w}_{t-H:t}$, served as a description of the memory. The authors proposed a gradient descent algorithm with $\mathcal{O}(\sqrt{T})$ and $\mathcal{O}(\log T)$ regret for convex and strongly convex loss functions, respectively. Just like standard OCO, the performance against a fixed comparator is less attractive in non-stationary environments, motivating the need to compete with time-varying comparators. Due to this, Zhao et al. (2022b) introduced the more general dynamic policy regret,

$$\text{D-Reg}(\mathbf{v}_{1:T}) \triangleq \sum_{t=H+1}^{T} f_t(\mathbf{w}_{t-H:t}) - \sum_{t=H+1}^{T} f_t(\mathbf{v}_{t-H:t}), \tag{2.3}$$

where $\mathbf{v}_{1:T}$ represents any sequence of changing comparators inside the domain. The authors established a minimax optimal $\mathcal{O}(\sqrt{T(1+P_T)})$ regret guarantee without the prior knowledge of P_T through a meta-base online ensemble method.

While traditional online learning primarily focus on the impact of current decisions, with little regard for their future influence, recent research is increasingly emphasizing the importance of online decision-making, which depicts the sequential interaction between the learner and the environments (Foster et al., 2021; Wang et al., 2022). Online decision making draws deep ties with the memory in online learning, which brings the impact of past decisions into the present. Apart from online non-stochastic control, the focus of this paper, there are other paradigms in online decision-making worth mentioning, such as online Markov decision processes (MDP) and rehearsal learning. In online MDP, the technique from online

learning with memory is also proved to play an important role (Zhao et al., 2022a). As for rehearsal learning, it focuses on enabling the learner to act proactively to prevent undesirable outcomes, emerging as a promising domain for further exploration (Zhou, 2022b).

3 Preliminaries

In this section, we formalize the problem setup and list some standard assumptions that will be used in the theoretical analysis.

3.1 Problem Setup

We study online non-stochastic control with partial feedback where only partially observed states and partially informed costs are accessible. Consider the following dynamical system:

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \boldsymbol{\xi}_t, \quad \mathbf{y}_t = C\mathbf{x}_t + \mathbf{e}_t, \tag{3.1}$$

where $A \in \mathbb{R}^{d_x \times d_x}, B \in \mathbb{R}^{d_x \times d_u}, C \in \mathbb{R}^{d_y \times d_x}$ are system transition matrices, $\mathbf{x}_t \in \mathbb{R}^{d_x}$ denotes the state, $\mathbf{u}_t \in \mathbb{R}^{d_u}$ is the learner's action guided by a certain policy, $\boldsymbol{\xi}_t \in \mathbb{R}^{d_x}$ and $\mathbf{e}_t \in \mathbb{R}^{d_y}$ are the disturbances chosen by an oblivious adversary. In round t, the learner only receives a partial observation $\mathbf{y}_t \in \mathbb{R}^{d_y}$ of the state, and a partially informed (bandit) cost $c_t(\mathbf{y}_t, \mathbf{u}_t)$, where $c_t : \mathbb{R}^{d_y} \times \mathbb{R}^{d_u} \mapsto \mathbb{R}^+$ is a convex and Lipschitz-continuous function adversarially chosen by the oblivious adversary. We adopt dynamic regret (1.3) as the performance measure. In the following, we conclude the capability and generality of our problem setup.

Capability. Online non-stochastic control is powerful in modeling due to its relaxed assumptions on the disturbances and cost functions, which enable the model to be applied to broader real-world applications. For instance, the model can tolerate misspecifications such as non-linearity and non-stationarity by treating unmodeled parts as non-stochastic disturbances. This is also validated via experiments (see Section 5). Furthermore, this paper considers the much weaker partial feedback. Specifically, our problem setup requires only partially informed costs and partially observed states and is thus closer to real-world decision-making tasks. At last, methods with dynamic regret guarantees are provably competitive with a sequence of time-varying policies and are thus more attractive than the standard static regret in non-stationary environments.

Generality. Our problem setup generalizes those of previous works (Agarwal et al., 2019; Cassel and Koren, 2020; Gradu et al., 2020; Simchowitz et al., 2020; Hazan et al., 2020; Zhao et al., 2022b) from various aspects including cost information, state observability, system knowability, and performance measure. Table 2 reports the comparison between our problem setup and those of the related works.

3.2 Assumptions

We list the assumptions used in this paper, which are common in the literature (Agarwal et al., 2019; Simchowitz et al., 2020; Hazan, 2020).

Assumption 1 (Boundedness)**.** The system transition matrices satisfy $\rho(A) \leq \kappa_A < 1, \|B\|_{\text{op}} \leq \kappa_B, \|C\|_{\text{op}} \leq \kappa_C$, where $\rho(\cdot)$ is the spectral radius and $\|\cdot\|_{\text{op}}$ denotes the matrix operator norm. And the disturbances are bounded as $\|\boldsymbol{\xi}_t\|_2, \|\mathbf{e}_t\|_2 \leq W$ for all $t \in [T]$.

Table 2: Comparisons of our work and previous ones regarding problem setups, including cost information, state observability, system knowability, and performance measure.

References	Partial Costs	Partial States	Unknown System	Dynamic Regret
Agarwal et al. (2019)	✗	✗	✗	✗
Cassel and Koren (2020)	✓	✗	✗	✗
Gradu et al. (2020)	✓	✗	✗	✗
Hazan et al. (2020)	✗	✗	✓	✗
Simchowitz et al. (2020)	✗	✓	✓	✗
Zhao et al. (2022b)	✗	✗	✗	✓
Ours	✓	✓	✓	✓

Note that Assumption 1 is without loss of generality since it can be extended to strongly stabilizable systems, where the system is instable, but can be stabilized by a linear controller, due to the reduction proposed in Appendix A of Cassel et al. (2022b).

Assumption 2 (Lipschitzness)**.** There exists a constant $L_c > 0$ for non-negative convex cost function c_t such that for all $(\mathbf{y}, \mathbf{u}), (\mathbf{y}', \mathbf{u}') \in \mathbb{R}^{d_y+d_u}$, and $R_c = \max\{\|[\mathbf{y}; \mathbf{u}]\|_2, 1\}$,

$$|c_t(\mathbf{y}, \mathbf{u}) - c_t(\mathbf{y}', \mathbf{u}')| \leq L_c R_c \left\|[\mathbf{y} - \mathbf{y}', \mathbf{u} - \mathbf{u}']^\top\right\|_2, \quad 0 \leq c_t(\mathbf{y}, \mathbf{u}) \leq L_c R_c^2.$$

Note that L_c is the intrinsic Lipschitz constant of the cost function. The $L_c R_c$ scaling of the Lipschitz constant aims to describe cost functions whose Lipschitz constant scales with radius (Simchowitz et al., 2020), e.g., quadratic costs.

4 Our Method

This section proposes our method for online non-stochastic control with partial feedback. We first reduce the problem to online learning in Section 4.1. In Section 4.2, we design the first online algorithm for the reduced problem and then provide both static and dynamic regret bounds whenever the environmental non-stationarity is known. To step further, we propose a meta-base online ensemble method with two different scheduling schemes to handle the unknown environmental non-stationarity in Section 4.3 and Section 4.4. Finally, we present the overall results for online non-stochastic control in Section 4.5.

4.1 Reduction to Online Learning

In this part, we reduce the online non-stochastic control problem to online convex optimization with memory by policy parameterization, specifically, through disturbance-response policies and a truncation operation. The reduction further allows us to update the policy via advanced online learning techniques.

First, we briefly introduce the disturbance-response policy (DRP). It is developed to handle partially observed states in the seminal work of Simchowitz et al. (2020) and follows the classical Youla parameterization (Youla et al., 1976) via the perspective of "Nature's y", defined as follows.

Definition 1 (Nature's y)**.** Nature's y (denoted by $\mathbf{y}^{\text{nat}}$) is the observation in the absence of any action on the system. In linear dynamical system with partial feedback (3.1), given disturbances $\boldsymbol{\xi}_{1:t}, \mathbf{e}_{1:t}$, $\mathbf{y}_t^{\text{nat}} \triangleq \mathbf{e}_t + \sum_{i=1}^{t-1} CA^{i-1}\boldsymbol{\xi}_{t-i}$ holds.

Intuitively, Nature's y is an external observation of the cumulative impact of disturbances. Consequently, we formulate the disturbance-response policy, which considers the influence of Nature's y in the limited past.

Definition 2. Given Nature's $\mathbf{y}^{\text{nat}}_{t-m+1:t}$, a disturbance-response policy, parameterized by a m-length tuple of matrices $M = (M^{[0]}, \ldots, M^{[m-1]}) \in \mathcal{M}$, chooses the action as $\mathbf{u}_t(M) = \sum_{i=0}^{m-1} M^{[i]}\mathbf{y}^{\text{nat}}_{t-i}$, where $\mathcal{M}$ denotes the domain of policy parameters.

DRP is powerful in handling partially observed states and encompasses many policy classes of interest. Choosing $\Pi = \Pi_{\text{DRP}}$ in (1.3) forms the performance measure used in our work. We defer the derivation of Definition 1 and the relation between the observation $\mathbf{y}$ and Nature's $\mathbf{y}^{\text{nat}}$ to Appendix A.2.

DRP parametrizes the observation $\mathbf{y}_t$ and action $\mathbf{u}_t$ as affine functions of its parameters $M_{1:t}$, which makes the cost $c_t(\mathbf{y}_t(M_{1:t-1}), \mathbf{u}_t(M_t))$ convex in $M_{1:t}$. Denoting by $h_t(M_{1:t})$ the parameterized loss, the control problem seems to be transformed into pure online learning with memory, with regret $\sum_{t=1}^{T} h_t(M_{1:t}) - \sum_{t=1}^{T} h_t(M^{\star}_{1:t})$. However, one obstacle that prohibits using OCO with memory is the time-varying memory length. To address this issue, a truncation operation (Agarwal et al., 2019; Simchowitz et al., 2020) is proposed to artificially erase the effect of actions of more than H rounds before. Intuitively, the rationality behind the truncation operation is that in stable systems (Assumption 1), the impact of the actions in the far past can be nearly ignored. The truncated observation, action, and the corresponding cost are defined as follows.

Definition 3. In round t, the truncated action $\widetilde{\mathbf{u}}_t$, observation $\widetilde{\mathbf{y}}_t$, cost f_t are defined as

$$\widetilde{\mathbf{u}}_t\left(M_t\right) \triangleq \sum_{i=0}^{m-1} M_t^{[i]}\mathbf{y}^{\text{nat}}_{t-i}, \quad \widetilde{\mathbf{y}}_t\left(M_{t-H:t-1}\right) \triangleq \mathbf{y}^{\text{nat}}_t + \sum_{i=1}^{H} G^{[i]}\widetilde{\mathbf{u}}_{t-i}\left(M_{t-i}\right),$$
$$f_t\left(M_{t-H:t}\right) \triangleq c_t\left(\widetilde{\mathbf{y}}_t\left(M_{t-H:t-1}\right), \widetilde{\mathbf{u}}_t\left(M_t\right)\right),$$

where $M_{t-H:t} \in \mathcal{M}^{H+1}$ are policy parameters, and G is the Markov operator of the system, a tuple of matrices of length H, with each entry $G^{[i]} \triangleq CA^{i-1}B$ for $i \in [H]$.

The Markov operator G describes how the system transforms the impact of actions into observations, reflecting certain properties of the system. The following lemma shows that, by carefully choosing the memory length H, the truncation error between the control cost and truncated cost can be neglected in terms of the time horizon T.

Lemma 1 (Lemma C.3 of Simchowitz et al. (2020))**.** *Under Assumption 2, in known systems, choosing memory length $H = \Theta(\log T)$ gives $\varepsilon \le \mathcal{O}(1/T)$, where $\varepsilon \triangleq \max_{t\in[T]} |\varepsilon_t|$ and $\varepsilon_t \triangleq |c_t(\mathbf{y}_t, \mathbf{u}_t) - f_t(M_{t-H:t})|$ denotes the truncation error of round t.*

It is noteworthy to point out that Lemma 1 only holds in known systems, where the truncated cost f_t is actually parametrized by the Markov operator G and Nature's $\mathbf{y}^{\text{nat}}_{t-H-m+1:t}$, i.e., $f_t(\cdot|G, \mathbf{y}^{\text{nat}}_{t-H-m+1:t})$. As will be discussed in Section 4.5, in unknown systems, the inaccurately estimated parameters will further import bias into the truncation error, which needs more involved analysis.

Through the disturbance-response policy and the truncation operation, we reduce online non-stochastic control to *bandit* convex optimization with memory and *inexact* feedback.

The feedback is both bandit and inexact due to the bandit nature of control costs and the truncation error, respectively. For ease of understanding and generality, we focus on the vector domain $\mathcal{K}$ instead of the policy parameter domain $\mathcal{M}$. Specifically, in round t, the learner submits $\mathbf{w}_t$ and receives an inexact loss value $f_t(\mathbf{w}_{t-H:t}) + \varepsilon_t$. The learner's performance is measured by dynamic policy regret (2.3), restated as follows,

$$\text{D-Reg}(\mathbf{v}_{1:T}) \triangleq \sum_{t=H}^{T} f_t(\mathbf{w}_{t-H:t}) - \sum_{t=H}^{T} f_t(\mathbf{v}_{t-H:t}), \tag{4.1}$$

where $\mathbf{v}_{1:T}$ denotes a sequence of time-varying comparators. The truncated costs inherit some nice properties from the control costs c_t such as convexity and coordinate-wise Lipschitzness, and thus (4.1) can be further decomposed into three parts,

$$\text{D-Reg}(\mathbf{v}_{1:T}) \leq \underbrace{\sum_{t=H}^{T} \widetilde{f}_t(\mathbf{w}_t) - \widetilde{f}_t(\mathbf{v}_t)}_{\text{Unary-Regret}} + \lambda \underbrace{\sum_{t=2}^{T} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2}_{\text{Switching-Cost}} + \lambda \underbrace{\sum_{t=2}^{T} \|\mathbf{v}_t - \mathbf{v}_{t-1}\|_2}_{\text{Path-Length}}, \tag{4.2}$$

where $\lambda = L(H+1)^2/2$, L is the Lipschitz constant of $f_{1:T}$, $\widetilde{f}_t : \mathcal{K} \to \mathbb{R}$ a simplified notation when all decisions are the same, namely, $\widetilde{f}_t(\mathbf{w}) \triangleq f_t(\mathbf{w}, \ldots, \mathbf{w})$. The goal of standard OCO is to only optimize the unary regret, i.e., the first term in (4.2), to catch up with the changing environments. However, the memory issue raised from online decision-making problems imports a unique term called *switching cost*, the variation of the decision sequence $\mathbf{w}_{1:T}$, preventing the decisions from moving too fast. Thus there exists a tradeoff between the unary regret and the switching cost, requiring decisions to move neither too fast nor too slowly. The two parts need to be controlled simultaneously to obtain a small regret.

4.2 Base Algorithm

We first consider the unary regret in (4.2). To deal with the lack of gradient information, we follow the classical zeroth-order FKM method (Flaxman et al., 2005) in standard bandit convex optimization. Specifically, we aim to construct an unbiased gradient estimator $\widetilde{\mathbf{g}}_t = d\widetilde{f}_t(\bar{\mathbf{w}}_t + \delta \mathbf{s}_t)\mathbf{s}_t/\delta$ of $\nabla \bar{f}_t(\bar{\mathbf{w}}_t)$, where $\mathbf{s}_t$ is a random unit vector from a sphere $\mathbb{S} \subseteq \mathbb{R}^d$, $\delta > 0$ is the magnitude of perturbation, $\bar{\mathbf{w}}_t \in (1-\alpha)\mathcal{K}$ is a shrunk decision to ensure the final decision $\mathbf{w}_t = \bar{\mathbf{w}}_t + \delta \mathbf{s}_t \in \mathcal{K}$ and $\bar{f}_t(\bar{\mathbf{w}}_t) \triangleq \mathbb{E}_{\mathbf{s}_t \in \mathbb{S}}[\widetilde{f}_t(\bar{\mathbf{w}}_t + \delta \mathbf{s}_t)]$ is the smoothed version of $\widetilde{f}_t$. The key is to obtain $\widetilde{f}_t(\bar{\mathbf{w}}_t + \delta \mathbf{s}_t)$, which is, however, inaccessible in the control-reduced online learning problem, leading us to the two issues below.

Regardless of the truncation error for a while, we need the value of $\widetilde{f}_t(\mathbf{w}_t)$, requiring the decisions of the last H rounds to be the same, i.e., $\mathbf{w}_{t-H} = \ldots = \mathbf{w}_t$. To achieve this, we artificially divide the whole time horizon into mini-batches, within which the decisions remain the same (Dekel et al., 2012; Cassel and Koren, 2020). Specifically, in each round, we choose a random bit b_t independently from a pre-defined Bernoulli distribution. Only if we choose a 1 with H consecutive 0 before, we estimate the gradient with $f_t(\mathbf{w}_{t-H:t})$. Note that although mini-batching slows down the algorithmic update, it will not increase the regret too much (at most $3H$ times) (Cassel and Koren, 2020, Lemma 11).

Algorithm 1 Base Algorithm

Input: Memory H, dimension d, domain $\mathcal{K}$, shrinkage α, perturbation δ, step size η.

1 Initialize the corresponding variables.

2 Initialize $\mathbf{w}_1, \ldots, \mathbf{w}_H$, any feasible decisions for the first H rounds.

for $t = H+1, \ldots, T$ **do**

3 Submit decision $\mathbf{w}_t$ and receive cost $f_t(\mathbf{w}_{t-H:t}) + \varepsilon_t$.

4 Draw a random bit $b_t \sim \text{Bernoulli}(1/H)$.

if $t \geq H$ and $b_t \prod_{i=1}^{H-1}(1-b_{t-i}) = 1$ **then**

5 Estimate the gradient via (4.3).

6 Update the decision via (4.4).

else

7 Maintain the decision $\mathbf{w}_{t+1} = \mathbf{w}_t$.

end

end

The above statement relies on the accessibility of $f_t(\mathbf{w}_{t-H:t})$, which does not hold here since the learner only receives a control cost $c_t(\mathbf{y}_t, \mathbf{u}_t)$, leading to a *biased* gradient estimator:

$$\widetilde{\mathbf{g}}_t = \frac{d}{\delta} \cdot c_t(\mathbf{y}_t, \mathbf{u}_t) \cdot \mathbf{s}_t = \frac{d}{\delta} \cdot (f_t(\mathbf{w}_{t-H:t}) + \varepsilon_t) \cdot \mathbf{s}_t. \tag{4.3}$$

Fortunately, we find that the bias can be quantified and bounded, stated in the following lemma, whose proof can be found in Appendix B.1.

Lemma 2 (Gradient Estimation Bias)**.** *Letting d, δ be the dimension and perturbation magnitude, $\mathbf{s} \in \mathbb{S}$ be a random unit vector, and $\bar{f}$ be the smoothed version of f, if the gradient is estimated as $\widetilde{\mathbf{g}} = d\,(f(\bar{\mathbf{w}} + \delta\mathbf{s}) + \varepsilon)\,\mathbf{s}/\delta$, then $\|\mathbb{E}[\widetilde{\mathbf{g}}] - \nabla \bar{f}(\bar{\mathbf{w}})\|_2 \leq d\varepsilon/\delta$ holds.*

We now have a solution to optimize the unary regret in (4.2). The switching cost, fortunately, can be bounded directly via gradient descent under static regret (Cassel and Koren, 2020). Thus we can simply run gradient descent with the gradient estimator $\widetilde{\mathbf{g}}_t$ (4.3):

$$\bar{\mathbf{w}}_{t+1} = \Pi_{(1-\alpha)\mathcal{K}}\left[\bar{\mathbf{w}}_t - \eta \widetilde{\mathbf{g}}_t\right]. \tag{4.4}$$

Algorithm 1 concludes the algorithm. The learner first submits a decision and receives an inexact cost value in Line 3. Then it draws a random bit from a pre-defined Bernoulli distribution in Line 4. If the condition after Line 4 is satisfied, the learner conducts gradient descent with the gradient estimator in Line 5 and Line 6. Otherwise, it maintains the previous decision in Line 7. Algorithm 1 enjoys the following theoretical guarantee, and the proof can be found in Appendix B.2.

Theorem 1. *Suppose the domain $\mathcal{K}$ satisfies $r\mathbb{B} \subseteq \mathcal{K} \subseteq R\mathbb{B} \subseteq \mathbb{R}^d$, the loss function f_t is convex, L-coordinate-wise Lipschitz continuous and satisfies $f_t(\cdot) \leq C_f$. Given perturbation magnitude δ and step size η, the expected dynamic regret of Algorithm 1 is at most*

$$\mathbb{E}[\text{D-Reg}(\mathbf{v}_{1:T})] \leq \frac{3H(7R^2 + RP_T)}{4\eta} + \frac{3d^2C_f^2\eta T}{2\delta^2} + \frac{6dR}{\delta} + \frac{\lambda d C_f \eta T}{H\delta} + \frac{L_{\text{eff}}\delta T}{H} + \lambda P_T,$$

where path length $P_T = \sum_{t=2}^{T} \|\mathbf{v}_t - \mathbf{v}_{t-1}\|_2$ *measures the non-stationarity of environments and* $L_{\text{eff}} \triangleq (3L + RL/r + 2\lambda/H)$ *denotes the effective Lipschitz constant.*

Corollary 1. *Under the same assumptions as Theorem 1, setting the perturbation magnitude* $\delta^\star = (3dC_f/L_{\text{eff}})^{1/2}(7H^3R^2/(2T))^{1/4}$ *and step size* $\eta^\star = \delta^\star R/(dC_f)(7H/(2T))^{1/2}$ *gives*

$$\mathbb{E}[\text{S-Reg}_T] = \mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{w}_{t-H:t}) - \min_{\mathbf{v}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{v},\ldots,\mathbf{v})\right] \leq \mathcal{O}(H^{3/4}T^{3/4}).$$

Remark 1. In Theorem 1, if the path length P_T is known in advance, by setting perturbation magnitude $\delta^\star = (3dC_f/L_{\text{eff}})^{1/2}(H^3(7R^2 + RP_T)/(2T))^{1/4}$ and step size $\eta^\star = \delta^\star/(dC_f)(H(7R^2 + RP_T)/(2T))^{1/2}$ optimally, our method achieves $\mathcal{O}(H^{3/4}T^{3/4}(1 + P_T)^{1/4})$ dynamic regret for any comparator sequence $\mathbf{v}_{1:T} \in \mathcal{K}^T$. ¶

Remark 2. In non-stochastic control, the memory length is usually chosen as $H = \Theta(\log T)$, so the result of Corollary 1 matches the $\mathcal{O}(T^{3/4})$ bound given by FKM method (Flaxman et al., 2005) for standard bandit convex optimization, up to poly-logarithmic factors in T. Note that this result is suboptimal compared with the lower bound $\Omega(\sqrt{T})$ (Shamir, 2013). Despite the optimal attainable results by recent breakthroughs (Bubeck et al., 2017; Lattimore, 2020), they are computationally expensive, not efficiently implementable in practice, and generally too involved to analyze the switching cost in our problem. ¶

The parameter configuration in Remark 1 requires prior knowledge of the environmental non-stationarity measure P_T, which is unavailable in applications. To handle this, we design two different scheduling schemes to realize novel meta-base online ensemble structures, based on the weighted combination or optional selection strategies to schedule base learners along with carefully designed regularized surrogate costs to address the memory issue.

4.3 Meta Algorithm I: Weighted Combination

Since the non-stationarity measure P_T is unknown, we guess its possible values, leading to multiple possibly optimal perturbation magnitudes and step sizes. A natural idea is to employ multiple base learners (Algorithm 1) to explore different parameters. Theorem 1 shows that the base learner with the truly optimal parameter configuration (the only right guess) will outperform the others, allowing us to choose the one with the smallest regret.

Note that in our problem, the learner *cannot* maintain multiple perturbation magnitudes since the partial feedback prohibits multiple queries of a cost function. As a result, we fix the perturbation magnitude as $\delta^\star$ and only explore the optimal step size. Specifically, we equip the base learners with different step sizes from a candidate pool $\mathcal{H} = \{\eta_1, \ldots \eta_N\}$, which is a discretization of the theoretically optimal step size $\eta^\star$, using the natural boundary of path length $P_T \in [0, 2RT]$. The specific values of $\delta^\star, \eta^\star, \mathcal{H}$ will be illuminated later.

Again, since the bandit feedback only allows *one* query in each round, simply maintaining multiple base learners and running them parallelly are not permitted. To deal with this, we propose our first scheduling scheme (meta learner), based on the *weighted combination* of base learners' decisions. Specifically, in round t, the learner first combines the i-th base learner's decision $\bar{\mathbf{w}}_{t,i}$ with weight $p_{t,i}$ as $\bar{\mathbf{w}}_t = \sum_{i=1}^{N} p_{t,i}\bar{\mathbf{w}}_{t,i}$, then submits a perturbed decision $\mathbf{w}_t = \bar{\mathbf{w}}_t + \delta^\star \mathbf{s}_t$, receives a cost and finally updates the weight to $p_{t+1,i}$ for all $i \in [N]$. Meanwhile, the i-th base learner updates its local decision to $\bar{\mathbf{w}}_{t+1,i}$ with step size η_i.

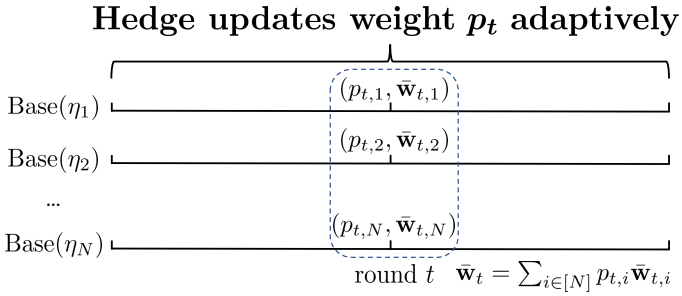


Figure 1: Illustration of our Pronc-Combine algorithm. The overall algorithm follows a two-layer meta-base online ensemble structure. Base(η) denotes the base algorithm with step size η. The base learners update in a parallel way, and the meta learner (Hedge) updates the weight $\boldsymbol{p}_t$ via the surrogate cost defined in (4.5).

The next question comes: how do we update the two-layer structure with only one cost value? In the following, we illuminate this problem with a simple derivation. In round t, we have to optimize the instantaneous regret $\bar{f}_t(\bar{\mathbf{w}}_t) - \bar{f}_t(\bar{\mathbf{v}}_t)$, where $\bar{f}_t$ is the smoothed function of unary loss $\widetilde{f}_t$. By plugging in the definition of the combined decision $\bar{\mathbf{w}}_t$, it holds that

$$\bar{f}_t(\bar{\mathbf{w}}_t) - \bar{f}_t(\bar{\mathbf{v}}_t) \leq \sum_{i=1}^{N} p_{t,i}\bar{f}_t(\bar{\mathbf{w}}_{t,i}) - \bar{f}_t(\bar{\mathbf{v}}_t) \leq \sum_{i=1}^{N} p_{t,i}\langle \nabla \bar{f}_t(\bar{\mathbf{w}}_{t,i}), \bar{\mathbf{w}}_{t,i} - \bar{\mathbf{v}}_t\rangle,$$

where both inequalities come from the convexity of the smoothed function $\bar{f}_t$. The analysis shows that N gradients $\nabla \bar{f}_t(\bar{\mathbf{w}}_{t,1}), \nabla \bar{f}_t(\bar{\mathbf{w}}_{t,2}), \ldots, \nabla \bar{f}_t(\bar{\mathbf{w}}_{t,N})$, i.e., N queries of cost functions, are required to update the base learners, which is prohibited by bandit feedback. To fix this issue, inspired by Zhao et al. (2021a), we start by adopting the convexity of $\bar{f}_t$ at the very first step of the analysis:

$$\bar{f}_t(\bar{\mathbf{w}}_t) - \bar{f}_t(\bar{\mathbf{v}}_t) \leq \langle \nabla \bar{f}_t(\bar{\mathbf{w}}_t), \bar{\mathbf{w}}_t - \bar{\mathbf{v}}_t\rangle = \sum_{i=1}^{N} p_{t,i}\langle \widetilde{\mathbf{g}}_t, \bar{\mathbf{w}}_{t,i} - \bar{\mathbf{v}}_t\rangle + \langle \nabla \bar{f}_t(\bar{\mathbf{w}}_t) - \widetilde{\mathbf{g}}_t, \bar{\mathbf{w}}_t - \bar{\mathbf{v}}_t\rangle,$$

where the second term above is caused by the biased gradient estimator and can be controlled via Lemma 2. Accordingly, by importing a surrogate loss $\ell_t(\cdot) \triangleq \langle \widetilde{\mathbf{g}}_t, \cdot\rangle$, the first term above can be rewritten as $\sum_{i=1}^{N} p_{t,i}(\ell_t(\bar{\mathbf{w}}_{t,i}) - \ell_t(\bar{\mathbf{v}}_t))$, where the gradients of all base learners are the same as $\nabla \ell_t(\bar{\mathbf{w}}_{t,1}) = \ldots = \nabla \ell_t(\bar{\mathbf{w}}_{t,N}) = \widetilde{\mathbf{g}}_t$, allowing them to run gradient descent with the same gradient. It is worthy noting that although the adversary becomes completely adaptive after this operation, because the estimated gradient $\widetilde{\mathbf{g}}_t$ relies on the decision of the current round, we can utilize deterministic full-information algorithms to optimize the first term above. Regarding the second term, which represents the gap between the true gradient $\nabla \bar{f}_t(\bar{\mathbf{w}}_t)$ and the estimated gradient $\widetilde{\mathbf{g}}_t$, taking expectation over it is sufficient due to Lemma 2.

As mentioned at the end of Section 4.1, how to balance the unary regret and the switching cost is the key issue when analyzing the dynamic regret of OCO with memory. We give a simple analysis to show that just ignoring the switching cost would lead to linear regret in the time horizon T. Formally, by definition of the meta-decision $\bar{\mathbf{w}}_t = \sum_{i=1}^{N} p_{t,i}\bar{\mathbf{w}}_{t,i}$, a

one-step switching cost $\|\bar{\mathbf{w}}_t - \bar{\mathbf{w}}_{t-1}\|_2$ can be bounded by

$$\|\bar{\mathbf{w}}_t - \bar{\mathbf{w}}_{t-1}\|_2 \leq R\,\|\boldsymbol{p}_t - \boldsymbol{p}_{t-1}\|_1 + \sum_{i=1}^{N} p_{t,i}\,\|\bar{\mathbf{w}}_{t,i} - \bar{\mathbf{w}}_{t-1,i}\|_2\,,$$

where $\boldsymbol{p}_t \triangleq (p_{t,1}, \ldots, p_{t,N})$. A more detailed derivation is deferred in (B.7). The switching cost of the base learner with the maximum step size satisfies $\|\bar{\mathbf{w}}_{t,N} - \bar{\mathbf{w}}_{t-1,N}\|_2 \leq \mathcal{O}(\eta_N)$. Considering the optimal step size $\eta^\star$ in Remark 1, since the maximum path length could be $P_T = \mathcal{O}(T)$, the maximum step size $\eta_N = \mathcal{O}(1)$. Summing over time horizon T makes the overall regret $\mathcal{O}(T)$, which is unacceptable. To address the memory issue, inspired by the study in the full-information setup (Zhao et al., 2022b), we add a *regularization term* in the surrogate cost of each base learner to punish the one with large switching cost,

$$\ell_{t,i}(\bar{\mathbf{w}}_{t,i}) \triangleq 3H\langle \widetilde{\mathbf{g}}_t, \bar{\mathbf{w}}_{t,i}\rangle + \lambda\,\|\bar{\mathbf{w}}_{t,i} - \bar{\mathbf{w}}_{t-1,i}\|_2\,. \tag{4.5}$$

Later, the overall regret can be decomposed into *meta-regret* and *base-regret*, incurred by the meta/base learner, respectively (see Appendix B.3 for details of the regret decomposition),

$$\begin{aligned}
\textsc{Meta-Reg} &= \sum_{t\in S}(\langle \boldsymbol{p}_t, \boldsymbol{\ell}_t\rangle - \ell_{t,i}) + \lambda R \sum_{t\in S}\|\boldsymbol{p}_t - \boldsymbol{p}_{t-1}\|_1\,,\\
\textsc{Base-Reg} &= 3H\sum_{t\in S}\langle \widetilde{\mathbf{g}}_t, \bar{\mathbf{w}}_{t,i} - \bar{\mathbf{v}}_t\rangle + \lambda\sum_{t\in S}\|\bar{\mathbf{w}}_{t,i} - \bar{\mathbf{w}}_{t-1,i}\|_2\,,
\end{aligned}$$

where $\boldsymbol{\ell}_t \triangleq (\ell_{t,1}(\bar{\mathbf{w}}_{t,1}), \ldots, \ell_{t,N}(\bar{\mathbf{w}}_{t,N}))$ denotes the surrogate loss vector and S is the set of mini-batches. Notice that the meta-regret is the static regret of a Prediction with Expert Advice problem (Cesa-Bianchi and Lugosi, 2006), incorporated with the switching cost. Thus, we can use Hedge (Freund and Schapire, 1997), a basic expert-tracking method, as the meta learner, with the following update rule:

$$p_{t+1,i} \propto p_{t,i}\exp(-\eta_{\text{meta}}\ell_{t,i}(\bar{\mathbf{w}}_{t,i})), \tag{4.6}$$

where η_{meta} denotes the learning rate of the meta algorithm. All base learners run gradient descent (4.4) with the gradient estimator (4.3) and the step sizes from $\mathcal{H} = \{\eta_1, \ldots \eta_N\}$.

Combining all the ingredients, Algorithm 2 concludes our first online ensemble algorithm, named Pronc-Combine. The learner broadcasts the gradient estimator to all base learners for updates and constructs surrogate losses as the input of the meta learner in Line 6 and Line 7. The meta learner updates in Line 8. The learner gives the decision of the next round in Line 9 and Line 10. Besides, we note that $\bar{\mathbf{w}}_t$ is the weighted combination of the base decisions. It is only used to generate the perturbed final decision $\mathbf{w}_t$ and can be abandoned afterward. The i-th base learner only needs to maintain its own local decision $\bar{\mathbf{w}}_{t,i}$. The following theorem shows the overall dynamic regret bound of our proposed Pronc-Combine algorithm. The proof can be found in Appendix B.3.

Theorem 2. *Under the same assumptions of Theorem 1, define the perturbation magnitude $\delta^\star$, the step size $\eta^\star$ as*

$$\delta^\star = \sqrt{\frac{3dC_f}{L_{\text{eff}}}}\left(\frac{7HR^2}{T}\right)^{1/4}, \quad \eta^\star = \frac{\delta^\star}{dC_f}\sqrt{\frac{H(7R^2 + RP_T)}{T}}, \tag{4.7}$$

Algorithm 2 PRONC-COMBINE

Input: Memory length H, dimension d, domain $\mathcal{K}$, shrinkage coefficient α, perturbation magnitude $\delta^\star$, step size pool $\mathcal{H} = \{\eta_1, \ldots, \eta_N\}$, learning rate of meta learner η_{meta}.

Initialize the corresponding variables.
Initialize $\mathbf{w}_1, \ldots, \mathbf{w}_H$, any feasible decisions for the first H rounds.
for $t = H+1, \ldots, T$ **do**
  Submit decision $\mathbf{w}_t$ and receive cost $f_t(\mathbf{w}_{t-H:t}) + \varepsilon_t$.
  Draw a random bit $b_t \sim \text{Bernoulli}(1/H)$.
  if $t \geq H$ and $b_t \prod_{i=1}^{H-1}(1-b_{t-i}) = 1$ **then**
    Estimate gradient via (4.3) with perturbation magnitude $\delta^\star$.
    for $i = 1, \ldots, N$ **do**
      Base-learner updates via (4.4).
      Construct the surrogate loss as (4.5).
    end
    Meta-learner updates its weights via (4.6).
    Obtain weighted combined decision $\bar{\mathbf{w}}_{t+1} = \sum_{i=1}^N p_{t+1,i} \bar{\mathbf{w}}_{t+1,i}$.
    Draw random unit vector $\mathbf{s}_{t+1} \in \mathbb{S}$ and perturb $\bar{\mathbf{w}}_{t+1}$ to $\mathbf{w}_{t+1} = \bar{\mathbf{w}}_{t+1} + \delta^\star \mathbf{s}_{t+1}$.
  else
    Maintain $p_{t+1,i} = p_{t,i}, \bar{\mathbf{w}}_{t+1,i} = \bar{\mathbf{w}}_{t,i}, \mathbf{w}_{t+1} = \mathbf{w}_t$ for $i \in [N]$.
  end
end

and the corresponding step size pool $\mathcal{H}$ *as*

$$\mathcal{H} = \left\{ \eta_i \,\middle|\, \eta_i = 2^{i-1} \frac{\delta^\star R}{dC_f} \sqrt{\frac{7H}{T}}, i \in [N] \right\}, \tag{4.8}$$

where $N = \lceil \log_2 (1 + 2T/7) / 2 \rceil + 1$ *denotes the number of candidate step sizes. Our* PRONC-COMBINE *(Algorithm 2) ensures that for any comparator sequence* $\mathbf{v}_{1:T} \in \mathcal{K}^T$*,*

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=H}^T f_t(\mathbf{w}_{t-H:t}) - \sum_{t=H}^T f_t(\mathbf{v}_{t-H:t}) \right] \\
\leq\ & \frac{20\lambda^{3/2} d C_f R^{3/2}}{\delta^\star} \sqrt{\frac{T}{H}} \ln\left(\frac{1}{2} \log\left(1 + \frac{P_T}{7R} \right) + 2 \right) + \frac{3H(7R^2 + RP_T)}{2\eta^\star} \\
&\quad + \frac{3d^2 C_f^2 \eta^\star T}{2{\delta^\star}^2} + \frac{6\varepsilon d R T}{\delta^\star} + \frac{\lambda d C_f \eta^\star T}{H\delta^\star} + L_{\text{eff}} \delta^\star T + \lambda P_T \\
=\ & \mathcal{O}(\min\{T^{3/4}(1+P_T)^{1/2}, T\}).
\end{aligned}$$

Remark 3. Our result recovers the dynamic regret of standard bandit convex optimization without memory (Zhao et al., 2021a) by setting the memory length $H = 1$. When environments change severely (path length $P_T \geq \Omega(\sqrt{T})$), the above result becomes vacuous, which is left as an important future direction to investigate. Indeed, even for the standard bandit convex optimization, or the simpler multi-armed bandits (Auer et al., 2002), obtaining the optimal dynamic regret for large non-stationarity remains open (Foster et al., 2020). ¶

Remark 4. We finally discuss the relationship and difference between our work and the prior study (Zhao et al., 2022b) in terms of the techniques. Our Pronc-Combine is mainly a combination of Zhao et al. (2021a) and Zhao et al. (2022b), with a careful adaptation to the partial-feedback non-stochastic control. Specifically, our algorithm additionally necessitates the design of linearized surrogate costs (4.5) to broadcast the gradient estimator to all the base learners, making the parallel update feasible in the online ensemble structure under the limited bandit feedback. Moreover, this linearization operation will introduce a bias term related to the difference between the expectation of the gradient estimator and the true gradient, which is unique in the control-based bandit convex optimization problem and requires more involved analyses, especially in unknown systems. Furthermore, in Section 4.4, we present another online ensemble algorithm called Pronc-Select that admits serial updates, which is fundamentally different from the fashion of the parallel update in Pronc-Combine and thus evidently differs from the prior work (Zhao et al., 2022b). The new update borrows the idea of the Bandits-over-Bandits mechanism from the literature of non-stationary linear bandits (Cheung et al., 2019) but requires additional technical modifications. As far as we know, this is the *first* work to adapt those techniques in the non-stochastic control, substantially improving the result of Pronc-Combine and even the best known dynamic regret of bandit convex optimization in most cases. ¶

4.4 Meta Algorithm II: Optional Selection

In this section, we propose another scheduling scheme based on the *optional selection* strategy. Briefly, the meta learner does not consider all base learners simultaneously but chooses only one for prediction and update. Concretely, our method adopts an ensemble mechanism called *Bandits-over-Bandits (BOB)*, originally proposed by Cheung et al. (2019) for designing parameter-free algorithms in non-stationary stochastic linear bandits. To the best of our knowledge, our work is the *first* to leverage this technique in (adversarial) bandit convex optimization. Moreover, as will be shown, this structure indeed helps in giving a better dynamic regret bound than the best known result in certain conditions.

In the following, we describe the BOB mechanism in detail. Generally, it is an ensemble structure with multiple base learners exploring the parameter space and a meta learner tracking the optimal parameter adaptively. In contrast to the parallel update strategy (i.e., Pronc-Combine in Section 4.3), BOB updates the base learners *serially*. Concretely, BOB performs in episodes. In each episode, only *one* base learner is chosen according to some selection criteria and returns its cumulative loss to the meta learner when the episode ends.

The serial update way brings unique benefits to the bandit convex optimization problem. Recall that in Pronc-Combine, we use a fixed perturbation magnitude $\delta^\star$ in (4.7) since the bandit feedback prohibits multiple queries within a single round. While in BOB, different perturbation magnitudes can be adopted in different episodes since only one base learner is active in each episode. In other words, the BOB mechanism allows *decoupled* base learners to explore both the perturbation magnitude space and the step size space, which intuitively leads to a better dynamic regret guarantee.

Now it is time to design the meta learner. A natural idea is to run a Multi-Armed Bandit (MAB) algorithm, such as Exp3 (Auer et al., 2002), by treating base learners as arms. By the no-regret guarantee of MAB algorithms, which states that, on average the

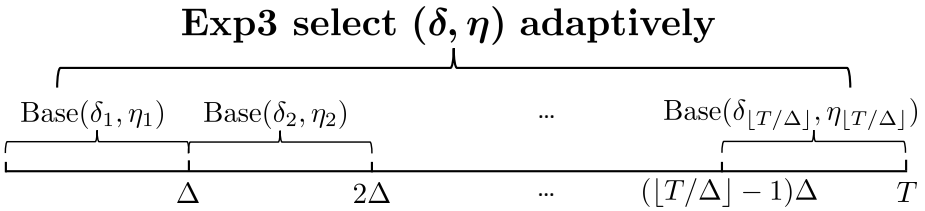


Figure 2: Illustration of our PRONC-SELECT algorithm. The overall algorithm follows a two-layer meta-base online ensemble structure. Base(δ, η) denotes the base algorithm with perturbation magnitude δ and step size η. The base learners update in a serial way, and the meta learner (Exp3) updates the weight $\boldsymbol{p}_t$ via the surrogate cost defined in (4.9).

performance of the algorithm is almost as good as that of the best arm, it seems that our problem is perfectly solved. However, as pointed out by Agarwal et al. (2017), this reasoning is flawed as the base learners are not static arms. In a valid adversarial MAB problem, two algorithms should receive the same loss if they select the same arm, no matter which arms they chose before. While in BOB, the performance of an arm is affected by previous choices. In other words, if previously selected arms have already achieved a small enough regret, even if the current arm is not good, it will still perform well. To fix this, the BOB mechanism *restarts* the decision at the beginning of each episode to erase the effect of previous choices.

Using the BOB mechanism directly as a black box is infeasible due to the existence of the memory, i.e., switching costs. To handle this, we design a novel surrogate loss for the meta learner to enforce low switching costs explicitly. Specifically, in episode i, denote by (δ_i, η_i) the parameter (arm) that Exp3 chooses, Δ_i the time steps in episode i, and $\{w_t(\delta_i, \eta_i)\}_{t \in \Delta_i}$ the decision sequence produced by the arm, we define the surrogate loss of arm (δ_i, η_i) as

$$\ell_i(\delta_i, \eta_i) \triangleq 3H \sum_{t \in \Delta_i} \widetilde{f}_t(\mathbf{w}_t(\delta_i, \eta_i)) + \lambda \sum_{t \in \Delta_i} \|\mathbf{w}_t(\delta_i, \eta_i) - \mathbf{w}_{t-1}(\delta_i, \eta_i)\|_2 , \tag{4.9}$$

which is the cumulative unary loss along with the switching cost in episode i. The last term is a regularization term responsible for punishing the base learner with large switching costs. With such a novel surrogate loss, the overall regret of all episodes $\{\Delta_i\}_{i=1}^{\lceil |S|/\Delta \rceil}$ can be decomposed into the *meta-regret* and *base-regret*, incurred by the meta learner and base learners, respectively (see Appendix B.4 for details of the regret decomposition),

$$\textsc{Meta-Reg} = \sum_{i=1}^{\lceil |S|/\Delta \rceil} \ell_i(\delta_i, \eta_i) - \sum_{i=1}^{\lceil |S|/\Delta \rceil} \ell_i(\delta^\star, \eta^\star),$$

$$\textsc{Base-Reg} = \sum_{i=1}^{\lceil |S|/\Delta \rceil} \sum_{t \in \Delta_i} 3H(\widetilde{f}_t(\mathbf{w}_t(\delta^\star, \eta^\star)) - \widetilde{f}_t(\mathbf{v}_t)) + \lambda \|\mathbf{w}_t(\delta^\star, \eta^\star) - \mathbf{w}_{t-1}(\delta^\star, \eta^\star)\|_2 .$$

It is noteworthy to point out that the base-regret is again the dynamic regret over the unary losses along with the switching cost, which our base algorithm can optimize with sound guarantees (Theorem 1), and the meta-regret is the static regret of an adversarial MAB problem, which can be optimized by Exp3 (Auer et al., 2002). For completeness, we

restate the algorithmic details of Exp3 here, whose main idea is constructing an unbiased loss estimator and passing it to Hedge (4.6). First, define an unbiased loss vector via the importance weighting to recover the losses of all arms,

$$\widehat{\ell}_i(\delta,\eta) \triangleq \frac{\ell_i(\delta,\eta)}{p_i(\delta,\eta)}\mathbf{1}\{(\delta,\eta)=(\delta_i,\eta_i)\}, \tag{4.10}$$

where $p_i(\delta,\eta)$ denotes the probability of choosing arm (δ,η), such that the loss of each arm is unbiased as $\mathbb{E}[\widehat{\ell}_i(\delta,\eta)] = (1-p_i(\delta,\eta))\times 0 + p_i(\delta,\eta)\times\frac{\ell_i(\delta,\eta)}{p_i(\delta,\eta)} = \ell_i(\delta,\eta)$. Thus, the problem is reduced to the full information setting, where the weights can be updated via the multiplicative rule of Hedge:

$$p_{i+1}(\delta,\eta) \propto p_i(\delta,\eta)\exp(-\eta_{\text{meta}}\widehat{\ell}_i(\delta,\eta)), \tag{4.11}$$

where η_{meta} is the learning rate of the meta learner.

Equipping BOB with Exp3 and the switching-cost-regularized surrogate loss (4.9), along with the base learners, forms our second algorithm, named Pronc-Select, whose details are concluded in Algorithm 3. Inside an episode, a base learner updates via gradient descent with the gradient estimator and the chosen parameters (δ_i,η_i) in Line 5 and Line 6. When one episode ends, the meta learner receives the cumulative surrogate loss, constructs an unbiased loss estimator, updates its weights, and chooses the arm of the next episode in Lines 7-10. Finally, the mechanism restarts the decision randomly in Line 11.

At last, we consider the parameter configuration of base learners. We discretize the optimal step size $\eta^\star$ and perturbation magnitude $\delta^\star$ into candidate step size pool $\mathcal{H}_\eta = \{\eta_1,\dots,\eta_{N_\eta}\}$ and candidate perturbation magnitude pool $\mathcal{H}_\delta = \{\delta_1,\dots,\delta_{N_\delta}\}$ using the natural boundary of $P_T \in [0,2RT]$ such that $N_\eta, N_\delta \approx \log T$. The specific values of $\eta^\star$ and $\delta^\star$ will be illuminated later. A natural idea is to group the possible values of the two variables freely, resulting in $N_\delta \times N_\eta \approx (\log T)^2$ base learners, which is not as time-efficient as Pronc-Combine, since the latter only has $\mathcal{O}(\log T)$ base learners, as shown in (4.8). Fortunately, we discover that the number of base learners can be reduced due to the inner connection between the step size and the perturbation magnitude. Concretely speaking, Theorem 1 shows that given certain perturbation magnitude δ_i and step size η_j, the dynamic regret of the base algorithm is related to $\frac{3H(7R^2+RP_T)}{4\eta_j} + \frac{3\eta_j d^2 C_f^2 T}{2\delta_i^2} + L_{\text{eff}}\delta_i T$, telling that the optimal perturbation magnitude and step size satisfy $\delta_i = (3d^2C_f^2\eta_j/L_{\text{eff}})^{1/3}$. As a result, we bind each step size with a certain perturbation magnitude as above, reducing the number of base learners from $(\log T)^2$ to $\log T$. The following theorem shows the theoretical guarantee of our proposed Pronc-Select, and its proof can be found in Appendix B.4.

Theorem 3. *Under the same assumptions of Theorem 1, define the perturbation magnitude $\delta^\star$, the step size $\eta^\star$, the restarting period Δ as*

$$\eta^\star = \sqrt{\frac{3}{L_{\text{eff}}dC_f}}\left(\frac{H(7R^2T^{\frac{1}{3}}+RP_T)}{2T}\right)^{3/4}, \quad \delta^\star = \left(\frac{3d^2C_f^2\eta^\star}{L_{\text{eff}}}\right)^{1/3}, \quad \Delta = T^{2/3}, \tag{4.12}$$

and the corresponding parameter pool $\mathcal{H}$ as

$$\mathcal{H} = \left\{(\eta_i,\delta_i)\,\middle|\,\eta_i = \sqrt{\frac{3R^3}{L_{\text{eff}}dC_fT}}\left(\frac{7H}{2}\right)^{3/4}, \delta_i = \left(\frac{3d^2C_f^2\eta_i}{L_{\text{eff}}}\right)^{1/3}, i\in[N]\right\}, \tag{4.13}$$

Algorithm 3 PRONC-SELECT

Input: Memory length H, dimension d, domain $\mathcal{K}$, parameter pool $\mathcal{H}$ (4.13), learning rate of meta learner η_{meta}.

Initialize the corresponding variables.
Initialize $\mathbf{w}_1, \ldots, \mathbf{w}_H$, any feasible decisions for the first H rounds.
for $t = H+1, \ldots, T$ **do**
  Submit decision $\mathbf{w}_t$ and receive cost $f_t(\mathbf{w}_{t-H:t}) + \varepsilon_t$.
  Draw a random bit $b_t \sim \text{Bernoulli}(1/H)$.
  if $t \geq H$ and $b_t \prod_{i=1}^{H-1}(1-b_{t-i}) = 1$ **then**
    if inside episode i **then**
      Estimate the gradient via (4.3) with perturbation magnitude δ_i.
      Base-learner updates its decision to $\bar{\mathbf{w}}_{t+1}$ via (4.4).
    else
      Receive surrogate loss $\ell_i(\delta_i, \eta_i)$ via (4.9).
      Construct unbiased loss vector $\widehat{\ell}_i$ via (4.10).
      Meta-learner updates its weights via (4.11).
      Choose the arm of the next episode as $(\delta_{i+1}, \eta_{i+1}) \sim p_{i+1}$.
      Restart decision $\bar{\mathbf{w}}_{t+1} \in (1-\alpha)\mathcal{K}$ randomly.
    end
    Draw random unit vector $\mathbf{s}_{t+1} \in \mathbb{S}$ and perturb $\bar{\mathbf{w}}_{t+1}$ to $\mathbf{w}_{t+1}$.
  else
    Maintain $p_{t+1,i} = p_{t,i}, \bar{\mathbf{w}}_{t+1,i} = \bar{\mathbf{w}}_{t,i}, \mathbf{w}_{t+1} = \mathbf{w}_t$ for $i \in [N]$.
  end
end

where $N = \lceil 3\log_2((1+2T^{2/3})/7)/4 \rceil + 1$ *denotes the number of candidate parameters. Our* PRONC-SELECT *(Algorithm 3) ensures that for any comparator sequence* $\mathbf{v}_{1:T} \in \mathcal{K}^T$,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=H}^{T} f_t(\mathbf{w}_{t-H:t}) - \sum_{t=H}^{T} f_t(\mathbf{v}_{t-H:t})\right] \\
&\leq 2(C_f + 2\lambda R)\sqrt{\frac{T\Delta \log T \log\log T}{H}} + \frac{\frac{21R^2T}{\Delta} + 3HRP_T}{4\eta^\star} + \frac{3d^2C_f^2\eta^\star T}{2{\delta^\star}^2} \\
&\quad + \frac{6dR}{\delta^\star} + \frac{\lambda d C_f \eta^\star T}{H\delta^\star} + L_{\text{eff}}\delta^\star T + \frac{T}{H\Delta} + \lambda P_T \\
&= \mathcal{O}(T^{3/4}(1+P_T)^{1/4} + T^{5/6}(\log T)^{1/2}).
\end{aligned}$$

Comparing Theorem 2 (PRONC-COMBINE) with Theorem 3 (PRONC-SELECT), we find that the two algorithms are preferred in different situations:

- PRONC-COMBINE is better in mildly changing environments, specifically, path length $P_T \leq \mathcal{O}(T^{1/6})$. Consider a completely stationary environment, two algorithms attain $\mathcal{O}(T^{3/4})$ and $\widetilde{\mathcal{O}}(T^{5/6})$ static regret, respectively. The performance of PRONC-SELECT is not ideal since it has to restart repeatedly, which is unnecessary in stationary environments and thus leads to a degeneration in regret bound.

Algorithm 4 PRONC (for PaRtial feedback Online Non-stochastic Control)

Input: Memory H, dimension d, domain $\mathcal{M}$, Markov operator G.

1 Initialize corresponding variables.

2 Initialize $\mathbf{u}_1, \dots, \mathbf{u}_H$, any feasible output control signals for the first H rounds.

for $t = H+1, \dots, T$ **do**

3 Observe $\mathbf{y}_t$.

4 Compute Nature's $\mathbf{y}_t^{\text{nat}} = \mathbf{y}_t - \sum_{i=1}^{H} G^{[i]} \mathbf{u}_{t-i}$.

5 Submit action $\mathbf{u}_t(M_t) = \sum_{i=0}^{m-1} M_t^{[i]} \mathbf{y}_{t-i}^{\text{nat}}$ and receive a cost $c_t(\mathbf{y}_t, \mathbf{u}_t)$.

6 Update the policy parameter to M_{t+1} with PRONC-COMBINE (Algorithm 2) or PRONC-SELECT (Algorithm 3).

end

- PRONC-SELECT is preferred in environments with large non-stationarity (path length $P_T \geq \Omega(T^{1/6})$) because intuitively it can just switch to the best base learner but not has to take care of all of them simultaneously as PRONC-COMBINE does.

We finally remark that Theorem 3 also implies a new dynamic regret for standard bandit convex optimization by setting the memory length as $H = 1$. Specifically, the result gives an $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/4} + T^{5/6})$ dynamic regret, improving upon the best known result of $\mathcal{O}(T^{3/4}(1+P_T)^{1/2})$ (Zhao et al., 2021a), whenever the environmental non-stationarity measure P_T is large. In particular, when $P_T \geq \Omega(T^{1/6})$, our result is tighter than the existing $\mathcal{O}(T^{3/4}(1+P_T)^{1/2})$ regret. Furthermore, when $P_T \geq \Omega(T^{1/3})$, our result is essentially $\mathcal{O}(T^{3/4}(1+P_T)^{1/4})$, which is as good as Theorem 1 but does not require the path length P_T as the algorithm input. This byproduct might be of independent interest.

4.5 Back to Online Non-stochastic Control

In Section 4.3 and Section 4.4, we design two online ensemble algorithms for bandit convex optimization with memory and inexact feedback. In this part, we apply these results back to online non-stochastic control. For ease of understanding, we make some common notations at the front of this part: denote by $\pi_{1:T} \in \Pi_{\text{DRP}}$ a sequence of time-varying DRP policies, parameterized via $M_{1:T}$ and $P_T \triangleq \sum_{t=2}^{T} \|M_t - M_{t-1}\|_{\text{F}}$ the corresponding path length.

Since the reduction from non-stochastic control to online learning is already solved, applying the results back is straightforward, concluded in Algorithm 4. In each round, the learner first gets a partial observation of the state in Line 3, computes the Nature's y in Line 4, submits a DRP action, and receives the corresponding cost in Line 5. After that, the learner feeds the cost and other parameters into PRONC-COMBINE (Algorithm 2) or PRONC-SELECT (Algorithm 3) for policy update in Line 6. Theorem 4 presents the theoretical guarantees of our method for online non-stochastic control with partial feedback. The proof is deferred to Appendix B.5.

Theorem 4. *Under Assumptions 1 and 2, with memory length $H = \Theta(\log T)$,*

- *Our weighted combination method* PRONC-COMBINE *ensures*

$$\mathbb{E}\left[\sum_{t=1}^{T} c_t(\mathbf{y}_t, \mathbf{u}_t) - \sum_{t=1}^{T} c_t(\mathbf{y}_t^{\pi_t}, \mathbf{u}_t^{\pi_t})\right] \leq \widetilde{\mathcal{O}}(\min\{T^{3/4}(1+P_T)^{1/2}, T\});$$

Algorithm 5 Unknown System Estimation

Input: Estimation rounds T_0, memory length H.

for $t = 1, \dots, T_0$ **do**

1 Submit random action $\mathbf{u}_t = \mathcal{N}(\mathbf{0}, I^{d_u \times d_u})$.

end

2 Obtain estimated Markov operator $\widehat{G}$ via (4.14) and send it to Algorithm 4.

- *With restarting period $\Delta = \mathcal{O}(T^{2/3})$, our optional selection* PRONC-SELECT *ensures*

$$\mathbb{E}\left[\sum_{t=1}^{T} c_t(\mathbf{y}_t, \mathbf{u}_t) - \sum_{t=1}^{T} c_t(\mathbf{y}_t^{\pi_t}, \mathbf{u}_t^{\pi_t})\right] \leq \widetilde{\mathcal{O}}(T^{3/4}(1 + P_T)^{1/4} + T^{5/6}).$$

Remark 5. Similar to the discussions in Section 4.4, PRONC-COMBINE is better in slowly changing environments, while PRONC-SELECT outperforms facing large non-stationarity. Besides, in completely stationary environments (path length $P_T = 0$), our results degenerate to $\widetilde{\mathcal{O}}(T^{3/4})$ and $\widetilde{\mathcal{O}}(T^{5/6})$ respectively, between which, the $\widetilde{\mathcal{O}}(T^{3/4})$ static regret for online non-stochastic control with partial feedback is also new. It is equal to the regret bound of fully observed bandit linear control problem (Gradu et al., 2020; Cassel and Koren, 2020) and can recover the result with full gradient information (Simchowitz et al., 2020). ¶

Extension to Unknown Systems. Besides known systems, we further extend our results to unknown ones, where the system transition matrices A, B, C in (3.1) are unknown. This problem is usually encountered in real applications and thus motivates the related methods with strong theoretical guarantees. Similar to many model-based reinforcement learning methods, we follow the *explore-then-exploit* paradigm. Briefly speaking, we adopt the way of Simchowitz et al. (2020), who use randomly generated actions to perturb the system and estimate the Markov operator G (see Definition 2) via least mean square:

$$\widehat{G} = \operatorname*{arg\,min}_{G \in (\mathbb{R}^{d_y \times d_u})^H} \sum_{t=H+1}^{T_0} \left\| \mathbf{y}_t - \sum_{i=1}^{H} G^{[i]} \mathbf{u}_{t-i} \right\|_2^2, \tag{4.14}$$

where T_0 is the estimation rounds and $\mathbf{u}_{1:T_0}$ are random actions. Finally, we feed the estimated Markov operator $\widehat{G}$ into Algorithm 4 and obtain Algorithm 5 for unknown systems.

In known systems, the truncation error is ignorable (see Lemma 1). However, the issue becomes harder in unknown systems due to the connection between the truncation error and the quality of the gradient estimator. Specifically, the truncated cost f_t is a function of DRPs $M_{t-H:t}$ parameterized by Markov operator G and Nature's $\mathbf{y}^{\text{nat}}_{t-H-m+1:t}$ (see Definition 3). Thus in unknown systems, the truncated cost is a function with the the estimated Markov operator and Nature's ys as parameters, i.e., $f_t(\cdot|\widehat{G}, \widehat{\mathbf{y}}^{\text{nat}}_{t-H-m+1:t})$. It imports more bias in the gradient estimator, which can further propagate in the online ensemble structure. With careful analysis, we establish Theorem 5 to show that our method can handle unknown systems with sound theoretical guarantees. The proof can be found in Appendix B.6.

Theorem 5. *Under Assumptions 1 and 2, for sufficiently large T, with memory length $H = \Theta(\log T)$ and estimation rounds $T_0 = \mathcal{O}(T^{4/5})$,*

- *Our weighted combination* Pronc-Combine *ensures that*

$$\mathbb{E}\left[\text{D-Reg}(\pi_{1:T})\right] \leq \widetilde{\mathcal{O}}(\min\{T^{4/5}(1+P_T)^{1/2}, T\});$$

- *With restarting period* $\Delta = \mathcal{O}(T^{2/3})$*, our optional selection* Pronc-Select *ensures*

$$\mathbb{E}\left[\text{D-Reg}(\pi_{1:T})\right] \leq \widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/4} + T^{5/6}).$$

Remark 6. The two algorithms are also preferred in different cases. When $P_T \leq \mathcal{O}(T^{1/15})$, Pronc-Combine is better, when $P_T \geq \Omega(T^{1/15})$, Pronc-Select outperforms the other, and they are comparable when $P_T = \Theta(T^{1/15})$. ¶

Remark 7. The result of Pronc-Combine is not as good as that in Theorem 4 due to the tradeoff between the estimation and truncation error. Similar dilemmas appear even in the full and semi-partial feedback setting: (1) $\widetilde{\mathcal{O}}(\sqrt{T}) \to \widetilde{\mathcal{O}}(T^{2/3})$ emerges in the static regret analysis under full feedback (Hazan et al., 2020); (2) $\widetilde{\mathcal{O}}(\text{poly}(\log T)) \to \widetilde{\mathcal{O}}(\sqrt{T})$ appears in partially observed systems with strongly convex, smooth losses and semi-non-stochastic disturbances (Simchowitz et al., 2020). There seems to be no price of system estimation in Pronc-Select because the degeneration is still dominated by the $\widetilde{\mathcal{O}}(T^{5/6})$ term caused by the BOB mechanism. ¶

5 Experiment

In this section, we validate the performance of our proposed method in synthetic linear and simulated nonlinear environments, aiming to answer the following three questions as well:

- whether online non-stochastic control can handle model misspecifications such as nonlinearity and non-stationarity;
- whether the meta-base aggregation helps in unknown non-stationary environments;
- whether the switching cost regularizer can deal with the memory issue raised in decision-making problems;

Contenders and Configurations. Since this problem is newly introduced, there are no existing methods to compete with. As a result, we design some baselines and skylines to verify the effectiveness of certain components. Concretely, we compare our algorithms, denoted by *PRONC.Combine* and *PRONC.Select*, with *two baselines*: (a) *BGD.Control* is mainly built on the work of Cassel and Koren (2020), which considers the static regret of bandit linear control and runs a simple bandit gradient descent algorithm; (b) *PBGD.Control* updates its policy using PBGD (Zhao et al., 2021a), an online ensemble method for bandit convex optimization without switching cost regularizers. We equip the baselines with disturbance-response policy to make them capable of dealing with partially observed states. Furthermore, we adopt *four skylines* that receive full information to validate our method's capability for partial feedback. Specifically, (c) *Grad.Combine* and *Grad.Select* have full information of the cost functions, served to measure the quality of the gradient estimator; (d) *Known.Combine* and *Known.Select* have access to the true system transition dynamics, regarded as a skyline for the system estimation procedure. Note that the skylines are

actually *infeasible* in the partial feedback setting and are only employed to illustrate how well our method can perform.

We report the average results with standard deviations of 5 independent runs to obtain convincing results. Only the randomness of the perturbation is preserved. All hyper-parameters are set to be theoretically optimal except the learning rate of the meta learners, which are scaled by constants to speed up the learning process. To make *PRONC.Select* more stable, we use a variant of the traditional Exp3 algorithm, called Exp3-IX (Neu, 2015), which enjoys the same regret guarantee as Exp3, but holds with high probability.

The rest of this section is organized as follows. We investigate the performance and robustness of our method in a synthetic time-varying linear dynamical system in Section 5.1 and three simulated nonlinear reinforcement learning tasks: pendulum, cartpole, and data center cooling in Section 5.2. Section 5.3 reports the time efficiency of our method.

5.1 Synthetic Time-Varying LDS

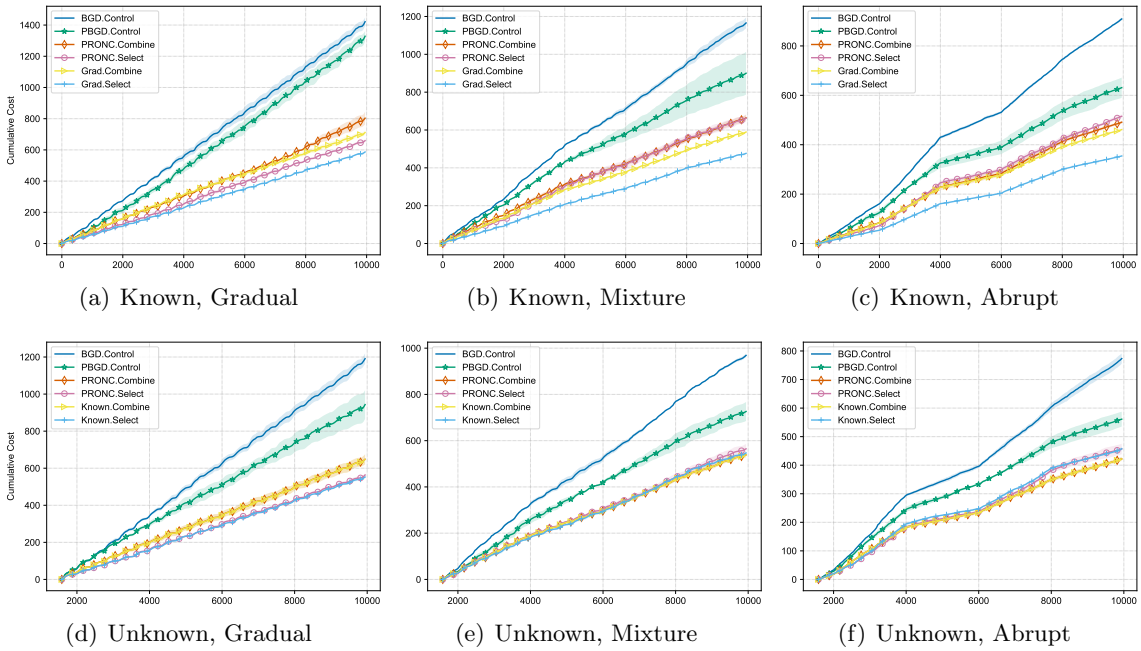
In this part, we show that online non-stochastic control can deal with *time-varying* linear dynamical systems with state transition function $\mathbf{x}_{t+1} = A_t\mathbf{x}_t + B_t\mathbf{u}_t + \boldsymbol{\xi}_t$. Here we suppose the disturbance $\boldsymbol{\xi}_t$ is Gaussian, the transition matrices A_t, B_t follow a semi-time-varying style: $A_t = A + \Delta_t^A, B_t = B + \Delta_t^B$, where A, B are fixed and accessible, and Δ_t^A, Δ_t^B are changing, zero-mean Gaussian and unavailable, served as a kind of model misspecification. Although this system is time-varying, it can still be modelled as a time-invariant one with transition equation $\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \boldsymbol{\xi}'_t$, where $\boldsymbol{\xi}'_t = \boldsymbol{\xi}_t + \Delta_t^A\mathbf{x}_t + \Delta_t^B\mathbf{u}_t$ denotes the adversarial disturbances. Such dynamical systems are more representative and thus more challenging to control.

We design three changing patterns of the online cost functions to simulate the adversarial nature of the non-stochastic control setting. Specifically, we use a changing quadratic cost $c_t(\mathbf{y}_t, \mathbf{u}_t) = \mathbf{y}_t^\top R_t\mathbf{y}_t + \mathbf{u}_t^\top P_t\mathbf{u}_t$, where R_t, P_t are time-varying and can change gradually, abruptly, or in a mixture of the former ways:

- Gradual change (large path length P_T): the parameters R_t, P_t follow the form of $R_t = a_tI, P_t = b_tI$, where $a_t = \sin(t/(10\pi)), b_t = \sin(t/(20\pi))$ are changing sinusoidally;
- Abrupt change (small path length P_T): the whole time horizon is split into 5 stages, equipped with 5 fixed cost functions parameterized via $\{(R_i, P_i)\}_{i\in[5]}$.
- Mixture change (medium path length P_T): the parameters R_t, P_t are fully mixture of the above changing styles.

We also conduct experiments in the unknown system setting, where A, B are unavailable.

Results. Figure 3 plots the cumulative cost curves of our method and the contenders. Smaller cumulative cost indicates better performance. Results in different cases validate the supremacy of our proposed *PRONC.Combine* and *PRONC.Select*. The comparison with *BGD.Control* shows the strong adaptability and robustness of the two-layer online ensemble framework, which answers the second question at the beginning of Section 5. The comparison with *PBGD.Control* reveals that the switching cost regularizer indeed helps obtain small switching costs, thus small cumulative costs, which answers the third question. The comparison with the skylines *Grad.Combine* and *Grad.Select* shows that the one-point



(a) Known, Gradual (b) Known, Mixture (c) Known, Abrupt

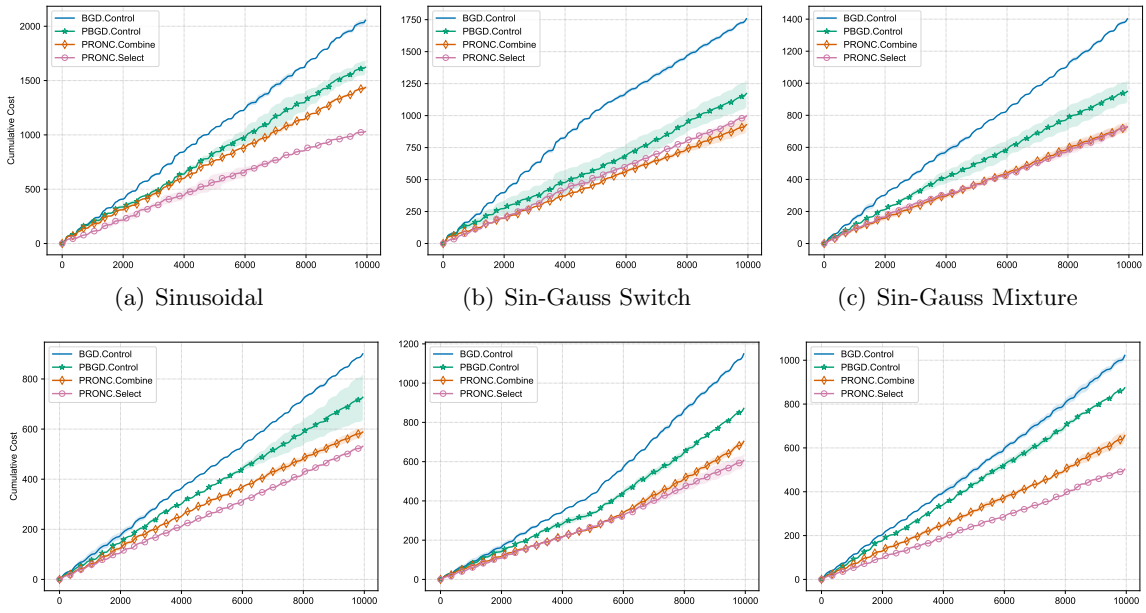
(d) Unknown, Gradual (e) Unknown, Mixture (f) Unknown, Abrupt

Figure 3: Cumulative costs of all methods in the synthetic time-varying LDS environment. Smaller cumulative cost indicates better performance. *BGD.Control* is only equipped with one base learner, *PBGD.Control* is equipped with the meta-base ensemble framework but without the switching cost regularizer, *PRONC.Combine* and *PRONC.Select* are our proposed algorithms. Four skyline methods, named by the keywords 'Grad' and 'Known', have access to the true loss gradients or the true system transition matrices.

gradient estimator is not ideal due to its high variance, inversely proportional to the perturbation. However, as mentioned in Remark 2, designing implementable BCO algorithms with tighter regret is still open. At last, the comparison with the skylines *Known.Combine* and *Known.Select* validates the accuracy of the system estimation (Algorithm 5).

In addition, we observe that, in both known and unknown systems, *PRONC.Select* is preferable when the cost functions are changing gradually (corresponding to large path length P_T), Pronc-Combine is better when the cost functions follow an abrupt changing style (small path length P_T), and the two algorithms are comparable in the mixture case (medium path length P_T), matching our theoretical results in Section 4.4 and Remark 6.

More Structured Disturbances. Furthermore, we validate the robustness of our algorithms under different kinds of structured disturbances. We first investigate the sinusoidal disturbances ($w_t = \sin(t/(20\pi))$) and Gaussian random walk disturbances ($w_t = \mathcal{N}(w_{t-1}, 1/T)$), and further consider the switch/mixture between them. 'A-B Switch' means that the disturbances follow distribution A in the first half horizon and switch to B in the remaining rounds. 'A-B Mixture' means the disturbance of each round is randomly chosen between A and B. Figure 4 presents the results, where our algorithms outperform the baselines when facing various kinds of disturbances, showing the robustness of our method.



(a) Sinusoidal (b) Sin-Gauss Switch (c) Sin-Gauss Mixture

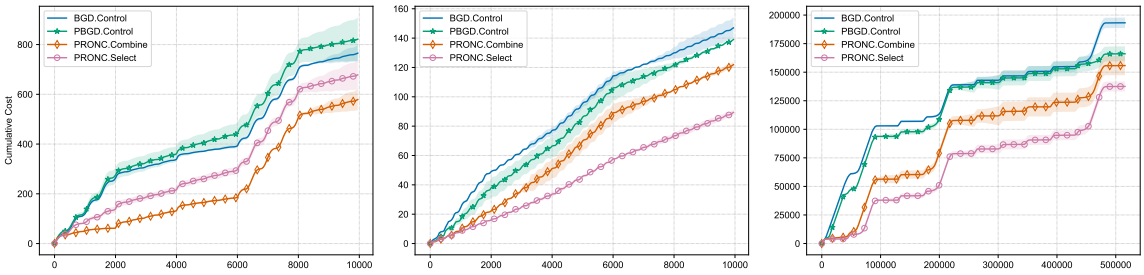
(d) Gaussian Random Walk (e) GRW-Gauss Switch (f) GRW-Gauss Mixture

Figure 4: Cumulative costs of all methods on different kinds of noises in the known synthetic time-varying LDS with gradually changing cost functions. 'Switch/Mixture' means switch or mixture of two kinds of disturbances. 'Gauss' stands for Gaussian noise, 'Sin' is the sinusoidal noise, and 'GRW' denotes the Gaussian random walk noise. *BGD.Control* is only equipped with one base learner, *PBGD.Control* is equipped with the meta-base ensemble framework but without the switching cost regularizer, *PRONC.Combine* and *PRONC.Select* are our proposed algorithms.

5.2 Simulated Nonlinear Environments

We further conduct experiments in simulated *nonlinear* environments in this part. Even though our method and theory are designed for linear dynamical systems, we can approximate nonlinear systems with piecewise linear ones and then treat the approximation error as system noises, so online non-stochastic control remains applicable. In this way, we restart the algorithm periodically, treating the system inside each period as linear and estimating the system transition matrices using Algorithm 5. By doing so, we have demonstrated the modeling power of online non-stochastic control by applying it to nonlinear problems. In the following, we conduct experiments in three simulated nonlinear tasks to examine the effectiveness of our proposed method.

The first application is the simple frictionless *pendulum* environment, a nonlinear but stable system. The goal is to make the pendulum stable in a vertical position. Its state is a 2-dimensional vector $\mathbf{x}_t = [\theta_t, \dot{\theta}_t]^\top$, where θ_t stands for the deviation angle normalized between $[-\pi, \pi]$ and $\dot{\theta}_t$ is the rotational velocity. The action is a scalar $\mathbf{u}_t = \ddot{\theta}_t$ representing the torque applied on the system. Denote by g the gravity, l, m the length and mass of the



(a) Pendulum (b) Cartpole (c) Data Center

Figure 5: Cumulative costs of different methods in three simulated nonlinear tasks: pendulum, cartpole and data center cooling. Smaller cumulative cost indicates better performance. *BGD.Control* is only equipped with one base learner, *PBGD.Control* is equipped with the meta-base ensemble framework but without the switching cost regularizer, *PRONC.Combine* and *PRONC.Select* are our proposed algorithms.

pole, and τ the time between state updates, the dynamics of the pendulum follows:

$$\theta_{t+1} = \theta_t + \tau\dot{\theta}_t, \quad \dot{\theta}_{t+1} = \dot{\theta}_t + \tau\left(\frac{3g\sin(\theta_t+\pi)}{2l} + \frac{3\ddot{\theta}_t}{ml^2}\right).$$

The second application is the *cartpole* environment, also known as the inverted pendulum, a commonly used benchmark consisting of a nonlinear and unstable system popularized by OpenAI Gym (Brockman et al., 2016). Concretely, a pole is attached by an unactuated joint to a cart, which moves along a frictionless track. The pole starts upright, and the goal is to prevent it from falling over by increasing and reducing the cart's velocity. The state consists of 4 statistics: the cart's position x, the cart's velocity $\dot{x}$, the pole's angle θ, and the pole's velocity at tip $\dot{\theta}$. The action u is a continuous value between $[-1, 1]$ where negative means pushing the cart to the left, and positive means the opposite direction. The transition function of the cartpole is a complicated nonlinear equation set. Denote by g the gravity, l_p, m_p the length and mass of the pole, m_c the mass of the cart, and τ the time between state updates, the dynamical function of the cartpole has the following form:

$$\theta_{\text{acc}} = \frac{g\sin\theta - \cos\theta\cdot\frac{u+l_p m_p\dot{\theta}^2\sin\theta}{m_p+m_c}}{l_p\left(\frac{4}{3} - \frac{m_p(\cos\theta)^2}{m_p+m_c}\right)}, \; x_{\text{acc}} = \frac{u + l_p m_p\dot{\theta}^2\sin\theta}{m_p+m_c} - \frac{m_p l_p\theta_{\text{acc}}\cos\theta}{m_p+m_c},$$

$$x_{t+1} = x_t + \tau\dot{x}_t, \; \dot{x}_{t+1} = \dot{x}_t + \tau x_{\text{acc}}, \; \theta_{t+1} = \theta_t + \tau\dot{\theta}_t, \; \dot{\theta}_{t+1} = \dot{\theta}_t + \tau\theta_{\text{acc}},$$

which is highly nonlinear and thus brings significant challenge to online control.

The last simulated application is the *data center cooling*. As mentioned in Section 1, data center cooling is a natural application of online non-stochastic control with partial feedback. The goal is to keep the data center's temperature within an acceptable range with the minimum electricity cost. The state consists of three statistics: the temperature d_t, the number of users n_t and the data transmission rate r_t. The control signal is a scalar ranging from -3 to 3 (negative means cooling down and positive means heating up). For

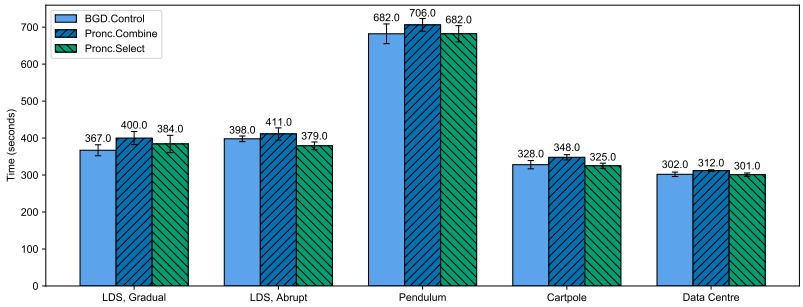


Figure 6: Comparisons of different algorithms in terms of running time.

simplicity, we use $|u_t|$ to represent the electricity cost. The number of users and the data transmission rate vary randomly, and the temperature change is influenced by the intrinsic temperature and the control signal. The current intrinsic temperature is related to the atmospheric temperature of the current time (month), denoted by a_t, the current number of users, and the current data transmission rate. Specifically, the transition function of the temperature is

$$d_{t+1} = d_t + (a_{t+1} - a_t) + 1.25(n_{t+1} - n_t) + 1.25(r_{t+1} - r_t) + u_t.$$

The time horizon consists of the seconds of the whole year ($T \approx 500000$).

Results. Figure 5 plots the cumulative costs of our method and contenders in three simulated nonlinear environments. Smaller cumulative cost indicates better performance. To approximate the systems' nonlinearity, we restart all methods 5, 5, and 12 times in pendulum, cartpole, and data center cooling, respectively. The results show the supremacy of our proposed *PRONC.Combine* and *PRONC.Select* and the modeling power of the online non-stochastic control in simulated nonlinear reinforcement learning tasks.

5.3 Running Time

In the end, we compare the time efficiency of our proposed algorithms, *PRONC.Combine* and *PRONC.Select*, with *BGD.Control*, the most time-efficient method due to only one base learner. Figure 6 plots the average running time with standard deviations of 5 independent runs. Overall, our method is almost as time efficient as *BGD.Control*. To be more rigorous, *BGD.Control* is the best in terms of running time since it is equipped with only one base learner. *PRONC.Select* is comparable because it updates serially, while *PRONC.Combine* is relatively not that efficient since it has to consider all base learners simulated, that is, do multiple projection operations, which is usually time-consuming.

6 Conclusion

This paper investigates online non-stochastic control with partial feedback, where the learner can only receive bandit cost values and partially observed states. The problem setup is ubiquitous in real-world decision-making and control applications and strictly generalizes exceptional cases studied disparately by previous works. We start by extending

the work of Cassel and Koren (2020) to partially observed states through disturbance-response policy parameterization and obtain the first online method for this problem with $\widetilde{\mathcal{O}}(T^{3/4})$ static regret and $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/4})$ dynamic regret whenever the path length P_T is known a priori. To further adapt to unknown non-stationary environments, we design a novel two-layer meta-base online ensemble method by treating the algorithm above as the base learner. And on top of that, we design two meta-combiners to simultaneously handle the unknown environmental variation and the memory issue arising from online control. Our algorithms, Pronc-Combine and Pronc-Select, enjoy $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/2})$ and $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/4}+T^{5/6})$ dynamic regret respectively *without* knowing the non-stationarity in advance. As a byproduct, we give a new dynamic regret bound for standard bandit convex optimization that is more advantageous than the best known result when facing large environmental non-stationarity and might be of independent interest. We further extend our results to unknown systems and obtain $\widetilde{\mathcal{O}}(T^{4/5}(1+P_T)^{1/2})$ and $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/4}+T^{5/6})$ dynamic regret. Finally, empirical studies in both synthetic linear and simulated nonlinear environments validate the effectiveness and efficiency of our proposed method and support our theoretical findings.

There are several questions worth for future research. The first one is to investigate the optimality of our results. Specifically, using static regret as the performance measure, the lower bound of bandit convex optimization shifts from $\Omega(\sqrt{T})$ to $\Omega(T^{2/3})$ due to the presence of the switching cost (Cesa-Bianchi et al., 2013; Dekel et al., 2014). This raises questions about the optimality of the $\widetilde{\mathcal{O}}(T^{3/4})$ regret in our study. Moreover, even if achieving the optimal static regret, it remains challenging to obtain an optimal dynamic regret, because the unknown path length P_T makes the problem essentially hard in the partial feedback scenario. At last, though our results for Pronc-Combine becomes worse in unknown systems, the guarantees for Pronc-Select remain constant, prompting us to question if the Pronc-Combine results can be improved by more dedicated analysis or more refined system estimation techniques.

Besides, recently there is a proposal called "Learnware" which advocates to exploit all kinds of trained machine learning models, submitted by developers all over the world to a *learnware market*, to enable future users not to build their own machine learning application from scratch, without disclosing the data of developers and users (Zhou, 2016). The key is a carefully designed *Learnware specification* which enables the identification and reassemble of helpful models without data disclosure (Zhou and Tan, 2023). The helpful models may be identified in an online fashion based on partially observed output (e.g., only output of a small number of "anchor" learners rather than all learnwares on user data) and partial information (e.g., only overall performance of these anchor learnwares rather than detailed predictions on every data instances). Thus, some inspirations may be obtained from studies of online non-stochastic control with partial feedback.

Acknowledgments

This research was supported by the National Science Foundation of China (61921006, 62206125), JiangsuSF (BK20220776), National Postdoctoral Program for Innovative Talent, and the Collaborative Innovation Center of Novel Software Technology and Industrialization. We are grateful for the anonymous reviewers for their helpful comments.

Appendix A. Preliminaries

In this section, we introduce some preliminaries about the notations, some nice properties of the disturbance response policy, and some projection issues.

A.1 Notations

In this part, we restate the definitions of norms and inner product and define some symbols for abbreviation.

For vectors, $\|\cdot\|, \|\cdot\|_*$ denote the general norm and its corresponding dual norm. For matrices, $\|\cdot\|_{\mathrm{F}}, \|\cdot\|_{\mathrm{op}}$ denote the Frobenius-norm and the operator norm. For tuple of matrices, such as $G \in (\mathbb{R}^{d_y \times d_u})^H, M \in (\mathbb{R}^{d_u \times d_y})^m$, its $\|\cdot\|_{\ell_1,\mathrm{op}}$ norm is defined as $\|M\|_{\ell_1,\mathrm{op}} = \sum_{i=1}^m \left\|M^{[i]}\right\|_{\mathrm{op}}$.

For any two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, their inner product is defined as $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^d \mathbf{x}_i \mathbf{y}_i$, where $\mathbf{x}_i$ denotes the i-th entry of $\mathbf{x}$.

Recall that G is a tuple of matrices, where $G^{[i]} = CA^{i-1}B$ for $i \geq 1$ (see Definition 3). For representation simplicity, we define the R_G, R_{nat} to bound $\|G\|_{\ell_1,\mathrm{op}}$ and $\|\mathbf{y}^{\mathrm{nat}}\|_2$ respectively using the T-independent variables defined in Assumptions 1-2, formally,

$$\left\|\mathbf{y}_t^{\mathrm{nat}}\right\|_2 \leq \left\|\mathbf{e}_t + \sum_{i=1}^{t-1} CA^{i-1}\boldsymbol{\xi}_{t-i}\right\|_2 \leq W + \kappa_C W \sum_{i=1}^{t-1} \kappa_A^{i-1} \leq W + \frac{\kappa_C W}{1-\kappa_A} \triangleq R_{\mathrm{nat}},$$

$$1 + \|G\|_{\ell_1,\mathrm{op}} = 1 + \sum_{i=1}^{\infty} \left\|CA^{i-1}B\right\|_{\mathrm{op}} \leq 1 + \frac{\kappa_B \kappa_C}{1-\kappa_A} \triangleq R_G.$$

Let $\mathcal{M} \subseteq (\mathbb{R}^{d_u \times d_y})^m$ be the feasible domain for DRP parameter M. Without loss of generality, we assume $\mathcal{M}$ contains the ball of radius r_M centered at the origin and is contained in the ball of radius R_M, i.e., $r_M \leq \|M\|_{\mathrm{F}} \leq R_M$ holds for any $M \in \mathcal{M}$. Furthermore, using relationship between different norms, it holds that

$$\|M\|_{\ell_1,\mathrm{op}} \leq \sqrt{m}\,\|M\|_{\mathrm{F}} \leq \sqrt{m} R_M \triangleq R_{\mathcal{M}}.$$

A.2 Disturbance Response Policy

In this part, we analyze the derivation of the Nature's $\mathbf{y}_t^{\mathrm{nat}}$ and the relationship between the true observation $\mathbf{y}_t$ and $\mathbf{y}_t^{\mathrm{nat}}$ (see Lemma 3).

To analyze Nature's y, we denote by $\mathbf{x}_t^{\mathrm{nat}}$ the state without any actions implemented on the system, namely, $\mathbf{x}_t^{\mathrm{nat}} = A\mathbf{x}_{t-1}^{\mathrm{nat}} + \boldsymbol{\xi}_{t-1}$. The formulation of Nature's $\mathbf{y}_t^{\mathrm{nat}}$ (see also Definition 1) satisfies

$$\begin{aligned}
\mathbf{y}_t^{\mathrm{nat}} &= C\mathbf{x}_t^{\mathrm{nat}} + \mathbf{e}_t = C\left(A\mathbf{x}_{t-1}^{\mathrm{nat}} + \boldsymbol{\xi}_{t-1}\right) + \mathbf{e}_t = CA\mathbf{x}_{t-1}^{\mathrm{nat}} + C\boldsymbol{\xi}_{t-1} + \mathbf{e}_t \\
&= CA^2\mathbf{x}_{t-2}^{\mathrm{nat}} + CA\boldsymbol{\xi}_{t-2} + C\boldsymbol{\xi}_{t-1} + \mathbf{e}_t \\
&= \dots \\
&= CA^t\mathbf{x}_0^{\mathrm{nat}} + \sum_{i=1}^{t-1} CA^{i-1}\boldsymbol{\xi}_{t-i} + \mathbf{e}_t
\end{aligned}$$

$$= \mathbf{e}_t + \sum_{i=1}^{t-1} CA^{i-1}\boldsymbol{\xi}_{t-i}.$$

The above derivation shows that the Nature's $\mathbf{y}_t^{\text{nat}}$ is actually a combination of the past system disturbances $\boldsymbol{\xi}_{1:t}, \mathbf{e}_{1:t}$. With the above result, Lemma 3 presents the following relationship between the observation $\mathbf{y}_t$ and the Nature's $\mathbf{y}_t^{\text{nat}}$.

Lemma 3. *For any linear dynamical system with partial feedback* (3.1) *subject to actions* $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathbb{R}^{d_u}$*, it holds that* $\mathbf{y}_t = \mathbf{y}_t^{\text{nat}} + \sum_{i=1}^{t-1} CA^{i-1}B\mathbf{u}_{t-i}$.

Proof We decompose the observation $\mathbf{y}_t$ by the dynamics of linear dynamical system with partial feedback (3.1),

$$\begin{aligned}
\mathbf{y}_t &= C\mathbf{x}_t + \mathbf{e}_t = C\left(A\mathbf{x}_{t-1} + B\mathbf{u}_{t-1} + \boldsymbol{\xi}_{t-1}\right) + \mathbf{e}_t \\
&= CA\mathbf{x}_{t-1} + CB\mathbf{u}_{t-1} + C\boldsymbol{\xi}_{t-1} + \mathbf{e}_t \\
&= CA\left(A\mathbf{x}_{t-2} + B\mathbf{u}_{t-2} + \boldsymbol{\xi}_{t-2}\right) + CB\mathbf{u}_{t-1} + C\boldsymbol{\xi}_{t-1} + \mathbf{e}_t \\
&= CA^2\mathbf{x}_{t-2} + CAB\mathbf{u}_{t-2} + CA\boldsymbol{\xi}_{t-2} + CB\mathbf{u}_{t-1} + C\boldsymbol{\xi}_{t-1} + \mathbf{e}_t \\
&= \ldots \\
&= CA^t\mathbf{x}_0 + \mathbf{e}_t + \sum_{i=1}^{t-1} CA^{i-1}\boldsymbol{\xi}_{t-i} + \sum_{i=1}^{t-1} CA^{i-1}B\mathbf{u}_{t-i} \\
&= \mathbf{y}_t^{\text{nat}} + \sum_{i=1}^{t-1} CA^{i-1}B\mathbf{u}_{t-i},
\end{aligned}$$

where the last step is by assuming $\mathbf{x}_0 = 0$, without loss of generality. ■

Lemma 3 shows that the current observation $\mathbf{y}_t$ is a combination of the impact of both the past disturbances (Nature's y) and the past actions $\mathbf{u}_{1:t-1}$.

A.3 Projection Issues

In bandit convex optimization, in order to have enough wiggle space for perturbation, we project the decision to a shrunk set. In this part, we introduce some basic properties about the relationship between a domain and its shrunk set.

Let $\mathcal{K} \subseteq \mathbb{R}^d$ be a closed and convex domain, and without loss of generality, we assume $\mathcal{K}$ contains the ball of radius r at the origin and is contained in the ball of radius R, i.e.,

$$r\mathbb{B} \subseteq \mathcal{K} \subseteq R\mathbb{B}, \tag{A.1}$$

where $\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 \leq 1\}$. The following results hold. First, for any point $\mathbf{y} \in (1-\alpha)\mathcal{K}$ where $\alpha \in (0,1)$ denotes the shrinkage parameter, the ball of radius αr centered at $\mathbf{y}$ belongs to domain $\mathcal{K}$, i.e., $(1-\alpha)\mathcal{K} + \alpha r\mathbb{B} \subseteq (1-\alpha)\mathcal{K} + \alpha\mathcal{K} \subseteq \mathcal{K}$ holds since $r\mathbb{B} \subseteq \mathcal{K}$ and $\mathcal{K}$ is convex. In practice, we only have perturbation magnitude δ and radius r, and need to compute the shrinkage via $\alpha = \delta/r$. Before presenting the second property, we define the smoothed version of a Lipschitz continuous function.

Definition 4 (Smoothed Function)**.** Assuming that $\mathcal{K}$ satisfies (A.1), a smoothed function of f, denoted by $\bar{f}$, is defined as $\bar{f}(\mathbf{y}) = \mathbb{E}_{\mathbf{s}\in\mathbb{S}}\left[f(\mathbf{y}+\delta\mathbf{s})\right]$ for any $\mathbf{y} \in (1-\alpha)\mathcal{K}$, where $\alpha = \delta/r$ and $\mathbf{s}$ is drawn from a uniform distribution over the unit sphere $\mathbb{S} \subseteq \mathbb{R}^d$.

The second property (Lemma 4) reveals the relationship between a function and its smoothed version, which can be derived immediately using the Lipschitzness of f.

Lemma 4. *If $\mathcal{K}$ satisfies* (A.1) *and f is L-Lipschitz, and let $\bar{f}$ be the smoothed version of f (Definition 4), then for any $\mathbf{y} \in (1-\alpha)\mathcal{K}$, $\left|\bar{f}(\mathbf{y}) - f(\mathbf{y})\right| \leq L\delta$.*

Appendix B. Analysis

In this section, we give detailed proofs of the results stated in Section 4, including the proof of the gradient bias lemma, the regret bound of the base algorithm, theoretical guarantees for PRONC-COMBINE and PRONC-SELECT in bandit convex optimization with memory setting, as well as in online non-stochastic control in known and unknown systems.

B.1 Proof of Lemma 2

Proof We expand the gradient estimator $\widetilde{\mathbf{g}}$ using its definition

$$\mathbb{E}\left[\widetilde{\mathbf{g}}\right] = \mathbb{E}\left[\frac{d}{\delta}(f(\mathbf{w})+\varepsilon)\mathbf{s}\right] = \mathbb{E}\left[\frac{d}{\delta}f(\bar{\mathbf{w}}+\delta\mathbf{s})\mathbf{s} + \frac{d\varepsilon\mathbf{s}}{\delta}\right] = \nabla\bar{f}(\bar{\mathbf{w}}) + \mathbb{E}\left[\frac{d\varepsilon\mathbf{s}}{\delta}\right],$$

where the last expression is due to the unbiased gradient estimation established in Flaxman et al. (2005). The last term $\mathbb{E}\left[d\varepsilon\mathbf{s}/\delta\right]$ does not simply equal to 0 because ε also has a dependence on $\mathbf{s}$. Finally, we finish the proof by

$$\left\|\mathbb{E}\left[\widetilde{\mathbf{g}}\right] - \nabla\bar{f}(\bar{\mathbf{w}})\right\|_2 \leq \left\|\mathbb{E}\left[\frac{d\varepsilon\mathbf{s}}{\delta}\right]\right\|_2 \leq \frac{d\varepsilon}{\delta},$$

where the second step holds due to Jensen's inequality. ■

B.2 Proof of Theorem 1

Proof To begin with, using Lemma 9 to build a relationship between the time horizon T and the mini-batching update rounds S, the dynamic regret can be written as

$$\begin{aligned}\mathbb{E}[\text{D-REG}(\mathbf{v}_{1:T})] &\leq \mathbb{E}\left[\sum_{t=1}^{T}(\widetilde{f}_t(\mathbf{w}_t) - \widetilde{f}_t(\mathbf{v}_t)) + \lambda\sum_{t=H}^{T}\left\|\mathbf{w}_t - \mathbf{w}_{t-1}\right\|_2 + \lambda\sum_{t=2}^{T}\left\|\mathbf{v}_t - \mathbf{v}_{t-1}\right\|_2\right] \\ &\leq 3H\mathbb{E}\left[\sum_{t\in S}\widetilde{f}_t(\mathbf{w}_t) - \sum_{t\in S}\widetilde{f}_t(\mathbf{v}_t)\right] + \lambda\mathbb{E}\left[\sum_{t\in S}\left\|\mathbf{w}_t - \mathbf{w}_{t-1}\right\|_2\right] + \lambda P_T,\end{aligned}$$

where the first inequality comes from (4.2), the second inequality is because of Lemma 9, the property of mini-batches and the definition of path length P_T. To be more rigorous, we use the notation $\{\mathbf{w}_t(\delta,\eta)\}$ to denote the decision sequence generated by the algorithm

given certain perturbation magnitude δ and step size η, and sometimes we will simplify $\mathbf{w}_t(\delta,\eta)$ as $\mathbf{w}_t$ for simplicity of representation. The base-regret can be decomposed as

$$\begin{aligned}\mathbb{E}[\text{D-Reg}(\mathbf{v}_{1:T})] \le 3H\underbrace{\mathbb{E}\left[\sum_{t\in S}\bar{f}_t(\bar{\mathbf{w}}_t)-\sum_{t\in S}\bar{f}_t(\bar{\mathbf{v}}_t)\right]}_{\text{term (a)}}+\lambda\underbrace{\mathbb{E}\left[\sum_{t\in S}\|\bar{\mathbf{w}}_t-\bar{\mathbf{w}}_{t-1}\|_2\right]}_{\text{term (b)}}\\ +3H\underbrace{\mathbb{E}\left[\sum_{t\in S}\widetilde{f}_t(\mathbf{w}_t)-\bar{f}_t(\bar{\mathbf{w}}_t)\right]}_{\text{term (c)}}+3H\underbrace{\mathbb{E}\left[\sum_{t\in S}\bar{f}_t(\bar{\mathbf{v}}_t)-\widetilde{f}_t(\mathbf{v}_t)\right]}_{\text{term (d)}}+2\lambda\underbrace{\sum_{t\in S}\|\mathbf{w}_t-\bar{\mathbf{w}}_t\|_2}_{\text{term (e)}}+\lambda P_T,\end{aligned} \tag{B.1}$$

where $\bar{f}_t$ denotes the smoothed version of $\widetilde{f}_t$ (see Definition 4), $\bar{\mathbf{w}}_t, \bar{\mathbf{v}}_t$ are the scaled decision and comparator such that $\mathbf{w}_t = \bar{\mathbf{w}}_t + \delta\mathbf{s}_t$ and $\bar{\mathbf{v}}_t = (1-\alpha)\mathbf{v}_t$. In above, term (a) is the regret on loss functions $\{\bar{f}_t\}_{t\in S}$, term (b) is the switching cost of the decision sequence $\{\bar{\mathbf{w}}_t\}_{t\in S}$, term (c) and term (d) are the gap between the unary losses $\{\widetilde{f}_t\}_{t\in S}$ and their smoothed versions $\{\bar{f}_t\}_{t\in S}$, and term (e) measures the gap between $\{\mathbf{w}_t\}_{t\in S}$ and $\{\bar{\mathbf{w}}_t\}_{t\in S}$. Now we bound these terms one by one. First, using the Lipschitzness of f_t and Lemma 4 to bound the difference between a function and its smoothed version, term (c) satisfies that

$$\text{term (c)} = \mathbb{E}\left[\sum_{t\in S}\left(\widetilde{f}_t(\mathbf{w}_t)-\widetilde{f}_t(\bar{\mathbf{w}}_t)+\widetilde{f}_t(\bar{\mathbf{w}}_t)-\bar{f}_t(\bar{\mathbf{w}}_t)\right)\right] \le 2LH\delta\mathbb{E}[|S|] \le 2L\delta T,$$

where the last step uses $\mathbb{E}[|S|] \le \lceil T/H\rceil$ (see Lemma 8). Similarly, the Lipschitzness of $\widetilde{f}_t$ gives the upper bound of term (d),

$$\begin{aligned}\text{term (d)} &= \mathbb{E}\left[\sum_{t\in S}\left(\bar{f}_t(\mathbf{v}_t)-\widetilde{f}_t(\bar{\mathbf{v}}_t)+\widetilde{f}_t(\bar{\mathbf{v}}_t)-\widetilde{f}_t(\mathbf{v}_t)\right)\right] \le \mathbb{E}\left[\sum_{t\in S}\left(\widetilde{L}\alpha R+\widetilde{L}\delta\right)\right]\\ &= (R/r+1)LH\delta\mathbb{E}[|S|] \le (R/r+1)L\delta T.\end{aligned}$$

It is easy to verify that term (e) $\le 2\lambda\delta T/H$. By the update rule (4.4), term (b) satisfies

$$\text{term (b)} = \mathbb{E}\left[\sum_{t\in S}\|\bar{\mathbf{w}}_t(\delta,\eta)-\bar{\mathbf{w}}_{t-1}(\delta,\eta)\|_2\right] \le \mathbb{E}\left[\sum_{t\in S}\|\eta\widetilde{\mathbf{g}}_t(\delta,\eta)\|_2\right] \le \frac{dC_f\eta T}{H\delta}. \tag{B.2}$$

At last, we investigate term (a), which can be further decomposed into two parts by exploiting the convexity of $\bar{f}_t$,

$$\begin{aligned}\text{term (a)} \le \mathbb{E}\left[\sum_{t\in S}\langle\nabla\bar{f}_t(\bar{\mathbf{w}}_t),\bar{\mathbf{w}}_t-\bar{\mathbf{v}}_t\rangle\right] &= \mathbb{E}\left[\sum_{t\in S}\langle\widetilde{\mathbf{g}}_t,\bar{\mathbf{w}}_t-\bar{\mathbf{v}}_t\rangle\right]\\ &+\mathbb{E}\left[\sum_{t\in S}\langle\nabla\bar{f}_t(\bar{\mathbf{w}}_t)-\widetilde{\mathbf{g}}_t,\bar{\mathbf{w}}_t-\bar{\mathbf{v}}_t\rangle\right].\end{aligned} \tag{B.3}$$

Lemma 7 gives the upper bound of the first term above,

$$\mathbb{E}\left[\sum_{t\in S}\langle\widetilde{\mathbf{g}}_t,\bar{\mathbf{w}}_t-\bar{\mathbf{v}}_t\rangle\right] \le \frac{7R^2+RP_S}{4\eta}+\frac{d^2C_f^2\eta T}{2H\delta^2} \le \frac{7R^2+RP_T}{4\eta}+\frac{d^2C_f^2\eta T}{2H\delta^2}, \tag{B.4}$$

where P_S denotes the path length upon S and the last step is due to $P_S \leq P_T$. To bound the second term in (B.3), we denote by $\mathcal{F}_t \triangleq \{\mathbf{w}_1, f_1, \ldots, \mathbf{w}_t, f_t\}$ the history up to round t. Consequently, it holds that

$$\mathbb{E}\left[\sum_{t\in S}\langle \nabla \bar{f}_t(\bar{\mathbf{w}}_t) - \widetilde{\mathbf{g}}_t, \bar{\mathbf{w}}_t - \bar{\mathbf{v}}_t\rangle\right] = \mathbb{E}\left[\mathbb{E}\left[\sum_{t\in S}\langle \nabla \bar{f}_t(\bar{\mathbf{w}}_t) - \widetilde{\mathbf{g}}_t, \bar{\mathbf{w}}_t - \bar{\mathbf{v}}_t\rangle \,\middle|\, \mathcal{F}_t\right]\right].$$

The inner expectation can be bounded as

$$\begin{aligned}&\mathbb{E}\left[\sum_{t\in S}\langle \nabla \bar{f}_t(\bar{\mathbf{w}}_t) - \widetilde{\mathbf{g}}_t, \bar{\mathbf{w}}_t - \bar{\mathbf{v}}_t\rangle \,\middle|\, \mathcal{F}_t\right] = \sum_{t\in S}\langle \nabla \bar{f}_t(\bar{\mathbf{w}}_t) - \mathbb{E}[\widetilde{\mathbf{g}}_t|\mathcal{F}_t], \bar{\mathbf{w}}_t - \bar{\mathbf{v}}_t\rangle \\ \leq\ & \frac{T}{H}\cdot \max_{t\in[T]} \left\|\nabla \bar{f}_t(\bar{\mathbf{w}}_t) - \mathbb{E}[\widetilde{\mathbf{g}}_t|\mathcal{F}_t]\right\|_2 \cdot 2R \leq \frac{2d\varepsilon RT}{H\delta},\end{aligned}$$

where the last step is due to Lemma 2. Considering the outer expectation, the bias term can be bounded by

$$\mathbb{E}\left[\sum_{t\in S}\langle \nabla \bar{f}_t(\bar{\mathbf{w}}_t) - \widetilde{\mathbf{g}}_t, \bar{\mathbf{w}}_t - \bar{\mathbf{v}}_t\rangle\right] \leq \frac{2d\varepsilon RT}{H\delta}. \tag{B.5}$$

Plugging all the bounds into (B.1), we have

$$\mathbb{E}[\text{D-REG}(\mathbf{v}_{1:T})] \leq \frac{3H(7R^2 + RP_T)}{4\eta} + \frac{3d^2C_f^2\eta T}{2\delta^2} + \frac{6d\varepsilon RT}{\delta} + \frac{\lambda d C_f \eta T}{H\delta} + L_{\text{eff}}\delta T + \lambda P_T, \tag{B.6}$$

where $L_{\text{eff}} \triangleq (3L + RL/r + 2\lambda/H)\delta T$ denotes the effective Lipschitz constant. ■

B.3 Proof of Theorem 2

Proof The dynamic regret can be decomposed similarly as in Theorem 1,

$$\begin{aligned}\mathbb{E}[\text{D-REG}(\mathbf{v}_{1:T})] \leq\ & \underbrace{3H\mathbb{E}\left[\sum_{t\in S}\langle \widetilde{\mathbf{g}}_t, \bar{\mathbf{w}}_t - \bar{\mathbf{v}}_t\rangle\right] + \lambda\mathbb{E}\left[\sum_{t\in S}\|\bar{\mathbf{w}}_t - \bar{\mathbf{w}}_{t-1}\|_2\right]}_{\text{TERM (A)}} \\ & + 3H\,\underbrace{\mathbb{E}\left[\sum_{t\in S}\langle \nabla \bar{f}_t(\bar{\mathbf{w}}_t) - \widetilde{\mathbf{g}}_t, \bar{\mathbf{w}}_t - \bar{\mathbf{v}}_t\rangle\right]}_{\text{BIAS-TERM}} + \lambda P_T + L_{\text{eff}}\delta T,\end{aligned}$$

where $\bar{f}_t$ is the smoothed version of f_t (see Definition 4), $\bar{\mathbf{w}}_t$ is a shrunk version of $\mathbf{w}_t$ such that $\mathbf{w}_t = \bar{\mathbf{w}}_t + \delta\mathbf{s}_t \in \mathcal{K}$ holds that any $\mathbf{s}_t \in \mathbb{S}$ and L_{eff} is the effective Lipschitz constant. The bias term is at most $2d\varepsilon RT/(\delta^\star H)$ as in (B.5). TERM (A) can be further decomposed into two parts by importing an intermediate term $\ell_{t,i}(\bar{\mathbf{w}}_{t,i})$ (4.5),

$$\text{TERM (C)} \leq \sum_{t\in S}\mathbb{E}[\underbrace{\lambda\|\bar{\mathbf{w}}_t - \bar{\mathbf{w}}_{t-1}\|_2 + 3H\langle \widetilde{\mathbf{g}}_t, \bar{\mathbf{w}}_t\rangle - \left(\lambda\|\bar{\mathbf{w}}_{t,i} - \bar{\mathbf{w}}_{t-1,i}\|_2 + 3H\langle \widetilde{\mathbf{g}}_t, \bar{\mathbf{w}}_{t,i}\rangle\right)}_{\triangleq M_{t,i}}]$$

$$+\sum_{t\in S}\mathbb{E}[\underbrace{\lambda\left\|\bar{\mathbf{w}}_{t,i}-\bar{\mathbf{w}}_{t-1,i}\right\|_2+3H\langle\widetilde{\mathbf{g}}_t,\bar{\mathbf{w}}_{t,i}-\bar{\mathbf{v}}_t\rangle}_{\triangleq B_{t,i}}].$$

The term $M_{t,i}$ can be further transformed by exploiting the structure of $\bar{\mathbf{w}}_t$,

$$\begin{aligned}\left\|\bar{\mathbf{w}}_t-\bar{\mathbf{w}}_{t-1}\right\|_2&=\Big\|\sum_{i=1}^N p_{t,i}\bar{\mathbf{w}}_{t,i}-\sum_{i=1}^N p_{t-1,i}\bar{\mathbf{w}}_{t-1,i}\Big\|_2\\&\le\Big\|\sum_{i=1}^N p_{t,i}\bar{\mathbf{w}}_{t,i}-\sum_{i=1}^N p_{t,i}\bar{\mathbf{w}}_{t-1,i}\Big\|_2+\Big\|\sum_{i=1}^N p_{t,i}\bar{\mathbf{w}}_{t-1,i}-\sum_{i=1}^N p_{t-1,i}\bar{\mathbf{w}}_{t-1,i}\Big\|_2\\&=\sum_{i=1}^N p_{t,i}\left\|\bar{\mathbf{w}}_{t,i}-\bar{\mathbf{w}}_{t-1,i}\right\|_2+R\left\|\boldsymbol{p}_t-\boldsymbol{p}_{t-1}\right\|_1.\end{aligned}\tag{B.7}$$

As a result, we have the following upper bound for $M_{t,i}$,

$$\begin{aligned}M_{t,i}&=\lambda\left\|\bar{\mathbf{w}}_t-\bar{\mathbf{w}}_{t-1}\right\|_2+3H\langle\widetilde{\mathbf{g}}_t,\bar{\mathbf{w}}_t\rangle-\lambda\left\|\bar{\mathbf{w}}_{t,i}-\bar{\mathbf{w}}_{t-1,i}\right\|_2-3H\langle\widetilde{\mathbf{g}}_t,\bar{\mathbf{w}}_{t,i}\rangle\\&\le\lambda\left(\sum_{i=1}^N p_{t,i}\left\|\bar{\mathbf{w}}_{t,i}-\bar{\mathbf{w}}_{t-1,i}\right\|_2+R\left\|\boldsymbol{p}_t-\boldsymbol{p}_{t-1}\right\|_1\right)+3H\sum_{i=1}^N p_{t,i}\langle\widetilde{\mathbf{g}}_t,\bar{\mathbf{w}}_{t,i}\rangle\\&\quad-\lambda\left\|\bar{\mathbf{w}}_{t,i}-\bar{\mathbf{w}}_{t-1,i}\right\|_2-3H\langle\widetilde{\mathbf{g}}_t,\bar{\mathbf{w}}_{t,i}\rangle\\&=\lambda R\left\|\boldsymbol{p}_t-\boldsymbol{p}_{t-1}\right\|_1+\langle\boldsymbol{p}_t,\boldsymbol{\ell}_t\rangle-\ell_{t,i}.\end{aligned}$$

Summing $M_{t,i}$ over the time horizon S, we can transform $\sum_{t\in S}M_{t,i}$ into

$$\sum_{t\in S}M_{t,i}\le\lambda R\sum_{t\in S}\left\|\boldsymbol{p}_t-\boldsymbol{p}_{t-1}\right\|_1+\sum_{t\in S}\langle\boldsymbol{p}_t,\boldsymbol{\ell}_t\rangle-\sum_{t\in S}\ell_{t,i},$$

where $\boldsymbol{p}_t=[p_{t,1},\ldots,p_{t,N}]^\top$ and $\boldsymbol{\ell}_t=[\ell_{t,1},\ldots,\ell_{t,N}]^\top$. Notice that

- $\mathbb{E}[\sum_{t\in S}M_{t,i}]$ represents the *meta-regret*: the static regret of a Prediction with Expert Advice (PEA) problem with switching cost;
- $\mathbb{E}[\sum_{t\in S}B_{t,i}]$ represents the *base-regret*: the dynamic regret of the i-th base learner compared with a sequence of time-varying comparators $\bar{\mathbf{v}}_{1:T}$ with switching cost.

We start by analyzing the base-regret, which can be bounded intermediately by (B.4) and (B.2) in the proof of Theorem 1,

$$\mathbb{E}\left[\sum_{t\in S}B_{t,i}\right]\le\frac{3H(7R^2+RP_T)}{4\eta_i}+\frac{3d^2C_f^2\eta_iT}{2\delta^{\star2}}+\frac{\lambda dC_f\eta_iT}{H\delta^\star}.$$

By the construction of the step size pool (4.8), there must exist an index $i^\star$ such that $\eta_{i^\star}\le\eta^\star\le2\eta_{i^\star}$, by choosing $i=i^\star$, the regret of the $i^\star$-th base learner can be bounded as

$$\begin{aligned}\mathbb{E}\left[\sum_{t\in S}B_{t,i^\star}\right]&\le\frac{3H(7R^2+RP_T)}{4\eta_{i^\star}}+\frac{3d^2C_f^2\eta_{i^\star}T}{2\delta^{\star2}}+\frac{\lambda dC_f\eta_{i^\star}T}{H\delta^\star}\\&\le\frac{3H(7R^2+RP_T)}{2\eta^\star}+\frac{3d^2C_f^2\eta^\star T}{2\delta^{\star2}}+\frac{\lambda dC_f\eta^\star T}{H\delta^\star}.\end{aligned}\tag{B.8}$$

Besides, it is easy to find that the index of the optimal base learner satisfies

$$\eta_{i^\star} \le \eta^\star \Rightarrow 2^{i^\star-1}\frac{\delta^\star R}{dC_f}\sqrt{\frac{7H}{T}} \le \frac{\delta^\star}{dC_f}\sqrt{\frac{H(7R^2+RP_T)}{T}} \Rightarrow i^\star \le \frac{1}{2}\log_2\left(1+\frac{P_T}{7R}\right)+1. \quad \text{(B.9)}$$

The next lemma shows the regret of the meta learner of PRONC-COMBINE, and the proof is deferred to the end of this part.

Lemma 5 (Meta Regret of PRONC-COMBINE)**.** *Suppose the domain's radius is bounded by R, and the ℓ_2-norm of the estimated gradient is bounded by G_f. By setting learning rate $\eta_{meta} = \sqrt{H/(2\lambda RT)}$, the regret of the meta learner of* PRONC-COMBINE *satisfies*

$$\mathbb{E}\left[\sum_{t\in S} M_{t,i}\right] \le 20\lambda^{3/2}R^{3/2}G_f \ln(i+1)\sqrt{\frac{T}{H}}.$$

Thus we can upper bound the meta-regret by Lemma 5 and (B.9),

$$\begin{aligned}\mathbb{E}\left[\sum_{t\in S} M_{t,i^\star}\right] &\le \frac{20\lambda^{3/2}dC_fR^{3/2}}{\delta^\star}\sqrt{\frac{T}{H}}\ln(i^\star+1)\\ &\le \frac{20\lambda^{3/2}dC_fR^{3/2}}{\delta^\star}\sqrt{\frac{T}{H}}\ln\left(\frac{1}{2}\log\left(1+\frac{P_T}{7R}\right)+2\right).\end{aligned} \quad \text{(B.10)}$$

Plugging in the results from (B.10) and (B.8), we have

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=H}^{T} f_t(\mathbf{w}_{t-H:t}) - \sum_{t=H}^{T} f_t(\mathbf{v}_{t-H:t})\right]\\ \le\ &\frac{20\lambda^{3/2}dC_fR^{3/2}}{\delta^\star}\sqrt{\frac{T}{H}}\ln\left(\frac{1}{2}\log\left(1+\frac{P_T}{7R}\right)+2\right)+\frac{3H(7R^2+RP_T)}{2\eta^\star}\\ &+\frac{3d^2C_f^2\eta^\star T}{2{\delta^\star}^2}+\frac{6\varepsilon dRT}{\delta^\star}+\frac{\lambda dC_f\eta^\star T}{H\delta^\star}+L_{\text{eff}}\delta^\star T+\lambda P_T.\end{aligned} \quad \text{(B.11)}$$

By setting the optimal perturbation magnitude $\delta^\star$ and step size $\eta^\star$ as

$$\delta^\star = \sqrt{\frac{3dC_f}{L_{\text{eff}}}}\left(\frac{7HR^2}{T}\right)^{1/4}, \quad \eta^\star = \frac{\delta^\star}{dC_f}\sqrt{\frac{H(7R^2+RP_T)}{T}},$$

the overall regret is about $\mathcal{O}(\min\{T^{3/4}(1+P_T)^{1/2}, T\})$. ■

Proof [of Lemma 5] Notice that meta-regret is the static regret of the prediction with expert advice (PEA) problem with switching cost. Denote by $L_{\max} = \max_{t\in[T]}\|\boldsymbol{\ell}_t\|_\infty$ the maximum infinity norm of $\boldsymbol{\ell}_t$. For technical considerations, we adopt a non-uniform weight initialization by setting $\boldsymbol{p}_1 \in \Delta_N$ with $p_{1,i} = (N+1)/(i(i+1)N)$. Following the standard analysis of Hedge (Freund and Schapire, 1997) and the non-uniform weight initialization (Cesa-Bianchi and Lugosi, 2006, Exercise 2.5), the unary part of the meta-regret satisfies

$$\mathbb{E}_S\left[\sum_{t\in S}\langle \boldsymbol{p}_t, \boldsymbol{\ell}_t\rangle - \sum_{t\in S}\ell_{t,i}\right] \le L_{\max}\left(\frac{\ln(1/p_{1,i})}{\eta_{\text{meta}}} + \eta_{\text{meta}}\mathbb{E}_S[|S|]\right), \quad \text{(B.12)}$$

where $\mathbb{E}_S[\cdot]$ is taken over the randomness of mini-batches S. Next, due to the stability of mirror-descent/FTRL based methods (Shalev-Shwartz, 2012, Lemma 2.10) (Hedge is actually an instance of FTRL with the negative-entropy regularizer), the one-step switching cost of Hedge satisfies that

$$\|\boldsymbol{p}_t - \boldsymbol{p}_{t-1}\|_1 \leq \eta_{\text{meta}} \|\boldsymbol{\ell}_{t-1}\|_\infty \leq \eta_{\text{meta}} L_{\max}. \tag{B.13}$$

Further, $L_{\max}$ can be bounded as

$$\begin{aligned} L_{\max} = \|\boldsymbol{\ell}_t\|_\infty &= \max_{t\in[T]} \max_{i\in[N]} |\ell_{t,i}| \leq \max_{t\in[T]} \max_{i\in[N]} \left(\lambda \|\bar{\mathbf{w}}_{t,i} - \bar{\mathbf{w}}_{t-1,i}\|_2 + 3H|\langle \widetilde{\mathbf{g}}_t, \bar{\mathbf{w}}_{t,i}\rangle|\right) \\ &\leq 2\lambda R + 3H \max_{t\in[T]} \|\widetilde{\mathbf{g}}_t\|_2 \max_{i\in[N]} \|\bar{\mathbf{w}}_{t,i}\|_2 \leq 2\lambda R + 3HG_f R \leq 5\lambda G_f R. \end{aligned}$$

Combining (B.12) and (B.13), we have that

$$\begin{aligned} \mathbb{E}_S\left[\sum_{t\in S} M_{t,i}\right] &\leq L_{\max}\left(\frac{\ln(1/p_{1,i})}{\eta_{\text{meta}}} + (1+\lambda R)\eta_{\text{meta}}\mathbb{E}_S[|S|]\right) \\ &\leq 5\lambda G_f R\left(\frac{\ln(1+i)}{\eta_{\text{meta}}} + \frac{2\lambda R\eta_{\text{meta}} T}{H}\right) \\ &= 20\lambda^{3/2} R^{3/2} G_f \ln(i+1)\sqrt{\frac{T}{H}}. \end{aligned}$$

The last step holds by setting learning rate of meta learner as $\eta_{\text{meta}} = \sqrt{H/(2\lambda RT)}$. ■

B.4 Proof of Theorem 3

Proof Dividing the time horizon S into $\lceil |S|/\Delta \rceil$ episodes, denoted by $\{\Delta_i\}_{i=1}^{\lceil |S|/\Delta\rceil}$, the regret can be decomposed as

$$\begin{aligned} \mathbb{E}[\text{D-Reg}(\mathbf{v}_{1:T})] \leq 3H \cdot \mathbb{E}&\left[\sum_{i=1}^{\lceil |S|/\Delta\rceil} \sum_{t\in\Delta_i} \widetilde{f}_t(\mathbf{w}_t(\delta_i,\eta_i)) - \widetilde{f}_t(\mathbf{v}_t)\right] \\ &+ \lambda\mathbb{E}\left[\sum_{i=1}^{\lceil |S|/\Delta\rceil} \sum_{t\in\Delta_i} \|\mathbf{w}_t(\delta_i,\eta_i) - \mathbf{w}_{t-1}(\delta_i,\eta_i)\|_2\right] + \frac{\mathbb{E}[|S|]}{\Delta} + \lambda P_T, \end{aligned}$$

where $\lambda = L(H+1)^2/2$ and $\{\mathbf{w}_t(\delta_i,\eta_i)\}_{t\in\Delta_i}$ is the decision sequence produced with perturbation magnitude δ_i and step size η_i in episode Δ_i. The $\mathbb{E}[|S|]/\Delta$ term above is due to the burn-in loss of the restart strategy. Note that the first two terms are the dynamic regret with switching loss of our algorithm, which is in general hard to analyze due to the two-layer online ensemble structure. We deal with this through a novel regret decomposition:

$$3H \cdot \mathbb{E}\left[\sum_{i=1}^{\lceil |S|/\Delta\rceil} \sum_{t\in\Delta_i} \widetilde{f}_t(\mathbf{w}_t(\delta_i,\eta_i)) - \widetilde{f}_t(\mathbf{v}_t)\right] + \lambda\mathbb{E}\left[\sum_{i=1}^{\lceil |S|/\Delta\rceil} \sum_{t\in\Delta_i} \|\mathbf{w}_t(\delta_i,\eta_i) - \mathbf{w}_{t-1}(\delta_i,\eta_i)\|_2\right]$$

$$
\begin{aligned}
&\leq 3H \cdot \mathbb{E}\left[\sum_{i=1}^{\lceil |S|/\Delta \rceil} \sum_{t\in\Delta_i} \widetilde{f}_t(\mathbf{w}_t(\delta_i,\eta_i)) - \widetilde{f}_t(\mathbf{w}_t(\delta^\star,\eta^\star)) + \widetilde{f}_t(\mathbf{w}_t(\delta^\star,\eta^\star)) - \widetilde{f}_t(\mathbf{v}_t)\right] \\
&\qquad + \lambda\mathbb{E}\left[\sum_{i=1}^{\lceil |S|/\Delta \rceil} \sum_{t\in\Delta_i} \|\mathbf{w}_t(\delta_i,\eta_i) - \mathbf{w}_{t-1}(\delta_i,\eta_i)\|_2\right] \\
&= \underbrace{\mathbb{E}\left[\sum_{i=1}^{\lceil |S|/\Delta \rceil} \sum_{t\in\Delta_i} \left(3H\widetilde{f}_t(\mathbf{w}_t(\delta_i,\eta_i) + \lambda\|\mathbf{w}_t(\delta_i,\eta_i) - \mathbf{w}_{t-1}(\delta_i,\eta_i)\|_2\right)\right]}_{\text{TERM (A-I)}} \\
&\qquad \underbrace{- \mathbb{E}\left[\sum_{i=1}^{\lceil |S|/\Delta \rceil} \sum_{t\in\Delta_i} \left(3H\widetilde{f}_t(\mathbf{w}_t(\delta^\star,\eta^\star) + \lambda\|\mathbf{w}_t(\delta^\star,\eta^\star) - \mathbf{w}_{t-1}(\delta^\star,\eta^\star)\|_2\right)\right]}_{\text{TERM (A-II)}} \\
&\quad \underbrace{+ 3H \cdot \mathbb{E}\left[\sum_{i=1}^{\lceil |S|/\Delta \rceil} \sum_{t\in\Delta_i} \widetilde{f}_t(\mathbf{w}_t(\delta^\star,\eta^\star)) - \widetilde{f}_t(\mathbf{v}_t) + \lambda \sum_{i=1}^{\lceil |S|/\Delta \rceil} \sum_{t\in\Delta_i} \|\mathbf{w}_t(\delta^\star,\eta^\star) - \mathbf{w}_{t-1}(\delta^\star,\eta^\star)\|_2\right]}_{\text{TERM (B)}},
\end{aligned}
$$

where $\delta^\star, \eta^\star$ denote the optimal perturbation magnitude and step size, which are unknown to the algorithm. By defining a surrogate loss for the meta learner as

$$
\ell_i(\delta_i,\eta_i) \triangleq 3H \sum_{t\in\Delta_i} \widetilde{f}_t(\mathbf{w}_t(\delta_i,\eta_i)) + \lambda \sum_{t\in\Delta_i} \|\mathbf{w}_t(\delta_i,\eta_i) - \mathbf{w}_{t-1}(\delta_i,\eta_i)\|_2 , \tag{B.14}
$$

TERM (A-I) along with TERM (A-II) can be rewritten as

$$
\text{TERM (A)} \triangleq \mathbb{E}\left[\sum_{i=1}^{\lceil |S|/\Delta \rceil} \ell_i((\delta_i,\eta_i)) - \sum_{i=1}^{\lceil |S|/\Delta \rceil} \ell_i((\delta^\star,\eta^\star))\right].
$$

Notice that

- TERM (A) represents the *meta-regret*: the static regret of a Multi-Armed Bandit problem, where each arm is a tuple of perturbation magnitude and step size, and its loss is the cumulative loss defined in (4.9). It measures the performance of the parameters chosen by our algorithm against that of the best one in hindsight.

- TERM (B) represents the *base-regret*: the dynamic regret of the best base learner with switching cost, compared to a sequence of time-varying comparators $\mathbf{v}_{1:T}$.

First we investigate the meta-regret. The lemma below states the regret bound of Exp3.

Lemma 6 (Theorem 3.1 of Auer et al. (2002))**.** *With the optimal tuning, Exp3 ensures* $\mathbb{E}[R_T] \leq 2\sqrt{TN\log N}$*, where* N *denotes the number of arms.*

The above lemma gives the meta-regret an upper bound of

$$\textsc{term}\,(\textsc{a}) \le 2L_{\max}\sqrt{\frac{\mathbb{E}[|S|]}{\Delta}N\log N} = 2(C_f+2\lambda R)\sqrt{\frac{T\Delta\log T\log\log T}{H}}, \tag{B.15}$$

where $L_{\max} \le \Delta(C_f + 2\lambda R)$ is the maximum value of the surrogate loss (B.14). Plugging $\delta = \delta^\star, \eta = \eta^\star, T = \Delta$ into the regret of the base algorithm (B.6), the base-regret satisfies

$$\begin{aligned}
\textsc{term}\,(\textsc{b}) &= \mathbb{E}\left[3H\sum_{i=1}^{\lceil|S|/\Delta\rceil}\sum_{t\in\Delta_i}\widetilde{f}_t(\mathbf{w}_t(\delta^\star,\eta^\star)) - \widetilde{f}_t(\mathbf{v}_t) + \lambda\sum_{t\in\Delta_i}\|\mathbf{w}_t(\delta^\star,\eta^\star) - \mathbf{w}_{t-1}(\delta^\star,\eta^\star)\|_2\right] \\
&\le \mathbb{E}\left[\sum_{i=1}^{\lceil|S|/\Delta\rceil}\frac{3H(7R^2+RP_{\Delta_i})}{4\eta^\star} + \frac{3d^2C_f^2\eta^\star\Delta}{2\delta^2} + \frac{6d\varepsilon R\Delta}{\delta} + \frac{\lambda dC_f\eta\Delta}{H\delta} + L_{\text{eff}}\delta\Delta\right] \\
&\le \frac{\frac{21R^2T}{\Delta}+3HRP_T}{4\eta^\star} + \frac{3d^2C_f^2\eta^\star T}{2\delta^2} + \frac{6d\varepsilon RT}{\delta} + \frac{\lambda dC_f\eta T}{H\delta} + L_{\text{eff}}\delta T,
\end{aligned}$$

where P_{Δ_i} denotes the path length in the i-th episode. Plugging in $\textsc{term}\,(\textsc{a})$ and $\textsc{term}\,(\textsc{b})$, the dynamic regret of Pronc-Select follows,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=H}^{T} f_t(\mathbf{w}_{t-H:t}) - \sum_{t=H}^{T} f_t(\mathbf{v}_{t-H:t})\right] \\
&\le 2(C_f+2\lambda R)\sqrt{\frac{T\Delta\log T\log\log T}{H}} + \frac{\frac{21R^2T}{\Delta}+3HRP_T}{4\eta^\star} + \frac{3d^2C_f^2\eta^\star T}{2\delta^2} \\
&\quad + \frac{6d\varepsilon RT}{\delta} + \frac{\lambda dC_f\eta T}{H\delta} + L_{\text{eff}}\delta T + \frac{T}{H\Delta} + \lambda P_T.
\end{aligned} \tag{B.16}$$

Finally, by setting the parameters $\eta^\star, \delta^\star, \Delta$ as

$$\eta^\star = \sqrt{\frac{3}{L_{\text{eff}}dC_f}}\left(\frac{H\left(7R^2T^{1/3}+RP_T\right)}{2T}\right)^{3/4}, \quad \delta^\star = \left(\frac{3d^2C_f^2\eta^\star}{L_{\text{eff}}}\right)^{1/3}, \quad \Delta = T^{2/3},$$

we obtain the final dynamic regret guarantee of $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/4} + T^{5/6})$. ■

B.5 Proof of Theorem 4

Proof In known systems, we begin with the following regret decomposition:

$$\begin{aligned}
&\mathbb{E}\left[\text{D-Reg}(\pi_{1:T})\right] = \mathbb{E}\left[\sum_{t=1}^{T} c_t\left(\mathbf{y}_t,\mathbf{u}_t\right) - \sum_{t=1}^{T} c_t\left(\mathbf{y}_t^{\pi_t},\mathbf{u}_t^{\pi_t}\right)\right] \\
&= \underbrace{\sum_{t=1}^{T_1} c_t\left(\mathbf{y}_t,\mathbf{u}_t\right)}_{\textsc{term}\,(\textsc{a})} + \underbrace{\sum_{t=T_1+1}^{T} c_t\left(\mathbf{y}_t,\mathbf{u}_t\right) - f_t\left(M_{t-H:t}\right)}_{\textsc{term}\,(\textsc{b})} + \underbrace{\mathbb{E}\left[\sum_{t=T_1+1}^{T} f_t\left(M_{t-H:t}\right) - f_t\left(M^\star_{t-H:t}\right)\right]}_{\textsc{term}\,(\textsc{c})}
\end{aligned}$$

$$+ \underbrace{\sum_{t=T_1+1}^{T} \widetilde{f}_t\left(M^\star_{t-H:t}\right) - c_t\left(\mathbf{y}_t(M^\star_{1:t-1}), \mathbf{u}_t(M^\star_{1:t})\right)}_{\text{TERM (D)}},$$

where $T_1 \triangleq 2H$ denotes the burn-in period and $f_{1:T}$ are the truncation cost functions (see Definition 3). Using Lemma 10, we have TERM (A) $\leq \mathcal{O}(H)$. With the truncation lemma (see Lemma 1), and by setting $H = \Theta(\log T)$, we have TERM (A) + TERM (B) + TERM (D) $\leq \widetilde{\mathcal{O}}(1)$.

TERM (C) is actually the dynamic regret of bandit convex optimization with memory and inexact feedback over the policy parameter domain $\mathcal{M}$. Since the truncated cost functions $f_{1:T}$ are convex and Lipschitz continuous, inherited from the control cost functions $c_{1:T}$ (see Lemma 11), we leverage Theorem 2 and Theorem 3 to bound TERM (C) respectively. For PRONC-COMBINE, TERM (C) $\leq \widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/2})$, so the overall regret is of order $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/2})$. For PRONC-SELECT, TERM (C) $\leq \widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/4} + T^{5/6})$, as a result the overall regret is $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/4} + T^{5/6})$. ■

B.6 Proof of Theorem 5

Proof In unknown systems, the regret can be decomposed as follows:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} c_t(\mathbf{y}_t, \mathbf{u}_t) - c_t(\mathbf{y}_t^{\pi_t}, \mathbf{u}_t^{\pi_t})\right] = \mathbb{E}\left[\sum_{t=1}^{T} c_t(\mathbf{y}_t, \mathbf{u}_t) - c_t(\mathbf{y}_t(M^\star_{0:t-1}), \mathbf{u}_t(M^\star_t))\right] \\
\leq\ & \underbrace{\sum_{t=1}^{T_2} c_t(\mathbf{y}_t, \mathbf{u}_t)}_{\text{TERM (A)}} + \underbrace{\sum_{t=T_2+1}^{T} c_t(\mathbf{y}_t, \mathbf{u}_t) - f_t(M_{t-H:t}|\widehat{G}, \widehat{\mathbf{y}}^{\text{nat}}_{1:t})}_{\text{TERM (B)}} \\
&\quad \underbrace{+ \mathbb{E}\left[\sum_{t=T_2+1}^{T} f_t(M_{t-H:t}|\widehat{G}, \widehat{\mathbf{y}}^{\text{nat}}_{1:t}) - \widetilde{f}_t(M^\star_{t-H:t}|\widehat{G}, \widehat{\mathbf{y}}^{\text{nat}}_{1:t})\right]}_{\text{TERM (C)}} \\
&\quad \underbrace{+ \sum_{t=T_2+1}^{T} f_t(M^\star_{t-H:t}|\widehat{G}, \widehat{\mathbf{y}}^{\text{nat}}_{1:t}) - c_t(\mathbf{y}_t(M^\star_{0:t-1}), \mathbf{u}_t(M^\star_t))}_{\text{TERM (D)}},
\end{aligned}$$

where $T_2 \triangleq m + 2H + T_0$ denotes the burn-in period. Leveraging Lemma 13 to bound the burn-in cost (TERM (A)), and using Lemma 12 and Lemma 14 to analyze the truncation error (TERM (B) and TERM (D)), it holds that

$$\text{TERM (A)} \leq \mathcal{O}(T_0), \quad \text{TERM (B)} + \text{TERM (D)} \leq \mathcal{O}\left(\frac{T}{\sqrt{T_0}}\right),$$

Pronc-Combine. Note that in unknown systems, the truncation error ε is no more $\mathcal{O}(T^{-1})$ but $\mathcal{O}(T_0^{-1/2})$ (see Lemma 12 and Lemma 14). Leveraging (B.11) with $\varepsilon \leq \mathcal{O}(T_0^{-1/2})$,

along with the other three terms in the regret decomposition, we have

$$\mathbb{E}\left[\text{D-Reg}(\pi_{1:T})\right] = \mathcal{O}\left(T_0 + \frac{1+P_T}{\eta^\star} + \frac{\eta^\star T}{\delta^{\star 2}} + \frac{T}{\delta^\star\sqrt{T_0}} + \delta^\star T + P_T\right).$$

By setting memory length H, estimation round T_0, perturbation $\delta^\star$ and step size $\eta^\star$ as

$$H = \Theta(\log T), \quad T_0 = \mathcal{O}(T^{4/5}), \quad \delta^\star = \mathcal{O}(T^{-1/5}), \quad \eta^\star = \delta^\star\left(\frac{1+P_T}{T}\right)^{1/2},$$

we obtain an $\widetilde{\mathcal{O}}\left(T^{4/5}(1+P_T)^{1/2}\right)$ regret guarantee.

Pronc-Select. The difference of analysis only lies in term (c). Leveraging (B.16) with $\varepsilon \le \mathcal{O}(T_0^{-1/2})$, along with the other three terms in the regret decomposition, it holds that

$$\mathbb{E}\left[\text{D-Reg}(\pi_{1:T})\right] \le \mathcal{O}\left(T_0 + \frac{T}{\delta^\star\sqrt{T_0}} + \frac{\frac{T}{\Delta}+P_T}{\eta^\star} + \frac{\eta^\star T}{\delta^{\star 2}} + \sqrt{T\Delta} + \frac{T}{\Delta} + \delta^\star T\right).$$

By setting the memory length H, the restarting period Δ, the estimation round T_0, the perturbation magnitude $\delta^\star$ and the step size $\eta^\star$ optimally as

$$H = \Theta(\log T), \quad \Delta = \mathcal{O}(T^{2/3}), \quad T_0 = \mathcal{O}(T^{4/5}),$$
$$\delta^\star = \left(\left(\frac{1}{\Delta}+\frac{P_T}{T}\right)^{1/2} + T_0^{-1/2}\right)^{1/2}, \ \eta^\star = \delta^\star\left(\frac{1}{\Delta}+\frac{P_T}{T}\right)^{1/2},$$

we obtain an $\widetilde{\mathcal{O}}(T^{3/4}(1+P_T)^{1/4} + T^{5/6})$ expected dynamic regret guarantee. Note that in the unknown setting, the relationship between the optimal step size $\eta^\star$ and the optimal perturbation magnitude $\delta^\star$ does not exist anymore, thus the number of the base learners increases from $\mathcal{O}(\log T)$ to $\mathcal{O}((\log T)^2)$ in this case. ■

Appendix C. Supporting Lemmas

In this section, we list some basic supporting lemmas often used in online non-stochastic control and online learning.

- Lemma 7 presents the dynamic regret of OGD.
- Lemma 8 describes the basic property of the mini-batching approach.
- Lemma 9 builds a relationship between the regret over the whole time horizon T and the mini-batching update set S.
- Lemma 10 captures the cost before the algorithm attains meaningful regret guarantees (referred to as burn-in cost).
- Lemma 11 bounds the domain diameter of DRP parameters and the Lipschitz constant of the truncate cost functions.

- Lemma 12 shows the relationship between the estimation rounds T_0 and the estimation accuracy ε_G;
- Lemma 13 gives an upper bound of the cost incurred in unknown systems when the algorithm is not running due to insufficient memory, namely, burn-in cost;
- Lemma 14 depicts the gap between the true cost and the truncated cost on the estimated system transition matrices. This gap consists of not only the truncation error but also the approximation error between the truncated cost of the true system transition and the estimated one.

Lemma 7 (Dynamic Regret of OGD (Zinkevich, 2003)). *Consider the online gradient descent $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}\left[\mathbf{x}_t - \eta \nabla f_t(\mathbf{x}_t)\right]$. Suppose the feasible domain $\mathcal{K}$ is bounded, i.e., $\|\mathbf{x} - \mathbf{y}\|_2 \leq D$ for any $\mathbf{x}, \mathbf{y} \in \mathcal{K}$, and meanwhile, the gradients of the online functions are bounded, i.e., $\|\nabla f_t(\cdot)\|_2 \leq G$ for any $t \in [T]$, then the dynamic regret 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 $\mathbf{u}_{1:T} \in \mathcal{K}^T$, whose path length is $P_T \triangleq \sum_{t=2}^{T} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2$.

Lemma 8 (Lemma 10 of Cassel and Koren (2020)). *Suppose random bits $b_{1:T}$ are drawn in advance. Let $t_0 = 0$ and $t_i = \min\{t \geq t_{i-1} + H \mid b_t \prod_{i=1}^{H-1}(1 - b_{t-i}) = 1\}$ for $i \geq 1$. Denoting by $S = \{t_i | H \leq t_i \leq T\}$ the rounds indicating the mini-batching updates, we have (i) $|S| \leq \lfloor T/H \rfloor$; and (ii) $\mathbb{E}[t_i - t_{i-1}] = \mathbb{E}[t_1] \leq 3H, \forall i \in [|S|]$.*

Lemma 9 (Bridging Regrets over Two Time Horizons (Cassel and Koren, 2020, Lemma 11)). *Suppose mini-batching method chooses random bits $b_{1:T}$ from Bernoulli$(1/H)$ independently, then for any function sequence $f_{H:T}$, decision sequence $\mathbf{w}_{H:T}$ and comparator sequence $\mathbf{v}_{H:T}$, it holds that*

$$\mathbb{E}\left[\sum_{t=H}^{T} f_t(\mathbf{w}_t) - \sum_{t=H}^{T} f_t(\mathbf{v}_t)\right] \leq 3H \cdot \mathbb{E}\left[\sum_{t \in S} f_t(\mathbf{w}_t) - \sum_{t \in S} f_t(\mathbf{v}_t)\right],$$

where S stands for the set containing the rounds of mini-batching updates.

Lemma 10 (Burn-in Cost of Known Systems (Simchowitz et al., 2020, Lemma C.2)). *Under Assumption 2, the burn-in cost of the algorithm can be bounded by $4L_c(m+H)R_G^2 R_{\mathcal{M}}^2 R_{\text{nat}}^2$.*

Lemma 11 (Lipschitz/Diameter Bounds of Known Systems (Simchowitz et al., 2020, Lemma C.5)). *Under Assumption 2, denote by $D_M = \max_{M_1, M_2 \in \mathcal{M}} \|M_1 - M_2\|_{\text{F}}$, L_f the Lipschitz constant of the surrogate cost f_t, then it holds that*

$$D_M \leq 2\sqrt{d_{\min}} R_{\mathcal{M}}, \; L_f = 2L_c \sqrt{m} R_G^2 R_{\mathcal{M}} R_{\text{nat}}^2,$$

where $d_{\min} = \min\{d_y, d_u\}$.

Lemma 12 (Guarantee for Algorithm 5 (Simchowitz et al., 2020, Theorem 7)). *Let $\delta \in (e^{-T}, T^{-1})$, $T_0, d_u \le T$, and $\psi_G(H) \le \frac{1}{10}$. For universal constants c, C_{est}, define*

$$\varepsilon_G(T_0, \delta) = C_{est}\frac{h^2 R_{\text{nat}}}{\sqrt{T_0}}C_\delta, \qquad \text{where } C_\delta \triangleq \sqrt{d_{\max} + \log\tfrac{1}{\delta} + \log(1 + R_{\text{nat}})}.$$

and suppose that $T_0 \ge ch^4 C_\delta^4 R_{\mathcal{M}}^2 R_G^2$. Then with probability $1 - \delta - T_0^{-\log^2 T_0}$, Algorithm 5 satisfies the following bounds

(i) For all $t \in [T_0]$, $\|\mathbf{u}_t\|_2 \le R_{u,est}(\delta) \triangleq 5\sqrt{d_u + 2\log(3/\delta)}$;

(ii) The estimation error is bounded as

$$\left\|\widehat{G} - G\right\|_{\ell_1,\text{op}} \le \|\widehat{G}^{[0:h]} - G^{[0:h]}\|_{\ell_1,\text{op}} + R_{u,est}\psi_G(H) \le \varepsilon_G(T_0, \delta) \le \frac{1}{2}\max\{R_{\mathcal{M}}R_G, R_{u,est}\}.$$

Lemma 13 (Burn-in Cost of Unknown Systems (Simchowitz et al., 2020, Lemma D.3)). *Under Assumption 2, suppose Lemma 12 holds, and define $\bar{R}_u(\delta) \triangleq 2\max\{R_{u,est}(\delta), R_{\mathcal{M}}R_{\text{nat}}\}$, we have that $\sum_{t=1}^{T'} c_t(\mathbf{y}_t, \mathbf{u}_t) \le 4L_c T' R_G^2 \bar{R}_u^2$, where T' denotes the burn-in period.*

Lemma 14 (Approximation Lemma for Unknown System (Simchowitz et al., 2020, Lemma D.5)). *Under Assumption 2, suppose Lemma 12 holds, then it holds that*

$$\sum_{t=T'+1}^{T} c_t(\mathbf{y}_t, \mathbf{u}_t) - \sum_{t=T'+1}^{T} f_t(M_{t-1-H:t}|\widehat{G}, \widehat{\mathbf{y}}_{1:t}^{nat}) \lesssim L_c T R_G R_{\mathcal{M}}^2 R_{\text{nat}}^2 \varepsilon_G,$$

where $f_t(M_{t-1-H:t}|\widehat{G}, \widehat{\mathbf{y}}_{1:t}^{nat})$ denotes the truncated cost based on the estimated $\widehat{G}$ and $\widehat{\mathbf{y}}_{1:t}^{nat}$.

References

Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret bounds for the adaptive control of linear quadratic systems. In *Proceedings of the 24th Annual Conference on Learning Theory (COLT)*, pages 1–26, 2011.

Alekh Agarwal, Haipeng Luo, Behnam Neyshabur, and Robert E. Schapire. Corralling a band of bandit algorithms. In *Proceedings of the 30th Conference on Learning Theory (COLT)*, pages 12–38, 2017.

Naman Agarwal, Brian Bullins, Elad Hazan, Sham M. Kakade, and Karan Singh. Online control with adversarial disturbances. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 111–119, 2019.

Oren Anava, Elad Hazan, and Shie Mannor. Online learning for adversaries with memory: Price of past mistakes. In *Advances in Neural Information Processing Systems 28 (NIPS)*, pages 784–792, 2015.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

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

Dheeraj Baby and Yu-Xiang Wang. Optimal dynamic regret in LQR control. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pages 24879–24892, 2022.

Yong Bai, Yu-Jie Zhang, Peng Zhao, Masashi Sugiyama, and Zhi-Hua Zhou. Adapting to online label shift with provable guarantees. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pages 29960–29974, 2022.

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

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *ArXiv preprint*, arXiv:1606.01540, 2016.

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

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with GPT-4. *ArXiv preprint*, arXiv:2303.12712, 2023.

Xin-Qiang Cai, Peng Zhao, Kai Ming Ting, Xin Mu, and Yuan Jiang. Nearest neighbor ensembles: An effective method for difficult problems in streaming classification with emerging new classes. In *Proceedings of the 19th International Conference on Data Mining (ICDM)*, pages 970–975, 2019.

Asaf Cassel and Tomer Koren. Bandit linear control. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pages 8872–8882, 2020.

Asaf Cassel, Alon Cohen, and Tomer Koren. Efficient online linear control with stochastic convex costs and unknown dynamics. In *Proceedings of the 35th Conference on Learning Theory (COLT)*, pages 3589–3604, 2022a.

Asaf Cassel, Alon Cohen, and Tomer Koren. Rate-optimal online convex optimization in adaptive linear control. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pages 7410–7422, 2022b.

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

Nicolò Cesa-Bianchi, Pierre Gaillard, Gábor Lugosi, and Gilles Stoltz. Mirror descent meets fixed share (and feels no regret). In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 989–997, 2012.

Nicolò Cesa-Bianchi, Ofer Dekel, and Ohad Shamir. Online learning with switching costs and other adaptive adversaries. In *Advances in Neural Information Processing Systems 26 (NIPS)*, pages 1160–1168, 2013.

Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Learning to optimize under non-stationarity. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1079–1087, 2019.

Alon Cohen, Avinatan Hasidim, Tomer Koren, Nevena Lazic, Yishay Mansour, and Kunal Talwar. Online linear quadratic control. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 1029–1038, 2018.

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.

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, Ambuj Tewari, and Raman Arora. Online bandit learning against an adaptive adversary: from regret to policy regret. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 1747–1754, 2012.

Ofer Dekel, Jian Ding, Tomer Koren, and Yuval Peres. Bandits with switching costs: $T^{2/3}$ regret. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC)*, pages 459–467, 2014.

Claude-Nicolas Fiechter. PAC adaptive control of linear systems. In *Proceedings of the 10th Annual Conference on Computational Learning Theory (COLT)*, pages 72–80, 1997.

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.

Dylan J. Foster, Akshay Krishnamurthy, and Haipeng Luo. Open problem: Model selection for contextual bandits. In *Proceedings of the 33rd Annual Conference Computational Learning Theory (COLT)*, pages 3842–3846, 2020.

Dylan J. Foster, Sham M. Kakade, Jian Qian, and Alexander Rakhlin. The statistical complexity of interactive decision making. *ArXiv preprint*, arXiv:2112.13487, 2021.

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

Jim Gao. Machine learning applications for data center optimization. *Google White Paper*, 2014.

Sascha Geulen, Berthold Vöcking, and Melanie Winkler. Regret minimization for online buffering problems using the weighted majority algorithm. In *Proceedings of the 23rd Conference on Learning Theory (COLT)*, pages 132–143, 2010.

Gautam Goel and Babak Hassibi. Competitive control. *IEEE Transactions on Automatic Control*, 68(9):5162–5173, 2023.

Gautam Goel, Naman Agarwal, Karan Singh, and Elad Hazan. Best of both worlds in online control: Competitive ratio and policy regret. In *Proceedings of the 5th Learning for Dynamics and Control (L4DC)*, pages 1345–1356, 2023.

Paula Gradu, John Hallman, and Elad Hazan. Non-stochastic control with bandit feedback. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pages 10764–10774, 2020.

Paula Gradu, Elad Hazan, and Edgar Minasyan. Adaptive regret for control of time-varying dynamics. In *Proceedings of the 5th Learning for Dynamics and Control (L4DC)*, pages 560–572, 2023.

Lei Guo. *Time-Varying Stochastic Systems, Stability and Adaptive Theory*. Science Press, second edition, 2020.

Lei Guo and Lennart Ljung. Performance analysis of general tracking algorithms. *IEEE Transactions on Automatic Control*, 40(8):1388–1402, 1995.

András György and Gergely Neu. Near-optimal rates for limited-delay universal lossy source coding. *IEEE Transactions on Information Theory*, 60(5):2823–2834, 2014.

András György and Csaba Szepesvári. Shifting regret, mirror descent, and matrices. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 2943–2951, 2016.

J Harold, G Kushner, and George Yin. Stochastic approximation and recursive algorithm and applications. *Application of Mathematics*, 35(10), 1997.

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

Elad Hazan. Lecture Notes: Online and Nonstochastic Control Theory, 2020.

Elad Hazan, Adam Kalai, Satyen Kale, and Amit Agarwal. Logarithmic regret algorithms for online convex optimization. In *Proceedings of the 19th Annual Conference on Learning Theory (COLT)*, volume 4005 of *Lecture Notes in Computer Science*, pages 499–513. Springer, 2006.

Elad Hazan, Sham M. Kakade, and Karan Singh. The nonstochastic control problem. In *Proceedings of the 31st International Conference on Algorithmic Learning Theory (ALT)*, pages 408–421, 2020.

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

Bo-Jian Hou, Lijun Zhang, and Zhi-Hua Zhou. Learning with feature evolvable streams. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 1417–1427, 2017.

Chenping Hou and Zhi-Hua Zhou. One-pass learning with incremental and decremental features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(11):2776–2792, 2018.

Andrew Jacobsen and Ashok Cutkosky. Parameter-free mirror descent. In *Proceedings of the 35th Conference on Learning Theory (COLT)*, pages 4160–4211, 2022.

Donald E Kirk. *Optimal Control Theory: An Introduction*. Courier Corporation, 2004.

Tor Lattimore. Improved regret for zeroth-order adversarial bandit convex optimisation. *Mathematical Statistics and Learning*, 2(3):311–334, 2020.

Nevena Lazic, Craig Boutilier, Tyler Lu, Eehern Wong, Binz Roy, M. K. Ryu, and Greg Imwalle. Data center cooling using model-predictive control. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 3818–3827, 2018.

Lennart Ljung and Torsten Söderström. *Theory and Practice of Recursive Identification*. MIT press, 1983.

Haipeng Luo and Robert E. Schapire. Achieving all with no parameters: AdaNormalHedge. In *Proceedings of the 28th Annual Conference Computational Learning Theory (COLT)*, pages 1286–1304, 2015.

Haipeng Luo, Mengxiao Zhang, Peng Zhao, and Zhi-Hua Zhou. Corralling a larger band of bandits: A case study on switching regret for linear bandits. In *Proceedings of the 35th Conference on Learning Theory (COLT)*, pages 3635–3684, 2022.

Jason R Marden and Jeff S Shamma. Game theory and control. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:105–134, 2018.

Neri Merhav, Erik Neat, Gadiel Seroussi, and Marcelo J. Weinberger. On sequential strategies for loss functions with memory. *IEEE Transactions on Information Theory*, 48(7): 1947–1958, 2002.

Xin Mu, Kai Ming Ting, and Zhi-Hua Zhou. Classification under streaming emerging new classes: A solution using completely-random trees. *IEEE Transactions on Knowledge and Data Engineering*, 29(8):1605–1618, 2017.

Gergely Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. In *Advances in Neural Information Processing Systems 28 (NIPS)*, pages 3168–3176, 2015.

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 Computational Learning Theory (COLT)*, pages 3–24, 2013.

Guanya Shi, Yiheng Lin, Soon-Jo Chung, Yisong Yue, and Adam Wierman. Online optimization with memory and competitive control. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pages 20636–20647, 2020.

Max Simchowitz. Making non-stochastic control (almost) as easy as stochastic. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pages 18318–18329, 2020.

Max Simchowitz, Karan Singh, and Elad Hazan. Improper learning for non-stochastic control. In *Proceedings of 33rd Conference on Learning Theory (COLT)*, pages 3320–3436, 2020.

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

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

Hamdy A Taha. *Operations Research: An Introduction*, volume 7. Prentice hall Upper Saddle River, NJ, 2003.

Han Wang, Yang Yu, and Yuan Jiang. Review of the progress of communication-based multi-agent reinforcement learning. *Science China Information Sciences*, 52, 2022.

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.

Yu-Hu Yan, Peng Zhao, and Zhi-Hua Zhou. Fast rates in time-varying strongly monotone games. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 39138–39164, 2023.

Dante Youla, Hamid Jabr, and Jr Bongiorno. Modern wiener-hopf design of optimal controllers–part ii: The multivariable case. *IEEE Transactions on Automatic Control*, 21(3): 319–338, 1976.

Jianjun Yuan and Andrew G. Lamperski. Trading-off static and dynamic regret in online least-squares and beyond. In *Proceedings of The 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 6712–6719, 2020.

Lijun Zhang. Online learning in changing environments. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5178–5182, 2020.

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, 2018.

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, Wei Jiang, Shiyin Lu, and Tianbao Yang. Revisiting smoothed online learning. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pages 13599–13612, 2021.

Mengxiao Zhang, Peng Zhao, Haipeng Luo, and Zhi-Hua Zhou. No-regret learning in time-varying zero-sum games. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 26772–26808, 2022a.

Yu-Jie Zhang, Zhen-Yu Zhang, Peng Zhao, and Masashi Sugiyama. Adapting to continuous covariate shift via online density ratio estimation. *ArXiv preprint*, arXiv:2302.02552, 2023.

Zhiyu Zhang, Ashok Cutkosky, and Ioannis Ch. Paschalidis. Adversarial tracking control via strongly adaptive online learning with memory. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 8458–8492, 2022b.

Peng Zhao. *Online Ensemble Theories and Methods for Robust Online Learning*. PhD thesis, Nanjing University, Nanjing, China, 2021. Advisor: Zhi-Hua Zhou.

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)*, pages 48–59, 2021.

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

Peng Zhao, Guanghui Wang, Lijun Zhang, and Zhi-Hua Zhou. Bandit convex optimization in non-stationary environments. *Journal of Machine Learning Research*, 22(125):1 – 45, 2021a.

Peng Zhao, Xinqiang Wang, Siyu Xie, Lei Guo, and Zhi-Hua Zhou. Distribution-free one-pass learning. *IEEE Transactions on Knowledge and Data Engineering*, 33(3):951–963, 2021b.

Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Adaptivity and non-stationarity: Problem-dependent dynamic regret for online convex optimization. *ArXiv preprint*, arXiv:2112.14368, 2021c.

Peng Zhao, Long-Fei Li, and Zhi-Hua Zhou. Dynamic regret of online Markov decision processes. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 26865–26894, 2022a.

Peng Zhao, Yu-Xiang Wang, and Zhi-Hua Zhou. Non-stationary online learning with memory and non-stochastic control. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2101–2133, 2022b.

Peng Zhao, Yan-Feng Xie, Lijun Zhang, and Zhi-Hua Zhou. Efficient methods for non-stationary online learning. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pages 11573–11585, 2022c.

Peng Zhao, Yu-Hu Yan, Yu-Xiang Wang, and Zhi-Hua Zhou. Non-stationary online learning with memory and non-stochastic control. *Journal of Machine Learning Research*, 24(206): 1–70, 2023.

Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. CRC press, 2012.

Zhi-Hua Zhou. Learnware: on the future of machine learning. *Frontiers in Computer Science*, 10(4):589–590, 2016.

Zhi-Hua Zhou. Open-environment machine learning. *National Science Review*, 9(8): nwac123, 2022a.

Zhi-Hua Zhou. Rehearsal: Learning from prediction to decision. *Frontiers of Computer Science*, 16(4):164352, 2022b.

Zhi-Hua Zhou and Zhi-Hao Tan. Learnware: Small models do big. *Science China Information Sciences*, 2023. doi: 10.1007/s11432-023-3823-6.

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.