Learning in the Presence of Adaptive Behavior

William Brown

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2024

Abstract

Learning in the Presence of Adaptive Behavior

William Brown

Algorithms for repeated (or "online") decision-making are predominantly studied under the assumption that feedback is either statistical (determined by fixed probability distributions) or adversarial (changing over time in a potentially worst-case manner). Both of these assumptions ignore a phenomenon commonly present in repeated interactions with other agents, in which the space of our possible future outcomes is shaped in a structured and potentially *predictable* manner by our history of prior decisions.

In this thesis, we consider online decision problems where the feedback model is *adaptive* rather than purely statistical or adversarial. One such example is a repeated game played against an opponent who uses a learning algorithm of their own; here, we give a characterization of possible outcome spaces which unifies disparate equilibrium notions, and serves as a basis for designing new algorithms. We then consider the task of providing recommendations to an agent whose preferences adapt based on the recommendation history, where we explore algorithmic tradeoffs in terms of the structure of this adaptivity pattern. We conclude by offering a general framework and algorithmic toolkit for approaching adaptive problems of this form.

Table of Contents

Acknowledgements

First and foremost, I am incredibly grateful for the guidance of my advisors, Tim Roughgarden and Christos Papadimitriou, during my time at Columbia. Tim offered countless invaluable perspectives on navigating the worlds of and beyond academic research, Christos provided sage inspiration to write bold introductions and to work on "the problems that keep me up at night", and together they gave me a tremendous amount of freedom to venture down various rabbit holes (fruitful or otherwise) throughout graduate school.

I would like to thank all of my past coauthors, notably Arpit Agarwal, Jon Schneider, and Kiran Vodrahalli, whose collaborations contribute to this thesis. I also thank my friends and colleagues in the Columbia Theory Group, at Morgan Stanley, and at Amazon, as well as Daniel Hsu and Christian Kroer for graciously offering to participate in my thesis committee.

I would additionally like to thank a number of mentors from before my time at Columbia, by whom my research journey has been substantially shaped. I owe an immense amount of gratitude to the dynamic duo of Michael Kearns and Aaron Roth, who guided my first forays into research at Penn and greatly inspired the way I have approached new problems ever since; to Sanjeev Khanna and Rajiv Gandhi, who fostered my initial excitement for theory and algorithms; and to Larry Rudolph and David Daly, for their supervision in formative summer internships.

Further, I thank my all of my friends outside of academia for putting up with my ramblings about machine learning over the years — you know who you are. I am also quite grateful for the company provided by my cat, Bingo, during late nights of writing. Last but certainly not least, I thank my parents and my brother for their unwavering support at every step along this journey.

Dedication

To my parents Megan and David, my brother Ian, and my cat Bingo.

Chapter 1: Introduction

1.1 Grappling with Adaptivity

Consider the following problems:

- Select videos to recommend to a user on a streaming platform.
- Choose a trading strategy to employ in an asset market.
- Determine optimal prices for commodity goods.
- Design selection criteria for access to desirable-but-scarce resources.

These problems are typically encountered not as isolated events but as part of ongoing repeated interactions with other agents, and algorithmic solutions are often desired for each. Indeed, there is a long tradition in algorithm design which considers repeated or *online* decision problems of this form. There, we are faced with a sequence of objective functions f_t (often "rewards" or "losses"), revealed to us (in whole or in part) only after we select our strategy x_t for the current round of interaction, and we aim to optimize our cumulative objective value $\sum_{t=1}^{T} f_t(x_t)$ over T rounds. For such problems, it is commonly assumed that objectives are either *statistical*, where f_t is drawn from a fixed distribution in each round, or *adversarial*, where f_t can change arbitrarily across rounds. However, for each of the problems mentioned above, neither assumption quite captures typical agent interaction dynamics. If the agents in these settings are strategic, rational, or otherwise *adaptive*, we should expect our resulting objectives f_t to shift over time in a manner which voids statistical assumptions, yet we should also expect these shifts to be *structured* rather than arbitrary or fully adversarial, as governed by the agents' own objectives.

It is also worth noting that the algorithmic methods employed in statistical and adversarial settings are often quite similar, resembling carefully-calibrated forms of gradient descent; in either

case, both settings overlook the possibility for feedback loops or *causality* in objectives, where each f_t may be shaped by our choice of strategies in previous rounds, and the performance criteria considered for these algorithms typically ignore any potential *counterfactual* objectives that we may have faced, had we chosen an alternate sequence of strategies. When objectives are shaped by repeated interactions with adaptive agents, this should prompt us to ask: *can we do better?*

This is the central question we address in this thesis. Concretely, we examine a number of repeated interaction domains where agent strategies may adapt over time as a function of our own. In each, we aim to characterize the space of *possible outcomes* for which our resulting objectives will apply, to design algorithms which *optimize* our objectives over the appropriate outcome space, and to identify potential *barriers* to further algorithmic improvements. Broadly speaking, each problem we consider concerns repeated decision-making over T rounds in the following form:

- We choose some *strategy* x_t in each round t;
- The *system* (of one or more agents) chooses a strategy y_t in a manner which may depend on our own sequence of strategies $\{x_1, \dots, x_t\}$;
- Our objective for the round is given by $f_t(x_t, y_t)$, depending potentially on both strategies.

We will aim to optimize our cumulative objective $\sum_{t=1}^{T} f_t(x_t, y_t)$ over the sequence of rounds, and the *complete* functions f_t will be presumed to follow either the statistical or adversarial assumptions mentioned previously. However, these assumptions will not apply for the *partial* functions $f_t(\cdot, y_t)$, owing to the adaptivity patterns of the system, and our algorithms will aim to *influence* the sequence of system strategies $\{y_1, \dots, y_t\}$ through the choices of our own. Often, we will also not know the precise manner in which y_t depends on $\{x_1, \dots, x_t\}$, and our approaches will require elements of *learning* via repeated interaction as well.

On the surface, if we view the system's strategy y_t as corresponding to some notion of "state", this general interaction structure somewhat resembles problems encountered in reinforcement learning or optimal control. However, the settings we are interested in will often feature both high-dimensional and continuous strategy spaces, as well as nonlinear *dynamics* governing the

updates to the system's strategy, which will preclude direct application of standard algorithmic techniques for optimization in tabular or linear Markov Decision Processes, or in linear dynamical systems. In contrast, our approaches will directly leverage properties which we often expect from "reasonable" adaptive agent behavior in the appropriate context (such as "recency bias", "preference for familiarity", or structural properties of online decision algorithms which agents may be using). In Chapter 4, we formalize this connection to dynamical systems and provide algorithms which can be applied to a broad range of problems involving adaptive agents, albeit leveraging a largely distinct set of techniques and assumptions to other recent works in online control.

As in many online learning problems, our aim will typically be to minimize our *regret* with respect to the optimal cumulative objective value obtained within some *target space*. However, in contrast to the standard "best fixed strategy" benchmark, we find that accommodating adaptivity in the manner we have described generally requires delicate choices for appropriate targets, which may depend intimately on the structure of agent adaptivity within the problems we encounter. We analyze our algorithms along a number of dimensions, namely: the *feasible* target set over which we can minimize regret, the *rate* at which we approach the optimal target value, the computational *efficiency* of our algorithms, and the *assumptions* needed to enable regret minimization. In several cases, we also identify *impossibility results* for further algorithmic improvements along these lines, highlighting the importance of the choice of targets and assumptions for problems of this form.

1.2 Reading this Thesis

The central results of this thesis are contained in Chapters 2, 3, and 4, with longer proofs deferred to the appendices. Each chapter is intended to be relatively self-contained, although Chapter 4 certainly benefits from the context provided by the preceding two. We briefly summarize the contributions of the thesis in Section 1.3. There, we refer to some pieces of relevant technical terminology in an informal manner as needed, with formal definitions and notation presented in Section 1.4 as well as the chapters themselves. Likewise, relevant prior work is discussed in each chapter, with key background included in Section 1.5 alongside pointers to useful reference texts.

1.3 Our Contributions

1.3.1 Reward vs. Regret

In Chapter 2, we consider the problem of optimizing cumulative rewards in a repeated (normal-form) game against an opponent who uses a learning algorithm of their own, alongside a number of more general questions related to tradeoffs between *reward* and *regret* in repeated gameplay between two agents. To facilitate our analysis, we introduce a notion of *generalized equilibrium* which allows for asymmetric regret constraints, and yields polytopes of feasible values for each agent and pair of regret constraints, where we show that any such equilibrium is reachable by a pair of algorithms which maintain their regret guarantees against arbitrary opponents. As a central example, we highlight the case where one agent uses a *no-swap-regret* algorithm and the other may use any algorithm, which need not possess any regret constraints. We show that the corresponding set of generalized equilibria captures an extension of *Stackelberg* equilibria with a matching optimal value, and that there exists a wide class of games where a player can significantly increase their utility by deviating from a no-swap-regret algorithm against a no-swap learner (in fact, almost any game without pure Nash equilibria is of this form). Additionally, we make use of generalized equilibria to consider tradeoffs in terms of the opponent's algorithm choice. We give a tight characterization for the maximal reward obtainable against *some* no-regret learner, yet we also show a class of games in which this is bounded away from the value obtainable against the class of common "mean-based" no-regret algorithms. Finally, we consider the question of learning reward-optimal strategies via repeated play with a no-regret agent when the game is initially unknown. Again we show tradeoffs depending on the opponent's learning algorithm: the Stackelberg strategy is learnable in exponential time via repeated play with any no-regret agent (and in polynomial time with any no-*adaptive*-regret agent) for any game where it is learnable via queries, and there are games where it is learnable in polynomial time against any no-swap-regret agent but requires exponential time against a mean-based no-regret agent.

The results of this chapter highlight the central promise of leveraging the structured adaptivity of

other agents in our optimization, as opposed to viewing our opponent in a repeated game as "merely adversarial". In the oft-studied scenario of simultaneous no-swap-regret learning in games, agents are guaranteed — or alternatively, doomed — to reach a correlated equilibrium via the time-average distribution of joint strategies played, and so no agent can receive an average reward higher than that in any correlated equilibrium for the game. When the Stackelberg value of the game for an agent is larger than this, which we show occurs under quite general conditions, then one can strictly improve outcomes by exploiting the adaptivity patterns of the opponent's algorithm.

1.3.2 Adaptivity Structures for Agent Preferences

In Chapter 3, we consider the problem of providing sequences of recommendations — articles to read, videos to watch, or items to purchase — to agents using an online platform. Historically, recommendation systems have been predominantly studied from a statistical perspective, wherein preferences are presumed to be fixed for each user and correlated across the population of users; there, the goal is typically to predict what content users will most prefer (possibly with exogenous constraints) by leveraging preference data from "similar" users. However, as the social media age has made abundantly clear, our content preferences are not static, and can often be directly influenced by our prior interactions with online platforms. We introduce a model for studying online recommendations when user preferences are *adaptive* based on their history of past selections, and where we aim to optimize our rewards as a function of the *choices* of the user, but we are constrained to show them a *menu* of many options in each interaction. This history-dependent adaptivity of user preferences violates the population-level statistical assumptions used by approaches such as collaborative filtering, and so we instead focus on interactions with a *single* user over T rounds, where this optimization problem already becomes nuanced and non-trivial.

In our setting, there is a universe of n items, and in each round we show a size-k menu of these items to an agent, who then chooses a single one. The choice of the agent is a probabilistic function of their *preference scores* for each item, and these scores are themselves functions of the history of the agent's past choices. Our rewards in each round are determined by (adversarially) time-varying

functions of the user's choice, and our goal will be to optimize these cumulative rewards with respect to some target set. The choice distribution over any item menu we show the user depends on their current preferences, and we will not know the user's preference functions in advance; our approaches will involve *learning* the agent's preference dynamics, *steering* the agent's preference scores, and implicitly *optimizing* over the item distributions which the agent will choose from.

There are some subtleties which emerge here in terms of our choice of target set: for many desirable targets, such as the best fixed menu or the best "stable" item distribution, minimizing regret turns out to be impossible or computationally intractable. A number of properties relating to the structure of agent preference dynamics will shape both the target sets we can feasibly optimize over and the rates at which we can minimize regret. Notably, our results will depend on the "memory horizon" of the agent as a function of T, the "exploration level" of the agent, as well as the agent's degree of "preference for familiarity". For quite broad ranges of these values, we obtain sublinear regret algorithms for appropriate convex target sets (as subsets of the item simplex $\Delta(n)$), and we highlight a number of barriers to further expanding these targets. Concretely, our algorithms in this chapter operate by guiding the user to conduct regret minimization over the target set on our behalf through our choices of recommendation menus, as we cannot directly force the agent to choose any particular item. We also observe that the target sets we consider are synergistic with another common exogenous goal for online platforms: showing a *diverse* set of content to users.

1.3.3 A Unified Approach

In Chapter 4, we introduce a framework for algorithmic optimization in a broad class of problems involving repeated interactions with adaptive agents, building upon the techniques developed for the recommendations problem in Chapter 3. We refer to these as "online Stackelberg" problems, alluding to the leader-follower optimization objective in Stackelberg games, though in many of our applications the agent behavior is not assumed to strictly follow a best-response pattern; rather, we consider a range of patterns for agent dynamics which are "in the spirit of" best-response behavior (e.g. no-regret dynamics as considered in Chapter 2, or preference-proportional sampling

as in Chapter 3). We show that many problems of this form can be cast as instances of online control with dynamics which may be nonlinear, yet which satisfy a useful property we refer to as *local controllability*, and where objectives can be well-approximated by surrogate convex losses over a bounded state space which represents the current behavior of the agents. Departing from the techniques often used in online linear control, we instead aim to implicitly optimize over the behavior space of the agent using tools from online convex optimization, and we give a sequence of regret minimization algorithms with results spanning a range of conditions related to computational efficiency, assumptions on the knowledge and structure of dynamics or losses, and the possibility of adversarial disturbances to the dynamics of the system.

To display the applicability of this framework, we show that a range of previously-studied problems can be cast as instances of online nonlinear control problems with dynamics satisfying the local controllability condition, which allows direct application of our algorithms and in many cases yields novel results for regret minimization. This includes a variant of the "optimizing against learners" problem considered in Chapter 2, wherein we aim to "steer" the behavior of an opponent's learning algorithm in a time-varying game, as well as the adaptive recommendations problem from Chapter 3 for both of the primary regret benchmarks we considered there. Additionally, we show applications to the problem of "strategic classification" or "performative prediction" in an online setting, where agents can dynamically manipulate their individual features to obtain desired labelings by a prediction model, as well as a problem related to revenue optimal adaptive-pricing for real-valued goods, wherein demand as well as production costs may vary over time.

1.4 Notation and Preliminaries

Let $[n]$ denote the integers 1 through n, let $\mathbb{R}^n$ denote the real numbers in n dimensions, and let $\Delta(n)$ denote the probability simplex in $\mathbb{R}^n$. Unless specified otherwise, we will assume throughout that functions defined over $\mathbb{R}^n$ are continuous and differentiable everywhere, with gradients denoted by $\nabla f(x)$. We will often make reference to convex sets and functions, as well as Lipschitz continuity.

Definition 1 (Convex Sets)**.** *A set $\mathcal{X} \subseteq \mathbb{R}^n$ is convex if for any two points $x, x' \in \mathcal{X}$ and $\alpha \in [0, 1]$,*

we have that $\alpha x + (1-\alpha)x' \in \mathcal{X}$.

Definition 2 (Convex Functions)**.** *A function* $f : \mathcal{X} \to \mathbb{R}$ *is convex if for any two points* $x, x' \in \mathcal{X}$ *and* $\alpha \in [0,1]$*, we have that* $f(\alpha x + (1-\alpha)x') \leq \alpha f(x) + (1-\alpha) f(x')$.

Definition 3 (Lipschitz Continuity)**.** *A function* $f : \mathcal{X} \to \mathbb{R}$ *is L-Lipschitz continuous (or simply L-Lipschitz) with respect to a norm* $\|\cdot\|$ *if* $|f(x) - f(x')| \leq L \|x - x'\|$ *for any two points* $x, x' \in \mathcal{X}$.

Many of the results in this thesis are concerned with *regret minimization*, where the aim is typically to compete with the optimum strategy within a fixed reference class over T rounds, with the per-round performance gap vanishing as T grows. For an online decision problem where in each round we choose a point $x_t \in \mathcal{X} \subseteq \mathbb{R}^n$, and our loss $f_t(x_t)$ is given by a function $f_t : \mathcal{X} \to \mathbb{R}$ revealed following our choice of x_t, we measure regret with respect to the optimal fixed $x^* \in \mathcal{X}$.

Definition 4 (Regret)**.** *Over T rounds, for a sequence of points* $x_1, \ldots, x_T \in \mathcal{X}$ *and losses* $f_1, \ldots, f_t : \mathcal{X} \to \mathbb{R}$*, the regret with respect to* $\mathcal{X}$ *is given by*

$$\mathsf{Reg}(T) = \max_{x^* \in \mathcal{X}} \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*).$$

If $\mathcal{X}$ and each f_t are convex, this is the setting of online convex optimization (OCO). if $\mathcal{X} = \Delta(n)$ and each f_t is a linear function taking values in $[0,1]$ this is known as the "experts problem", referring to the goal of tracking the performance of the best of a set of n experts who give predictions in each round (e.g. whether to buy or sell an asset). There are quite a number of algorithms — Hedge, Online Mirror Descent, Follow the Regularized Leader, Follow the Perturbed Leader, EXP3, and more — which are well-studied for a variety of regret minimization problems. A particularly simple example of such an algorithm is Online Gradient Descent, first analyzed for the OCO setting by Zinkevich [1], which we state here for illustration along with its regret bound. Perhaps surprisingly if one is unfamiliar with regret minimization in adversarial settings, this algorithm is essentially identical to gradient-based methods which are often used for static or statistical optimization problems.

Algorithm 1 Online Gradient Descent (OGD)

Input: convex G-Lipschitz losses $f_1, \ldots, f_T$, step size η
Set $x_1 = \mathbf{0}$
for $t = 1$ to T **do**
  Play x_t and observe cost $f_t(x_t)$
  Let $y_{t+1} = x_t - \eta \nabla f_t(x_t)$
  Set $x_{t+1} = \Pi_{\mathcal{X}}(y_{t+1})$
end for

Proposition 1 (Regret Bound for OGD [1])**.** *For a set $\mathcal{X}$ with diameter at most D, a sequence of G-Lipschitz losses $f_1, \ldots, f_T$, and step size $\eta = \frac{D}{G\sqrt{T}}$, the regret of* OGD *with respect to $\mathcal{X}$ is at most*

$$\max_{x^* \in \mathcal{X}} \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*) \leq \frac{D^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\nabla_t\|^2 \leq GD\sqrt{T}.$$

We say that any algorithm with $o(T)$ regret is a "no-regret" algorithm, and often the optimal obtainable bound will be of the form $O(\sqrt{T})$ — there are lower bounds of $\Omega(\sqrt{T})$ known for both the OCO and experts settings. Here, $O(\cdot)$ is the standard asymptotic upper-bound notation, hiding constant factors and lower-order terms, and $o(\cdot)$ denotes asymtotic dominance by the argument; we use $\tilde{O}(\cdot)$ when polylogarithmic factors are hidden as well. $\Omega(\cdot)$ and $\omega(\cdot)$ denote the analogous asymptotic lower bounds, and $\Theta(\cdot)$ denotes that both lower and upper bounds hold asymptotically (up to polylogarithmic factors with $\tilde{\Theta}(\cdot)$). In this thesis we will also discuss notions of "swap regret" and "policy regret", both of which consider regret comparators given by functions (which take a counterfactual action or other time-dependent context as an argument, respectively) rather than a fixed point $x^* \in \mathcal{X}$. If our algorithm observes only the value of $f_t(x_t)$ rather than the entire function f_t, we refer to this as "bandit feedback"; there are many no-regret algorithms known for the bandit setting as well. Finally, as we will discuss in greater detail in Chapter 2, there are intimate connections between no-regret algorithms and equilibria in games. Notably, if a no-regret guarantee holds for both players in a $m \times n$-action game under a joint action distribution $\sigma \in \Delta([m] \times [n])$, we say that σ is a *coarse correlated equilibrium*, and σ is a *correlated equilibrium* if a no-swap-regret guarantee holds for both players (up to an approximation threshold).

1.5 Additional Background

Perhaps the most technically relevant reference text for this thesis, which proved an invaluable resource throughout the process of writing and research, is "Introduction to Online Convex Optimization" by Hazan [2], which contains thorough coverage of important concepts and algorithms related to online learning and regret minimization. For a comprehensive overview of topics at the intersection of game theory and computer science, including the previously-discussed connections between learning and equilibria, we recommend "Algorithmic Game Theory" by Nisan, Roughgarden, Tardos, and Vazirani [3]. Two notable recent lines of research, which share some conceptual similarities with the results of this thesis (but largely distinct technical approaches), are the problems of online nonstochastic (linear) control and performative prediction; see "Introduction to Online Nonstochastic Control" by Hazan [4] for a manuscript containing primary results for the former, and "Performative Prediction: Past and Future" by Hardt and Mendler-Dünner [5] for a survey of important works on the latter. We discuss these connections in detail in Chapter 4.

Chapter 2: Is Learning in Games Good for the Learners?

The results of this chapter are based on joint work with Jon Schneider and Kiran Vodrahalli [6].

2.1 Overview

How should two rational agents play a repeated, possibly unknown, game against one another? One natural answer – barring any knowledge of the game, or the capacity to compute potentially computationally intractible equilibria – is that they should employ some sort of learning algorithm to learn how to play over time. Indeed, there is a vast literature which studies what happens when all the players in a repeated game run (some specific type of) learning algorithms to select their actions. For example, when all players in a game simultaneously run no-(swap)-regret learning algorithms, it is known that the average strategy profile of the learners converges to a (coarse) correlated equilibrium [7, 8, 9]. More recent works have studied how to design algorithms that converge to these equilibria at faster rates [10, 11, 12], performance guarantees of such equilibria compared to the optimal possible welfare [13, 14], and the specific dynamics of such learning algorithms [15, 16, 17].

In contrast, relatively little attention has been devoted to whether it is actually *in the interest of these agents* to run these specific learning algorithms. For example, in a setting where all agents are running no-swap-regret learning algorithms, when can an agent significantly improve their reward by deviating and running a different type of algorithm? And if they can, what other algorithm should the agent deviate to instead?

2.1.1 Our Results

We explore the following questions (and others) in the case where two agents repeatedly play a normal-form, general-sum game for T rounds.

When does reward trade off with regret? While maximizing reward is often viewed as the objective of regret minimization against arbitrary adversaries, tensions may emerge when playing against another learning agent, as one's choice of actions has a causal effect on future loss functions as determined by the opponent's algorithm. In such cases, as we discuss further below, it turns out that demanding a stronger regret guarantee (e.g. asking for no-swap-regret instead of no-external-regret) may ultimately result in lower reward for an agent. To analyze these tradeoffs, in Section 2.2 we introduce a notion of *generalized* (Φ_A, Φ_B)*-equilibrium*, where Φ_A and Φ_B are the sets of "deviation strategies" over which players A and B minimize regret (e.g. fixed actions or swap functions), presented as an asymmetric extension of the linear Φ-equilibria considered by [18]. Each pair of strategy sets (Φ_A, Φ_B) generates a polytope of (Φ_A, Φ_B)-equilibria, where any point then yields a reward value for each player. We show that all such points are feasible: for any game and any (Φ_A, Φ_B)-equilibrium φ, there is a pair of algorithms which converge to φ while maintaining their respective Φ_A-regret and Φ_B-regret guarantees against arbitrary opponents. Further, deviating to a strategy with fewer constraints Φ_A can often result in *strictly* improved reward for player A. As a concrete application, we consider the question of deviating from simultaneous no-swap-regret play.

When is no-swap-regret play a stable equilibrium? What should you do if you know that your opponent in a repeated game is running a no-swap-regret algorithm to select their actions? In [19], the authors show that one utility-optimizing response (up to additive $o(T)$ factors) is to play a static (mixed) strategy (your *Stackelberg strategy*) and obtain the *Stackelberg value* of the game. However, determining your Stackelberg strategy requires some knowledge of the game, and acquiring this knowledge from repeated play may be difficult (we address the latter issue in

Section 2.5). In comparison, it is relatively straightforward to also run a no-swap-regret learning algorithm. This raises the question: are there games where you obtain significantly less utility (i.e., at least $\Omega(T)$ less utility) by running a no-swap-regret learning algorithm instead of playing your Stackelberg strategy?

We show that the answer is *yes*, such games exist and are relatively common. In fact, we provide an efficient algorithmic characterization of the games G for which both players playing a no-swap-regret learning algorithm is an ($o(T)$-)approximate Nash equilibrium of the entire repeated "meta-game". The exact characterization is presented in Section 2.3 and is somewhat subtle, e.g., there are slightly different characterizations depending on whether we insist all pairs of no-swap-regret algorithms lead to approximate equilibria or only one specific pair, corresponding to best-case and worst-case values in the generalized equilibrium polytope. One consequence of both characterizations, however, is that for *almost all* games (in a measure-theoretic sense, considering arbitrarily small perturbations), in order for it to be an approximate equilibrium for both players to play a low-swap regret strategy, the game G must possess a *pure* Nash equilibrium. That is, in any game without a pure Nash equilibrium, it is possible for at least one of the parties to do significantly better by switching from no-swap-regret learning to playing their Stackelberg strategy.

Finally, we additionally show there are some games where playing a no-(external)-regret algorithm against another no-swap-regret learner weakly dominates playing a no-swap-regret algorithm, regardless of the specific choice of algorithms. This counters the intuition that stronger regret guarantees protect a player from worse outcomes.

Optimizing reward against no-regret learners. What if our opponent is not running a no-swap-regret algorithm, but simply a no-(external)-regret algorithm? In this case, it is still possible to obtain at least the Stackelberg value of the game by playing our Stackelberg strategy (no-regret algorithms are also guaranteed to eventually learn and play the best response to this strategy). However, unlike in the no-swap-regret setting, there exist specific games and no-regret algorithms where it is possible to obtain *significantly* ($\Omega(T)$) more than the Stackelberg value by playing a

specific dynamic strategy. This phenomenon was first observed in [19], where a specific game is given for which it is possible to obtain $\Omega(T)$ more than Stackelberg when playing against any no-regret algorithm in the family of *mean-based learning algorithms* (including algorithms such as multiplicative weights and EXP3). However, many questions remain unanswered, such as understanding in which games it is possible to outperform playing one's Stackelberg strategy, and by how much.

In Section 2.4 we present some answers to these questions for the case of generic no-regret algorithms. Specifically, we first show that if player B is running (any) no-regret algorithm, the utility of player A (regardless of what strategy they employ) is upper bounded by $\mathsf{Val}_A(\emptyset, \mathcal{E}) \cdot T + o(T)$, where $\mathsf{Val}_A(\emptyset, \mathcal{E})$ is what we call the the *unconstrained-external (equilibrium) value* of the game for player A, which is given by the solution to a linear program. We then show that this upper bound is asymptotically tight: there exists a no-regret algorithm $\mathcal{L}$ such that if player B is playing according to $\mathcal{L}$, then the player A can obtain utility at least $\mathsf{Val}_A(\emptyset, \mathcal{E}) \cdot T - o(T)$ by playing an appropriate strategy in response.

Note that this characterization does not completely resolve the question of [19] – it requires the construction of a fairly specific no-regret algorithm $\mathcal{L}$, and it is still open what is possible against specific (classes of) no-regret algorithms (e.g., multiplicative weights or mean-based algorithms). In fact, we show a property of games which, when satisfied, implies that it is impossible to obtain the unconstrained-external value against a mean-based learner.

Learning the Stackelberg strategy through repeated play. Finally, we address the question of how hard it is to actually identify the Stackelberg strategy in a game against a learning opponent. Given full knowledge of the game G, finding an agent's Stackelberg strategy is simply a computational problem which turns out to be efficiently solvable by solving several small LPs (see [20]). However, if the game (in particular, the opponent's reward matrix) is unknown, an agent must learn the Stackelberg strategy over time. Existing work on learning Stackelberg strategies (e.g., [21, 22]) generally assumes access to a *best-response oracle* for the game (i.e., for a specific mixed strategy,

how will an opponent best-respond?). In contrast, if our opponent is playing a specific no-regret learning algorithm, they may not immediately best respond to the strategies we play! This raises the following two questions. First, when is it possible to learn the Stackelberg equilibrium of a game while playing against a learning opponent? Second, is it easier to learn this equilibrium when playing against certain classes of learning algorithms?

In Section 2.5 we begin by showing that it is indeed possible to convert any best-response query algorithm for finding Stackelberg equilibria via best-response queries to an adaptive strategy that learns Stackelberg equilibria via repeated play against a generic no-regret learner, albeit potentially at the cost of an exponential blow-up in the number of rounds, e.g. for a query algorithm which makes Q best-response queries, simulating it against a no-regret learner may require $T = \exp(Q)$ rounds of play. For the special case of opponents with no-*adaptive*-regret algorithms (such as online gradient descent), we show that only $T = \text{poly}(Q)$ rounds are required in the worst case. However, in general we show that exponential runtime can be necessary. In particular, we give an example of a game with M actions where it is possible to learn the Stackelberg equilibrium in $\text{poly}(M)$ rounds when playing against any no-swap-regret learning algorithm, but where it requires at least $\exp(M)$ rounds to learn this equilibrium when playing a mean-based no-regret algorithms.

2.1.2 Related Work

The broader literature on no-regret learning in repeated games is substantial, covering many equilibrium convergence results varying assumptions. A recent line of work [23, 19, 24] considers problems related to optimizing one's reward when competing against a no-regret learner in a game. We extend these questions to consider the relationship and regret for an optimizer, as well as to settings where properties of the game are initially unknown, and give a series of separation results in terms of various notions of equilibrium. Also relevant is the literature on analysis of no-regret trajectory dynamics, in particular [16] which shows a game in which no-regret dynamics outperform the reward of the Nash equilibrium. Additionally, there is also prior work considering regret minimization problems involving either best-responding or otherwise strategic agents (see

e.g. [25, 26]), as well as work considering alternate regret notions or behavior models for repeated Stackelberg games (e.g. [27, 28]).

2.1.3 Notation and Preliminaries

Throughout this chapter, we consider two-player bimatrix games $G = (A, B)$, where player A ("the optimizer") has action set $\mathcal{A} = \{a_1, \ldots, a_M\}$ and player B ("the learner") has action set $\mathcal{B} = \{b_1, \ldots, b_N\}$. When the optimizer plays action a_i and the learner plays action b_j, the players receive rewards $u_A(a_i, b_j)$ and $u_B(a_i, b_j)$, respectively. We assume that the magnitude of each utility is bounded by a constant. The sets of mixed strategies for each player are denoted by $\Delta(\mathcal{A})$ and $\Delta(\mathcal{B})$, respectively; when the optimizer plays a mixed strategy $\alpha \in \Delta(\mathcal{A})$ and the learner plays $\beta \in \Delta(\mathcal{B})$, the expected reward for the optimizer is given by $u_A(\alpha, \beta) = \sum_{i=1}^{M} \sum_{j=1}^{N} \alpha_i \beta_j u_A(a_i, b_j)$, with $u_B(\alpha, \beta)$ defined analogously. An action $b \in \mathcal{B}$ is a *best response* to a strategy $\alpha \in \Delta(\mathcal{A})$ if $b \in \operatorname{argmax}_{b' \in \mathcal{B}} u_B(\alpha, b')$. We let $\mathsf{BR}(\alpha)$ be the set of all such actions for player B, and likewise $\mathsf{BR}(\beta)$ for player A.

2.2 Generalized Equilibria and No-Φ-Regret Learning

Here we introduce the notions of Φ-regret and generalized equilibria, which we use to analyze the regret and reward of players in repeated bimatrix games under varying assumptions regarding the choice of regret benchmarks, the algorithms used, and the structure of the game.

Originally introduced in [29] and extended to general convex settings in [18], we consider the formulation of *linear* Φ-regret as it relates to bimatrix games. Given a sequence of action pairs $(a_{i_1} b_{j_1}), \ldots, (a_{i_T} b_{j_T})$ for $T > 0$ and some set of functions Φ, where each $f \in \Phi$ maps actions $\mathcal{A}$ to action profiles in $\Delta(\mathcal{A})$, we say that the Φ-*regret* for the optimizer (or analogously for the learner) is

$$\mathsf{Reg}_\Phi(T) = \max_{f \in \Phi} \sum_{t=1}^{T} u_A(f(a_{i_t}), b_{j_t}) - u_A(a_{i_t}, b_{j_t}).$$

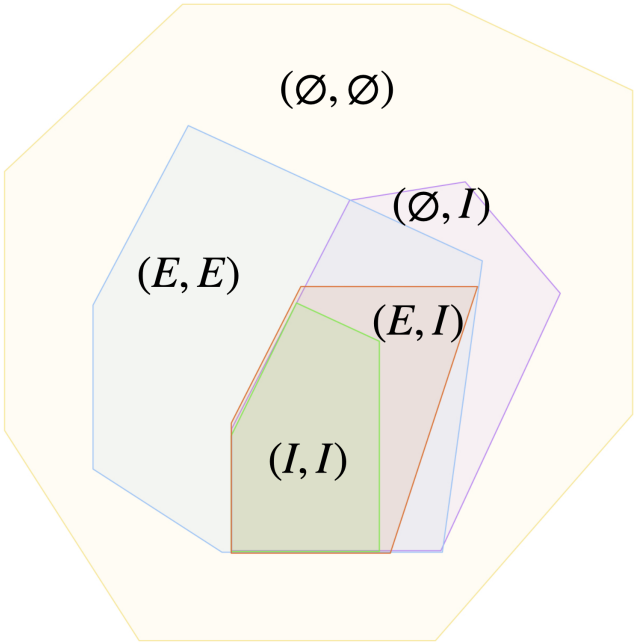

Figure 2.1: Examples of generalized equilibria involving internal, external, and unconstrained regret. Here, (I, I) is equivalent to the set of correlated equilibria, and (E, E) is equivalent to the set of coarse correlated equilibria.

Definition 5 (No-Φ-Regret Learning)**.** *We say a learning algorithm $\mathcal{L}$ is a* no-Φ-regret *algorithm if, for some constant $c < 1$, we have that $\mathsf{Reg}_\Phi(\mathcal{L}, T) = O(T^c)$, where $\mathsf{Reg}_\Phi(\mathcal{L}, T)$ is the Φ-regret corresponding to the action sequence played by $\mathcal{L}$.*

Some notable sets of regret comparator functions Φ are the constant maps $\mathcal{E}$ (corresponding to *external regret*), where all input actions are mapped to the same output action, and the "swap functions" $\mathcal{I}$ (corresponding to *internal regret*[1]), which contain all single swap maps $f_{ij} : [M] \to [M]$ where $f(i) = j$ and $f(i') = i'$ for $i' \neq i$. Imposing these constraints on players in a game results in a *(coarse) correlated equilibrium*, which are instances of our notion of *generalized equilibrium*.

Definition 6 (Generalized (Φ_A, Φ_B)-Equilibria)**.** *A (Φ_A, Φ_B)-*equilibrium *$\varphi \in \Delta(\mathcal{A} \times \mathcal{B})$ in a two-player game is a joint distribution over action profiles (a, b) such that player A cannot increase their expected reward by deviating with some strategy in Φ_A and player B cannot benefit by deviating with some strategy in Φ_B.*

[1]We refer to internal and swap regret interchangeably, as our focus is primarily on rates with respect to T.

In contrast to the Φ-equilibria considered by [29, 18], here we allow constraints to be asymmetric between players. While many equilibrium notions for two-player games impose symmetric regret constraints on each player (e.g. Nash, correlated, and coarse correlated equilibria), this need not always be the case. In Section 2.3, we highlight Stackelberg equilibria as a motivating example for considering more general notions of asymmetric equilibria from the perspective of Φ-regret, to determine when one should deviate from simultaneous no-swap play, and in Section 2.4 we characterize the maximum reward attainable against no-regret learners in terms of asymmetric equilibria.

We say that the *value* of a game G for player A of a certain equilibrium class (Φ_A, Φ_B), denoted $\mathsf{Val}_A(\Phi_A, \Phi_B)$ is the maximum reward obtainable by player A at some (Φ_A, Φ_B)-equilibrium (with $\mathsf{Val}_B(\Phi_A, \Phi_B)$ defined symmetrically for player B). Likewise, we say that the *min-value* of a game for a player and an equilibrium class, denoted by e.g. $\mathsf{MinVal}_A(\Phi_A, \Phi_B)$ for player A, is the minimum reward for a player over all (Φ_A, Φ_B)-equilibria in a game. These capture the range of feasible average rewards under repeated play via (Φ_A, Φ_B)-regret dynamics.

Proposition 2. *For a repeated game over T rounds where player A uses a no-Φ_A-regret algorithm and player B uses a no-Φ_B-regret algorithm, the average rewards obtained by each player are upper bounded by* $\mathsf{Val}_A(\Phi_A, \Phi_B) + o(1)$ *and* $\mathsf{Val}_B(\Phi_A, \Phi_B) + o(1)$*, respectively, and lower bounded by* $\mathsf{MinVal}_A(\Phi_A, \Phi_B) - o(1)$ *and* $\mathsf{MinVal}_B(\Phi_A, \Phi_B) - o(1)$.

Proof. The set of (Φ_A, Φ_B)-equilibria includes all strategy profile distributions in which both constraints are satisfied. If a player receives substantially more or less than the corresponding value, this would imply a violation of the regret constraints for at least one of the players' learning algorithms. □

We consider an ε-approximate (Φ_A, Φ_B)-equilibrium to be a joint profile distribution where each constraint is satisfied up to additive error ε, and observe that no-Φ-regret dynamics convergence to appropriate notions of generalized equilibria, connecting Definitions 5 and 6 as follows.

Proposition 3. *Suppose after T rounds of a game where player A plays a no-Φ_A-regret algorithm and player B plays a no-Φ_B-regret algorithm, player A has average Φ_A-regret $\leq \varepsilon$ and player B has average Φ_B-regret $\leq \varepsilon$. Let $\varphi^t := p_A^t \times p_B^t$ denote the joint distribution over both players' actions at time t and $\varphi := \frac{1}{T}\sum_{t=1}^{T} \varphi^t$ denote the time-averaged history over joint player action distributions. Then, φ is an ε-approximate (Φ_A, Φ_B)-equilibrium where*

$$\begin{aligned}
\mathop{\mathbb{E}}_{(a,b)\sim\varphi}\left[u_A\left(a,b\right)\right] &\geq \mathop{\mathbb{E}}_{(a,b)\sim\varphi}\left[u_A\left(f_A(a),b\right)\right] - \varepsilon, \text{ and} \\
\mathop{\mathbb{E}}_{(a,b)\sim\varphi}\left[u_B\left(a,b\right)\right] &\geq \mathop{\mathbb{E}}_{(a,b)\sim\varphi}\left[u_B\left(a,f_B(b)\right)\right] - \varepsilon
\end{aligned}$$

for every possible deviation $f_A \in \Phi_A, f_B \in \Phi_B$. Likewise, if players A and B repeatedly play strategies corresponding to an (Φ_A, Φ_B)-equilibrium, then player A is no-Φ_A-regret and player B is no-Φ_B-regret.

Proof. The statement follows by observing that

$$\begin{aligned}
\mathop{\mathbb{E}}_{(a,b)\sim\varphi}\left[u_{\{A,B\}}\left(a,b\right)\right] &= \frac{1}{T}\sum_{t=1}^{T}\mathop{\mathbb{E}}_{(a,b)\sim\varphi^t}\left[u_{\{A,B\}}\left(a,b\right)\right] \\
\mathop{\mathbb{E}}_{(a,b)\sim\varphi}\left[u_A\left(f_A(a),b\right)\right] &= \frac{1}{T}\sum_{t=1}^{T}\mathop{\mathbb{E}}_{(a,b)\sim\varphi^t}\left[u_A\left(f_A(a),b\right)\right] \\
\mathop{\mathbb{E}}_{(a,b)\sim\varphi}\left[u_B\left(a,f_B(b)\right)\right] &= \frac{1}{T}\sum_{t=1}^{T}\mathop{\mathbb{E}}_{(a,b)\sim\varphi^t}\left[u_B\left(a,f_B(b)\right)\right]
\end{aligned}$$

which in turn are equivalent to the time-averaged utility of the play of players A and B, the time-averaged utility for player A under a deviation f_A, and the time-averaged utility for player B under a deviation f_B. Applying the definition of average Φ-regret and applying the given bounds on the Φ-regret yields the conclusion of the first direction. The reverse direction follows by reversing the steps. □

A general construction for no-Φ-regret algorithms is given in [18], which immediately implies feasibility of dynamics which converge to *some* instance of any class of (Φ_A, Φ_B)-equilibria in a game, possibly requiring agents to use differing algorithms if constraints are asymmetric. We show

that a much stronger claim is true: such a pair of algorithms exists for *any* (Φ_A, Φ_B)-equilibrium Ψ in a game. These algorithms also satisfy a "best-of-both-worlds" property, in that they converge to Ψ when played together, yet simultaneously maintain their corresponding regret guarantees against arbitrary adversaries.

Theorem 1. *Consider any game G. Suppose there exists a no-Φ_A-regret learning algorithm $\mathcal{L}_A$ and a no-Φ_B-regret learning algorithm $\mathcal{L}_B$. For any particular (Φ_A, Φ_B)-equilibrium Ψ in a game G, there exists a pair of learning algorithms $(\mathcal{L}_A^*(\Psi), \mathcal{L}_B^*(\Psi))$ such that:*

- *The empirical sequence of play when Player A uses $\mathcal{L}_A^*(\Psi)$ and Player B uses $\mathcal{L}_B^*(\Psi)$ converges to Ψ.*
- *$\mathcal{L}_A^*(\Psi)$ and $\mathcal{L}_B^*(\Psi)$ are no-Φ_A-regret and no-Φ_B-regret, respectively, against arbitrary adversaries.*

We give the full proof for Theorem 1 in Appendix A.1. Our approach is for the algorithms to initially implement a "round robin"-style schedule of strategies which converges to Ψ; additionally, each player also aims to detect when their opponent disobeys the schedule by tracking their Φ-regret with respect to Ψ, and after $o(T)$ violations the player can deviate indefinitely to playing a standalone no-Φ-algorithm for all remaining rounds. This ensures that Ψ is reached under joint play with $(\mathcal{L}_A^*(\Psi), \mathcal{L}_B^*(\Psi))$ yet both algorithms maintain their respective guarantees against arbitrary opponents. Several of our results throughout the chapter make use of Theorem 1. Here we state a notable immediate implication for equilibrium selection.

Corollary 1.1. *For any equilibrium scoring function $\Gamma : \Delta(\mathcal{A} \times \mathcal{B}) \to \mathbb{R}$ with a unique optimum computable in finite time, there exists a pair of learning algorithms $(\mathcal{L}_A^*, \mathcal{L}_B^*)$ such that:*

- *The empirical distribution when player A uses $\mathcal{L}_A^*$ and player B uses $\mathcal{L}_B^*$ converges to* $\operatorname{argmax}_\Psi \Gamma(\Psi)$.
- *$\mathcal{L}_A^*$ and $\mathcal{L}_B^*$ are no-Φ_A-regret and no-Φ_B-regret, respectively, against arbitrary adversaries.*

Proof. First optimize Γ over Ψ in finite time to find the unique optimum; then apply Theorem 1 to the resulting desired equilibrium. □

Corollary 1.1 allows for optimizing for objectives such as total welfare or min-max utility for both players, and imposing conditions on generalized equilibria beyond Φ-regret constraints (e.g. product constraints for Nash equilibria) by assigning arbitrarily low scores to invalid strategy profiles.

In subsequent sections, we will primarily focus on the function classes $\mathcal{E}$ and $\mathcal{I}$ corresponding to external and internal regret as mentioned above, as the well empty set $\emptyset$ corresponding to unconstrained regret; in Section 2.5 we additionally will consider the case when the game G is initially unknown. Before continuing, we note that each player's values for any (Φ_A, Φ_B)-equilibrium class can be expressed via a linear program, whose size is polynomial in the game dimensions for these function classes of interest.

Proposition 4. *For any game G and constraints (Φ_A, Φ_B), both $\mathsf{Val}_A(\Phi_A, \Phi_B)$ and $\mathsf{Val}_B(\Phi_A, \Phi_B)$ are computable via linear programs with MN variables and $\mathrm{poly}(M, N, |\Phi_A|, |\Phi_B|)$ constraints. When Φ_A and Φ_B belong to $\{\emptyset, \mathcal{E}, \mathcal{I}\}$, the number of constraints is $\mathrm{poly}(M, N)$.*

Proof. By definition, the set of (Φ_A, Φ_B)-equilibria φ is a sub-polytope of $\Delta(\mathcal{A} \times \mathcal{B})$ defined via the following linear constraints:

- For each $f_A \in \Phi_A$, we have that

$$\sum_{i \in [M]} \sum_{j \in [N]} \varphi_{ij} \, u_A(a_i, b_j) \geq \sum_{i \in [M]} \sum_{j \in [N]} \varphi_{ij} \, u_A(a_{f(i)}, b_j).$$

- For each $f_B \in \Phi_B$, we have that

$$\sum_{i \in [M]} \sum_{j \in [N]} \varphi_{ij} \, u_B(a_i, b_j) \geq \sum_{i \in [M]} \sum_{j \in [N]} \varphi_{ij} \, u_B(a_{f(i)}, b_j).$$

The value $\mathsf{Val}_A(\Phi_A, \Phi_B)$ corresponds to the element φ of this polytope that maximizes

$$\sum_{i \in [M]} \sum_{j \in [N]} \varphi_{ij}\, u_A(a_i, b_j).$$

Optimizing this linear function over the above polytope can be done in time $\text{poly}(M, N, |\,\Phi_A\,|, |\,\Phi_B\,|)$ via any linear program solver. Computing $\mathsf{Val}_B(\Phi_A, \Phi_B)$ can be likewise done efficiently.

For player A, the regret comparator function sets $\emptyset$, $\mathcal{E}$, and $\mathcal{I}$ contain 0, M, and M^2 elements respectively. In all three of these cases $|\,\Phi_A\,| = \text{poly}(M)$; likewise, in all three of these cases $|\,\Phi_B\,| = \text{poly}(N)$ (and thus we can efficiently compute these values when $\Phi_A, \Phi_B \in \{\emptyset, \mathcal{E}, \mathcal{I}\}$). □

2.2.1 Reward Separations

In general, the polytopes for different notions of generalized equilibria will be distinct for non-degenerate games and non-trivially different regret definitions. With respect to optimal values, these equilibrium classes are often distinct as well, and in Theorem 2 we show that there indeed exist games where the generalized equilibrium values do not collapse. The separations we show here consider the equilibrium cases either where both players have identical regret constraints, or where player A is unconstrained. We note that while inspecting other cases, we identified similar examples for several other generalized equilibrium pairs, and we expect that strict separations exist between any distinct pair of generalized equilibria for the three regret notions we consider, in any direction not immediately precluded by the regret constraints. We are mostly interested in cases where B is constrained, and A may be constrained or unconstrained.

Theorem 2. *For each of the following, there exists a* 4×4 *game* G *with rewards in* $\{0, 1, 2\}$ *where:*

1. $\mathsf{Val}_A(\emptyset, \mathcal{E}) > \mathsf{Val}_A(\emptyset, \mathcal{I}) > \mathsf{Val}_A(\mathcal{E}, \mathcal{E}) > \mathsf{Val}_A(\mathcal{I}, \mathcal{I})$

2. $\mathsf{Val}_A(\emptyset, \mathcal{E}) > \mathsf{Val}_A(\mathcal{E}, \mathcal{E}) > \mathsf{Val}_A(\emptyset, \mathcal{I}) > \mathsf{Val}_A(\mathcal{I}, \mathcal{I})$

Proof. We prove both results by exhibiting a game with the desired chain of inequalities, which we found by searching random examples of 4×4 games with values constrained in $\{0, 1, 2\}$ and

computing the various values of the games with a linear programming library. The numerical values are easy to check with computation. The game $G_1 := (M_{A_1}, M_{B_1})$ satisfies the conditions for the first chain of inequalities, and the game $G_2 := (M_{A_2}, M_{B_2})$ satisfies the conditions for the second chain of inequalities. First we instantiate the game G_1:

$$M_{A_1} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 2 & 2 & 0 & 2 \\ 0 & 2 & 0 & 0 \end{bmatrix} \qquad M_{B_1} := \begin{bmatrix} 0 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

The corresponding values for game G_1 are simple to check:

1. $\mathsf{Val}_A(\emptyset, \mathcal{E}) = 8/5$.
2. $\mathsf{Val}_A(\emptyset, \mathcal{I}) = 4/3$.
3. $\mathsf{Val}_A(\mathcal{E}, \mathcal{E}) = 1$.
4. $\mathsf{Val}_A(\mathcal{I}, \mathcal{I}) = 0$.

Then we instantiate the game G_2:

$$M_{A_2} := \begin{bmatrix} 2 & 0 & 1 & 0 \\ 2 & 1 & 1 & 0 \\ 0 & 2 & 1 & 2 \\ 2 & 0 & 2 & 1 \end{bmatrix} \qquad M_{B_2} := \begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 2 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 1 & 1 \end{bmatrix}$$

The corresponding values for game G_2 are simple to check:

1. $\mathsf{Val}_A(\emptyset, \mathcal{E}) = 13/7$.
2. $\mathsf{Val}_A(\mathcal{E}, \mathcal{E}) = 12/7$.
3. $\mathsf{Val}_A(\emptyset, \mathcal{I}) = 5/3$.

4. $\mathsf{Val}_A(\mathcal{I}, \mathcal{I}) = 4/3$.

□

Our results in Section 2.3 illustrate a particularly stark separation of this form, in which it can often be *dominant* to deviate to a strategy where Φ-regret constraints are violated.

2.3 Stability of No-Swap-Regret Play

Here we address the following question: when is it the case that for two players in a game, it is an approximate (Nash) equilibrium for both players to play no-swap-regret strategies? More specifically, imagine a "metagame" where at the beginning of this repeated game, both players simultaneously announce and commit to a specific adaptive (and possibly randomized) algorithm they intend to run to select actions to play in the repeated game G for the next T rounds. In this metagame, for which games G is it an $o(T)$-approximate Nash equilibrium for both players to play a no-swap-regret learning algorithm?

Of course, the answer to this question might depend on *which* specific no-swap-regret learning algorithm the agents declare. We therefore attempt to understand the following two questions:

- **Necessity:** For which games G is it the case that there exists *some* pair of no-swap-regret algorithms which form a $o(T)$-approximate Nash equilibrium? (Equivalently, when is it *never* the case that playing no-swap-regret algorithms forms an approximate Nash equilibrium?)
- **Sufficiency:** For which games G is it the case that *all* pairs of no-swap regret algorithms form $o(T)$-approximate Nash equilibria?

A central element of our analysis will be to consider the *Stackelberg equilibria* of a game.

Definition 7 (Stackelberg Equilibria)**.** *The* Stackelberg equilibrium *of a game G for player A is the pair of strategies (α, b) given by* $\operatorname{argmax}_{\alpha \in \Delta(\mathcal{A}),\ b \in \mathsf{BR}(\alpha)} u_A(\alpha, b)$*, and the resulting expected utility for player A is the* Stackelberg value *of the game, denoted* Stack_A*.* Stack_B *is defined symmetrically.*

We can relate Stackelberg equilibria to our notions of generalized equilibria.

Proposition 5. *For any game G, we have that* $\mathsf{Stack}_A = \mathsf{Val}_A(\emptyset, \mathcal{I})$.

Proof. Observe that under any strategy (α, b) where $b \in \mathsf{BR}(\alpha)$, player B cannot have any swap-regret, and so any Stackelberg equilibrium is also a $(\emptyset, \mathcal{I})$-equilibrium. Further, the marginal distributions over the optimal $(\emptyset, \mathcal{I})$-equilibrium for player A over each b_i cannot have distinct expected value for player A, as otherwise this would contradict optimality, and so an optimal $(\emptyset, \mathcal{I})$-equilibrium is either a single Stackelberg equilibrium or a mixture of Stackelberg equilibria with equal value. □

Here, any joint distribution over action profiles where player B has zero swap regret constitutes a $(\emptyset, \mathcal{I})$-equilibrium for a game, and the optimal value for such an equilibrium for player A coincides with the Stackelberg value. Further, each equilibrium set can be optimized over via a linear program.

Note that each value definition allows for tiebreaking in favor of player A. In general, simply playing the Stackelberg strategy α may not suffice to obtain Stack_A if the best response for Player B is not unique. However, there are a number of mild conditions which are each sufficient to ensure the existence of an approximate Stackelberg strategy α' which yields a unique best response for player B and obtains $\mathsf{Stack}_A - \varepsilon$ for any $\varepsilon > 0$. Here we consider a minimal such condition (essentially, no action is weakly dominated without also being strictly dominated).

Assumption 1. *In a game G, for each b, either* $\mathsf{BR}(\alpha) = \{b\}$ *for some* α*, or* $b \notin \mathsf{BR}(\alpha)$ *for all* α*. Likewise, for each a, either* $\mathsf{BR}(\beta) = \{a\}$ *for some* β*, or* $a \notin \mathsf{BR}(\beta)$ *for all* β*.*

We provide an efficient algorithmic procedure to answer both questions of necessity and sufficiency for a specific game G satisfying Assumption 1. To do this, recall that when two players both employ no-swap regret strategies, they asymptotically (time-average) converge to some correlated equilibrium (here, corresponding to an $(\mathcal{I}, \mathcal{I})$-equilibrium). On the other hand, by defecting from playing a no-swap regret strategy (while the other player continues playing their no-swap regret strategy), a player can guarantee their Stackelberg value for the game. Moreover, as shown by [19], this is the *optimal* (up to $o(T)$ additive factors) best response to an opponent running a no-swap regret strategy. It thus suffices to understand how the utility a player might receive under a correlated

equilibrium compares to the utility they receive under their Stackelberg strategy. For a fixed game G, let $\mathsf{Stack}_A = \mathsf{Val}_A(\emptyset, \mathcal{I})$ be the Stackelberg value for the first player, and $\mathsf{Stack}_B = \mathsf{Val}_B(\mathcal{I}, \emptyset)$ be the Stackelberg value for the second player. We have the following theorem.

Theorem 3. *Fix a game G satisfying Assumption 1. The following two statements hold:*

1. *There exists some pair of no-swap-regret algorithms that form an $o(T)$-approximate Nash equilibrium in the metagame iff there exists a correlated equilibrium φ in G such that $u_A(\varphi) = \mathsf{Stack}_A$ and $u_B(\varphi) = \mathsf{Stack}_B$.*

2. *Any pair of no-swap-regret algorithms form an $o(T)$-approximate Nash equilibrium in the metagame iff for all correlated equilibria φ in G, $u_A(\varphi) = \mathsf{Stack}_A$ and $u_B(\varphi) = \mathsf{Stack}_B$.*

Moreover, given a game G, it is possible to efficiently (in polynomial time in the size of G) check whether each of the above cases holds.

Proof. We obtain both claims by leveraging the construction in Theorem 1: best-case and worst-case correlated equilibria are feasible by some pair of no-swap-regret algorithms, and both players must simultaneously achieve close to their Stackelberg value for deviating to not be preferable.

We begin with the first claim. To prove the forward direction, if there exists such a φ, then choose a pair of low-swap-regret algorithms $(\mathcal{L}_A, \mathcal{L}_B)$ such that the time-averaged trajectory over T rounds is guaranteed to asymptotically converge to φ (this is possible by either the results of [8], or our Theorem 1). That is, if the two players play strategy φ_t at round $t \in [T]$, then $\hat{\varphi} = \frac{1}{T}\sum_t \varphi_t$ satisfies $||\hat{\varphi} - \varphi||_\infty = o(1)$. It follows that $\sum_t u_A(\varphi_t) \geq T \cdot u_A(\varphi) - o(T) = T \cdot \mathsf{Stack}_A - o(T)$ and therefore player A has at most an $o(T)$ incentive to deviate (by [19], they can obtain at most $\mathsf{Stack}_A\, T + o(T)$ against $\mathcal{L}_B$). Symmetric logic holds for player B.

To prove the reverse direction, assume $\mathcal{L}_A$ and $\mathcal{L}_B$ are no-swap-regret algorithms such that $(\mathcal{L}_A, \mathcal{L}_B)$ is an $o(T)$-approximate Nash equilibrium in the metagame. Since they are no-swap-regret, the time-averaged play of these two algorithms for T rounds must converge to an $o(1)$-approximate correlated equilibrium $\hat{\varphi}_T$; moreover, since $(\mathcal{L}_A, \mathcal{L}_B)$ is an $o(T)$-approximate Nash equilibrium, $\hat{\varphi}_T$ must have the property that $u_A(\hat{\varphi}_T) \geq \mathsf{Stack}_A - o(1)$ and $u_B(\hat{\varphi}_T) \geq \mathsf{Stack}_B - o(1)$.

Taking the limit as $T \to \infty$ and selecting a convergent subsequence of the $\hat{\varphi}_T$, this shows there must exist a correlated equilibrium φ with the desired properties.

Likewise, similar logic proves the second claim with the following modifications. In the forward direction, we can now choose any pair of low-swap-regret algorithms $(\mathcal{L}_A, \mathcal{L}_B)$, and any correlated equilibrium φ they asymptotically converge to is guaranteed to have the property that $u_A(\varphi) = \mathsf{Stack}_A$ and $u_B(\varphi) = \mathsf{Stack}_B$. In the reverse direction, since any correlated equilibrium is implementable by some pair of low-regret algorithms (again, by Theorem 1), the same logic shows that all correlated equilibria φ must satisfy $u_A(\varphi) = \mathsf{Stack}_A$ and $u_B(\varphi) = \mathsf{Stack}_B$.

Finally, to see that these two conditions are efficiently checkable, note that: i. the two values Stack_A and Stack_B are efficiently computable given the game G, and ii. the set of correlated equilibria φ form a convex polytope defined by a small (poly(N, M)) number of linear constraints (see Proposition 4). In particular, since $u_A(\varphi)$ and $u_B(\varphi)$ are simply linear functions of φ for a given game G, we can efficiently check whether there exists any point in this polytope where $u_A(\varphi) = \mathsf{Stack}_A$ and $u_B(\varphi) = \mathsf{Stack}_B$. □

The characterization in Theorem 3 is algorithmically useful, but sheds little direct light on in which games or how often we would expect playing no-swap-regret to be an approximate equilibrium. It turns out that for many games, playing no-swap-regret is *not* an equilibrium; below we will show that for almost all games, if G does not have a pure Nash equilibrium, at least one player has an incentive to deviate to their Stackelberg strategy.

Definition 8. *A property P of a game holds for* almost all *games if, given any game G, property P holds with probability* 1 *for the game G' formed by starting with G and perturbing each of the entries $u_A(a_i, b_j)$ and $u_B(a_i, b_j)$ by independent uniform random variables in the range $[-\varepsilon, \varepsilon]$ (for any choice of ε). In other words, the property holds for almost all choices of the $2MN$ utility values that define a game (with respect to the uniform measure on this space).*

We can show that if a correlated equilibrium has the same utility for a player as their Stackelberg value (a consequence of Theorem 3), then the correlated equilibrium must be a convex combination

of valid Stackelberg equilibria. In almost all games, both players have unique Stackelberg equilibria (and Assumption 1 holds), which implies that this correlated equilibrium must actually be the Stackelberg strategy for both players simultaneously. This implies that it is a pure Nash equilibrium (since one action in a generic Stackelberg equilibrium is always pure).

Theorem 4. *For almost all games G, if G does not have a pure Nash equilibrium, then there does not exist a pair of no-swap-regret algorithms which form a $o(T)$-approximate Nash equilibrium in the metagame for G.*

Proof. We will show that (for almost all games G) if there is a correlated equilibrium φ such that $u_A(\varphi) = \mathsf{Stack}_A$ and $u_B(\varphi) = \mathsf{Stack}_B$, then there exists a simultaneous unique Stackelberg equilibrium for both players in G, which must be a pure Nash equilibrium. Combined with Theorem 3, this implies the theorem statement.

We will rely on the following fact: in almost all games G, both players have a unique Stackelberg strategy. To see this, consider the following method for computing A's Stackelberg strategy. For each pure strategy b_j for player B, consider the convex set $A_j \subseteq \Delta(\mathcal{A})$ containing the mixed strategies for player A which induce b_j as a best response (i.e., $A_j = \{\alpha \in \Delta(\mathcal{A}) \mid b_j \in \mathsf{BR}(\alpha)\}$). Then, for each $j \in [N]$, compute the strategy $\alpha_j \in A_j$ which maximizes $u_A(\alpha_j, b_j)$. The Stackelberg value Stack_A is then given by $\max_j u_A(\alpha_j, b_j)$. In order for this to stem from a unique Stackelberg equilibrium, it is enough that: 1. the maximum utility is not attained by more than one j, and 2. for each j, the optimizer $\alpha_j \in A_j$ is unique.

These two properties are guaranteed to hold in almost all games. To see this, first note that the convex sets A_j are determined entirely by the utilities u_B, so we will treat these as fixed. Now, given any convex set A_j, the extremal point in a randomly perturbed direction will be unique with probability 1 – but since α_j is simply the extremal point of A_j in the direction specified by $u_A(\cdot, b_j)$ (which is a randomly perturbed direction), so α_j is unique in almost all games. Finally, if we perturb the magnitude of each of the utilities $u_A(\cdot, b_j)$ (keeping the direction the same), the maximizer $\max_j u_A(\alpha_j, b_j)$ will also be unique almost surely.

Let (α_A, b_A) be the Stackelberg equilibrium for player A and let (a_B, β_B) be the Stackelberg

equilibrium for player B. Now, consider the aforementioned correlated equilibrium $\varphi \in \Delta(A \times B)$. We will begin by decomposing it into its marginals based on its first coordinate; that is, we will write $\varphi = \sum_{i=1}^{M} \lambda_i(a_i, \beta_i)$ for some mixed strategies $\beta_i \in \Delta(\mathcal{B})$ and weights λ_i (with $\sum_i \lambda_i = 1$). By the definition of correlated equilibria, note that each a_i belongs to $\mathsf{BR}(\beta_i)$. But this means that $u_B(a_i, \beta_i) \leq \mathsf{Stack}_B$, with equality holding iff $(a_i, \beta_i) = (a_B, \beta_B)$ (due to uniqueness of Stackelberg). Therefore, in order for $u_B(\varphi) = \mathsf{Stack}_B$, we must have that $\varphi = (a_B, \beta_B)$. By symmetry, we must also have that $\varphi = (\alpha_A, b_A)$. If both these are true, then φ is a pure strategy correlated equilibrium of the game, and is hence a pure strategy Nash equilibrium (and moreover, is also the Stackelberg equilibrium for both A and B). □

Note that although Theorem 4 holds for almost all games, there are some important classes of games (most notably, zero-sum games) in the measure zero subset omitted by this theorem statement that both a) do not have pure Nash equilibria and b) have the property that playing no-swap-regret algorithms is an approximate equilibrium in the metagame (in particular, for zero-sum games, the Stackelberg value collapses to the value of the unique Nash equilibrium). Still, Theorem 4 shows that there are very wide classes of games for which playing no-swap-regret algorithms is not stable from the perspective of the agents.

2.3.1 Deviation to Weaker Regret Constraints

We have seen from Theorem 4 that if two players are playing no-swap-regret strategies against one another, it is often in the interest of each player to switch to playing their Stackelberg strategy (in particular, this is true whenever the game does not have a pure Nash equilibrium). However, as we will see in Section 2.5, learning one's Stackelberg strategy in such a game can be difficult. It is therefore natural to wonder if there are beneficial deviations to computationally efficient strategies. In particular, we can ask whether there are games where efficient deviations – e.g., to algorithms with *weaker* regret guarantees – lead to strictly more utility for the deviating player. Is it ever in a player's interest to weaken their regret benchmark, and e.g. switch from playing a no-swap-regret algorithm to a no-external-regret algorithm? We show in the following theorem that this can be

true in a fairly strong sense: there are games G where $\mathsf{Val}_A(\mathcal{E}, \mathcal{I}) > \mathsf{MinVal}_A(\mathcal{E}, \mathcal{I}) = \mathsf{Val}_B(\mathcal{I}, \mathcal{I})$. That is, we exhibit a game G where if player A switches from playing a no-swap-regret algorithm to *any* no-external-regret algorithm, their asymptotic utility never decreases and sometimes strictly increases – i.e., there is no downside to switch (and potentially a high upside).

Theorem 5. *There exists a game G where $\mathsf{MinVal}_A(\mathcal{E}, \mathcal{I}) \geq \mathsf{Val}_A(\mathcal{I}, \mathcal{I})$ and $\mathsf{Val}_A(\mathcal{E}, \mathcal{I}) \geq \mathsf{Val}_A(\mathcal{I}, \mathcal{I})$.*

Proof. Consider the game G specified by the two payoff matrices

$$M_A := \begin{bmatrix} 0 & 0 & 2 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \qquad M_B := \begin{bmatrix} 2 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{bmatrix}.$$

The corresponding values for this game are simple to compute:

1. $\mathsf{Val}_A(\mathcal{I}, \mathcal{I}) = \mathsf{MinVal}_A(\mathcal{I}, \mathcal{I}) = 0$.
2. $\mathsf{MinVal}_A(\mathcal{E}, \mathcal{I}) = 0$.
3. $\mathsf{Val}_A(\mathcal{E}, \mathcal{I}) = 1$.

□

2.4 Optimal Rewards Against No-Regret Learners

Here, we characterize the feasibility of optimizing one's reward against no-(external)-regret learners in terms of generalized equilibria. In contrast to the case of no-swap-regret learners, as shown by [19] there are games in which one can obtain $\Omega(T)$ more than the Stackelberg value over T rounds against certain no-regret algorithms by playing an appropriate adaptive strategy. A major remaining open question from this line of work is determining the best feasible reward and corresponding optimal strategy against no-regret agents in arbitrary games. We resolve this question when considering the maximum over all possible no-regret algorithms: for any game,

we can compute an upper bound on the feasible reward against any no-regret algorithm, and we show that there exists a specific no-regret algorithm against which we can obtain this reward via an efficiently implementable strategy.

Theorem 6. *For any game G, there exists a no-regret algorithm $\mathcal{L}$ and a strategy for player A such that the total reward of player A converges to $\mathsf{Val}_A(\emptyset, \mathcal{E}) \cdot T \pm o(T)$ when player B uses $\mathcal{L}$.*

Proof. By Theorem 1, there is a pair of $\emptyset$-regret and $\mathcal{E}$-regret algorithms $\mathcal{L}_A^*$ and $\mathcal{L}_B^*$ which converge to a $(\emptyset, \mathcal{E})$-equilibrium for which player A obtains $\mathsf{Val}_A(\emptyset, \mathcal{E})$. By Proposition 2, this is optimal over all no-external-regret algorithms, as any adaptive strategy constitutes a no-$\emptyset$-regret algorithm. By Proposition 4 we can identify the optimal $(\emptyset, \mathcal{E})$-equilibrium in $\text{poly}(M, N)$ time, which is sufficient to implement the algorithms $\mathcal{L}_A^*$ and $\mathcal{L}_B^*$ efficiently for any desired T. □

However, we additionally show that this bound is often unattainable against many standard no-external-regret algorithms. A property of many such algorithms (including Multiplicative Weights, Follow the Perturbed Leader, and Exp-3) is that they are *mean-based*, as formulated by [23].

Definition 9 (Mean-Based Learning)**.** *Let $\sigma_{i,t}$ be the cumulative reward resulting from playing action i for the first t rounds. An algorithm $\mathcal{L}$ is* γ-mean-based *if, whenever $\sigma_{i,t} \leq \sigma_{j,t} - \gamma T$, the probability that the algorithm selects action i in round $t+1$ is at most γ, for some $\gamma = o(1)$.*

These algorithms resemble "smoothed" variants of Follow the Leader; they only play actions with probability higher than $o(1)$ if their cumulative reward thus far is not too far from optimal, and hence never play dominated strategies. However, in general, $(\emptyset, \mathcal{E})$-equilibria may contain *dominated* strategies, as is also the case for coarse correlated equilibria. This allows us to show the following via a concrete example.

Theorem 7. *Against any mean-based no-regret algorithm for player B, there are games where a T-round reward of $(\mathsf{Val}_A(\emptyset, \mathcal{I}) + \varepsilon) \cdot T$ cannot be reached by any adaptive strategy for player A, for any $\epsilon > 0$. However, for this same game, $\mathsf{Val}_A(\emptyset, \mathcal{E}) > \mathsf{Val}_A(\emptyset, \mathcal{I})$.*

	b_1	b_2	b_3
a_1	1, 1	0, 0	3, 0
a_2	0, 0	1, 1	0, 0

Figure 2.2: Game where $\mathsf{Val}_A(\emptyset, \mathcal{E}) > \mathsf{Val}_A(\emptyset, \mathcal{I}) = \mathsf{MBRew}_A$

Proof. Let MBRew_A denote the maximal reward obtainable by player A when player B uses a mean-based algorithm. Observe that b_3 is dominated for player B, and thus cannot be included in any $(\emptyset, \mathcal{I})$-equilibrium (by Theorem 8). Further, it will never be played by a mean-based learner for more than $o(T)$ rounds, as for any distribution over a_1 and a_2 the best response is either b_1 or b_2. As such, both $\mathsf{Val}_A(\emptyset, \mathcal{I})$ and MBRew_A are at most $1 + o(1)$; a reward of $1 - o(1)$ is obtainable by committing to either a_1 or a_2 for each round. However, we can see that the optimal $(\emptyset, \mathcal{E})$-equilibrium p for player A includes positive mass on (a_1, b_3), and yields an average reward of $\mathsf{Val}_A(\emptyset, \mathcal{E}) = 2$ for player A. Let p_1 be the probability on (a_1, b_1), let p_2 be the probability on (a_2, b_2), let p_3 be the probability on (a_1, b_3), and let p_0 be the remaining probability. The reward for player A is given by:

$$\mathsf{Rew}_A(p) = p_1 + p_2 + 3p_3$$

and p defines a $(\emptyset, \mathcal{E})$-equilibrium if

$$\mathsf{Rew}_B(p) \geq \mathsf{Rew}_B(p \to b_i)$$

for each b_i, which holds if:

$$p_1 + p_2 \geq p_1 + p_3;$$

$$p_1 + p_2 \geq p_2;$$

$$p_1 + p_2 \geq 0.$$

Only the first constraint is non-trivial, and so the optimal $(\emptyset, \mathcal{E})$-equilibrium for player A occurs by maximizing $p_1 + p_2 + 3p_3$ subject to $p_2 \geq p_3$, which yields a probability of 0.5 for both p_2 and p_3

(and 0 for p_1 and p_0), as well as an average reward of 2. As such, player A cannot obtain a reward approaching $\mathsf{Val}_A(\emptyset, \mathcal{E})$, as their per-round reward is at most $1 + o(1)$. □

To generalize beyond this example, we introduce a notion of *dominated-swapping external regret*, a more precise upper bound on the behavior of mean-based algorithms than the standard no-external-regret guarantee, which we use to characterize a class of games where this separation holds.

Definition 10 (Dominated-Swapping External Regret)**.** *For a game G, let $D(G)$ be the set of dominated strategies for player B, i.e. $b_i \in D(G)$ if $b_i \notin \mathsf{BR}(\alpha)$ for all $\alpha \in \Delta(M)$. For $j, k \in [N]$ define $g_{jk}(b_i)$ as:*

$$g_{jk}(b_i) = \begin{cases} b_j & b_i \notin D(G) \\ b_k & b_i \in D(G) \end{cases}$$

i.e. $g_{jk}(b_i)$ swaps b_i to b_k if b_i is dominated and plays b_j otherwise. Let $\mathcal{E}_{D(G)} = \{g_{jk} : j, k \in [N]\}$ be the set of dominated-swapping external regret *comparators.*

This definition leads to the following upper bound on the average reward which is achievable against a mean-based no-regret algorithm.

Theorem 8. *For any game G and mean-based no-regret algorithm used by player B, there is no strategy yielding an asymptotic average reward for player A of $\mathsf{Val}_A(\emptyset, \mathcal{E}_{D(G)}) + \varepsilon$, for any $\varepsilon > 0$.*

Proof. First, we observe that mean-based algorithms will never play a dominated strategy $b_i \in D(G)$ in more than $o(T)$ rounds. As b_i is dominated, there is some $\delta > 0$ such that for every $\alpha \in \Delta(M)$, there is some b_j where $u_B(\alpha, b_j) \geq u_B(\alpha, b_i) + \delta$. Let α_t denote the empirical distribution of player A's actions up to time t. After some window of $O(\gamma T) = o(T)$ rounds we will have the cumulative rewards $\sigma_{i,t}$ and $\sigma_{j,t}$ satisfy $\sigma_{i,t} < \sigma_{j,t} - \delta t < \sigma_{j,t} - \gamma T$ under any α_t for some b_j in each subsequent round, and so b_i will never be played in more than $o(T)$ rounds.

We can also see that any such no-$\mathcal{E}$-regret algorithm is a no-$\mathcal{E}_{D(G)}$-regret algorithm. Suppose such an algorithm had $\mathcal{E}_{D(G)}$-regret ϵT, for $\epsilon > 0$; then, there is some g_{jk} for which $U_B(\alpha_T, g_{jk}(\beta_T)) \geq U_B(\alpha_T, \beta_T) + \epsilon$. By the $\mathcal{E}$-regret guarantee this cannot occur if $j = k$, as any such function g_{jj} is equivalent to the fixed deviation rule for b_j. However, if this occurs for $j \neq k$, such an algorithm must have played dominated strategies in a total $\Omega(\epsilon T)$. This contradicts our assumption that no dominated strategy b_i is played in more than $o(T)$ rounds, and so any mean-based no-$\mathcal{E}$-regret algorithm is also a no-$\mathcal{E}_{D(G)}$-regret algorithm, against which player A cannot obtain average reward which converges to any amount higher than $\mathsf{Val}_A(\emptyset, \mathcal{E}_{D(G)}) + o(1)$. □

2.5 Learning Stackelberg Equilibria in Unknown Games

Our results thus far have highlighted the primacy of the Stackelberg reward as an objective for repeated play against a learner: it is optimal against a no-swap learner and can sometimes be optimal against a mean-based learner, and it is almost always attainable against any learner. However, until now our strategies have assumed knowledge of the entire game, which may be unrealistic in many settings for which learning in games is relevant, particularly in terms of our opponent's rewards.

Here, we consider the challenge of learning the Stackelberg strategy via repeated play against a no-regret learner when only our own rewards are known, which is unaddressed in the literature to our knowledge; much of the prior work on learning Stackelberg equilibria assumes a *query* model, where one can observe the best response $\mathsf{BR}(\alpha)$ played by an opponent for any queried mixed strategy α. While here we cannot immediately observe the best response of an opponent, as their actions are selected by a learning algorithm which may be slow to adapt to changes in our behavior, we give a reduction from query algorithms of this form to strategies for choosing our actions which enable us to *simulate* queries to $\mathsf{BR}(\alpha)$ against a learner, and we analyze the efficiency of this approach (in terms of rounds required for learning) under differing assumptions on the learner's algorithm.

For comparison of behavior across time horizons of varying lengths, it will be convenient for us to consider the notion of an *anytime* regret bound, which can be obtained from any base no-regret

algorithm via doubling methods, as well as often via learning rate decay.

Definition 11 (Anytime regret algorithms)**.** *An algorithm is an* anytime no-Φ-regret *algorithm if satisfies* $\mathsf{Reg}_\Phi(t) = O(t^c)$ *over the first* t *rounds, for some* $c < 1$ *and any* $t \leq T$.

We also recall the notion of adaptive regret; many no-external-regret algorithms such as Online Gradient Descent satisfy no-adaptive-regret bounds (see e.g. [30]).

Definition 12 (Adaptive regret algorithms)**.** *An algorithm* $\mathcal{L}$ *for player* B *is a* no-adaptive-Φ-regret *algorithm if* $\sup_{r,s\in[T]} \mathsf{Reg}_\Phi(\mathcal{L}, [r, s]) \leq O(T^c)$*, for some* $c < 1$*, where* $\mathsf{Reg}_\Phi(\mathcal{L}, [r, s]) = \max_{f\in\Phi} \sum_{t=r}^{s} u_B(a_{i_t}, f(b_{j_t})) - u_B(a_{i_t}, b_{j_t})$.

A key distinction between adaptive-regret algorithms like OGD and mean-based algorithms like FTRL is in in their "forgetfulness", and hence their ability to quickly adapt when rewards change. This has stark implications for the efficiency of learning Stackelberg equilibria, which we show can take *exponentially* longer against mean-based algorithms. As shown by [30], adaptive regret is closely connected with dynamic regret; we note that our results for adaptive-regret learners can also be extended directly to hold for dynamic-regret learners.

2.5.1 Simulating Query Algorithms

Our approach will be to compute an ε-approximate Stackelberg strategy by simulating best response queries against a learner, after which point we can obtain an average reward approaching $\mathsf{Stack}_A - \varepsilon$ in each subsequent round, calibrating ε in terms of T as desired. The query complexity for such algorithms can depend on the geometry of the best response regions of the game, and unfortunately, as shown by [22], there are "hard" game instances which require exponentially many queries. This issue arises when the best response regions may be quite small but non-empty, as even finding a point in each region is information-theoretically difficult. We restrict our attention to games in which this does not occur, and fortunately efficient query do algorithms exist for this case.

Assumption 2. *For a game G and any action b_i, we have that*

$$\Pr_{\alpha \sim \textsc{Unif}(\Delta(M))} [b_i \in \mathsf{BR}(\alpha)] \in \{0\} \cup [1/\text{poly}(\varepsilon^{-1}), 1],$$

i.e. the volume of each BR *region is either 0 or inverse polynomially large.*

Proposition 6 ([21, 22])**.** *For a game G satisfying Assumption 2, there is an algorithm which finds an ε-approximate Stackelberg strategy for player A with $Q = \text{poly}(M, N, 1/\varepsilon)$ queries to $\mathsf{BR}(\alpha)$.*

We note that while such algorithms can indeed obtain tighter approximation guarantees in terms of ε (e.g. $O(\log(1/\varepsilon))$), the query complexity is still inverse polynomially related to the best response region volumes; we consider only ε-approximate equilibria due to challenges which are inherent to the no-regret learning setting, as the precision with which we can simulate a query is constrained by our time horizon. The key to our approach is to play according to a mixed strategy α until it saturates the relevant window of the learner's history, which induces them to play a best response. Against no-adaptive-regret learners, a best response will be induced quickly, as their regret is bounded even over small windows. However, for arbitrary no-regret learners, we have no promises other than the cumulative regret bound, which may require saturating the entire history for each query.

Theorem 9. *Suppose $\mathcal{E} \subseteq \Phi$. For a game satisfying Assumption 2, there is an algorithm which finds an ε-approximate Stackelberg strategy in $\text{poly}(1/\varepsilon)^Q$ rounds against any anytime-no-Φ-regret learner, and in $\text{poly}(Q/\varepsilon)$ rounds against any no-adaptive-Φ-regret calibrated for $T = \Theta(\text{poly}(Q/\varepsilon))$, where $Q = \text{poly}(M, N, 1/\varepsilon)$.*

Proof. We recall that the SU algorithm from [21] finds initial points $\alpha^*(b_i)$ in each best response region via random sampling, which takes takes $1/\text{poly}(\varepsilon^{-1})$ queries in expectation. Then, upon calibrating for $O(\log(1/\varepsilon))$ bits of precision SU makes $\text{poly}(M, N, \log(1/\varepsilon))$ queries, each of which can be taken to be a point on some grid of spacing $1/\text{poly}(\varepsilon^{-1})$ within the simplex by the precision condition. The computed approximate Stackelberg strategy is then the optimal such point on the grid.

We first describe our strategy for simulating each query against an arbitrary anytime-no-regret learner; as $\mathcal{E} \subseteq \Phi$, we can restrict to considering only no-external-regret learners, as these regret constraints will always be satisfied. To implement a query q, greedily play the action whose historical frequency of play is the furthest below its target frequency in q. After $O(\text{poly}(1/\varepsilon))$ rounds, the historical distribution will be within $1/\text{poly}(\varepsilon^{-1})$ of q, and continuing the greedy selection strategy indefinitely will ensure that the history remains in a $1/\text{poly}(\varepsilon^{-1})$-ball around q. Let t_q be the time at which this occurs. After maintaining the greedy strategy for q for an additional $\omega(t_q^c)$ rounds, the anytime regret bound ensures that most frequently played item must indeed be the best response response to some point in the ball around q, provided that this ball is contained entirely inside some best response region R_j. For the sampling step, a taking sufficiently fine grid (but still $1/\text{poly}(\varepsilon^{-1})$) ensures that random sampling still suffices to find a point in each best response region even if our queries may be adversarially perturbed to neighboring points on the grid, as each region is convex and has volume at least $1/\text{poly}(\varepsilon^{-1})$. To address the issue for the line search steps, it suffices to take an additional step along each search conducted by SU before termination, where we then take each hyperplane boundary estimate to be one step inward along the grid from where our search terminates, maintaining a buffer between each hyperplane estimate in which all our points of uncertainty must lie. This adds at most a constant factor to our query complexity, and impacts our approximation by $1/\text{poly}(\varepsilon^{-1})$, which then yields us a runtime of $\text{poly}(1/\varepsilon)^Q$ rounds.

For the case of a no-adaptive-regret learner, suppose such an algorithm is calibrated for $T = O(Q^{C_1}(1/\varepsilon)^{C_2})$; then, over any window of length W its regret is at most $O\left((Q^{C_1}(1/\varepsilon)^{C_2})^c W^{-1}\right)$. Taking $W = \omega((Q^{C_1}(1/\varepsilon)^{C_2})^c)$ yields a per-round regret of at most $o(1)$ over the window, and so an algorithm must play a best response in $W - o(W)$ of the rounds. For sufficiently large C_1 and C_2, each W is large enough to yield the same precision we required for the anytime case, where now we greedily play the action whose frequency is furthest below its target *since our previous query terminated*, which allows us to again simulate the $O(Q)$ queries in $\text{poly}(Q/\varepsilon)$ rounds (accounting for the robustness checks) while yielding $\Theta(WQ) = o(T)$. □

2.5.2 Efficiency Separations for Mean-Based and No-Swap Algorithms

We show here that the exponential dependence for mean-based algorithms is necessary: there exist games where learning the Stackelberg strategy *requires* exponentially many rounds against a particular mean-based algorithm. The algorithm we introduce for purposes of this analysis, Rounded Mean-Based Doubling, is an extension of the classical Multiplicative Weight Updates algorithm (MWU; see [31] for a survey). It is modified to satisfy two key properties: it has an anytime $O(\sqrt{t})$ regret bound, and all actions with sufficiently far-from-maximal cumulative reward are played with probability 0.

Algorithm 2 Rounded Mean-Based Doubling (RMBD)

Initialize and run MWU for $T_1 := 2$ rounds and n actions.
Let $T_2 := 2T_1$ and $i := 2$.
while $T_i \leq T$ **do**
  Initialize MWU for T_i rounds and n actions.
  Simulate running MWU for T_{i-1} rounds, using the average of the first T_{i-1} rewards each round.
  For T_{i-1} rounds, run MWU with action probabilities rounded to multiples of $4\gamma = \tilde{O}(T_i^{-1/2})$.
  Let $T_{i+1} = 2T_i$ and $i := i + 1$.
end while

Lemma 1. *When running* RMBD *for T rounds, the following hold at any round $t \leq T$:*

- RMBD *has cumulative regret $\tilde{O}(n\sqrt{t})$;*
- *If action j has the highest cumulative reward and $\sigma_{i,t} \leq \sigma_{j,t} - \tilde{O}(\sqrt{t})$, then action i is played with probability 0 at round t.*

Proof. Let $C\sqrt{t}$ bound the regret of MWU over t rounds (where $C = O(\sqrt{\log n})$), and let $D = \sqrt{2}C + \tilde{O}(n)$. We can bound the regret of RMBD over T_i rounds by $D\sqrt{T_i}$ via induction (which holds trivially at T_1). Suppose it holds for some T_i. Let $R(T_i)$ be the true reward obtained by RMBD over T_i rounds, which is at least $\sigma_{j^*,T_i} - D\sqrt{T_i}$, where σ_{j^*,T_i} is the cumulative reward of the best action over T_i rounds. Consider our simulation of MWU over T_i rounds using the average reward function. As the reward function is identical each round, and the cumulative reward for

each action j is equivalent under averaging, the measured reward $\hat{R}(T_i)$ from the simulated run is at most σ_{j^*,T_i} after T_i rounds. Upon continuing to run this instance of MWU for an additional T_i rounds, the regret bound ensures that the total measured reward $\hat{R}(T_{i+1})$ is at least $\sigma_{j^*,2T_i} - C\sqrt{2T_i}$. Rounding probabilities contributes at most an additional $2n\gamma T_i$ to the regret; it suffices to implement rounding by reallocating probability mass from any $p_{i,t} < 2\gamma$ onto other actions arbitrarily, to avoid renormalization. The total reward of RMBD over $2T_i = T_{i+1}$ is given by its cumulative reward at T_i, as well as the additional reward obtained by the MWU instance over the next T_i rounds, and so we have that

$$\begin{aligned} R(T_{i+1}) &= R(T_i) + \hat{R}(T_{i+1}) - \hat{R}(T_i) \\ &\geq \sigma_{j^*,T_{i+1}} - D\sqrt{T_i} - C\sqrt{2T_i} - 2n\gamma T_i \\ &\geq \sigma_{j^*,T_{i+1}} - D\sqrt{T_{i+1}}, \end{aligned}$$

which yields the bound for every T_i. We can extend this to any $t \in [T_i, T_{i+1}]$ with at most a factor 2 increase to cumulative regret.

To bound the selection frequency of actions with suboptimal cumulative reward, we recall the mean-based analysis of MWU given in Theorem D.1 from [23], which shows that the selection frequency $p_{k,t}$ for action k at time t is at most $\gamma = \frac{2\log(\sqrt{T\log n})}{\sqrt{T\log n}}$ if $\sigma_{k,t} \leq \sigma_{j,t} - \gamma T$ for the action j with highest cumulative reward. As such, any action whose cumulative reward $\sigma_{k,t} \leq \sigma_{j,t} - \tilde{O}(\sqrt{t})$ will be played with probability 0. □

While exponential time will be necessary against RMBD for the games we construct, we show that it is still possible to efficiently learn the Stackelberg strategy against any no-*swap*-regret learner.

Theorem 10. *There is a distribution over games $\mathcal{D}$ such that for a sampled game G:*

- *For any no-swap-regret learner used by the opponent, there is a strategy for the leader which yields an average reward of* $\mathsf{Stack}_A - \varepsilon$ *in* $T = \mathrm{poly}(M/\varepsilon)$ *rounds.*
- *There is a mean-based no-regret algorithm such that, when used by the opponent, there is no*

strategy for the leader which yields an average reward of $\mathsf{Stack}_A - \varepsilon$ *over* T *rounds unless* $T = \exp(\Omega(M))$.

Our construction includes a set of actions for player B which are best responses to pure actions from player A, and one such pure strategy pair will necessarily constitute the Stackelberg equilibrium; identifying each best response suffices for player A to identify the Stackelberg strategy. The game also includes a number of *safety* actions for player B, which yield no reward for player A with any strategy, yet allow player B to "hedge" between multiple actions of player A. This poses a barrier to optimizing against a mean-based learner using RMBD: the history must be heavily concentrated on a single action to observe the best response, and as such the history length must grow by a constant factor for each observation. However, against a no-swap-regret learner, it suffices for the optimizer to only play each action for a polynomially long window in order to identify the learner's best response; we track the accumulation of a "swap-regret buffer" for any other action and show that it cannot be too large, limiting the number of rounds it can be played when it is not a current best response.

Chapter 3: Recommendations for Agents with Adaptive Preferences

The results of this chapter are based on joint work with Arpit Agarwal [32, 33].

3.1 Overview

Suppose you manage an online platform that repeatedly provides content recommendations to visitors, such as suggestions for videos to watch or items to purchase. Carefully-calibrated recommendation systems are an integral part of online platforms of this form across a variety of domains ranging from e-commerce and entertainment to news and social networks, and often our goal is to elicit favorable interactions from users, which yield us some reward (in the form of ad revenue, watch time, purchases, or other metrics). At any given time, users of such platforms often expect to be presented with a limited set of options to choose from (a "menu" of recommendations), selected by the platform from a much larger universe of content — as concrete examples, we might consider choosing accounts to show on the "Explore" or "For You" pages in social platforms like Instagram, Twitter, or TikTok, or alternatively selecting channels to recommend on the homepage of a video platform like YouTube. Upon receiving this menu of recommendations, the user will then make choices for interaction (likes, views, purchases) based on their *preferences*, which we generally will not know in advance, and which may not align with our own objectives as a platform (e.g. if a user prefers to watch videos which garner lower ad spend). Crucially, the preferences of the user are typically not fixed a priori, and in fact may be *adaptive* as a function of their history of interactions with the platform — it well-documented that user preferences change over time in response to content recommendations, which can lead to self-reinforcing patterns of content consumption with undesirable side-effects when recommendations are determined using traditional statistical approaches (which generally do not consider adaptive preferences) [34, 35, 36, 37]. These

preference-recommendation feedback loops — often termed "rabbit holes", "echo chambers", or "filter-bubbles" — can amplify bias, spur political polarization, and increase homogeneity in a manner which negatively impacts user experiences on the platform [38, 39, 40], and can also drive content providers to leave [41]. This motivates the design of recommendation algorithms that can account for adaptive preferences, which yields an intricate optimization problem: the recommendations we give to a user today will affect not only their immediate choices but also their downstream preferences, shifting their choices from other recommendations in the future.

In this chapter, we introduce a formal model for this problem and give a series of optimization algorithms as well negative results. Our aim is to eludicate the inherent difficulties posed by adaptivity and understand when optimization can be provably feasible, and to this end we make a number of simplifications to the vast design landscape for real-world recommendation systems. Departing from the fixed-preference statistical settings of "classical" recommendation methods such as collaborative filtering, where preference information is aggregated across a large population of agents, we instead consider an ongoing sequence of interactions with a single agent, which may also be more appropriate for the personalization patterns and interaction frequencies observed in modern platforms. In our setting, we assume that there is a total universe of n items, and that in each round (for T rounds total) we must recommend a size-k menu $K_t \subseteq [n]$ of items (for a fixed k) to the agent, who will then choose a single item from the menu. The choice of the agent will be probabilistic, and will depend on their current preferences for each item in the menu, which are in turn determined by functions of the agent's choice history; as we take the universe of items to be fixed and allow repeated recommendations across rounds, it is perhaps more natural for us to view items as corresponding to accounts, channels, or brands rather than posts, videos, or atomic purchases. We will soon characterize the agent's choice model more explicitly; in short, the agent will have "preference scores" $f_i(v_t) \in [0, 1]$ for each item $i \in [n]$, where $v_t \in \Delta(n)$ is what we refer to as the "memory vector" of the agent at round t (which is a representation of their historical item choice distribution, possibly recency-biased), and choice probabilities in each round will simply be proportional to the preference scores for the k items shown in the menu. Upon observing the

agent's selection $i_t \in K_t$, we then receive bandit reward feedback $r_t(i_t) \in [0, 1]$, where r_t is linear over $\Delta(n)$ and can be chosen adversarially in each round, and our goal will be to optimize our rewards from the agent's choices over T rounds.

We evaluate our algorithms from the perspective of regret minimization with respect to some *target set* S, and we strive for per-round computational efficiency as well. We will allow the preference scoring functions f_i to be quite general, and we will see shortly, this results in are a number of nuances emerging with regards to our choice of target set. Observe that for $k = 1$ the problem collapses to the classical adversarial multi-armed bandit problem, as the agent will choose the single shown item deterministically. We therefore assume that $k > 1$, which yields drastically more intricate choice dynamics due to the role of the agent's adaptive preferences; we are now playing a bandit problem in which we cannot choose items directly, and the set of *instantaneously realizable distributions* of items we may induce (when considering any distribution over menus we might choose to sample from ourselves in a given round) is shifting as a function of the history according to unknown and nonlinear dynamics. As a result, in general it will not be possible to do as well as the best individual item in terms of reward; even if scores were always *uniform*, an agent would never choose an item more than once every k rounds. Several other desirable target sets turn out to be computationally infeasible even offline, or suffer from linear regret lower bounds, and it will ultimately be most appropriate to consider target sets which are (convex) subsets of the simplex over items $\Delta(n)$ in order to obtain sublinear regret.

Given a target set $S \subseteq \Delta(n)$, our high-level approach will be to formulate online recommendation as a nonlinear reinforcement learning problem over the space of agent preferences, where the "state" corresponds to the agent's memory vector v_t and our "'actions" are the choices of menus we recommend in each round. While nonlinear high-dimensional reinforcement learning problems are often hopelessly intractable, our salvation will come from the synchrony between v_t and the agent's choices i_t which determine our per-round rewards $r_t(i_t)$, and through our choices of menus we will be able to guide the agent to implicitly conduct online convex optimization over S on our behalf. A key dimension of the problem affecting both our regret rates as well as our feasible targets for

sublinear optimization will be the "memory horizon" of the agent, as determined by the discount factor γ applied to each prior round when constructing the memory vector v_t from the agent's sequence of past choices. When the preference functions f_i are unknown, we will need to *learn* the agent's current preferences in tandem with our optimization process, which is aided by the stability of v_t provided by longer memory horizons (e.g. growing with T, possibly sublinearly); yet in some cases we will need to *steer* the agent's memory towards more favorable preferences (in terms of the item choice distributions which we can induce), which becomes more difficult when the memory horizon is large (such as in the "uniform memory" case with no discounting, i.e. $\gamma = 1$).

3.1.1 Our Results

The organization and contributions of this chapter are summarized as follows:

- In Section 3.2 we introduce our interaction model for providing recommendations to agents with adaptive preferences along with key definitions and formalisms which we use throughout the chapter, such as:
  - the set of *instantaneously realizable distributions* of items at a given memory vector ($\mathrm{IRD}(v_t)$) and the set of *everywhere* instantaneously realizable distributions (EIRD);
  - memory horizons and discounted-memory update rules; and
  - smoothness and dispersion properties for preference functions.
- In Section 3.3 we give a series of negative results which motivate the consideration of EIRD as a potential target set for regret minimization as well as a primary focus on memory horizons which scale with T, including:
  - a linear regret lower bound for competing with the best fixed menu distribution;
  - a linear regret lower bound for competing with the IRD set for the uniform vector;
  - an NP-hardness-of-approximation result for optimizing over "stabilizable" memory vectors (i.e. contained in their own IRD set); and

- a superpolynomial (in n) regret lower bound for EIRD when the agent's memory horizon is sufficiently small.

- In Section 3.4, we give a sequence of preliminary results which build towards our primary algorithms in Sections 3.5 and 3.6, showcasing the structure of our approach under increasingly relaxed informational assumptions, including:

  - an explicit characterization of $\mathrm{IRD}(v)$ sets in terms of preference scores $\{f_i(v)\}$, which enables efficient sparse construction of a menu distribution which realize an item distribution $x \in \mathrm{IRD}(v)$, and also implies non-emptiness of EIRD when $f_i(v) \geq k/n$ is satisfied for every $i \in [n]$ and $v \in \Delta(n)$;
  - an algorithm for regret minimization over EIRD when both the functions $\{f_i\}$ and the EIRD set are known explicitly, under full-information (vs. bandit) feedback for r_t;
  - a similar algorithm which removes the requirement for advance knowledge of EIRD, provided that the functions $\{f_i\}$ are known;
  - an algorithm for bandit regret minimization which does not need to know EIRD in advance, and additionally obtains graceful performance degradation in terms of imprecision in our knowledge of preferences $\{f_i\}$; and
  - characterization and examples for the class of *locally learnable* functions $\{f_i\}$ which enables a "learn-then-optimize" approach with $\tilde{O}(T^{3/4})$ regret for uniform-memory agents, along with an argument that restricting to this class is necessary in order to use such an approach whenever T is sub-exponential in n.

  Additionally, we show that EIRD contains all sufficiently high-entropy distributions, highlighting the synergy between the goals of regret minimization and avoiding auxiliary harms caused by a lack of diversity in recommendations.

- In Section 3.5, leveraging the tools from the previous section, we give an algorithm which obtains $\tilde{O}(T^{1-c/4})$ regret with respect to the EIRD set, provided that the agent's memory

horizon is at least $\Omega(T^c)$ for any $c \in (0, 1]$ and that their preference functions $\{f_i\}$ are Lipschitz. Bypassing the previously-mentioned restrictions to locally learnable functions for a "learn-then-optimize" approach, we instead maintain continually-updating local constant approximations of preferences whose accuracy is preserved in the neighborhood around the current memory vector, recovering the $\tilde{O}(T^{3/4})$ rate for uniform-memory agents under a much broader class of preference functions.

- In Section 3.6 we consider the goal of optimizing over regret benchmarks can include points substantially beyond the boundaries of EIRD under additional restrictions on scoring functions, and we give algorithms which can minimize regret with respect to nearly the simplex $\Delta(n)$ under appropriate conditions. To accomplish this, we:
  - introduce the notion of *scale-bounded* functions $f_i(v)$ whose values are bounded within a multiplicative factor of $\lambda + (1 - \lambda)v_i$, where λ can be arbitrarily small;
  - consider a target set $\Delta^{\phi}(n)$ which we refer to as the *ϕ-smoothed simplex*, where $\Delta(n)$ is mixed with $\phi = O(\lambda)$ uniform noise; and
  - give an algorithm which obtains $\tilde{O}(T^{1-c/2} + T^{1/2+c/2})$ regret with respect to $\Delta^{\phi}(n)$ whenever an agent has a memory horizon of $\Theta(T^c)$ and has preferences functions which are scale-bounded.

  Additionally, we give an algorithm for this setting which obtains $o(T)$ regret over $\Delta^{\phi}(n)$ under constant memory horizons if reward distributions for r_t cannot change too frequently.

Taken as a whole, in addition to providing a model of recommendations for adaptive agents and several algorithms for efficient regret minimization, our results highlight a number of delicate tradeoffs between the choice of target set, the speed at which an agent's memory is updated, and structural assumptions on preference functions, which may be informative for agent preference modeling more broadly.

3.1.2 Related Work

Here we briefly summarize several lines of research which share conceptual or technical connections to the setting and results of this chapter.

Empirical investigations of recommendation feedback loops. There is a long line of work on studying bias, feedback loops, and "echo-chamber" effects at the population level in recommendation systems [34, 36, 37]. A substantial body of evidence has emerged in recent years indicating that recommendation systems can create feedback loops which drive negative social consequences. [39] observed that users accessing videos with extreme political views are likely to get caught in an "ideological bubble" in just a few clicks, and [42] explore the role of recommendation algorithms in creating distrust and amplifying political polarization on social media platforms. By investigating a real-world e-commerce dataset, [38] study the way in which recommendation systems drive agents' self-reinforcing preferences and lead them into "echo chambers" where they are separated from observing a diversity of content. [43] conduct a meta-analysis over many datasets which focuses specifically on the "rabbit hole" problem by means of exploring "taste distortion" of agents who observe recommendations which are more extreme than their current preferences. Therealso is prior work leveraging reinforcement learning for recommendations to maximize long-run rewards [44, 45, 46], typically with a focus on empirical evaluation. Such results motivate investigating these dynamics from game-theoretic and learning-theoretic foundations.

Models of adaptivity in recommendation systems. A number of recent works from the recommendation systems literature have explored the role of collaborative filtering algorithms for various models of agent behavior, aiming to understand how feedback loops in recommendation patterns emerge, the harms they cause, and how they can be corrected [47, 48]. A common theme is homogenization of recommendations across a *population* of users, which can lead to exacerbation of biased utility distributions for minority groups [37], long-run utility degradation [40], and a lack of traffic to smaller content providers which results in them being driven to exit the platform

[41]. Our work indirectly addresses this phenomenon by encouraging diverse recommendations in many cases, but our primary focus is from the perspective of a single agent, who may be led down a "rabbit hole" by an algorithm which optimizes for their immediate engagement. There has also been works which consider preference dynamics in recommendation systems, with an emphasis on linear update models. [49, 50] consider a model for political preference dynamics where vector preferences drift towards the agreement or disagreement on randomly drawn issues, and [51] study a similar model in the context of personalized recommendations with single-item menus. A related model is also considered by [52] to study the influence of recommendations on genre formation, and interactions between payment incentives and self-reinforcing preferences are studied in a bandit setting by [53]. While some of this work has adopted the regret minimization perspective for multi-agent recommendation problems [50, 51, 52], strong assumptions are required on the recommendation setting and model of preference evolution, in which only a single recommendation is given each round, which then results in a linear update to preferences. In contrast, the setting we consider allows for unknown and potentially highly nonlinear preference dynamics which can express the complex interactions often present between items (e.g. substitute and complement effects, relevant sequential orderings, or genre correlations), as well as to accommodate the practical constraint faced by many systems in which an agent must be shown a menu of multiple items to choose from. This yields a problem which is non-trivial even in the case of a single agent. Here we also note that our discounted-memory and scale-bounded assumptions can be respectively interpreted as requiring that (i) agents are recency-biased, and (ii) preferences generally increase with familiarity, both of which are widely assumed in modeling of agent preferences (see [35] for a more in-depth overview).

Dueling bandits and other dynamic bandit problems. The "dueling bandits" problem, initially proposed as a model for similar recommendation systems challenges [54, 55], and which has been generalized for sets larger than two [56, 57], considers a similar setting in which bandit optimization is conducted with respect to the preference model of an agent, occasionally represented via an

explicit parametric form. Here, one presents a set of choices to an agent, then receives only *ordinal* feedback about the relative rewards of the choices, and must optimize recommendations with regret measured against the best individual choice. In contrast to our setting, these works consider preferences which are fully determined *a priori*, and do not change as a function of item history or exhibit preference feedback loops.

Problems where future rewards are causally affected by past actions have also been studied in the stochastic multi-armed bandit setting [58, 59, 60, 61, 62, 63, 64, 65, 66]. Most recent work on such problems has focused on specific models for reward evolution with motivations such as agent satiation, agent boredom, and congestion, and considers a single action being chosen in each round. In contrast, our setting allows for adversarial rewards, and preference evolution is determined by interplay between the multiple items we choose in each menu.

Some conceptual features of our setting resemble elements of other well-studied online problems as well, including the restricted exploration ability for limited switching problems (e.g. [67]), and the contracting target set for chasing nested convex bodies (e.g. [68]).

Online Stackelberg problems. A number of works in recent years explore online problems where an agent responds to the decision-maker's actions, influencing their reward. The performative prediction setting, introduced in [69], captures settings in which a deployed classifier results in changes to the distribution itself, in turn affecting performance. This work has been extended to handle stochastic feedback [70] and notably, to a no-regret variant [71] which involves learning mapping between classifiers and distribution shifts, which bears some conceptual similarities to our procedure for locally learning an agent's preference model, and well a simultaneous learning model by [72], who study the effects of using a learning rate which is either much faster or much slower than that of the agents which one aims to classify, drawing connections to equilibrium concepts as well. A parallel line of work has begun to explore the problem of designing optimal strategies in a repeated game against agents who *adapt* their strategies over time using a no-regret algorithm. In auction problems, [73] study the extent to which an auction designer can extract value from

bidders who use different kinds of no-regret algorithms. More generally, [74] connect this line of investigation to Stackelberg equilibria for normal-form games; our results in Chapter 2 fall into this literature as well. Further, the "revealed preferences" literature involves a similar requirement of learning a mapping between actions and agent choices [75, 76] in order to induce desired choices by agents. Our work extends this notion of strategizing against adaptive agents to a recommendations setting, with novel formulations of adaptivity and regret to suit the problem's constraints.

3.2 Preliminaries

Throughout the chapter, we use $\Delta(n)$ $\mathbf{u}_n$ to denote the uniform distribution on n items and $d_{TV}(v, v')$ for the total variation distance between distributions. We use the ℓ_2 norm unless specified otherwise (e.g. as $\|x\|_1$), and we use $B_\epsilon(x)$ to denote the ball of radius ϵ around a point x.

3.2.1 Interaction Model

Here, we introduce our setting and interaction model for the online recommendations problem for an agent with adaptive preferences. At any time $t \in [T]$, there is some *memory vector* $v_t \in \Delta(n)$, which expresses some function of the prior selections of the agent. The *preference model* of an agent is a mapping $M : \Delta(n) \to [0, 1]^n$ which assigns scores $M(v)_i = f_i(v)$ according to preference functions $f_i : \Delta(n) \to [0, 1]$ for each item. An instance of our problem is specified by a set of items $[n]$, a menu size $k < n$, a preference model M, a memory update rule U (to be defined in Section 3.2.3), and a sequence of reward vectors $r_1, \ldots, r_T$. In each round $t \in [T]$:

- the recommender chooses a menu K_t, consisting of k distinct items from $[n]$, which is shown to the agent;
- the agent selects one item $i_t \in K_t$, chosen at random according to the distribution given by:

$$p_t(i; K_t, v_t) = \frac{f_i(v_t)}{\sum_{j \in K_t} f_j(v_t)};$$

- the memory vector is updated to $v_{t+1} = U(v_t, i_t, t)$ by the update rule;

- the recommender receives reward $r_t(i_t) \in [0, 1]$ for the chosen item.

The initial $v_1 \in \Delta(n)$ can be chosen arbitrarily. We assume each f_i is unknown to the recommender, but that the memory update rule U is known. The goal of the recommender is to minimize regret over T rounds with respect to some *target set* $S \subseteq \Delta(n)$. For any such S, the regret of an algorithm $\mathcal{A}$ with respect to S is

$$\mathsf{Reg}_S(\mathcal{A}; T) = \mathbb{E}\left[\max_{x \in S} \sum_{t=1}^{T} \langle r_t, x \rangle - r_t(i_t)\right]$$

where i_t is the agent's item choice at time t resulting from $\mathcal{A}$, and where the expectation is taken over internal randomness of $\mathcal{A}$ as well as the agent's choices. We discuss challenges and tradeoffs related to our choice of target set in Section 3.3, which will motivate our restriction to considering targets of the form $S \subseteq \Delta(n)$ in "item space" (rather than e.g. the best fixed menu).

3.2.2 Realizable Distributions

For a preference model M and memory vector v, let $\mathrm{IRD}(v, M)$ denote the set of *instantaneously realizable distributions* for v, given by

$$\mathrm{IRD}(v, M) = \mathrm{convhull}\left\{p(K, v) : K \in \left[\binom{n}{k}\right]\right\}$$

where K is a k-item subset of $[n]$ and $p(K, v)$ denotes the item selection distribution of an agent with memory v conditioned on being shown a menu K, which is given by

$$p(i; K, v) = \frac{f_i(v)}{\sum_{j \in K} f_j(v)}$$

for each item i in K (and 0 otherwise), where f_i is the preference scoring function for any item i. Note that this expresses the possible item choice distributions of an agent with memory v resulting from all possible menu selection strategies by the recommender, as any distribution over menus yields a convex combination of item distributions $p(K, v)$. The set of *everywhere instantaneously*

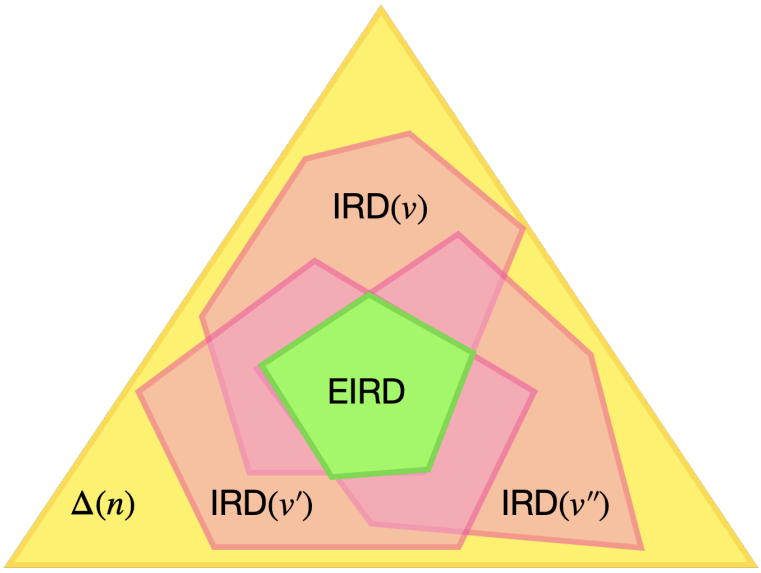

Figure 3.1: Illustration of the EIRD set and several IRD sets inside of the simplex.

realizable distributions is given by

$$\mathrm{EIRD}(M) = \bigcap_{v \in \Delta(n)} \mathrm{IRD}(v, M).$$

We often simply denote these sets as EIRD and $\mathrm{IRD}(v)$ when M is clear from context; we will consider EIRD as our regret minimization target set for several of our results. Note that EIRD is convex, as it is an intersection of convex $\mathrm{IRD}(v)$ sets. As a toy example, consider the case where preferences are fixed at $f_i(v) = 1$ for all v; here, every IRD set is equivalent to EIRD, which is given by the convex hull of all vectors in $\Delta(n)$ with mass $1/k$ on exactly k items.

We will assume that scoring functions f_i are bounded in $[\lambda, 1]$ for some constant $\lambda > 0$ which captures exploration on behalf of the agent, which we refer to as the "dispersion" parameter (i.e. such preferences are λ-dispersed). In Sections 3.3 and 3.5 we will assume $\lambda \geq k/n$, which ensures that EIRD is non-empty (and in particular contains $\mathbf{u}_n$); Section 3.6 allows λ to be arbitrarily small (alongside a minor reparameterization of the role of λ for notational convenience).

3.2.3 Discounted Memory Agents

Definition 13 (Discounted Memory Updating)**.** *Under the γ-discounted memory update rule U_γ, for some $\gamma \in [0, 1]$, when an item i_t is selected at round t, the memory vector v_t is updated to $v^{t+1} = U_\gamma(v_t, i_t, t)$, with*

$$v_{t+1}(i) = \frac{\sum_{s=1}^{t} \gamma^{t-s} \cdot \mathbf{1}(i = i_s)}{\sum_{s=1}^{t} \gamma^{t-s}}.$$

Throughout the chapter we consider agents whose memory update rules are γ-discounted. Taking $\gamma = 1$ yields the "uniform memory" case, where v_t is simply the empirical distribution of choices in $\Delta(n)$. As in many settings with discounting, we can view values γ closer to 1 as corresponding to larger "effective horizons" for memory; for any $\gamma \leq 1 - o(1)$ we refer to the quantity $1/(1-\gamma)$ as simply the *effective memory horizon*, often also expressed as $1/(1-\gamma) = T^c$ for some $c \in (0, 1]$; we overload $c = 1$ for the uniform memory case, and assume γ (or c) is known.

3.2.4 Smooth Preference Models

Some of our results consider preference models with the property that each scoring function is Lipschitz in addition to λ-dispersed, which we refer to as *smooth* preference models.

Definition 14 (Smooth Preference Models)**.** *A preference model M is (λ, L)-smooth if each scoring function f_i takes values in $[\lambda, 1]$ and is L-Lipschitz over $\Delta(n)$ with respect to the ℓ_1 norm.*

This property allows for quite a broad class of functions, and is satisfied by each of the "locally learnable" classes we consider in Section 3.4.6 (e.g. low-degree polynomials) with appropriate parameters. Despite its generality, we show in Section 3.5 that this assumption alone is sufficient to enable us to always maintain an accurate *local* approximation of the model, provided that the agent's memory vector does not change too quickly, by periodically implementing a query learning routine. For convenience, we assume that preference scores are always normalized to have a constant sum $\sum_i f_i(v) = C$ for some C and any v; this can be relaxed in each result up to $\text{poly}(n, L)$ factors.

3.3 Benchmarks and Barriers

As we saw in Chapter 2, where our space of outcomes depended on the opponent's choice of learning algorithm, here the landscape of feasibility for efficient regret minimization will depend on the agent's preference dynamics as well. As we alluded to previously, it is fairly immediate to see that the standard experts or adversarial bandits benchmark of the best fixed item will not be feasible, and alternative benchmarks must be considered.

Proposition 7. *Even if rewards r_t are fixed across rounds and preferences $\{f_i\}$ are given by known constant functions, no algorithm obtains $o(T)$ regret with respect to $\Delta(n)$ for any $k \geq 2$.*

Proof. For preferences fixed at $f_i(v) = 1$ for all v, every IRD set (and thus the EIRD set) is given by the convex hull of all vectors in $\Delta(n)$ with mass $1/k$ on exactly k items. If $r_t(1) = 1$ and $r_t(i) = 0$ for every $t \in [T]$ and $i > 1$, then no algorithm obtains reward more than $T/2$ in expectation (as the agent never chooses item 1 with probability more than $1/2$ in a round), whereas the maximum reward for a distribution $x \in \Delta(n)$ is T. □

3.3.1 A Regret Lower Bound for Fixed Menus

Another natural benchmark to target, in the spirit of the "policy regret" benchmarks which are considered for problems such as contextual bandits, online MDPs, or online control, would be to compete with the performance of the best fixed menu (or distribution over menus). Unfortunately, we show that sublinear regret is impossible in this case (for the uniform-memory setting in particular), owing to the slow-moving nature of the memory vector whose value may significantly impact the agent's feasible choice distributions (via the IRD set) and subsequently the rewards we can obtain.

Theorem 11. *There is no algorithm which obtains $o(T)$ regret against the best menu distribution in $\Delta\left(\binom{n}{k}\right)$ when $\gamma = 1$, even if the preference model is known and is given by univariate linear functions.*

Proof. Let $M = \{f_i\}$ be the λ-dispersed memory model where the functions for items (a, b, c), and every other item i, are given by:

- $f_a(v) = \lambda + (1-\epsilon)(1-v_b)$;
- $f_b(v) = \lambda + (1-\epsilon)v_b$;
- $f_c(v) = \lambda + (1-\epsilon)v_c$;
- $f_i(v) = \lambda + (1-\epsilon)(1-v_b)$ for $i \notin \{a, b, c\}$;

for some $\lambda > 0$ and $\epsilon > \lambda$. Let $k = 2$. Consider a sequence of rewards $\{f_t\}$ which yields reward α to items (a, b) for each round $t \leq t^*$ and 0 to the rest, then in each step after t^*, yields a reward of β for item c, a reward of 0 for item b, and reward of $-\beta$ for every other item. Note the total expected reward for the following distributions:

$$R_{(a,b)}(T) = \alpha t^* - \beta(T - t^*)/2;$$
$$R_{(b,c)}(T) = \alpha t^*/2 + \beta(T - t^*)/2;$$

The bound for $R_{(a,b)}(t^*)$ follows from symmetricity of the resulting stationary distribution, given by the unique solution $v_a = 0.5$ to the recurrence:

$$v_b = \frac{\lambda + (1-\epsilon)v_b}{2\lambda + (1-\epsilon)}$$

which is approached in expectation for large T regardless of initial conditions for any constant λ. Symmetricity also results in balanced expectations for each item in $R_{(b,c)}$.

Consider the distribution p_{t^*} played by an algorithm $\mathcal{A}$ over the first t^* rounds, where t^* is large enough to ensure concentration. If $p_{t^*,a} + p_{t^*,b} \leq 1 - \delta$ for some constant δ, then for $\beta = 0$ the algorithm has regret $\delta\alpha t^* = \Theta(T)$ for any $t^* = \Theta(T)$. Further, if regret is not bounded, the menu (a, b) must be played in nearly every round, as other item placed in the menu has positive selection probability. As such, the empirical probability of b must be close to $1/2$.

After t^*, the algorithm cannot obtain a per-round expected utility which matches that of (b, c)

up to δ until a round t where either:

$$\frac{\lambda + (1-\epsilon)p_{t,c}}{2\lambda + (1-\epsilon)(p_{t,b} + p_{t,c})} \geq 1/2 - \delta$$

or

$$\frac{\lambda + (1-\epsilon)p_{t,c}}{2\lambda + (1-\epsilon)(1 - p_{t,b} + p_{t,c})} \geq 1/2 - \delta,$$

which requires the total number of rounds in which c is chosen to approach $t^*/2 - C \cdot \delta t^*$, where C is a constant depending on ϵ and λ. Let $T = 3t^*/2$, and so this cannot happen for small enough constant δ, resulting in a regret of $\delta\beta T/3 - \alpha T/3$ with respect to (b, c), which is $\Theta(T)$ when $\delta\beta > \alpha$. □

Given this, we will instead focus on regret benchmarks in "item space" rather than "menu space"; in addition to the regret lower bounds for the best fixed menu distribution, another motivation for this is that our rewards are determined by the items chosen by the agent rather than our recommendations, and choice distributions (viewed as stationary distributions for a stochastic process over memory vectors) are not guaranteed to quickly stabilize even if we hold menu distributions fixed.

3.3.2 A Regret Lower Bound for IRD($\mathbf{u}_n$)

As we will see in Section 3.3, the uniform distribution $\mathbf{u}_n$ is always feasible under mild conditions on the dispersion parameter λ, and one might hope that this could enable choosing choosing IRD($\mathbf{u}_n$) as a regret minimization target set. However, a similar construction for uniform-memory agents shows a linear regret lower bound for this set as well.

Theorem 12. *There is no algorithm which obtains $o(T)$ regret against the best item distribution in* IRD($\mathbf{u}_n$) *when $\gamma = 1$, even if preferences are known and given by univariate linear functions.*

Proof. First we give an example for which obtaining $o(T)$ regret against IRD(v, M) for the uniform vector $\mathbf{u}_n$ is impossible. Consider the memory model $M = \{f_i\}$ where:

- $f_1(v) = \lambda + 0.5 + \frac{n}{n-1} \cdot (v_1 - \frac{1}{n}) \cdot (0.5 - \lambda)$;
- $f_2(v) = \lambda + 0.5(1 - v_1 + \frac{1}{n})$;
- $f_i(v) = 0.5 + \lambda$ for $i > 2$.

Observe that at the uniform distribution where $v_1 = \frac{1}{n}$, all items have a score of $0.5 + \lambda$. If $v_1 = 1$, we have that:

- $f_1(v) = 1$, and
- $f_2(v) = \lambda + \frac{0.5}{n}$.

If $v_1 = 0$, we have that:

- $f_1(v) = \lambda + 0.5 - \frac{0.5-\lambda}{n-1}$, and
- $f_2(v) = \lambda + 0.5 \cdot (1 + \frac{1}{n})$

As scores linearly interpolate between these endpoints for any v_1, M is λ-dispersed, and scores lie in $[\lambda, 1]$. Let $k = 2$. Consider reward functions which give reward $\alpha > 0$ for item 1 in each round up to $t^* = T/2$, giving reward 0 to each other item; after t^*, a reward of $\beta > 0$ is given for item 2 while the rest receive a reward 0. The distribution which assigns probability $1/2$ each to item 1 and 2, with all other items having probability 0, is contained in $\text{IRD}(\mathbf{u}_n)$, as one can simply play the menu with both items. This distribution yields a total expected reward of

$$R_v = \frac{\alpha t^*}{2} + \frac{\beta t^*}{2}$$

over T steps. Consider the performance of any algorithm $\mathcal{A}$ which results in item 1 being selected with an empirical probability p over the first t^* rounds. At $t = t^*$, we have $v_{t^*,1} = p$; its total reward over the first t^* rounds is $\alpha p t^*$. For sufficiently large n and small λ, the score for item 2 is approximated by $f_2(v) = 0.5(1 - p)$ up to any desired accuracy. In future rounds $t \geq t^*$, the value

$v_{t,1}$ is at least $\frac{pt^*}{t}$, and so the score for item 2 is at most

$$f_2(v) = 0.5(1 - \frac{pt^*}{t}).$$

Each other item has a score of at least 0.5, yielding an upper bound on the probability that item 2 can be selected even if it is always in the menu, as well as a maximum expected per-round reward

$$R_t = \beta \cdot \left(\frac{0.5(1 - \frac{pt^*}{t})}{1 - 0.5\frac{pt^*}{t}} \right).$$

At time $T = 2t^*$, the instantaneous reward is at most

$$R_T = \beta \cdot \left(\frac{2 - p}{4 - p} \right),$$

which is also a per-round upper bound for each $t \geq t^*$. This bounds the total reward for $\mathcal{A}$ by

$$R_{\mathcal{A}} = \alpha p t^* + \beta t^* \cdot \frac{2 - p}{4 - p}.$$

We now show that for any p there exists a β such that $R_v - R_{\mathcal{A}} = \Theta(T)$. For any $p \leq \frac{1}{3}$, we have

$$R_{\mathcal{A}} \leq \frac{\alpha t^*}{3} + \frac{\beta t^*}{2},$$

and for any $p > \frac{1}{3}$ we have:

$$R_{\mathcal{A}} \leq \alpha t^* + \frac{5\beta t^*}{11}.$$

In the first case, we immediately have $R_v - R_{\mathcal{A}} \geq T\alpha/6$ for any β. In the second case, let $\beta \geq 22\alpha$.

We then have:

$$\begin{aligned} R_v - R_{\mathcal{A}} &\geq \frac{\beta t^*}{22} - \frac{\alpha t^*}{2} \\ &\geq T\alpha/4. \end{aligned}$$

The value of β can be determined adversarially, and so there is no algorithm $\mathcal{A}$ which can obtain $o(T)$ regret against $\text{IRD}(v)$. □

Beyond the barriers of slow-moving memory, an added difficulty is posed by the fact that points in $\text{IRD}(\mathbf{u}_n)$ need not be contained in their own IRD set; for such points, retaining a stable choice distribution and corresponding memory vector may be impossible, as all choice distributions may result in a "drift" away from our target distribution. As such, one important desideratum for points in our target set will be that they are each contained in their own IRD set. We say that any vector $v \in \text{IRD}(v)$ is *stabilizable*. In general, it appears hopeless to attempt to compete with a distribution which is *not* in its own IRD set; any point which *is* inside its IRD set is (approximately) stabilizable under long memory horizons for smooth preferences if it can be initially reached.

3.3.3 Hardness of Approximation for Optimal Stabilizable Distributions

Unfortunately, restricting to all stabilizable distributions is still likely insufficient to yield a tractable benchmark if we also desire computational efficiency, even for a static problem where rewards and preferences are known, via a hardness-of-approximation reduction for the problem of simply *finding* the best stabilizable distribution. We show that an instance of the "Max Independent Set" problem, which is NP-hard to approximate, can be encoded in a preference model for an agent, wherein optimizing reward corresponds to selecting any independent set. We suppose there is only 1 item which receives positive reward and will always be in the menu, and the objective corresponds to maximizing its score. Our construction operates by interpreting assignments of weight to items as proposals for possible independent sets, and then efficiently checks a graph for edges between the corresponding vertices; the score of item 1 is then proportional to the size of any valid independent

set for the graph.

Theorem 13. *Unless* P = NP*, there is no polynomial time algorithm which takes as input a circuit representation of a preference model M and linear reward function r, and approximates the per-round reward $r(v)$ of the best distribution $v \in \text{IRD}(v, M)$ contained in its own* IRD *set within a $O(1/n)$ factor.*

Proof. The Maximum Independent Set (MIS) problem is Poly-APX-Hard (see e.g. [77]), and so there is no constant factor polynomial time approximation algorithm unless P = NP. Given a graph G with $n-1$ vertices (which we can assume to have a unique maximum independent set V^* with at most $(n-1)/2$ vertices), we construct a circuit for a preference model M for n items where identifying the optimal item distribution contained in its IRD set corresponds to identifying the maximum independent set S^*. For our reward function, we assume that item 1 yields a constant reward of 1, with all other items yielding a reward of 0. The objective is then to maximize the probability of item 1 being selected, but the score $f_1(v) \geq \lambda$ will only be maximized when the memory mass not placed on item i is (near-)uniformly allocated across items corresponding to the maximum independent set (the other scoring functions can be constant at λ).

We describe our preference model in terms of a circuit for f_1 with both arithmetic and Boolean gates, where all other functions f_i are presumed to be constant at λ, and note that translation to a pure Boolean circuit is feasible with at most polynomial blowup. Let $N = \{2, \ldots, n\}$ correspond to the vertices of G. For any memory vector v, let $V = \{j > 1 : v_j \geq \frac{1}{n}\}$. Our function f_1 is given by

$$f_1(v) = \begin{cases} \frac{|V|(1-\lambda)}{n-1} + \lambda & V \text{ is an independent set in } G \\ \lambda & V \text{ is not independent in } G \end{cases}.$$

To construct a circuit for f_1, first we include gates g_j which output 1 for each j if and only if $v_j \geq \frac{1}{n}$ (and 0 otherwise). Given these gates, we also include gates e_{ij} for each edge in the graph which output 1 if and only if i has an edge with j (where $1 - e_{ij}$ then denotes no edge). We can construct

a gate g_{valid} which outputs 1 if and only if our set V is an independent set by taking the AND of all gates $(e_{ij} \oplus (g_i \wedge g_j))$, as well as a gate g_{count} which gives the count of proposed independent nodes by summing over gates g_i. Taking the product of g_{valid} and g_{count} then yields a counter for the size of V if V is independent, and 0 otherwise, which can then be arithmetically scaled to yield $f_1(v)$.

Let $\lambda = 1/((n-1)^2/k + 1)$. Given an independent set V, the highest reward obtainable by a memory vector $v \in \text{IRD}(v, M)$ which corresponds to V under f_1 can be expressed via menu times, where we must have that

$$\frac{k\frac{v_1}{f_1(v)}}{\frac{v_1}{f_1(v)} + \frac{(1-v_1)}{\lambda}} \leq 1$$

by Lemma 2. Solving $(k-1)v_1 = (1-v_1)\left(1 + \frac{|V|(1-\lambda)}{\lambda(n-1))}\right)$ for v_1 gives us that

$$\begin{aligned} v_1 &= \frac{\frac{1}{k-1} + \frac{|V|(1-\lambda)}{\lambda(n-1)(k-1)}}{1 + \frac{1}{k-1} + \frac{|V|(1-\lambda)}{\lambda(n-1)(k-1)}} \\ &= \frac{\frac{k|V|}{n-1} + 1}{\frac{k|V|}{n-1} + k}. && (\lambda(n-1)^2/k = 1-\lambda) \end{aligned}$$

This is at most $1/2$ for any $|V| < (n-1)/2$, and so further it remains possible to allocate menu time which yields $v_i \geq \frac{1}{n}$ for each $i \in V$, yielding that v_1 the maximum feasible reward for some $v \in \text{IRD}(v, M)$ when $|V|$ is the size of the largest independent set. The gradient of v_1 in terms of $|V|$ is given by

$$\frac{\partial}{\partial |V|} \frac{|V| + (n-1)/k}{|V| + (n-1)} = \frac{(n-1)(1-1/k)}{(|V| + (n-1))^2}$$

which is $\Theta(1/n)$ for $|V| \in [1, n/2]$. Any polynomial time algorithm which approximates v_1 to within a $O(1/n)$ factor must result in a memory vector v' which corresponds to an independent set V' which approximates V^* within a constant factor, which would yield a polynomial time constant

factor approximation algorithm for Maximum Independent Set. □

This suggests we should not expect to compete with all stabilizable distributions; one interpretation of this barrier is due to non-convexity, as the set of stabilizable distributions above a particular reward threshold may be disconnected. Fortunately, the EIRD set is convex and all contained points are self-stabilizable by definition, and thus is a natural set to target. In fact, we expect that EIRD is essentially the *canonical* target for this problem, and we view the previous theorem as evidence that expanding to target sets nontrivially larger than EIRD will require carefully tailored assumptions on preference models (such as those considered in Section 3.6).

3.3.4 A Quasi-Polynomial Lower Bound for Short-Term Memory

As we will see in Section 3.4, the EIRD set is indeed feasible as a regret benchmark, provided that the agent's memory horizon grows with T. The latter restriction appears essentially necessary to have any hope of regret minimization for a nontrivial benchmark (at least when T is not too large), due to the difficulty of estimating preferences when the agent's memory vector may move significantly in each round. As such, throughout the chapter we will assume that the agent's memory horizon is at least T^c for some $c > 0$ if the preference functions $\{f_i\}$ are unknown.

Here we show a regret lower bound with respect to EIRD for any algorithm over a time horizon T which is quasi-polynomial in n by constructing preference models in which the optimal strategy depends delicately on the current memory vector, and which simultaneously induces fast exploration over a discrete state space. Our approach is to observe that every feasible memory vector encodes a unique truncated history of length $O(\log n)$, resulting in an implicit state space of size $O(n^{\log(n)})$. We design preference models in which the optimal policy is implementable by inducing the uniform distribution at each round, which lies inside EIRD, yet requires identifying a specific set of alternate items to place in the menu deterministically at each state to maximize the selection probability of item 1. We show that any competitive strategy also necessarily explores many states with high probability, and so any algorithm will frequently reach states where it cannot identify the optimal menu distribution, which is defined on a per-state basis by a random process.

Theorem 14. *For any $\gamma \in (0, 1/2)$, there is a set of (λ, L)- smooth preference models $\mathcal{M}$ with $\lambda = O(1/n)$ and $L = \text{poly}(n)$ such that any algorithm must have expected regret $\Omega(T)$ for any $T \in O(n^{\log(n)})$ when the preference model M is sampled uniformly from $\mathcal{M}$.*

Proof. We consider a set of models and reward functions where item 1 yields a reward of 1 in each round, with all other items yielding a reward of 0. For $\gamma = \frac{1}{2^c}$ for some constant $c > 1$, note that the weight of any step in memory is larger than the sum of weights of all preceding steps, and thus a memory vector v_t exactly encodes the history of item selections for the first $t-1$ rounds. Let $h = \log_{2^c}(n)$; For t sufficiently larger than h, the sum of weights of steps 1 through $t-h$ is $\Theta(1/n)$. We will consider states s which are subsets of the space of memory vectors corresponding to each possible history truncated to the previous h steps, and which are bounded apart by a distance of at least $O(1/n)$. We will abuse notation and represent each memory vector v_t as its rounded state s_t. The behavior of the memory model is constant over each state, and smoothly interpolates between states; the model can be defined arbitrarily for infeasible memory vectors to satisfy $L = \text{poly}(n)$ Lipschitzness. Our process for generating $\mathcal{M}$ is as follows:

- Let $k = n/2$;
- Let $\lambda = \frac{1}{n-k+1}$;
- For each $s \in [n]^h$, let G_s be a set of $k-2$ items sampled uniformly at random from $\{2, \ldots, n\}$;
- Let $f_i(s) = \lambda$ if $i = 1$ or $i \in G_s$, and $f_i(s) = 1$ otherwise.

Observe that the optimal strategy π^* at s is to include item 1, each item in G_s, and any arbitrary final item. Note that each of the $k-1$ items with score λ is selected with probability

$$\begin{aligned}\Pr[i \text{ chosen} | f_i(s) = \lambda, \pi^* \text{ played}] &= \frac{\lambda}{1 + (k-1)\lambda} \\ &= \frac{1}{n},\end{aligned}$$

and so the expected reward per round is 1 as well. Note that π^* is consistent with a menu distribution which chooses the final item (after 1 and G_s) uniformly at random, which generates the uniform

distribution. As such, the uniform distribution lies inside EIRD (t is straightforward to define scores for infeasible memory vectors such that feasibility holds for any $v \in \Delta(n)$). We can also see that any menu inconsistent with π^* has expected reward at most:

$$\begin{aligned}\frac{\lambda}{2+(k-2)\lambda} &= \frac{\frac{1}{n-k+1}}{\frac{2n-2k+2}{n-k+1}+\frac{k-2}{n-k+1}} \\ &= \frac{3}{4n},\end{aligned}$$

as some item not in G_s must be included. To lower bound the regret of any algorithm, consider an arbitrary time t and history of item selections. By time t, at most distinct states have been observed thus far. Consider the following cases:

- The algorithm plays a menu consistent with π^* in every step from t to $t+h-1$;
- The algorithm plays a menu inconsistent with π^* at some step from t to $t+h-1$.

Suppose t is less than $T = \frac{1}{2} \cdot (\frac{n}{2})^h = O(n^{\log(n)})$. In the former case, there is a uniform distribution over $\frac{n}{2}$ items chosen by the agent at each round, and so the maximum probability of any given state is at most $(\frac{n}{2})^h$. Given that the set G_s is generated independently at random for each state, an algorithm has no information about G_s for unvisited states, and thus cannot improve expected reward beyond that obtained by choosing a random hypothesis for G_s, which incurs $O(\frac{1}{n})$ regret at round $t+h$. In the latter case, the step in which a menu inconsistent with π^* is played additionally incurs $O(\frac{1}{n})$ regret. Each event occurs once at least once in expectation every h rounds while $t < T$, and thus any algorithm must have $O(\frac{1}{nh})$ expected regret per round up to T. □

3.4 Adaptive Recommendations via Nested Online Optimization

Here, we illustrate the approach taken by our primary algorithm for EIRD Section 3.5 via a sequence of "warm-up" methods requiring increasingly relaxed assumptions about the information available to us regarding rewards as well as the agent's preferences. We begin with a result on the

structure of IRD sets which we use to enable efficient menu selection in each round of our regret minimization algorithms, and which further implies useful properties of the EIRD set.

3.4.1 Characterizing IRD via Menu Times

If we know the agent's current preferences $\{f_i(v_t)\}$, one immediate characterization of the set $\text{IRD}(v_t)$ is given by the definition of IRD sets as the convex hull of item distributions for every size-k menu. However, applying this naively to test if a distribution x belongs to IRD — or to construct a menu distribution which realizes x as an instantaneous choice distribution for the agent — involves explicitly enumerating all $\binom{n}{k}$ menus and solving a linear program with $\Theta(n^k)$ variables, which quickly becomes intractable even for moderate values of k.

To overcome this barrier, we introduce a notion of the *menu time* required by each item in order to induce a particular item distribution x, which allows us to directly characterize IRD sets, as well as efficiently construct menu distributions at each round with a greedy approach. At a memory vector v, for a target distribution x, the menu time for item i is given by

$$\mu_i = \frac{k \cdot \frac{x_i}{f_i(v)}}{\sum_{j=1}^{n} \frac{x_j}{f_j(v)}}.$$

Observe that these quantities always satisfy $\sum_i \mu_i = k$ for any v and x. Informally, the menu time μ_i corresponds to the "fraction of time" in which each item i must be included in the menu in order to realize x (with inclusions weighted to normalize across the sum-of-menu-score denominators) — we show that a distribution x can be realized from a memory vector v if and only if $\max_i \mu_i \leq 1$.

Lemma 2. *An item distribution x belongs to $\text{IRD}(v, M)$ if and only if we have that the menu time μ_i for each item is at most 1. If this condition holds, there is a $\text{poly}(n)$ time algorithm $\texttt{MenuDist}(v, x, M)$ for constructing a menu distribution z such that $\mathbb{E}_{K\sim z}[p(K, v)] = x$.*

Proof Sketch. If $x \in \text{IRD}(v, M)$, there exists some menu distribution z which yields x; converting this menu distribution to μ_i values by "crediting" a menu in proportion with its mass and the inverse of the sum of its item scores results in a menu time vector satisfying $\sum_i \mu_i = k$ and $\max_i \mu_i \leq 1$.

Given a menu time vector satisfying these conditions, we can construct such a distribution by greedily choosing a menu of the k items with highest remaining menu time and "charging" their remaining menu times at the same rate, breaking ties for the kth highest by charging and including at fractional rates. The number of items tied for kth highest remaining μ_i increases by 1 at each stage, and the highest initial $k-1$ items (with $\mu_i \leq 1$) will be included non-fractionally until tied for kth highest. The mass of each added menu in our final distribution z will be allocated proportionally to the sum of scores of items in the menu. This allows cancellation of the terms for sums of menu scores, resulting in a menu distribution where the selection probability of an item is proportional to its score $f_i(v)$ and the number of (fractional) stages in which it was added to the menu. As the latter number of stages in which an item is added to a menu is proportional to its menu time, and its menu time is proportional to $x_i/f_i(v)$, the induced item choice distribution is then proportional to x_i. □

`MenuDist`(v, x, M) directly implements this menu distribution construction, and is used by our algorithms throughout Section 3.4 as well as in Sections 3.5 and 3.6.2. A full proof of Lemma 2 is given in Appendix B.2. As a corollary, this gives us a minimal sufficient condition for non-emptiness of EIRD, notably by showing inclusion of $\mathbf{u}_n$.

Proposition 8. *For any λ-dispersed preference model M with $\lambda \geq \frac{k}{n}$, $\mathrm{EIRD}(M)$ contains the uniform distribution $\mathbf{u}_n$.*

Proof. For any item i and preferences $f_j(v) \in [\lambda, 1]$ for all j, the menu time for $x_i = \frac{1}{n}$ is at most

$$\mu_i \leq \frac{k \cdot \frac{1/n}{k/n}}{\sum_{j=1}^{n} x_j} \leq 1.$$

□

The dispersion condition will play an important role in the analysis of our algorithm by enabling efficient exploration, but it additionally coincides with diversity constraints in appropriate regimes.

Assuming increasingly larger lower bounds on λ yields a growing region around $\mathbf{u}_n$ which will be contained in EIRD, including all distributions which are sufficiently "diversified" (in entropy).

3.4.2 High-Entropy Containment for EIRD

Highlighting the synergy between the goals of regret minimization and avoiding the auxiliary harms caused by a lack of diversity in recommendations discussed in Section 3.1, we show that EIRD contains (close approximations of) all sufficiently high-entropy distributions. As λ grows, the EIRD set will also contain uniform distributions over shrinking subsets of $[n]$, and taking mixtures of these distributions can approximate any high-entropy distribution.

Theorem 15 (High-Entropy Containment in EIRD)**.** *Consider the set of distributions $x \in S_\chi \subset \Delta(n)$ with entropy $H(x)$ at least $\log(n) - \chi$, and let $\tau \geq \exp(-\chi)$. Let M be a λ-dispersed preference model with $\lambda \geq \frac{k \exp(\chi/\tau)}{n}$. For any vector $v \in S_\chi$, there is a vector $v' \in \mathrm{EIRD}(M)$ such that $d_{TV}(v, v')$ is at most $O(\tau)$.*

Proof. Observe that Proposition 8 also implies that any uniform distribution over n/C items lies inside EIRD(M) for a λ-dispersed preference model M with $\lambda \geq \frac{Ck}{n}$, upon considering an artificially-restricted size-n/C universe of items. We make use of a lemma from [78], which we restate here.

Lemma 3 (Lemma 8 in [78])**.** *For a random variable A over $[n]$ with $H(A) \geq \log n - \chi$, there is a set of $\ell + 1 = O(\chi/\tau^3))$ distributions ψ_i for $i \in \{0, \ldots, n\}$ over a partition of the support of A which can be mixed together to generate A, where ψ_0 has weight $O(\tau)$, and where for each $i \geq 1$:*

1. $\log |\mathrm{supp}(\psi_i)| \geq \log n - \chi/\tau$.

2. *ψ_i is within total variation distance $O(\tau)$ from the uniform distribution on its support.*

Using this, we can explicitly lower bound the support of each ψ_i:

$$\begin{aligned}\log|\text{supp}(\psi_i)| &\geq \log(n) - \chi/\tau \\ &= \log(n) - \log(\exp(\chi/\tau)) \\ &= \log\left(\frac{n}{\exp(\chi/\tau)}\right).\end{aligned}$$

As such:

$$|\text{supp}(\psi_i)| \geq \frac{n}{\exp(\gamma/\tau)}.$$

Each uniform distribution over $\text{supp}(\psi_i)$ therefore lies inside $\text{EIRD}(M)$ for $\lambda \geq \frac{Ck}{n}$, provided that $C \geq \exp(\chi/\tau)$; the $O(\tau)$ bound on total variation distance is preserved under mixture, as well as when redistributing the mass of ψ_0 arbitrarily amongst the uniform distributions. □

3.4.3 Online Gradient Descent over EIRD

Suppose that we have direct access to the agent's preference functions $\{f_i\}$, as well as an explicit convex representation of EIRD. As we assume the agent's memory update rule is known, and we see their choice in each round, this allows us to maintain exact knowledge of v_t in every round as well, which is sufficient to implement $\texttt{MenuDist}(v_t, x_t, M)$ for any $x_t \in \text{EIRD}$. Given this, we can guide the agent to run online gradient descent over EIRD on our behalf if we observe each reward function r_t in full, computing gradient updates using the expected rewards $r_t(x_t)$ in each round.

Algorithm 3 OGD over EIRD

Input: linear rewards $r_1, \ldots, r_T \in [0,1]^n$, domain $\mathcal{X} = \text{EIRD} \subseteq \Delta(n)$, step size η,
Set $x_1 = \mathbf{0}$
for $t = 1$ to T **do**
  Select menu distribution $z_t = \texttt{MenuDist}(v_t, x_t, M)$, observe reward function r_t
  Let $y_{t+1} = x_t + \eta \nabla r_t(x_t)$
  Set $x_{t+1} = \Pi_{\mathcal{X}}(y_{t+1})$
end for

Proposition 9. *Running online gradient descent over* EIRD *by computing menu distributions to sample from in each round using* `MenuDist`(v_t, x_t, M) *yields expected regret* $O(\sqrt{nT})$ *over* EIRD.

Proof. This follows immediately from Lemma 2 and the regret bound for OGD, using the fact that rewards r_t are $\sqrt{n}$-Lipschitz over $\Delta(n)$ with respect to the ℓ_2 norm. □

However, even if the functions $\{f_i\}$ are known, it may be intractable to explicitly compute EIRD in advance, as it may involve taking an intersection of infinitely many sets (each defined by nonlinear functions over $\Delta(n)$).

3.4.4 Optimizing over Contracting Domains

Suppose we have access to preference functions $\{f_i\}$ but not EIRD. As rounds progress, we can maintain a superset of EIRD by taking intersections of the finitely many IRD(v_t) sets observed thus far, each of which can be represented with polynomially many linear constraints (by rearranging the menu time constraints for Lemma 2). We show that the analysis of online gradient descent extends directly to the setting of "contracting domains", where our action x_t in each round must belong to an adversarially-chosen subset of the domain from the previous round $\mathcal{K}_t \subseteq \mathcal{K}_{t-1}$ (revealed prior to our choice of each x_t), and where our regret is measured with respect to the final set $\mathcal{K}_T$. As such, if we receive full-information feedback for r_t we can run OGD over EIRD as before even if EIRD is not known in advance by taking the intersection of all past IRD(v_t) sets as our domain in each round, which will always completely contain EIRD.

Algorithm 4 Contracting Online Gradient Descent.

Input: sequence of contracting convex decision sets $\mathcal{K}_1, \ldots \mathcal{K}_T$, $x_1 \in \mathcal{K}_1$, step size η
Set $x_1 = \mathbf{0}$
for $t = 1$ to T **do**
  Play x_t and observe cost $\ell_t(x_t)$
  Update and project:

$$y_{t+1} = x_t - \eta \nabla \ell_t(x_t)$$
$$x_{t+1} = \Pi_{\mathcal{K}_{t+1}}(y_{t+1})$$

end for

Lemma 4. *For a sequence of contracting convex decision sets $\mathcal{K}_1, \ldots \mathcal{K}_T$, $x_1 \in \mathcal{K}_1$ each with diameter at most D, a sequence of G-Lipschitz losses $\ell_1, \ldots, \ell_T$, and parameter η, the regret of Algorithm 4 with respect to $\mathcal{K}_t$ is bounded by*

$$\sum_{t=1}^{T} \ell_t(x_t) - \min_{x^* \in \mathcal{K}_T} \sum_{t=1}^{T} \ell_t(x^*) \leq \frac{D^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\nabla_t\|^2 \leq GD\sqrt{T}$$

when $\eta = \frac{D}{G\sqrt{T}}$.

Proof. Let $x^* = \arg\min_{x \in \mathcal{K}_T} \sum_{t=1}^{T} \ell_t(x)$, and let $\nabla_t = \nabla \ell_t(x_t)$. First, note that

$$\ell_t(x_t) - \ell_t(x^*) \leq \nabla_t^\top (x_t - x^*)$$

by convexity; we can then upper-bound each point's distance from x^* by:

$$\|x_{t+1} - x^*\| = \left\| \Pi_{\mathcal{K}_{t+1}}(x_t - \eta \nabla \ell_t(x_t)) - x^* \right\| \leq \|x_t - \eta \nabla_t - x^*\|,$$

as $x^* \in \mathcal{K}_{t+1} \supseteq \mathcal{K}_T$. Then we have

$$\|x_{t+1} - x^*\|^2 \leq \|x_t - x^*\|^2 + \eta^2 \|\nabla_t\|^2 - 2\eta \nabla_t^\top (x_t - x^*)$$

and

$$\nabla_t^\top (x_t - x^*) \leq \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{2\eta} + \frac{\eta \|\nabla_t\|^2}{2}.$$

We can then conclude:

$$\begin{aligned}
\sum_{t=1}^{T} \ell_t(x_t) - \sum_{t=1}^{T} \ell_t(x^*) &\leq \sum_{t=1}^{T} \nabla_t^\top (x_t - x^*) \\
&\leq \sum_{t=1}^{T} \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\nabla_t\|^2 \\
&\leq \frac{\|x_T - x^*\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\nabla_t\|^2 \\
&\leq \frac{D^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\nabla_t\|^2 \\
&= GD\sqrt{T}. && (\eta = \tfrac{D}{G\sqrt{T}})
\end{aligned}$$

□

Proposition 10. *Running contracting online gradient descent over* EIRD *by sampling in each round from* $\texttt{MenuDist}(v_t, x_t, M)$ *and taking* $\mathcal{K}_t$ *to be the intersection of* $\text{IRD}(v_t)$ *and* $\mathcal{K}_{t-1}$ *in each round (with* $\mathcal{K}_1 = \Delta(n)$*) yields expected regret* $O(\sqrt{nT})$ *over* EIRD.

Proof. This follows from substituting Lemma 4 into the proof for Proposition 9 and noting that regret with respect to EIRD cannot increase by expanding our target set to $\mathcal{K}_T \supseteq$ EIRD. □

Next we consider a relaxation to bandit feedback.

3.4.5 A Useful Algorithm for Adversarial Bandits

Here, we introduce a new algorithm for adversarial bandit problems with a number of useful robustness properties, which serves as the centerpiece of our approach in Section 3.5. This algorithm, Deferred Bandit Gradient, can tolerate unobserved adversarial perturbations ξ_t to the action distribution x_t in each round (where x_t is corrupted to $y_t = x_t + \xi_t$ prior to sampling), and can accommodate contracting decision sets (where the action distribution x_t chosen in each round must lie in a set $\mathcal{K}_t$, with $\mathcal{K}_t \subseteq \mathcal{K}_{t-1}$). Both of these properties prove useful in our setting, as preference scoring estimates will inevitably have some imprecision, and we generally will not

know the shape of the EIRD set in advance. For online optimization in general, the contracting domains property appears challenging to obtain with algorithms resembling Follow the Regularized Leader (such as Hedge or EXP3), yet is much more straightforward with approaches resembling (projected) Online Gradient Descent. Yet, in contrast to "OGD-style" FKM algorithm for bandit convex optimization which obtains $O(T^{3/4})$ regret ([79]), our algorithm operates directly in the adversarial bandit setting for linear losses over the simplex, and leverages linearity to decrease the variance in gradient estimates (a bottleneck for the regret of FKM) by "deferring" the contribution of each reward observation across several future rounds.

Algorithm 5 Deferred Bandit Gradient

Input: sequence of rewards r_t, perturbation vectors $\xi_1, \ldots, \xi_T$ where $\left|\xi_{t,i}\right| \leq \frac{\epsilon}{n}$ at each round t, and contracting convex decision sets $\mathcal{K}_1, \ldots \mathcal{K}_T$ where $\mathcal{B}_\epsilon \subseteq \mathcal{K}_T$ for a given ϵ.
Set $x_1 = \mathbf{u}_n$, $H = \frac{n}{\epsilon}$
for $t = 1$ to T **do**
  Adversary perturbs distribution x_t to $y_t = x_t + \xi_t$
  Sample $i_t \sim y_t$, observe i_t and reward $r_t(i_t)$
  Let $\tilde{r}_t = \frac{e_{i_t}}{H} \cdot \frac{r_{t,i_t}}{x_{t,i_t}}$ and $\widetilde{\nabla}_t = \sum_{s=\max(t-H+1,1)}^{t} \frac{\tilde{r}_t}{H}$
  Let $\mathcal{K}_{t+1,\epsilon} = \{x | \mathbf{u}_n + \frac{1}{1-\epsilon}(x - \mathbf{u}_n) \in \mathcal{K}_{t+1}\}$
  Update $x_{t+1} = \Pi_{\mathcal{K}_{t+1,\epsilon}}[x_t + \eta \widetilde{\nabla}_t]$
end for

Theorem 16. *For a sequence of rewards $r_t, \ldots, r_T \in [0,1]^n$, contracting convex decision sets $\mathcal{K}_1, \ldots \mathcal{K}_T \subseteq \Delta(n)$ where $\mathbf{u}_n \in \mathcal{K}_T$, and perturbation vectors $\xi_1, \ldots, \xi_T$ satisfying $\left|\xi_{t,i}\right| \leq \frac{\epsilon}{n}$ for a given ϵ in each round t, Deferred Bandit Gradient obtains expected regret bounded by*

$$\max_{x^* \in \mathcal{K}_t} \sum_{t=1}^{T} r_t^\top x^* - \sum_{t=1}^{T} r_t^\top y_t \leq 2\eta n^2 T + \frac{\sqrt{2}}{\eta} + 3\sqrt{n}\epsilon T + \frac{n}{2\epsilon} + \sum_{t=1}^{T} \sum_{i=1}^{n} \frac{\left|\xi_{t,i}\right|}{x_{t,i}}.$$

We prove Theorem 16 in Appendix B.1, which builds upon the analysis for contracting OGD. We track the regret of contracting OGD over the expectations of the sequence of $\widetilde{\nabla}_t$ vectors, whose squared norms are small in expectation, and we show that this additionally tracks both the regret obtained by our algorithm and the reward of the optimum x^* in hindsight up to small error (as the total deviation between the sum of vectors for r_t and $\widetilde{\nabla}_t$ can be represented by a martingale). Note

that if the constraint $\left|\xi_{t,i}\right| \leq \frac{\epsilon x_{t,i}}{n}$ is satisfied in each round, DBG can be calibrated to obtain regret $O(n\sqrt{T} + \epsilon\sqrt{n}T)$.

Applying the approach from Proposition 9 with DBG in place of OGD allows us to optimize over EIRD with bandit feedback and, looking ahead, affords us flexibility in tolerating imperfect estimates of preference functions.

3.4.6 Learn-Then-Optimize for Locally Learnable Preferences

When preferences are unknown, one approach we might take to extend the previous algorithms is to first estimate functions $\hat{f}_i$ which well-approximate f_i for every $v \in \Delta(n)$ and then subsequently implement Deferred Bandit Gradient, taking advantage of the flexibility for imprecision it affords. This clearly will require some restrictions on the allowed functions f_i; we introduce a class of functions which are compatible with such a "Learn-Then-Optimize" approach, which we refer to as *locally learnable* functions. For a class of preference models to be locally learnable, one must be able to accurately estimate a model's preference scores everywhere in $\Delta(n)$ when only given access to samples in an arbitrarily small region around $\mathbf{u}_n$.

Definition 15 (Local Learnability)**.** *Let $\mathcal{M}$ be a class of preference models, and let*

$$\texttt{EIRD}(\mathcal{M}) = \bigcap_{M \in \mathcal{M}} \texttt{EIRD}(M).$$

Let v^* be a point in $\texttt{EIRD}(\mathcal{M})$, and V_α be the set of points within distance α from v^*, for α such that $V_\alpha \subseteq \texttt{EIRD}(\mathcal{M})$. $\mathcal{M}$ is h-locally learnable if there is some v^* and an algorithm $\mathcal{A}$ which, for any $M \in \mathcal{M}$ and any $\alpha > 0$, given query access to normalized score estimates $\hat{s}_v$ where $\left\|\hat{s}_v - M(v)/M_v^*\right\|_\infty \leq \beta$ for any $v \in V_\alpha$ (where $M_v^* = \sum_i M(v)_i$) and for some β, can produce a hypothesis model $\hat{M}$ such that $\left\|\hat{M}(x)/\hat{M}_x^* - M(x)/M_x^*\right\| \leq \epsilon$ for any $x \in \Delta(n)$ and $\epsilon = \Omega(\beta)$.

The local learnability condition, while encompassing several natural examples we discuss shortly, is indeed somewhat restrictive. In particular, it is not difficult to see that classes of piecewise functions, such as neural networks with ReLU activations, are not locally learnable.

However, this appears to be essentially a necessary assumption for any "Learn-Then-Optimize", given the cumulative nature of uniform memory in our setting. We show a runtime lower bound for any algorithm that hopes to learn an estimate $\hat{M}$ for the preference model M via *queries*. Even a recommender who can force the agent to pick a particular item each round, and exactly query the preference model for free at the current memory vector, may require exponentially many rounds to learn $\hat{M}$ up to small worst-case error if the points it must query are far apart.

Theorem 17 (Query Learning Lower Bound)**.** *Suppose the recommender can force the agent to select any item at each step t, and can query $M(v_t)$ at the current memory vector v_t. Let $\mathcal{A}_S$ be an algorithm which produces a hypothesis $\hat{M}$ by receiving queries $M(v)$ for each $v \in S$. For points v and v', let $d_{\max}(v, v') = \max_i v_i - v'_i$. Then, any sequence of item selections and queries by the recommender requires at least*

$$T \geq \min_{\sigma \in \pi(S)} \prod_{i=1}^{|S|-1} (1 + d_{\max}(\sigma(i), \sigma(i+1)))$$

rounds to run $\mathcal{A}(S)$, where $\pi(S)$ is the set of permutations over S and $\sigma(i)$ is the ith item in σ.

Proof. For any permutation σ, we can lower bound the steps required to move between any two vectors adjacent in the ordering in terms of $d_{\max}$ and the number of rounds elapsed thus far.

Lemma 5. *Consider two vectors v and v', where v is the current empirical item distribution after t steps. Reaching an empirical distribution of v' requires at least $t \cdot d_{\max}(v, v')$ additional steps.*

Proof. Let x be the histogram representation of v with total mass t, and let $j^* = \arg\max_j v_j - v'_j$, where $v_j - v'_j = d_{\max}(v, v')$. Let $x' = t' \cdot v'$ be the histogram representation of v' with total mass t', such that $x_{j^*} = x'_{j^*}$. Note that t' is the smallest total mass (or total number of rounds) where a histogram can normalize to v', as any subsequent histogram must have $x'_{j^*} \geq x_{j^*}$. As such, we must

have that $t' \cdot v'_{j^*} \geq t \cdot v_{j^*}$, implying that:

$$\begin{aligned}\frac{t'}{t} &\geq \frac{v_{j^*}}{v'_{j^*}} \\ &= \frac{v'_{j^*} + d_{\max}(v, v')}{v'_{j^*}} \\ &\leq 1 + d_{\max}(v, v').\end{aligned}$$

□

At least one round is required to reach the first vector in a permutation, and we can use the above lemma to lower-bound the rounds between any adjacent vectors in the ordering. Taking the minimum over all permutations gives us the result. □

Notably, this implies that if S contains m points which, for any pair (v, v') have both $d_{\max}(v, v') \geq \gamma$ and $d_{\max}(v', v) \geq \gamma$, at least $O\left((1+\gamma)^m\right)$ rounds are required.

There are indeed several interesting examples of model classes which are locally learnable, whose analyses we defer to Appendix B.3. In general, our approach is to query a grid of points inside the radius α ball around the uniform vector, estimate each function's parameters and show that the propagation of over the entire domain is bounded. Note that the normalizing constants for each query we observe may differ; for univariate functions, we can handle this by only moving a subset of values at a time, allowing for renormalization. For multivariate polynomials, we consider two distinct classes and give a separate learning algorithm for each; we can estimate ratios of scores directly for multilinear functions, and if scores are already normalized we can avoid rational functions altogether. Each local learning result we prove involves an algorithm which makes queries near the uniform vector, and taking $\lambda \geq k^2/n$ suffices to ensure that these queries can indeed be implemented via an appropriate sequence of menu distributions for any M in such a class (as a ball around $\mathbf{u}_n$ will be contained around EIRD).

Bounded-degree univariate polynomials.

Let $\mathcal{M}_{BUP}$ be the class of *bounded-degree univariate polynomial* preference models where:

- For each i, $M(v)_i = f_i(v_i)$, where f_i is a degree-d univariate polynomial which takes values in $[\lambda, 1]$ over the range $[0, 1]$ for some constant $\lambda > 0$.

Univariateness captures cases where relative preferences for an item depend only on the weight of that item in the agent's memory, i.e. there are no substitute or complement effects between items.

Lemma 6. *$\mathcal{M}_{BUP}$ is $O(d)$-locally learnable by an algorithm $\mathcal{A}_{BUP}$ with $\beta \leq O(\epsilon\lambda^2 \cdot (\frac{\alpha}{nd})^d)$.*

Bounded-degree multivariate polynomials.

Let $\mathcal{M}_{BMLP}$ be the class of *bounded-degree multilinear polynomial* preference models where:

- For each i, $M(v)_i = f_i(v)$, where f_i is a degree-d multilinear (i.e. linear in each item) polynomial which takes values in $[\lambda, 1]$ over $\Delta(n)$ for some constant $\lambda > 0$,

and let $\mathcal{M}_{BNMP}$ be the class of *bounded-degree normalized multivariate polynomial* preference models where:

- For each i, $M(v)_i = f_i(v)$, where f_i is a degree-d polynomial which takes values in $[\lambda, 1]$ over $\Delta(n)$ for some constant $\lambda > 0$, where $\sum_i f_i(v) = C$ for some constant C.

Together, these express a large variety of adaptivity patterns for preferences which depend on frequencies of many items items simultaneously. In particular, these can capture relatively intricate "rabbit hole" effects, in which some subsets of items are mutually self-reinforcing, and where their selection can discourage future selection of other subsets.

Lemma 7. *$\mathcal{M}_{BMLP}$ and $\mathcal{M}_{BNMP}$ are both $O(n^d)$-locally learnable, for $\beta \leq O(\frac{\epsilon^2}{\text{poly}(n(d/\alpha)^d))})$, and $\beta \leq \frac{\epsilon}{\alpha^d F(n,d)}$, respectively, where $F(n, d)$ is independent of other parameters.*

Univariate functions with sparse fourier representations. We can also allow for classes of functions where the minimum allowable α depends on some parameter. Functions with sparse Fourier representations are such an example, and naturally capture settings where preferences are somewhat cyclical, such as when an Agent goes through "phases" of preferring some type of content for a limited window. We say that a function $f : \mathbb{R} \to \mathbb{R}$ is ℓ-*sparse* if $f(x) = \sum_{i=1}^{\ell} \xi_i e^{2\pi \mathbf{i} \eta_i x}$ where $\eta_i \in [-F, F]$ denotes the i-th frequency and ξ_i denotes the corresponding magnitude. We say that an ℓ-sparse function f is $\hat{\alpha}$-separable when $\min_{i \neq j} |\eta_i - \eta_j| > \hat{\alpha}$. Let $\mathcal{M}_{SFR}(\hat{\alpha})$ be the class of univariate *sparse Fourier representation* preference models where:

- For each i , $M(v)_i = f_i(v_i)$, where f_i is a univariate ℓ-sparse and $\hat{\alpha}$-separable function which, over $[0, 1]$, is L-Lipschitz and takes values in $[\lambda, 1]$ for some constant $\lambda > 0$.

Lemma 8. *$\mathcal{M}_{SFR}(\hat{\alpha})$ is $\widetilde{O}(n\ell)$-locally learnable by an algorithm $\mathcal{A}_{SFR}$ with $\beta \leq O(\frac{\epsilon\lambda\alpha}{\sqrt{n\ell}})$ and any $\alpha \geq \tilde{\Omega}(1/\hat{\alpha})$.*

Learn-Then-Optimize. For the aforementioned classes we can obtain a sublinear regret algorithm over EIRD for uniform-memory agents. At a high level, this approach proceeds as follows:

- force the memory vector v_t towards $\mathbf{u}_n$ (up to accuracy ϵ) by showing only the least-frequently chosen items to the agent, then estimate preferences at v_t by showing the agent every item in $\tilde{\Theta}(1/\epsilon^2)$ rounds;
- repeat a similar procedure for the other Q points required by the local learnability query algorithm, in a ball around $\mathbf{u_n}$, with each movement step taking $\tilde{\Theta}(1/\epsilon^3)$ rounds;
- run Deferred Bandit Gradient over EIRD as we did previously, using the estimated preferences $\hat{M} = \{\hat{f}_i\}$ in each call to `MenuDist`.

This obtains a regret bound of $\tilde{\Theta}(T^{3/4})$ under appropriate local learnability parameters with $\epsilon = \tilde{\Theta}(T^{-1/4})$, as the learning stage will take $\tilde{\Theta}(1/\epsilon^3)$ rounds. We omit a complete analysis of this approach, as its guarantees are essentially subsumed by the algorithm we give in Section 3.5; further details are given in [32].

3.5 Targeting EIRD for Agents with Long Memory Horizons

Recalling the lower bound for short-term memory agents, we consider cases where $\gamma = 1 - o(1)$, with an effective memory horizon of $\Omega(T^c)$ for some $c \in (0, 1]$. Here, memory vectors change slowly, and every point in $\Delta(n)$ is well-approximated *some* sequence of item selections. Rather than local learnability, we assume only that preference functions are (λ, L)-smooth, with $\lambda \geq k/n$.

At a high level, our approach is to guide the agent to implicitly run Deferred Bandit Gradient on our behalf, over a contracting subset of $\Delta(n)$ which always contains EIRD. Periodically, we pause in order to refresh our estimates of the agent's current preferences, wherein all items are shown to the agent sufficiently often in order to accurately estimate preference scores near the current memory vector. We leverage smoothness of preferences to determine accuracy bounds on our score estimates as v_t updates. Given a target item distribution x_t for the agent to sample from (as selected by DBG), we can invoke the `MenuDist` routine with our score estimates to construct a menu distribution z_t which approximately induces a choice distribution of x_t (whose error is represented by the perturbations ξ_t for DBG).

As our approach relies on stability of memory vectors across rounds, our regret decays towards $\Theta(T)$ as the memory horizon vanishes relative to T; as we saw in Section 3.3, this essentially inevitable when the discount factor is sufficiently small (e.g. $\gamma < 1/2$). For memory horizons of T^c for any $c > 0$ we obtain strictly sublinear regret, and we recover the aforementioned $\tilde{O}(T^{3/4})$ rate for uniform memory with any smooth preferences rather than only for specific parametric classes.

Theorem 18. *For an agent with a (λ, L)-smooth preference model M for $\lambda \geq k/n$, and γ-discounted memory for $\gamma \geq 1 - \frac{1}{T^c}$ and $c \in (0, 1]$, Algorithm 6 obtains regret bounded by*

$$\max_{x^* \in \mathrm{EIRD}(M)} \sum_{t=1}^{T} r_t^\top x^* - \mathbb{E}\left[\sum_{t=1}^{T} r_t(i_t)\right] = \tilde{O}\left((n/\lambda)^{3/2} L^{1/4} \cdot T^{1-c/4}\right).$$

The complete proof of Theorem 18 is deferred to Appendix B.4. Building on the regret bound for DBG in Theorem 16, the central challenges are to show that preference estimates F_i are close

Algorithm 6 (Targeting EIRD for Smooth Models).

Let $c^* = \min(c, 3/4)$, $\epsilon = \tilde{O}(nL^{1/4}\lambda^{-3/2}T^{-c/4})$, $Q = \tilde{O}(\frac{n^2}{\lambda^4\epsilon^2})$, and $\eta = (nT)^{-1/2}$.
Initialize $q = 0$, $v_0 = v^* = \mathbf{u}_n$, $F_i = \frac{C}{n}$ for $i \in [n]$, $M^* = \{F_i\}$
Initialize DBG for ϵ, η.
while $t \leq T$ **do**
  if $t < T^{c^*}$ **then**
    Show arbitrary menu K_t to agent
  else if $t \geq T^{c^*}$ and either $\|v_t - v^*\|_1 \geq \frac{\lambda\epsilon}{2nL}$ or $q = 0$ **then**
    for $b \in \{0, \ldots, \lceil \frac{n-1}{k-1} \rceil\}$ **do**
      Show agent menu $K_b = \{1\} \cup \{b(k-1)+2, \ldots, (b+1)(k+1)+1\}$ for Q rounds
      Let $\hat{F}_i = (\#\text{ times } i_t = i)\,/(\#\text{ times } i_t = 1)$ within the Q rounds, for $i \in K_b$
    end for
    Set $F_i = \frac{C \cdot \hat{F}_i}{\sum_{j=1}^n \hat{F}_j}$ for each $i \in [n]$, $M^* = \{F_i\}$
    Set $v^* = v^t$, increment q by $\lceil \frac{n-1}{k-1} \rceil \cdot Q$, set $\mathcal{K}_{t-q+1} = \mathcal{K}_{t-q} \cap \text{IRD}(v^*, M^*)$ for DBG
  else
    Get x_{t-q} from DBG
    Let $z_t = \texttt{MenuDist}(v_t, x_{t-q}, M^*)$, sample menu $K_t \sim z_t$
    Show K_t to agent, update DBG with observed i_t and $r_t(i_t)$
  end if
  Set $v_{t+1} = U(v_t, i_t, t)$, for each round counted by q if necessary
end while

enough to each $f_i(v_t)$ to enable accurate choice targeting via `MenuDist`, and that the number of non-DBG rounds spent updating F_i (tracked by q) does not grow too quickly. While EIRD contains at least the uniform distribution (as implied by Lemma 2 when $\lambda \geq \frac{k}{n}$), it may not be particularly large in general. In Section 3.6, we identify conditions under which an alternate algorithmic approach allows us to compete with a much larger set of item distributions than EIRD.

3.6 Regret Minimization Beyond EIRD

Recall that one of our motivations for considering EIRD is the difficulty of exploration under uniform memory, as evidenced by lower bounds over the set of menu distributions as well as $\text{IRD}(\mathbf{u}_n)$, as the current memory cannot be quickly "washed away". However, considering discount factors of $\gamma \leq 1 - o(1)$ introduces the possibility that we might be able efficiently explore the space of feasible vectors and compete against item distributions which lie outside of EIRD, i.e. item distributions which are only feasible for a strict subset of all memory vectors. We identify a

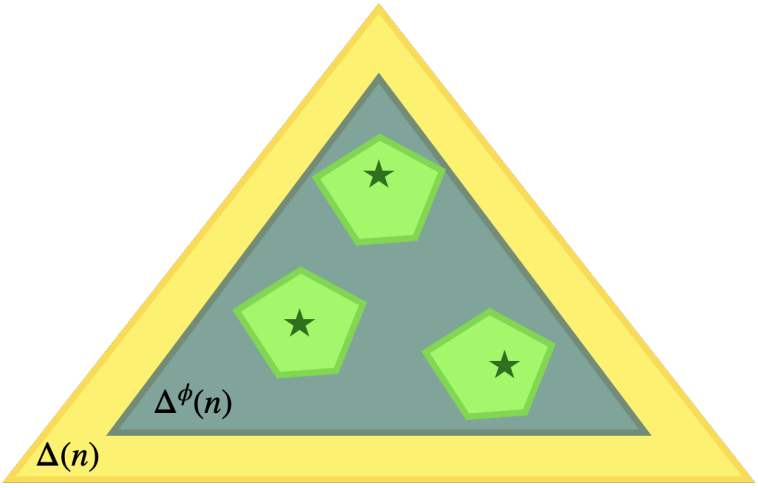

Figure 3.2: Illustration of the ϕ-smoothed simplex and several contained IRD sets, whose respective memory vectors are denoted by ($\star$).

structural property which enables this, wherein preference scoring function outputs cannot be too far multiplicatively from their item's weight in memory. We say that such functions are *scale-bounded*.

3.6.1 Scale-Bounded Preferences and the ϕ-Smoothed Simplex

Definition 16 (Scale-Bounded Functions)**.** *A preference scoring function* $f_i : \Delta(n) \to [\frac{\lambda}{\sigma}, 1]$ *is* (σ, λ)*-scale-bounded for* $\sigma \geq 1$ *and* $\lambda > 0$ *if*

$$\sigma^{-1}((1-\lambda)v_i + \lambda) \leq f_i(v) \leq \sigma((1-\lambda)v_i + \lambda).$$

We say that a preference model M is scale-bounded if each f_i is scale-bounded; in this case, the vector of scores $M(v)$ cannot stray too far from their values in v. When this property is satisfied, we can show (using the menu time approach from Lemma 2) that any point which is not too close to the boundary of $\Delta(n)$ is contained in its own IRD set. This motivates a target set of all such points, which we term the *ϕ-smoothed simplex*.

Definition 17 (ϕ-Smoothed Simplex)**.** *For any* $\phi \in [0, 1]$*, the ϕ-smoothed simplex is the set given by* $\Delta^{\phi}(n) = \{(1-\phi)x + \phi\mathbf{u}_n : x \in \Delta(n)\}$.

We show that a neighborhood around any such point is contained in $\Delta^{\phi}(n)$ as well; as preference

scores cannot be too far from an item's current memory vector weight, the required menu time for any item in a distribution x which is nearby v cannot be too large.

Lemma 9. *Let M be a (σ, λ)-scale-bounded preference model with $\sigma \leq \sqrt{n/(2k)}$. Then, $x \in \mathrm{IRD}(v, M)$ for any $x \in B_{\lambda\phi}(v) \cap \Delta^{\phi}(n)$ and any $v \in \Delta^{\phi}(n)$, provided that $\phi \geq 4k\lambda\sigma^2$.*

Proof. From Lemma 2, it suffices to show that the menu time μ_i for any such point x is at most 1. Given that v and x lie inside $\Delta^{\phi}(n)$ and that each f_i is pseudo-increasing, we have that

$$\frac{x_i}{((1-\lambda)v_i + \lambda)\sigma} \leq \frac{x_i}{f_i(v)} \leq \frac{\sigma x_i}{(1-\lambda)v_i + \lambda} \tag{3.1}$$

and that $x_i \leq \min(v_i + \lambda\phi/\sqrt{2}, 1 - \phi(n-1)/n)$ as $x \in B_{\lambda\phi}(v)$. Recall that the menu time μ_i for x is given by

$$\mu_i(x) = \frac{k \cdot \frac{x_i}{f_i(v)}}{\sum_{j=1}^{n} \frac{x_j}{f_j(v)}}.$$

First we show that this numerator is at most $k\sigma$. We have that

$$\begin{aligned} x_i &\leq v_i + \lambda\phi/\sqrt{2} \\ &\leq (1-\lambda)v_i + \lambda(1 - \phi(n-1)/n) + \lambda\phi/\sqrt{2} \\ &\leq (1-\lambda)v_i + \lambda && \text{(for } n \geq 4\text{)} \end{aligned}$$

which yields that $kx_i/f_i(v) \leq k\sigma$ by (3.1). We can also lower-bound the menu denominator by $k\sigma$. Let $\alpha_j = v_j - x_j$, where we have:

$$\sum_{j=1}^{n} \frac{x_j}{f_j(v)} \geq \sum_{j=1}^{n} \frac{v_j - \alpha_j}{((1-\lambda)v_j + \lambda)\sigma}.$$

Differentiating with respect to v_j for any term, we have:

$$\frac{\partial}{\partial v_j}\frac{v_j-\alpha_j}{((1-\lambda)v_i+\lambda)\sigma} = \frac{((1-\lambda)v_i+\lambda)\sigma-(v_j-\alpha_j)(1-\lambda)\sigma}{\sigma^2((1-\lambda)v_i+\lambda)^2}$$
$$= \frac{\lambda+\alpha_j(1-\lambda)}{\sigma((1-\lambda)v_i+\lambda)^2}$$

which is positive for any v_j, and so each numerator term is increasing for any fixed α_j (as $|\alpha_j| \le \lambda$). As such, each term is minimized over valid v_j and x_j when $x_j = \phi/n$ and $v_j = \phi/n + \lambda\phi/\sqrt{2}$, yielding

$$\begin{aligned}
\sum_{j=1}^{n}\frac{x_j}{f_j(v)} &\ge \frac{\phi}{((1-\lambda)(\phi/n+\lambda\phi/\sqrt{2})+\lambda)\sigma} \\
&\ge \frac{(k\sigma)\cdot\phi}{k\sigma^2\lambda + k\sigma^2(1-\lambda)\phi/n + k\sigma^2(1-\lambda)\phi\lambda/\sqrt{2}} \\
&\ge \frac{(k\sigma)\cdot\phi}{\phi/4+\phi/2+\phi^2/(4\sqrt{2})} && (\phi \ge 4k\lambda\sigma^2,\ \sigma^2 \le n/(2k)) \\
&\ge k\sigma. && (\phi \le 1)
\end{aligned}$$

As such, we have that

$$\frac{k\cdot\frac{x_i}{f_i(v)}}{\sum_{j=1}^{n}\frac{x_j}{f_j(v)}} \le 1,$$

and thus $x \in \mathrm{IRD}(v, M)$. □

Here, we consider $\Delta^{\phi}(n)$ as a target set for regret minimization when preferences are scale-bounded; the convexity of $\Delta^{\phi}(n)$ along with the stabilizability promised by Lemma 9 will allow us to sidestep the barrier results of Section 3.3 that apply to general preference models which may not be scale-bounded. We will no longer require explicit lower bounds on λ, and so we can take our regret benchmark to be approaching the entire simplex as λ approaches 0 with an appropriate choice of ϕ. This presents a stark constrast with the EIRD benchmark, as the scale-bounded property now suggests that it may be possible to persuade the agent to pick the best item in nearly every round,

rather than in $O(T/n)$ rounds (which may occur in Section 3.5, e.g. if some $f_i(v) = \lambda = k/n$ at every $v \in \Delta(n)$).

3.6.2 A No-Regret Algorithm for $\Delta^{\phi}(n)$

In contrast to Algorithm 6, where we considered each round as a step for a bandit optimization algorithm with interleaved learning stages, here we collapse multiple iterations of learning and targeting into a *single* step for Online Gradient Descent, run over $\Delta^{\phi}(n)$, where the agent's entire memory vector is moved in a desired direction. While we can no longer instantaneously realize any distribution in our target set, the ability to induce any choice distribution in a nearby ball enables exploration throughout $\Delta^{\phi}(n)$ via the agent's memory vector. Further, the scale-bounded condition tethers scores to memory weight, enabling reduced variance in estimating both rewards and preferences. However, this also yields a decay as c approaches 1 (in addition to 0), as memory does not update quickly enough to enable exploration. Theorems 11 and 12 from Section 3.3 imply that this is necessary: an adversary may shift the reward distribution in later rounds when we can no longer significantly move the entire memory vector, forcing linear regret.

Theorem 19 (Scale-Bounded Discounted Regret Bound)**.** *For any agent with a preference model M which is (σ, λ)-scale-bounded and $(\frac{\lambda}{\sigma}, L)$-smooth with $\sigma \leq \sqrt{n/(2k)}$, and with γ-discounted memory for $\gamma = 1 - \frac{1}{T^c}$ for $c \in (0, 1)$, Algorithm 7 obtains regret*

$$\mathsf{Reg}_{\Delta^{\phi}(n)}(\mathcal{A}_2; T) = \tilde{O}\left((n^4 L\left(T^{1-c/2} + T^{1/2+c/2}\right)\right)$$

with respect to the ϕ-smoothed simplex, for $\phi = 4k\lambda\sigma^2$.

Proof Sketch. In the "burn-in" stage, our goal is simply to push the memory vector towards $\mathbf{u}_n$; by first saturating memory on only k items for T^c rounds, we are then able to push all low-memory items towards $\frac{1}{n}$ at near-uniform rates, as memory now moves slowly and the k lowest values will remain close together by the scale-bounded condition, so no item can get "stuck" near 0 and we reach $\mathbf{u}_n$ in $\tilde{O}(T^c)$ rounds. In the "initial learning" stage, we are now promised that the uniform

Algorithm 7 (Targeting Δ^{ϕ} for Scale-Bounded Preferences).

Let $\epsilon = \tilde{O}(n^4 L \cdot \max(T^{-c/2}, T^{c/2-1/2}))$, $S = \tilde{O}(n^{3/2}T^c)$, $\eta = \tilde{O}(n^{-1/2}L \cdot T^{c/2-1/2})$.
`- burn-in -`
for $t = 1$ to T^c **do**
  Show agent menu $K = \{1, \ldots, k\}$
end for
while $\max_i \left|v_{t,i} - \frac{1}{n}\right| \geq \frac{\epsilon}{4n^2 L\sigma}$ **do**
  Show agent k items with smallest $v_{t,i}$, choosing randomly among ties up to T^{-c}
end while
`- initial learning -`
for T^c rounds **do**
  Let $F_{t,i} = \sigma^{-1}((1-\lambda)v_{t,i} + \lambda)$ if $v_{t,i} < \frac{1}{n}$, else $F_{t,i} = \sigma((1-\lambda)v_{t,i} + \lambda)$
  Let $z_t = \texttt{MenuDist}(v_t, \mathbf{u}_n, \{F_{t,i}\})$, show agent $K_t \sim z_t$
end for
Set $F_i = \sum_t \mathbf{1}(i_t = i) \cdot F_{t,i}/(\frac{1}{n}T^c)) \cdot C(\sum_{j=1}^n F_{t,j})^{-1}$, set $v^* = v_t$
`- optimization -`
Initialize OGD over $\Delta_{\epsilon}^{\phi}(n)$ for T/S rounds with η, set $x_1 = v^* := v_t$.
for $s = 1$ to T/S **do**
  Receive x_s from OGD
  for S rounds do **do**
    if $\|v_t - v^*\|_1 \geq \epsilon/(2nL)$ **then**
      Let $\tilde{v} = v_t$
      Show agent $K_t \sim z = \texttt{MenuDist}(v_t, \tilde{v}, \{F_i\})$ for T^c/L^2 rounds
      Set $F_i = \sum_t \mathbf{1}(i_t = i) \cdot F_i/(\tilde{v}_i T^c/L^2) \cdot C(\sum_{j=1}^n F_j)^{-1}$, set $v^* = v_t$
    end if
    Show agent $K_t \sim z_t = \texttt{MenuDist}(v^*, x_s, \{F_i\})$
  end for
  Set $\tilde{\nabla}_s = \sum_{h=t-S+1}^{t} e_{i_h} r_h(i_h)/(x_{h,i}S)$, update OGD
end for

distribution is is within our IRD set, and we can force memory to remain there by assuming pessimistic scores if $v_{t,i} < \frac{1}{n}$ and optimistic scores otherwise. By comparing observed selection frequencies to those indicated by our assumed scores, we obtain unbiased estimators for the true f_i values near $\mathbf{u}_n$. In the "optimization" stage, we batch $O(T^c)$ rounds into "steps" for Online Gradient Descent large enough to maintain locally accurate f_i estimates throughout, and alternate between progressing toward the chosen target and updating our scores, which further enables concentrated estimates of average reward vectors in each step and a regret bound akin to that for "slowed down" OGD. □

We allow λ to be arbitrarily small, and assume only that T is large enough to yield $\lambda \geq T^{-c/4}\,\mathrm{poly}(n)$; our bound has no dependence on λ or ϕ beyond this. Our optimal rate over c is again $\tilde{O}(T^{3/4})$, yet this time occurring when $c = \frac{1}{2}$, balancing improved variance reduction in learning with the need to quickly explore in memory space. The full proof is given in Appendix B.5.

3.6.3 Scale-Bounded Preferences with Short-Term Memory

When the discount factor of the agent is small enough that memory vectors may move rapidly, we lose the precision required by the algorithms in Section 3.5 in order to implement queries, and in fact the feasible state space may more closely resemble a discrete grid, with memory vectors encoding the sequence of items chosen over an effective horizon which is constant with respect to T. Nonetheless, for scale-bounded models we give an algorithm which we call EXP-ϕ, which obtains $o(T)$ regret with respect to $\Delta^{\phi}(n)$ for *any* value of $\gamma \in [0, 1)$ under an assumption about the restricted adversarial nature of rewards. Here, we assume that rewards are stochastic rather than adversarial for windows of length $o(T)$, but distributions may change adversarially between each window; we require a slightly larger lower bound on ϕ (yet still $O(\lambda)$).

The idea behind EXP-ϕ is to view each vertex of the smoothed simplex as an action for a multi-armed bandit problem, where each "pull" corresponds to several rounds. When we "commit" to playing an item in the menu for a sufficiently long time, while otherwise playing items with the smallest weight in memory, the scale-bounded property will ensure that the selection frequency

of that item gravitates towards its vertex in the smoothed simplex. Further, as we are no longer attempting to learn the preference model explicitly, we can relax the smoothness requirement.

Algorithm 8 (EXP-ϕ).

Initialize EXP3 to run for T/t_{hold} steps
while $t < T$ **do**
  Sample arm i^* from EXP3
  for $\tilde{O}(T^{2/3}/(1-\gamma))$ rounds **do**
    Let $K_t = \{i^*\} + \text{argmin}_{j\neq i}^{k-1} v_j$
  end for
  Update EXP3 with average reward of i^*
end while

Theorem 20. *For any agent with a preference model M which is (σ, λ)-scale-bounded for which each $f_i(v) \in [\lambda, 1]$ for $\lambda \geq \frac{\sigma^2 k}{n}$ and $\sigma \leq \sqrt{n/(2k)}$, and with γ-discounted memory for $\gamma \in [0, 1)$, when losses are drawn from a distribution which changes at most once every $t_{hold} = \tilde{O}\left(\frac{T^{2/3}}{1-\gamma}\right)$ rounds, Algorithm 7 obtains regret at most*

$$\mathsf{Reg}_{\Delta^\phi(n)}(\mathcal{A}_3; T) = \tilde{O}(T^{5/6})$$

with respect to $\Delta^\phi(n)$, for $\phi = 3\lambda k^3 \sigma^6$.

Proof. The key element of our analysis is to analyze the convergence of item frequencies during windows of length t_{hold} when a fixed target item i is held constant in the menu. For a given such window of length $t_{\text{hold}} = O(\frac{1}{\alpha^4(1-\gamma)})$, we can ensure that the accumulated reward approaches its expectation under the current reward distribution to within α. As we choose the $k-1$ smallest weights in memory, the total weight of items in memory other than i is at most $\frac{(k-1)(1-v_i)}{n}$; given a current memory vector v, the probability of selecting item i from a menu K_t is given by:

$$\begin{aligned}\Pr[i^* \text{ selected}] &= \frac{f_i(v)}{\sum_{K_t} f_j(v_i)} \\ &\geq \frac{(1-\lambda)v_i + \lambda}{(1-\lambda)v_i + \lambda + \left(\frac{(1-v_i)(1-\lambda)}{n} + \lambda\right)(k-1)\sigma^2}\end{aligned}$$

by the pseudo-increasing property. Our approach will be to analyze the expectation of $v_{t,i}$ over time, with $E_t = \mathbb{E}[v_{t,i}]$, and show that it approaches $1 - \phi(n-1)/n$, equal to the probability at the corresponding vertex of the smoothed simplex. A challenge is that, given a current expectation, there are many possible allocations of probabilities to values of $v_{t,i}$ which yield E_t. A second derivative test shows that the above probability function is concave for positive v_i when $\sigma^2 \leq n/(2k)$; note that both the numerator and denominator are positive and increasing linearly in v_i, and that the numerator is always smaller but grows faster in v_i. As such, we can apply Jensen's inequality and restrict our consideration to the extremal case where the expectation E_t is entirely composed of trials in which $v_{t,i} = 0$ or $v_{t,i} = 1$, which indeed occurs at $\gamma = 0$. We can define P_0 and P_1 as lower bounds on selection probabilities for each case:

$$\begin{aligned}
\Pr[i^* \text{ selected} | v_{t,i} = 1] &\geq \frac{1}{1 + (k-1)\lambda\sigma^2} \\
&\geq 1 - k\lambda\sigma^2 \\
&:= P_1;
\end{aligned}$$

$$\begin{aligned}
\Pr[i^* \text{ selected} | v_{t,i} = 0] &\geq \frac{\lambda}{\lambda + (\frac{1-\lambda}{n} + \lambda)(k-1)\sigma^2} \\
&= \frac{1}{1 + \left(\frac{1-\lambda}{n\lambda} + 1\right)(k-1)\sigma^2} \\
&\geq \frac{1}{1 + \left(k\sigma^2 + 1\right)(k-1)\sigma^2} \\
&\geq \frac{1}{2\sigma^2 k^4} \\
&:= P_0.
\end{aligned}$$

As such, we have that

$$\begin{aligned} E_{t+1} &= \mathbb{E}[v_{t+1}|E_t] \\ &\geq (1-\gamma)\left(E_t \cdot \Pr[i^* \text{ selected}|v_{t,i}=1] + (1-E_t)\cdot \Pr[i^* \text{ selected}|v_{t,i}=0]\right) + \gamma E_t \\ &\geq (1-\gamma)\left(E_t \cdot P_1 + (1-E_t)\cdot P_0\right) + \gamma E_t. \end{aligned}$$

We can solve for E_t^* such that $E_{t+1} = E_t$, i.e. where $E_t \cdot P_1 + (1-E_t)\cdot P_0 = E_t$, as:

$$\begin{aligned} E_t^* &= \frac{1}{1+2\lambda\sigma^6 k^3} \\ &\geq 1 - 2\sigma^6 k^3 \lambda \\ &\geq 1 - \phi(n-1)/n \qquad (n \geq 3) \end{aligned}$$

and further for a value E_t^α such that $E_{t+1} \geq (1-\gamma)(E_t+\alpha) + \gamma E_t$ as:

$$E_t^\alpha = E_t^* - 2\sigma^4 k^2 \alpha.$$

Note that the rate of growth of E_{t+1} is decreasing in t, and eventually reaches a fixed point; given that the rate of growth of E_t is linear in α when within $O(\alpha)$ of E_t^*, the cumulative number of rounds required to reach $E_t^* - O(\alpha)$ is at most $O(\frac{1}{\alpha(1-\gamma)})$. If we continue after for $O(\frac{1}{\alpha^2(1-\gamma)})$ rounds, these first rounds contribute at most α to the total summed expectation for the fraction of rounds in which item i is selected is at least $E_t^* - \alpha$; the fraction of each other item played also quicly approaches $\frac{1-E_t^*}{n-1}$ in expectation.

Treat each such batch of $O(\frac{1}{\alpha^2(1-\gamma)})$ rounds as a trial, and repeat for $\tilde{O}(1/\alpha^2)$ trials, resulting in a total of $t_{\text{hold}} = O(\frac{1}{\alpha^4(1-\gamma)})$ steps. We can treat each trial as independent, as resetting the memory vector to some lower value of v_i can only decrease expected reward under the pessimistic lower bounds we consider. By a Hoeffding bound, we have that the reward is within α from the expectation under the current distribution and the "arm" of the ϕ-smoothed simplex corresponding

to i. To conclude, observe that our total regret (using the $\tilde{O}(T^{1/2})$ bound for EXP3) is given by:

$$\begin{aligned}\mathsf{Reg}_{\Delta^{\phi}(n)}(\mathcal{A}_3; T) &= \tilde{O}(t_{\text{hold}} \cdot \left(\frac{T}{t_{\text{hold}}}\right)^{1/2} + \alpha T) \\ &= \tilde{O}(\frac{T^{1/2}}{\alpha^2} + \alpha T) \\ &= \tilde{O}(T^{5/6})\end{aligned}$$

upon setting $\alpha = O(T^{-1/6})$. □

3.7 Future Directions

While we have characterized the feasibility and impossibility for a broad range of objectives in this setting, a number of questions still remain unresolved. In particular:

- Can rates below $T^{3/4}$ be obtained either for EIRD or $\Delta^{\phi}(n)$ when preferences are unknown, for any memory horizon?
- Can stronger regret lower bounds be shown for $\gamma > 1/2$?
- Are there other natural alternatives to scale-bounded functions which enable regret minimization beyond EIRD?
- Are there practical approaches for extending this setting to consider many agents simultaneously, with improved per-agent efficiency?
- Can the structure leveraged by our algorithms be extended to other problem domains?

We conjecture that rates of $T^{1/2}$ over EIRD cannot be obtained without advance knowledge of preferences, as simultaneous learning and optimization appears significantly more challenging than optimization alone, although it is unclear whether e.g. $\tilde{O}(T^{2/3})$ rates are likely — if so, we expect a significantly more involved analysis of the intricacies of evolving memory dynamics would be required. Likewise, we expect that stronger lower are obtainable, but are perhaps quite delicate. We

leave open the questions regarding alternate preference assumptions and multi-agent extensions; the design space here is large, and it seems quite plausible that a variety of interesting models could be proposed. For the final question regarding generalizations to other problem domains, we offer an affirmative answer in the next chapter.

Chapter 4: Online Stackelberg Optimization via Nonlinear Control

The results of this chapter are based on joint work with Christos Papadimitriou and Tim Roughgarden [6].

4.1 Overview

Machine learning problems involving strategic or adaptive agents are commonly framed as Stackelberg games, wherein the leader aims to commit to an optimal strategy in anticipation of the follower's best response. This approach has been effectively applied to challenges ranging from performative feature manipulation [80, 81, 69, 82] and optimal pricing [75, 83, 84] to resource allocation in security games [85, 25, 86] and learning in tabular games [21, 87, 88, 89], often with a regret minimization objective. Additionally, several of these settings have been independently extended to account for agents that may update their strategies gradually over time rather than optimally responding in each round [90, 91, 73, 92, 6]. Despite their conceptual similarities, these problems have largely been approached as distinct areas of study, each with their own growing body of techniques. Our aim in this work is to offer a unifying perspective and algorithmic approach for problems of this form, through the lens of online control.

For the broad family of online "Stackelberg-style" optimization problems, the language of control is quite natural to adopt: we are navigating a dynamical system where states corresponding to agent strategies evolve as a function of our own actions, and where objectives which consider best-response stability can be expressed in terms of the stationary behavior of this system. Our results consider a general class of online control instances for representing such problems, which we introduce in Section 4.2, and in Section 4.3 we give a sequence of no-regret algorithms for these instances satisfying a range of robustness properties. In Section 4.4, we show that several

online optimization problems involving adaptive agents, including variants of online performative prediction (as in [93]), online recommendations (as in [33]), adaptive pricing (as in [75]), and learning in time-varying games (as in [94]) can be embedded in our framework and solved by our algorithms.

While there has been a great deal of recent progress in online linear control, yielding algorithms which can optimize over stabilizing linear policies even with general convex costs, adversarial disturbances, and unknown dynamics [95, 96, 97, 98], the required assumptions and regret benchmarks for these algorithms do not always type-check with the settings we are interested in. For the examples we consider, we will often wish to allow for nonlinear dynamics (e.g. encoding an agent's utility function) and explicitly bounded spaces (e.g. via projection into the simplex), and we will seek to compete with regret benchmarks which correspond to stable responses by the agent. Unfortunately, as we show in Proposition 12, the latter goal is incompatible with linear policies even under linear dynamics and in the absence of any disturbances: the performance of *every* linear policy can be $\Omega(T)$ worse than the best policy in the class of affine "state-targeting" policies.

In contrast, the orthogonal set of assumptions we identify enables tractable regret minimization even for *nonlinear* control problems and comports with the requirements of Stackelberg optimization across a wide range of settings, including the ability to compete with state-targeting policies. For convex and compact state and action spaces $\mathcal{X}$ and $\mathcal{Y}$, our first key assumption is that the dynamics $D(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathcal{Y}$ satisfy a notion of *local controllability*. While local controllability is well-studied for continuous-time and asymptotic control [99, 100, 101, 102], we are unaware of any prior applications to finite-time online optimization, and we adapt existing definitions to be appropriate for this setting. We say that $D(x, y)$ is *strongly* locally controllable if every state in a fixed-radius ball around y is reachable in a single round by an appropriate choice of x, and that $D(x, y)$ is *weakly* locally controllable if the reachable radius around y is allowed to vanish near the boundary of $\mathcal{Y}$. We also assume that our loss f_t in each round is determined (or well-approximated by) an adversarially-chosen convex function depending only on the state y_t.

When these conditions hold, we show in Theorem 21 that this is sufficient to obtain $O(\sqrt{T})$

regret with respect to the loss of the best fixed state, provided that dynamics are known and we have offline access to an oracle for non-convex optimization; the oracle call can be removed if the dynamics are *action-linear*, i.e. given locally by a function linear in x at each fixed y. If adversarial disturbances to the dynamics are present, our approach can be extended for both weakly (Theorem 22) and strongly (Theorem 23) locally controllable dynamics with additional regret scaling linearly in total disturbance magnitude, provided that each round's disturbance cannot be too large in the case of weak local controllability; we give lower bounds showing that each dependence on disturbance magnitude is tight. The aforementioned results all extend to the case where the dynamics (absent disturbances) are given by a known but time-dependent function $D_t(x, y)$. In Section 4.4 we show that each of the following, with appropriate assumptions, can be cast as a locally controllable instance with state-only convex surrogate losses:

- **Performative prediction:** Minimize prediction loss $\mathbb{E}_{z\sim p_t} f_t(x_t, z)$ for a classifier x_t, where the distribution p_t in each round is updated according to the prior classifier and distribution.
- **Adaptive recommendations:** Maximize the reward $f_t(i_t)$ when showing menus $K_t \subseteq [n]$ of size $k \ll n$ to an agent, whose choice $i_t \sim p(K_t, v_t)$ in each round depends on preferences which are influenced by choices in prior rounds (encoded in the "memory vector" v_t).
- **Adaptive pricing:** Maximize profit $\langle p_t, x_t\rangle$ - $c_t(x_t)$ for selling bundles of goods x_t to an agent at prices p_t and with costs c_t, where the agent's purchased bundle x_t is a function of their utility function, consumption rate, and existing reserves.
- **Repeated gameplay:** Maximize the reward $x_t^\top A_t y_t$ obtained from playing a sequence of time-varying games (A_t, B_t) against a no-regret learning agent.

In each case, application of our algorithms from Section 4.3 yields results which extend beyond the applicability regimes of prior work, such as by enabling relaxation of previous assumptions or a novel extension to adversarial or dynamic problem variants.

4.1.1 Related Work

Online control. Much of the recent progress in online control [95, 103, 97, 98] considers linear systems with general convex losses, benchmarking against a class of ("strongly stable") fast-mixing linear policies introduced for linear-quadratic control [104] by leveraging the framework of "OCO with memory" [105]. Results have also been shown for nonlinear policy classes via neural networks [106], and for nonlinear dynamics with oracles in episodic settings [107], via approximation with random Fourier features [108, 109], via adaptive regret for time-varying linear systems [110, 98], and via dynamic regret over actions in terms of disturbance "attenuation" [111]. For a further overview of online control and its historical context, see [4]. In contrast to the bulk of prior work in which states and actions are bounded implicitly via policy stability notions, we consider state and action spaces which are bounded explicitly, as enabled by nonlinearity in dynamics (e.g. via projection, or range decay of dynamics near the boundary). These works also view disturbances as intrinsic to the system, and account for their influence directly in regret benchmarks (the "optimal policy" will face the same sequence of disturbances in hindsight, regardless of state). Within the context of Stackelberg optimization where a fixed protocol largely determines an agent's strategy updates, we view the role of disturbances as more akin to adversarial *corruptions* as considered in reinforcement learning [112, 113]; while we incur linear dependence, our regret benchmarks are agnostic to alternate counterfactual disturbance sequences.

Strategizing against learners. Initially formulated within the context of repeated auctions [73], a recent line of work has considered the problem of optimizing long-run rewards in a repeated game against a no-regret learner across a range of tabular and Bayesian settings [92, 114, 6, 115] (including Chapter 2). While bounds on attainable reward have been known in terms of the Price of Anarchy [13, 116], this sequence of results has highlighted important connections with Stackelberg equilibria: the Stackelberg value of the game is attainable on average against any no-regret learner, and it is the maximum attainable value against many common no-regret algorithms (such as no-swap learners, as shown by [92]). This theme has emerged in other simultaneous learning settings

as well; notably, [117] show that long-run outcomes in strategic classification are shaped by relative learning rates between parties, which can designate either as the Stackelberg leader.

Nested convex optimization. The technique of identifying convex structure nested inside a more general problem has been applied broadly across a range of online optimization settings [118, 119, 120]. For repeated interaction problems involving an agent with unknown utility, such as optimal pricing, [75] identify utility conditions under which the non-convex objective over prices becomes convex in the space of agent actions, and where explorability properties resembling local controllability hold, which enables convex optimization by locally learning agent preferences; this "revealed preferences" approach has also been applied to strategic classification [81]. For our results in Chapter 3 concerning recommendations for agents with history-dependent preferences [121, 33], properties related to local controllability are leveraged to enable tractable optimization as well. We consider each of these settings as applications in Section 4.4.

4.2 Model and Preliminaries

Let $\mathcal{X}$ and $\mathcal{Y}$ be convex and compact subsets of Euclidean space, respectively denoting the action and state spaces, where we assume $\dim(\mathcal{X}) \geq \dim(\mathcal{Y})$. Further, we assume that $\mathcal{Y}$ contains a ball of radius r around the origin $\mathbf{0}$, and is contained in a ball of radius R around the origin.

An instance of our control problem consists of choosing a sequence of actions $\{x_t \in \mathcal{X}\}$ over T rounds, which will yield a sequence of states $\{y_t \in \mathcal{Y}\}$, and we will incur losses determined by adversarially chosen functions $\{f_t\}$. Let the initial state be $y_0 = \mathbf{0}$. In the basic version of our problem, upon choosing each x_t for rounds $t \in [T]$, we observe the state update to

$$y_t = D(x_t, y_{t-1}),$$

where $D : \mathcal{X} \times \mathcal{Y} \to \mathcal{Y}$ is an arbitrary continuous function which we refer to as the *dynamics* of our problem. We sometimes allow *disturbances* to the dynamics, where $y_t = D(x_t, y_{t-1}) + w_{t+1}$ for $\{w_t\}$ chosen adversarially. In some cases we allow *time-varying* dynamics $D : \mathcal{X} \times \mathcal{Y} \times [T] \to \mathcal{Y}$,

where the dynamics in each round are denoted by $D_t(x_t, y_{t-1})$.

Here and in Section 4.3, we assume that our loss in round is given by $f_t(y_t)$, where each f_t is a L-Lipschitz convex function revealed after playing x_t; we relax these assumptions for some of our applications in Section 4.4, e.g. to allow dependence on x_t as well. We generally measure will performance with respect to the best fixed state, and the regret for an algorithm $\mathcal{A}$ yielding $\{y_t\}$ is

$$\mathsf{Reg}_T(\mathcal{A}) = \sum_{t=1}^{T} f_t(y_t) - \min_{y \in \mathcal{Y}} \sum_{t=1}^{T} f_t(y).$$

In Proposition 12, we relate this benchmark to the class of "state-targeting" policies, which can sometimes be expressed by affine functions, and we compare their performance to linear policies. Throughout, we use $\|\cdot\|$ to donate the Euclidean norm, and we let $\mathcal{B}_\epsilon(y) = \{\hat{y} : \|y - \hat{y}\| \leq \epsilon\}$ denote the norm ball of radius ϵ around y. We let $\Pi_{\mathcal{Y}}(\cdot)$ denote Euclidean projection into the set $\mathcal{Y}$; $\mathbf{u}_n$ denotes the uniform distribution over n items, and $\Delta(n)$ denotes the probability simplex.

4.2.1 Locally Controllable Dynamics

A number of properties under the name "local controllability" have been considered for various continuous-time and asymptotic control settings [99, 100, 101, 102], generally relating to the notion that all states in a neighborhood around a given state are reachable. We give two formulations of local controllability for our setting, which we take as properties of the dynamics D holding over all inputs.

Definition 18 (Weak Local Controllability)**.** *For $\rho \in (0, 1]$, an instance $(\mathcal{X}, \mathcal{Y}, D)$ satisfies (weak) ρ-local controllability if for any $y \in \mathcal{Y}$ and $y^* \in \mathcal{B}_{\rho \cdot \pi(y)}(y)$, there is some x such that $D(x, y) = y^*$, where $\pi(y) = \min_{\hat{y} \in \mathrm{bd}(\mathcal{Y})} \|\hat{y} - y\|$ is the distance from y to the boundary of $\mathcal{Y}$.*

Definition 19 (Strong Local Controllability)**.** *For $\rho > 0$, an instance $(\mathcal{X}, \mathcal{Y}, D)$ satisfies strong ρ-local controllability if for any $y \in \mathcal{Y}$ and $y^* \in \mathcal{B}_\rho(y) \cap \mathcal{Y}$, there is some x such that $D(x, y) = y^*$.*

We often refer to weak local controllability simply as local controllability. This property ensures that there is always some action x_t which results in the next state y_t staying fixed at y_{t-1}, as well

as some action which moves the state to any point in a surrounding ball; in the weak case, the size of the reachable ball is allowed to decay as y_t approaches the boundary of $\mathcal{Y}$. The parameter ρ controls the speed at which we can navigate the state space: when $\rho = 1$ in the weak case (or $\rho \geq R$ in the strong case), we can always immediately reach some point on the boundary of $\mathcal{Y}$, yet for ρ close to zero we may only be able to move in a small neighborhood. Our results use local controllability to minimize regret over $\mathcal{Y}$ by reduction to online convex optimization. Up to a quantifier alternation which vanishes as ρ approaches 0, a property of this form is essentially necessary: competing with the best state y is impossible if we cannot remain in its neighborhood.

Proposition 11. *Suppose there is some $y \in \mathcal{Y}$ and values $\alpha, \beta > 0$ such that for all $\hat{y} \in \mathcal{B}_\alpha(y)$ and $x \in \mathcal{X}$, $D(x, \hat{y}) \notin \mathcal{B}_\beta(\hat{y})$. Then, there are losses such that $\mathsf{Reg}_T(\mathcal{A}) = \Omega(T)$ for any algorithm $\mathcal{A}$.*

Proof of Proposition 11. Without loss of generality, assume $\alpha \leq \beta/2$ and that T is even. Let $f_t = \|y_t - y\|$ for each t. Consider any round t where $y_{t-1} \in B_\alpha(y)$; then, for all actions x_t, we have that $y_t \notin \mathcal{B}_\alpha(y)$, as $\mathcal{B}_\alpha(y) \subseteq \mathcal{B}_\beta(y_{t-1})$; as such, we incur loss $f_t(y_t) \geq \alpha$ in round t. Now suppose $y_{t-1} \notin B_\alpha(y)$; then, we must have incurred loss at least $f_{t-1}(y_{t-1}) \geq \alpha$ in round $t-1$. As losses are non-negative, our total loss is at least $\alpha T/2$, as loss α is incurred at least every other round; given that the best fixed state $y^* = y$ incurs total loss 0, we have that $\mathsf{Reg}_{\mathcal{A}}(T) = \Omega(T)$ for any algorithm $\mathcal{A}$. □

4.2.2 States vs. Policies

While regret benchmarks in online control are typically expressed in terms of a reference class of policies, we note that there is a class of "state-targeting" policies which track the reward of fixed states (asymptotically, and up to the influence of disturbances), and which can be implemented if D is known; we maintain the formulation in terms of fixed states for clarity with respect to our motivations for Stackelberg optimization. Existing no-regret algorithms for online control typically compete with linear policies, and choose actions each round by implementing policies which are linear in multiple past states (as in e.g. [95]). Here, we show that all such policies can be arbitrarily suboptimal when compared to state-targeting policies, even for dynamics which are linear up to

projection and with fixed convex losses over states, as they may yield actions and states which remain fixed at $\mathbf{0}$ in every round even if the optimal state is always immediately accessible under the dynamics.

Proposition 12. *For an instance $(\mathcal{X}, \mathcal{Y}, D)$, let the class of state-targeting policies for $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$ be given by $\mathcal{P}_{\hat{\mathcal{Y}}} = \{P_{\hat{y}} : \hat{y} \in \hat{\mathcal{Y}}\}$ where $P_{\hat{y}}(y) = \operatorname{argmin}_{\{x \in \mathcal{X} : D(x,y) \in \hat{\mathcal{Y}}\}} \|D(x,y) - \hat{y}\|^2$. Define the regret of a policy class $\mathcal{P}$ as*

$$\mathsf{Reg}_T(\mathcal{P}) = \min_{P \in \mathcal{P}} \left(\sum_{t=1}^{T} f_t(y_t) \right) - \min_{y \in \mathcal{Y}} \left(\sum_{t=1}^{T} f_t(y) \right),$$

where y_t is updated by playing P at each round. For any ρ-locally controllable instance, there is a set $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$ for which $\mathsf{Reg}_T(\mathcal{P}_{\hat{\mathcal{Y}}}) = O(\sqrt{T\rho^{-1}})$. Further, for any class $\mathcal{P}_{\mathcal{K}}$ where each $K \in \mathcal{P}_{\mathcal{K}}$ is a matrix yielding actions $x_t = -Ky_{t-1}$, there is an instance where $\mathsf{Reg}_T(\mathcal{P}_{\mathcal{K}}) \geq \Omega(T)$ for $\rho = 1$.

Proof of Proposition 12. We begin by observing that for instances $(\mathcal{X}, \mathcal{Y}, D)$, the class of state-targeting policies contains a policy which obtains the reward of the best fixed state up to $O(\sqrt{T\rho^{-1}})$, for sufficiently large T. Consider the set $\hat{\mathcal{Y}} = \{y^* \in \mathcal{Y} : \pi(y^*) \geq (T\rho)^{-1/2}\}$. Note that the reward of any $y \in \mathcal{Y}$ is matched by some $y^* \in \hat{\mathcal{Y}}$ up to $O(\sqrt{T\rho^{-1}})$ for any fixed inner radius r, outer radius R, and Lipschitz constant L. For any such y^*, note that under the policy P_{y^*} when starting at $y_0 = 0$, the distance between y_t and y^* in each round t is updated to at most:

$$\|y_t - y^*\| \leq \max\left(0, \rho \cdot \pi(y_{t-1})\right).$$

It is straightforward to see that $\hat{\mathcal{Y}}$ is convex, and so our state y_t will never leave $\hat{\mathcal{Y}}$ on its path to y^*; as such, we reach y^* within $O(\sqrt{T\rho^{-1}})$ rounds, after which point our reward exactly tracks that of y^*. For some $y^* \in \hat{\mathcal{Y}}$, this yields a regret for P_{y^*} of at most $O(\sqrt{T\rho^{-1}})$ to the best fixed state in $\mathcal{Y}$.

Next, consider an instance where $\mathcal{X}$ and $\mathcal{Y}$ are both the unit ball in $\mathbb{R}^n$. With $y_0 = 0$, let the

dynamics be given by

$$y_t = \Pi_{\mathcal{Y}}\left(y_{t-1} + x_t\right).$$

Observe that this satisfies ρ-local controllability for any $\rho \leq 1$, as a ball of radius $\pi(y_{t-1})$ is always feasible around y_{t-1}. Let each loss $f_t = \|y - p\|^2$, for some $p \neq 0$. Immediately we can see that any matrix policy $K \in \mathcal{P}_{\mathcal{K}}$ has regret $\Omega(T)$, as the action $x_t = 0$ will be played in each round. □

If dynamics are linear up to projection with $D(x_t, y_{t-1}) = \Pi_{\mathcal{Y}}(By + Ax)$ for full-rank A, and $\dim(\mathcal{X}) = \dim(\mathcal{Y})$, note that $P_{\hat{y}}(y) = A^{-1}(\hat{y} - By)$ implements any $P_{\hat{y}}$ for sufficiently large $\mathcal{X}$.

4.3 No-Regret Algorithms for Locally Controllable Dynamics

Here we give a sequence of no-regret algorithms satisfying a range of robustness properties. Our primary algorithm NestedOCO, presented in Section 4.3.1, operates over known time-varying dynamics without disturbances and requires an offline non-convex optimization oracle, and we identify conditions in Section 4.3.2 which remove the oracle requirement. In Section 4.3.3 we give two algorithms, NestedOCO-BD and NestedOCO-UD, which allow adversarial disturbances to weakly and strongly locally controllable dynamics, respectively. In Section **??** we obtain comparable bounds for all of our algorithms under bandit feedback for linear losses.

4.3.1 Nonlinear Control via Online Convex Optimization

When dynamics satisfy local controllability and y_{t-1} is not too close to $\mathrm{bd}(\mathcal{Y})$, all points y_t in a ball around y_{t-1} are feasible with an appropriate x_t; this enables execution of an online convex optimization (OCO) algorithm over $\mathcal{Y}$ by playing the action x_t which yields a state update to the target y_t chosen at each iteration, computed via offline non-convex optimization. Here we assume that D is known and can be queried for any inputs, and that disturbances to the state are not present. We allow the dynamics to change over time, potentially as a function of previous actions x_s and losses f_s for $s < t$, provided that D_t can be determined in each round. We use

Follow the Regularized Leader (FTRL) as our OCO subroutine [122, 123], yet we note that it may be substituted for any OCO algorithm whose per-round step size is guaranteed to be sufficiently small (such as OGD with a constant learning rate); statements of the FTRL algorithm and its key properties are provided in Appendix C.1. We instantiate FTRL over a contracted space $\tilde{\mathcal{Y}} \subseteq \mathcal{Y}$, calibrated to ensure that the minimum loss over $\tilde{\mathcal{Y}}$ is close to that for $\mathcal{Y}$, yet where each step of FTRL lies within the feasible region ensured by (weak) local controllability.

Algorithm 9 Nested Online Convex Optimization (NestedOCO).

Let $\psi : \mathcal{Y} \to \mathbb{R}$ be γ-strongly convex with $\operatorname{argmin}_y \psi(y) = \mathbf{0}$ and $\max_{y,y'} |\psi(y) - \psi(y')| \leq G$
Let $\eta = (G\gamma)^{1/2}((1 + \frac{R}{r\rho})TL^2)^{-1/2}$
Let $\widetilde{\mathcal{Y}} = \{y : \frac{1}{1-\delta} y \in \mathcal{Y}\}$ for $\delta = \eta \frac{L}{r\rho\gamma}$
Initialize FTRL to run for T rounds over $\widetilde{\mathcal{Y}}$ with regularizer ψ and parameter η
for $t = 1$ to T **do**
  Let y^* be the point chosen by FTRL
  Use $\texttt{Oracle}(y_{t-1}, y^*)$ to compute $x_t = \operatorname{argmin}_x \|D_t(x, y_{t-1}) - y^*\|^2$
  Play action x_t
  Observe y_t and loss $f_t(y_t)$, update FTRL
end for

Theorem 21. *For a ρ-locally controllable instance $(\mathcal{X}, \mathcal{Y}, D)$ without disturbances and with D_t known at each t, the regret of* NestedOCO *for convex L-Lipschitz losses $f_t : \mathcal{Y} \to \mathbb{R}$ is at most*

$$\mathsf{Reg}_T(\textsc{NestedOCO}) \leq 2L\sqrt{(1 + R(r\rho)^{-1})TG\gamma^{-1}}$$

with respect to any state $y^* \in \mathcal{Y}$, with T queries made to a non-convex optimization oracle.

Proof. First we show that any point chosen by FTRL will be feasible under local controllability, by induction. It is straightforward to see that $\tilde{\mathcal{Y}}$ is convex and $\tilde{\mathcal{Y}} \subseteq \mathcal{Y}$; further, any $y \in \tilde{\mathcal{Y}}$ is bounded away from $\mathrm{bd}(\mathcal{Y})$. By the definition of $\tilde{\mathcal{Y}}$, we have that $y = (1 - \delta)y'$ for some $y' \in \mathcal{Y}$. Recall that $\mathcal{B}_r(\mathbf{0}) \subseteq \mathcal{Y}$, and note that $\mathcal{B}_{\delta r}(y) = \{y + \delta\hat{y} : \hat{y} \in \mathcal{B}_r(\mathbf{0})\}$. Let y'' be any point in $\mathcal{B}_r(\mathbf{0})$. By convexity of $\mathcal{Y}$, we then have that any point $(1 - \delta)y' + \delta y''$ lies in $\mathcal{Y}$, and so for any $y \in \tilde{\mathcal{Y}}$ we have that $\mathcal{B}_{r\delta}(y) \subseteq \mathcal{Y}$. Each y_{t-1} lies in $\tilde{\mathcal{Y}}$, and so we have that $\pi(y_{t-1}) \geq r\delta$; as such, any point y_t in $\mathcal{B}_{r\delta\rho}(y_{t-1}) \subseteq \mathcal{B}_{\rho\cdot\pi(y_{t-1})}(y_{t-1})$ is feasible. Given that $\eta\frac{L}{\gamma} \leq r\delta\rho$, by Proposition 15 we have that

$y_t \in \mathcal{B}_{r\delta\rho}(y_{t-1})$ in each round for the chosen point. Each action will be selected by solving for

$$\operatorname*{argmin}_{x_t \in \mathcal{X}} \|D(x_t, y_{t-1}) - y^*\|^2$$

via a call to $\texttt{Oracle}(y_{t-1}, y^*)$. Each call is guaranteed to have a solution which achieves an objective of 0 where $D(x_t, y_{t-1}) = y^*$ for some $y^* \in \mathcal{B}_{\rho \cdot \pi(y_{t-1})}(y_{t-1})$ by local controllability, yielding an exact state update to $y_t = y^*$ as we assume `Oracle` can solve arbitrary non-convex minimization problems. To bound the regret, first note that for any $y^* \in \mathcal{Y}$, we have

$$\sum_{t=1}^{T} f_t(y_t) \leq \eta \frac{TL^2}{\gamma} + \frac{G}{\eta} + \sum_{t=1}^{T} f_t((1-\delta)y^*)$$

by Proposition 14, as $(1-\delta)y^* \in \tilde{\mathcal{Y}}$ for any $y^* \in \mathcal{Y}$. Then, observe that for any $y^* \in \mathcal{Y}$, we have that

$$\begin{aligned}
\sum_{t=1}^{T} f_t((1-\delta)y^*) &\leq \sum_{t=1}^{T} \left(f_t(y^*) + L\|\delta y^*\|\right) \\
&\leq \sum_{t=1}^{T} \left(f_t(y^*) + \delta LR\right).
\end{aligned}$$

Combining the previous claims, we have that

$$\begin{aligned}
\sum_{t=1}^{T} f_t(y_t) - f_t(y^*) &\leq \delta TLR + \eta \frac{TL^2}{\gamma} + \frac{G}{\eta} \\
&= \eta \left(1 + \frac{R}{r\rho}\right) \frac{TL^2}{\gamma} + \frac{G}{\eta} \\
&= 2\sqrt{\frac{(1+\frac{R}{r\rho})TGL^2}{\gamma}}
\end{aligned}$$

upon setting $\delta = \eta \frac{L}{r\rho\gamma}$ and $\eta = \sqrt{\frac{G\gamma}{(1+\frac{R}{r\rho})TL^2}}$, which yields the theorem. □

4.3.2 Efficient Updates for (Locally) Action-Linear Dynamics

While NestedOCO requires no assumptions on the dynamics beyond local controllability, there are large classes of dynamics for which the oracle call can be removed. We say that dynamics are *action-linear* if $y_x = D(x, y)$ is linear in x, for $y_x \in \text{int}(\mathcal{Y})$ (and arbitrary for $y_x \in \text{bd}(\mathcal{Y})$).

Proposition 13. *For a ρ-locally controllable and action-linear instance $(\mathcal{X}, \mathcal{Y}, D)$, the per-round optimization problem for* `Oracle`(y_{t-1}, y^*) *in* NestedOCO *is convex.*

Proof. For $y = y_{t-1} \in \widetilde{\mathcal{Y}} \subseteq \text{int}(\mathcal{Y})$, we have $D(x, y) = A_y \cdot x + b_y$ for some matrix A_y and vector b_y, and so we can solve $x_t = \text{argmin}_{x \in \mathcal{X}} \left\| A_y \cdot x + b_y - y^* \right\|^2$ efficiently. □

The class of action-linear dynamics is quite general, owing to the flexibility permitted by nonlinear parameterizations of (A_y, b_y) in terms of y. Here we show that local controllability holds for explicit families of instances when appropriate eigenvalue conditions are satisfied. As a simple yet general example of dynamics which are both action-linear and locally controllable, consider update rules in which a step is taken by applying a nonsingular matrix transformation to the action, where the matrix can be parameterized by the state, with projection back into $\mathcal{Y}$ if necessary.

Example 1. *Let both $\mathcal{X}$ and $\mathcal{Y}$ be given by the unit ball $\mathcal{B}_1(\mathbf{0})$ in $\mathbb{R}^n$. For any fixed y, let the updates from $D(x, y)$ be given by*

$$D(x, y) = \Pi_{\mathcal{Y}} \left(y + A_y \cdot x \right),$$

where each A_y is a square matrix with minimum absolute eigenvalue $\left|\lambda_n(A_y)\right| \geq \pi(y) \cdot \rho$ for some $\rho > 0$. Then, the instance $(\mathcal{X}, \mathcal{Y}, D)$ is action-linear and satisfies ρ-local controllability.

Proof. It is straightforward to see that $D(x, y)$ is action-linear. To show ρ-local controllability, let y^* be any point in $\mathcal{B}_{\rho \cdot \pi(y)}(y)$. It suffices to show that there is some $x^* \in \mathcal{X}$ such that $A_y \cdot x^* = y^* - y$. As A_y is non-singular, we can solve for $x^* = A_y^{-1}(y^* - y)$, where $\|y^* - y\| \leq \rho \cdot \pi(y)$ and $\left|\lambda_1(A_y^{-1})\right| \leq \frac{1}{\rho \cdot \pi(y)}$, and so we have that $x^* \in \mathcal{B}_1(\mathbf{0}) = \mathcal{X}$. □

We can also extend this to include state-parameterized generalizations of any linear system governed by nonsingular matrices over a bounded-radius state space (for a sufficiently large action space).

Example 2. *Let $\mathcal{Y}$ be given by the radius-R ball $\mathcal{B}_R(\mathbf{0})$ in $\mathbb{R}^n$, and let $\mathcal{X} = \mathcal{B}_{cR}(\mathbf{0})$. For any fixed y, let the updates from $D(x, y)$ be given by*

$$D(x, y) = \Pi_{\mathcal{Y}} \left(K_y \cdot y + A_y \cdot x \right),$$

where both K_y and A_y are square matrices. For any y, let $M_y = K_y - I$, and suppose we take c large enough such that $c \cdot \left|\lambda_n(A_y)\right| \geq \left|\lambda_1(M_y)\right| + \pi(y) \cdot \rho$ for some $\rho > 0$. Then, the instance $(\mathcal{X}, \mathcal{Y}, D)$ is action-linear and satisfies ρ-local controllability.

Proof. Here, again it is evident that $D(x, y)$ is action-linear, and so it suffices to show that there is some $x^* \in \mathcal{X}$ such that

$$\begin{aligned} K_y \cdot y + A_y \cdot x^* &= y + M_y \cdot y + A_y \cdot x^* \\ &= y^* \end{aligned}$$

for any y^* in $\mathcal{B}_{\rho \cdot \pi(y)}(y)$. As in the proof for Example 1, we have that $\left\|M_y \cdot y\right\| \leq R \cdot \left|\lambda_1(M_y)\right|$, and for large enough c there is some x^* such that $A_y \cdot x^* = \hat{y}$ for any $\hat{y}$ where $\|\hat{y}\| \leq R \cdot \left|\lambda_1(M_y)\right| + \pi(y) \cdot \rho$. Thus, any point $y^* \in \mathcal{B}_{R \cdot \left|\lambda_1(M_y)\right| + \pi(y) \cdot \rho}(y + M_y \cdot y)$ is feasible by some x^*, which contains the ball $\mathcal{B}_{\pi(y) \cdot \rho}(y)$. □

Note that our target y_t will always be near y_{t-1}; it will in fact suffice for computational efficiency if $D(x, y)$ is only approximately action-linear in the neighborhood around stabilizing actions for y.

4.3.3 Adversarial Disturbances

Our algorithm NestedOCO can be extended to accommodate adversarial disturbances, where the state is updated as $y_t = D(x_t, y_{t-1}) + w_t$, with $\{w_t\}$ chosen adversarially. In the weak local

controllability case, we show a sharp threshold effect in terms of whether or not $\|w_t\|$ is allowed to exceed the undisturbed distance from the boundary by a factor of $\frac{\rho}{1+\rho}$: if disturbances are bounded below this threshold, regret minimization remains feasible with a tight $\Theta(E)$ dependence on the total disturbance magnitude, yet if disturbances may exceed this, no sublinear regret rate is attainable even for a *constant* total disturbance magnitude. When ρ is small, an adversary can push us to the boundary faster than we can "undo" past disturbances, causing our feasible range to decay.

Theorem 22 (Bounded Disturbances for Weak Local Controllability)**.** *For any $\rho \in (0, 1]$, suppose that a sequence of adversarial disturbances w_t for a ρ-locally controllable instance $(\mathcal{X}, \mathcal{Y}, D)$ satisfies $\sum_{t=1}^{T} \|w_t\| \leq E$ and $\|w_t\| \leq \frac{\rho-\alpha\rho}{1+\rho} \cdot \pi\left(D(x_t, y_{t-1})\right)$, for some $\alpha \in \mathbb{R}$. If $\alpha > 0$, there is an algorithm* NestedOCO-BD *with regret for convex Lipschitz losses f_t bounded by*

$$\mathsf{Reg}_T(\textsc{NestedOCO-BD}) \leq O\left(\sqrt{T \cdot (\alpha\rho)^{-1}} + E\right),$$

and there is an instance where any algorithm $\mathcal{A}$ obtains $\mathsf{Reg}_T(\mathcal{A}) = \Omega(E)$. If $\alpha < 0$, there is an instance such that any algorithm $\mathcal{A}$ obtains $\mathsf{Reg}_T(\mathcal{A}) \geq \Omega(T)$ even when $E = O(1)$.

The maximum disturbance bound can be removed when dynamics are strongly locally controllable, as the ensured feasible range of the dynamics does not vanish at the boundary of the state space. For such instances, we can minimize regret (with tight $O(E \cdot \rho^{-1})$ dependence) even if disturbances are only implicitly bounded by the state space diameter (which is at least ρ, without loss of generality).

Theorem 23 (Unbounded Disturbances for Strong Local Controllability)**.** *For any $\rho > 0$ and strongly ρ-locally controllable instance $(\mathcal{X}, \mathcal{Y}, D)$ with disturbances w_t satisfying $\sum_{t=1}^{T} \|w_t\| \leq E$, there is an algorithm* NestedOCO-UD *with regret for convex Lipschitz losses f_t bounded by*

$$\mathsf{Reg}_T(\textsc{NestedOCO-UD}) \leq O\left(\sqrt{T} + E \cdot \rho^{-1}\right),$$

and there is an instance where any algorithm $\mathcal{A}$ obtains $\mathsf{Reg}_T(\mathcal{A}) \geq \Omega\left(E \cdot \rho^{-1}\right)$.

In each case, our lower bounds in terms of E hold for the same constants obtained by our algorithms, and our algorithms obtain the stated regret guarantees even when E is not known in advance. We present the algorithms and analysis for each theorem in Appendix C.2; both proceed by tracking deviations from an idealized trajectory without disturbances, and calibrating parameters to preserve sufficient reachability margin for applying corrections towards this trajectory in each round.

4.4 Applications at a Glance

We give several applications of our framework to online Stackelberg problems involving strategic or adaptive agents, each cast as an instance of online control with nonlinear dynamics where local controllability holds, and where our objectives are well-approximated by convex surrogate losses only over the state. Each application extends prior work by either allowing for more relaxed assumptions, unifying distinct problem instances, or giving a novel formulation to account for dynamic and adversarial behavior; analysis and comparison to related work is contained in Appendices C.3-C.6.

4.4.1 Online Performative Prediction

Performative prediction was introduced by [69] to capture settings in which the data distribution may shift as a function of the classifier itself. We consider the online formulation of Performative Prediction introduced in [93] as an instance of online convex optimization with unbounded memory, which we extend to accommodate a *stateful* variant of the problem (as in [91]) in which the update to the distribution is a function of both the classifier and the current distribution itself. Let $\mathcal{X} \subseteq \mathbb{R}^n$ denote our space of classifiers, and let p_0 be the initial distribution over $\mathbb{R}^n$. When a classifier x_t is deployed, the distribution is updated to

$$p_t = (1 - \theta)p_{t-1} + \theta\, \mathcal{D}(x_t, y_{t-1})$$

where $\mathcal{D}(x_t, y) = A(x_t, y_{t-1}) + \xi$, for a random variable $\xi \in \mathbb{R}^n$ with mean μ and covariance Σ, and with $y_t = A(x_t, y_{t-1})$, where A satisfies ρ-local controllability for some $\rho > 0$ and appropriate smoothness notions. We also assume there is some linear $s : \mathcal{X} \to \mathcal{Y}$ such that $A(x, y) = s(x)$ if $y = s(x)$. We then receive loss $\tilde{f}_t(x_t, p_t) = \mathbb{E}_{z \sim p_t}[f_t(x_t, z)]$, where each f_t is convex and Lipschitz.

This generalizes the model of [93], in which $A(x, y) = A \in \mathbb{R}^{n \times n}$ is taken to be a fixed matrix; there, ρ-local controllability is satisfied for some $\rho > 0$ provided that A is nonsingular. Their aim is to compete with the best fixed classifier by running regret minimization over $\mathcal{X}$. Here we run NestedOCO over $\mathcal{Y}$, taken over the range of s, which allows us to compete against the best fixed classifier as well by the properties of s; while the classifiers x_t we play will generally not result in stabilizing points of A, their excess loss compared to each $s^{-1}(y_t)$ is bounded.

Theorem 24 (Regret Minimization for Performative Prediction)**.** *For any $\theta > 0$, the dynamics for Online Performative Prediction are ρ-locally controllable, and* NestedOCO *obtains regret $O(\sqrt{T(\rho^{-1} + \theta^{-1})})$ with respect to the best fixed classifier.*

4.4.2 Adaptive Recommendations

Online interactions with economic agents of various types are ubiquitous, and the resulting control problems tend to be manifestly nonlinear; here we treat two diverse examples from this space. In the adaptive recommendations problem of Chapter 3 (first introduced in [121]) we provide menu recommendations repeatedly to an agent, whose choice distribution is a function of their past selections, while our reward in each round depends on adversarial losses over the agent's choice. In each round $t \in [T]$, we show the agent a (possibly randomized) menu K_t containing k (out of n) items, and the agent's instantaneous choice distribution conditioned on seeing K_t is

$$p_t(i; K_t, v_{t-1}) = \begin{cases} \frac{s_i(v_{t-1})}{\sum_{j \in K_t} s_j(v_{t-1})} & i \in K_t \\ 0 & i \notin K_t \end{cases}$$

where each $s_i : \Delta(n) \to [\lambda, 1]$ is the agent's *preference scoring function* for item i, for some $\lambda > 0$, taking as input the agent's *memory vector* $v \in \Delta(n)$. The memory vector updates each round as

$$v_t = (1 - \theta_t)v_{t-1} + \theta_t p_t,$$

where $\theta_t \in [\theta, 1]$ for $\theta > 0$ is a possibly time-dependent update speed, and we receive loss $f_t(p_t)$, where each f_t is convex and L-Lipschitz. Note that the set of feasible choice distributions when considering all menu distributions $x_t \in \Delta(\binom{n}{k})$ depends on the memory vector v_t. The regret benchmark considered by [121] is the intersection of all such sets, denoted the "everywhere instantaneously-realizable distribution" set $\text{EIRD} = \cap_{v \in \Delta} \text{IRD}(v)$, where $\text{IRD}(v)$ is the "instantaneously realizable distribution" set for v, given as the convex hull of the choice distributions $p(K_t)$ resulting from each menu $K_t \in [\binom{n}{k}]$ when v is the memory vector. It is shown that the set is non-empty when λ is not too small, and algorithms which minimize regret with respect to any distribution in EIRD are given in [121] and [33] under varying assumptions regarding the scoring functions and update speed.

While the prior work considers a bandit version of the problem with unknown dynamics, here we consider a full-feedback deterministic variant of the problem for simplicity, which further allows us to circumvent barriers posed by uncertainty [121, 33] and relax structural assumptions (e.g. on θ_t or s_i). We can cast this as an instance of our framework by taking $\mathcal{X} = \Delta(\binom{n}{k})$ and $\mathcal{Y} = \text{EIRD}$, where D expresses updates to the memory vector. We assume $v_0 = \mathbf{u}_n$, and we reparameterize to run our algorithm over $\Delta(n)$. We optimize surrogate losses $f_t^*(v_t)$, and bound excess regret from $f_t(p_t)$.

Theorem 25 (Regret Minimization over EIRD)**.** *For $\lambda > \frac{k-1}{n-1}$, the dynamics for Adaptive Recommendations over* EIRD *are θ-locally controllable, and* NestedOCO *obtains regret $O(\sqrt{T\theta^{-1}})$.*

In [33], a property for scoring functions is considered which enables regret minimization over a potentially much larger set of distributions than EIRD. A scoring function $s_i : \Delta(n) \to [\frac{\lambda}{\sigma}, 1]$ is

said to be (σ, λ)-*scale-bounded* for $\sigma > 1$ if, for all $v \in \Delta(n)$, we have that

$$\sigma^{-1}((1-\lambda)v_i + \lambda) \leq s_i(v) \leq \sigma((1-\lambda)v_i + \lambda).$$

The set considered is the ϕ-*smoothed simplex* $\Delta^\phi(n) = \{(1-\phi)v + \phi\mathbf{u}_n : v \in \Delta(n)\}$, for $\phi = \Theta(k\lambda\sigma^2)$, where it is shown that $\mathrm{IRD}(v)$ contains a ball around v for $v \in \Delta^\phi(n)$. We take $\mathcal{Y} = \Delta^\phi(n)$, which satisfies local controllability, and optimize over $f_t^*(v_t)$ with NestedOCO.

Theorem 26 (Regret Minimization over $\Delta^\phi(n)$)**.** *For (σ, λ)-scale-bounded scoring functions s_i, for any $\lambda > 0$ and $\sigma > 1$, the dynamics for Adaptive Recommendations over $\Delta^\phi(n)$ are $\Omega(\theta\lambda\phi)$-locally controllable, and* NestedOCO *obtains regret $O(\sqrt{T(\theta\lambda\phi)^{-1}})$.*

4.4.3 Adaptive Pricing

Here we consider an Adaptive Pricing problem for real-valued goods, formulated as a dynamic extension of the setting of [75] where purchase history and consumption affect demand. In each round we set per-unit price vectors $p_t \in \mathbb{R}^n_+$, and an agent buys some bundle of goods $x_t \in \mathbb{R}^n_+$, which results in us obtaining a reward $\langle p_t, x_t\rangle - c_t(x_t)$, where our production cost function c_t at each round is convex and L_c-Lipschitz, and may be chosen adversarially.

Departing from [75], we consider an agent who maintains goods reserves $y_{t-1} \in \mathbb{R}^n_{\geq 0}$ and consumes an adversarially chosen fraction $\theta_t \in [\theta, 1]$ of every good's reserve at each round (for some $\theta > 0$). The agent then chooses a bundle x_t to maximize their utility $g(p_t, x_t, y_t) = v(y_t) - \langle p_t, x_t\rangle$, where $y_t = (1-\theta_t)y_{t-1} + x_t$ is their updated reserve bundle. We make several regularity assumptions on the agent's valuation function $v : \mathbb{R}^n_+ \to \mathbb{R}_+$, all of which are satisfied by several classically studied utility families (which we discuss in Appendix 4.4.3). Notably, we assume that v is strictly concave and increasing, and homogeneous; the range is bounded under rationality.

Our aim will be to set prices which allow us to compete with the best *stable reserve policy*, e.g. against any pricing policy where the agent maintains the same reserve bundle $y_t = y^*$ at each round

for some y^* regardless of θ_t. We take an appropriate convex set of such bundles as our state space, for which we show that local controllability holds. Observe that to induce a purchase of $x_t = \theta_t y_{t-1}$, it suffices to set prices $p_t = \nabla v(y_{t-1})$, as we then have that $\nabla_{x_t}(v((1-\theta_t)y_{t-1} + x_t) - \langle p_t, x_t\rangle) = \mathbf{0}$. By homogeneity of v, we also have that $\langle \nabla v(y_t), \theta_t y_t\rangle = \theta_t k \cdot v(y_t)$ for some k, and we show that optimization via the concave surrogate rewards

$$f_t^*(y_t) = \theta_t k \cdot v(y_t) - c_t(\theta_t y_t)$$

will closely track our true rewards $f_t(p_t, x_t) = \langle p_t, x_t\rangle - c_t(x_t)$. While neither our true nor surrogate rewards will be Lipschitz, we extend NestedOCO to obtain sublinear regret over Hölder continuous losses by appropriately calibrating our step size (which may be of independent interest).

Theorem 27 (Regret Minimization over Stable Reserve Policies)**.** *For any $\theta > 0$, the dynamics for Adaptive Pricing can are θ-locally controllable, and* NestedOCO *obtains regret $o(T\theta^{-1})$ with respect to the best stable reserve policy.*

4.4.4 Steering Learners in Online Games

A recent line of work [92, 114, 6] explores maximizing rewards in a repeated game against a no-regret learner, and [94] study of no-regret dynamics in time-varying games. We consider these questions in unison, and aim to optimize reward against a no-regret learner for game matrices chosen adversarially and online.

Consider adversarial sequences of two-player $m \times n$ bimatrix games (A_t, B_t), where $m > n$; we assume that the convex hull of the rows of each B_t contains the unit ball. As Player A, we choose strategies $x_t \in \Delta(m)$ each round to maximize our reward against Player B, who chooses their strategies $y_t \in \Delta(n)$ according to a no-regret algorithm (in particular, online projected gradient descent). The game (A_t, B_t) is only revealed after both players have chosen strategies for round t. Our aim here is to illustrate the feasibility of *steering* the opponent's trajectory, and so we consider games where Player A's reward is predominantly a function only of Player B's actions. We assume

that $\left\|xA_t - xA_t^*\right\| \leq \delta_t$ for any $x \in \Delta(m)$, where each A_t^* is a matrix with identical rows, and that per-round changes to B_t are bounded, with $\|xB_t - xB_{t-1}\| \leq \epsilon_t$ for any $x \in \Delta(m)$. We measure the regret of an algorithm $\mathcal{A}$ with respect to *any* profile $(x, y) \in \Delta(m) \times \Delta(n)$, where

$$\mathsf{Reg}_T(\mathcal{A}) = \max_{(x,y)\in\Delta(m)\times\Delta(n)} \sum_{t=1}^{T} xA_t y - x_t A_t y_t.$$

When Player B plays OGD with step size $\theta = \Theta(T^{-1/2})$, their strategy updates each round as

$$y_{t+1} = \Pi_{\Delta(n)}\left(y_t + \theta(x_t B_t)\right),$$

with $y_1 = \mathbf{u}_n$, and yields regret $O(\sqrt{T})$ for Player B with respect to any $y \in \Delta(n)$ for the loss sequence $\{x_t B_t : t \in [T]\}$. To cast this in our framework, we consider $\Delta(n) = \mathcal{Y}$ as our state space, where we select actions x_{t-1} to induce desired updates to y_t and optimize over the surrogate losses $\{\mathbf{u}_m A_t^* y_t : t \in [T]\}$. While we do not see B_t prior to choosing each x_t, we view our update errors from instead selecting an action in terms of the dynamics resulting from B_{t-1} as adversarial disturbances and run NestedOCO-UD, as the dynamics are strongly locally controllable.

Theorem 28 (Regret Minimization in Online Games)**.** *For $\theta = \Theta(T^{-1/2})$, repeated play against* OGD *in online $m \times n$ games can be cast as a θ-strongly locally controllable instance of online control with nonlinear dynamics, for which* NestedOCO-UD *obtains regret $O(\sqrt{T} + \sum_t(\delta_t + \epsilon_t))$.*

References

[1] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ser. ICML'03, Washington, DC, USA: AAAI Press, 2003, 928–935, ISBN: 1577351894.

[2] E. Hazan, "Introduction to online convex optimization," *CoRR*, vol. abs/1909.05207, 2019. arXiv: 1909.05207.

[3] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. USA: Cambridge University Press, 2007, ISBN: 0521872820.

[4] E. Hazan and K. Singh, *Introduction to online nonstochastic control*, 2022. arXiv: 2211.09619 [cs.LG].

[5] M. Hardt and C. Mendler-Dünner, *Performative prediction: Past and future*, 2023. arXiv: 2310.16608 [cs.LG].

[6] W. Brown, J. Schneider, and K. Vodrahalli, *Is learning in games good for the learners?* 2023. arXiv: 2305.19496 [cs.GT].

[7] H. Moulin and J. P. Vial, "Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon," *International Journal of Game Theory*, vol. 7, pp. 201–221, 1978.

[8] D. P. Foster and R. V. Vohra, "Asymptotic calibration," *Biometrika*, vol. 85, no. 2, pp. 379–390, 1998.

[9] S. Hart and A. Mas-Colell, "A simple adaptive procedure leading to correlated equilibrium," *Econometrica*, vol. 68, no. 5, pp. 1127–1150, 2000.

[10] C. Daskalakis, A. Deckelbaum, and A. Kim, "Near-optimal no-regret algorithms for zero-sum games," in *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, SIAM, 2011, pp. 235–254.

[11] I. Anagnostides, C. Daskalakis, G. Farina, M. Fishelson, N. Golowich, and T. Sandholm, "Near-optimal no-regret learning for correlated equilibria in multi-player general-sum games," in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, 2022, pp. 736–749.

[12] C. Daskalakis, M. Fishelson, and N. Golowich, "Near-optimal no-regret learning in general games," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 604–27 616, 2021.

[13] A. Blum, M. Hajiaghayi, K. Ligett, and A. Roth, "Regret minimization and the price of total anarchy," in *Proceedings of the fortieth annual ACM symposium on Theory of computing*, 2008, pp. 373–382.

[14] J. Hartline, V. Syrgkanis, and E. Tardos, "No-regret learning in bayesian games," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[15] C. Daskalakis, R. M. Frongillo, C. H. Papadimitriou, G. Pierrakos, and G. Valiant, "On learning algorithms for nash equilibria.," in *SAGT*, Springer, 2010, pp. 114–125.

[16] K. Ligett and G. Piliouras, "Beating the best nash without regret," *SIGecom Exchanges*, vol. 10, no. 1, pp. 23–26, 2011.

[17] P. Mertikopoulos, C. Papadimitriou, and G. Piliouras, "Cycles in adversarial regularized learning," in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 2018, pp. 2703–2717.

[18] G. J. Gordon, A. Greenwald, and C. Marks, "No-regret learning in convex games," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08, Helsinki, Finland: Association for Computing Machinery, 2008, 360–367, ISBN: 9781605582054.

[19] Y. Deng, J. Schneider, and B. Sivan, "Strategizing against no-regret learners," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[20] V. Conitzer and T. Sandholm, "Computing the optimal strategy to commit to," in *Proceedings of the 7th ACM Conference on Electronic Commerce*, ser. EC '06, Ann Arbor, Michigan, USA: Association for Computing Machinery, 2006, 82–90, ISBN: 1595932364.

[21] J. Letchford, V. Conitzer, and K. Munagala, "Learning and approximating the optimal strategy to commit to," in *Algorithmic Game Theory*, 2009.

[22] B. Peng, W. Shen, P. Tang, and S. Zuo, "Learning optimal strategies to commit to," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'19/IAAI'19/EAAI'19, Honolulu, Hawaii, USA: AAAI Press, 2019, ISBN: 978-1-57735-809-1.

[23] M. Braverman, J. Mao, J. Schneider, and M. Weinberg, "Selling to a no-regret buyer," in *Proceedings of the 2018 ACM Conference on Economics and Computation*, 2018, pp. 523–538.

[24] Y. Mansour, M. Mohri, J. Schneider, and B. Sivan, "Strategizing against learners in bayesian games," in *Conference on Learning Theory*, PMLR, 2022, pp. 5221–5252.

[25] M.-F. Balcan, A. Blum, N. Haghtalab, and A. D. Procaccia, "Commitment without regrets: Online learning in stackelberg security games," in *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, ser. EC '15, Portland, Oregon, USA: Association for Computing Machinery, 2015, 61–78, ISBN: 9781450334105.

[26] Z. Huang, J. Liu, and X. Wang, "Learning optimal reserve price against non-myopic bidders," *CoRR*, vol. abs/1804.11060, 2018. arXiv: 1804.11060.

[27] D. Goktas, J. Zhao, and A. Greenwald, *Robust no-regret learning in min-max stackelberg games*, 2022.

[28] N. Haghtalab, T. Lykouris, S. Nietert, and A. Wei, *Learning in stackelberg games with non-myopic agents*, 2022. arXiv: 2208.09407 [cs.GT].

[29] A. Greenwald and A. Jafari, "A general class of no-regret learning algorithms and game-theoretic equilibria," in *Learning Theory and Kernel Machines*, B. Schölkopf and M. K. Warmuth, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 2–12, ISBN: 978-3-540-45167-9.

[30] L. Zhang, T. Yang, R. Jin, and Z.-H. Zhou, "Dynamic regret of strongly adaptive methods," in *International Conference on Machine Learning*, 2017.

[31] S. Arora, E. Hazan, and S. Kale, "The multiplicative weights update method: A meta-algorithm and applications," *Theory of Computing*, vol. 8, no. 6, pp. 121–164, 2012.

[32] A. Agarwal and W. Brown, "Diversified recommendations for agents with adaptive preferences," in *NeurIPS*, 2022.

[33] A. Agarwal and W. Brown, "Online recommendations for agents with discounted adaptive preferences," in *ALT*, 2024.

[34] S. Flaxman, S. Goel, and J. M. Rao, "Filter Bubbles, Echo Chambers, and Online News Consumption," *Public Opinion Quarterly*, vol. 80, no. S1, pp. 298–320, Mar. 2016. eprint: https://academic.oup.com/poq/article-pdf/80/S1/298/17120810/nfw006.pdf.

[35] M. Curmei, A. A. Haupt, B. Recht, and D. Hadfield-Menell, "Towards psychologically-grounded dynamic preference models," in *RecSys '22: Sixteenth ACM Conference on Recommender Systems, Seattle, WA, USA, September 18 - 23, 2022*, J. Golbeck *et al.*, Eds., ACM, 2022, pp. 35–48.

[36] H. Abdollahpouri, R. Burke, and B. Mobasher, *Managing popularity bias in recommender systems with personalized re-ranking*, 2019. arXiv: 1901.07555 [cs.IR].

[37] M. Mansoury, H. Abdollahpouri, M. Pechenizkiy, B. Mobasher, and R. Burke, "Feedback loop and bias amplification in recommender systems," *CoRR*, vol. abs/2007.13019, 2020. arXiv: 2007.13019.

[38] Y. Ge *et al.*, "Understanding echo chambers in e-commerce recommender systems," *CoRR*, vol. abs/2007.02474, 2020. arXiv: 2007.02474.

[39] D. O'Callaghan, D. Greene, M. Conway, J. Carthy, and P. Cunningham, "Down the (white) rabbit hole: The extreme right and online recommender systems," *Social Science Computer Review*, vol. 33, no. 4, pp. 459–478, 2015.

[40] A. J. B. Chaney, B. M. Stewart, and B. E. Engelhardt, "How algorithmic confounding in recommendation systems increases homogeneity and decreases utility," *CoRR*, vol. abs/1710.11214, 2017. arXiv: 1710.11214.

[41] M. Mladenov, E. Creager, O. Ben-Porat, K. Swersky, R. S. Zemel, and C. Boutilier, "Optimizing long-term social welfare in recommender systems: A constrained matching approach," *CoRR*, vol. abs/2008.00104, 2020. arXiv: 2008.00104.

[42] T. Hassan, "Trust and trustworthiness in social recommender systems," in *Companion Proceedings of The 2019 World Wide Web Conference*, ser. WWW '19, San Francisco, USA: Association for Computing Machinery, 2019, 529–532, ISBN: 9781450366755.

[43] X. Zhao, Z. Zhu, and J. Caverlee, "Rabbit holes and taste distortion: Distribution-aware recommendation with evolving interests," in *Proceedings of the Web Conference 2021*, ser. WWW '21, Ljubljana, Slovenia: Association for Computing Machinery, 2021, 888–899, ISBN: 9781450383127.

[44] E. Ie *et al.*, "Slateq: A tractable decomposition for reinforcement learning with recommendation sets," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, S. Kraus, Ed., ijcai.org, 2019, pp. 2592–2599.

[45] R. Zhan *et al.*, "Towards content provider aware recommender systems: A simulation study on the interplay between user and provider utilities," in *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, J. Leskovec, M. Grobelnik, M. Najork, J. Tang, and L. Zia, Eds., ACM / IW3C2, 2021, pp. 3872–3883.

[46] M. Chen, A. Beutel, P. Covington, S. Jain, F. Belletti, and E. H. Chi, "Top-k off-policy correction for a REINFORCE recommender system," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC,*

Australia, February 11-15, 2019, J. S. Culpepper, A. Moffat, P. N. Bennett, and K. Lerman, Eds., ACM, 2019, pp. 456–464.

[47] K. Burghardt and K. Lerman, "Emergent instabilities in algorithmic feedback loops," *CoRR*, vol. abs/2201.07203, 2022. arXiv: 2201.07203.

[48] A. Sinha, D. F. Gleich, and K. Ramani, "Deconvolving feedback loops in recommender systems," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16, Barcelona, Spain: Curran Associates Inc., 2016, 3251–3259, ISBN: 9781510838819.

[49] J. Hazla, Y. Jin, E. Mossel, and G. Ramnarayan, "A geometric model of opinion polarization," *CoRR*, vol. abs/1910.05274, 2019.

[50] J. Gaitonde, J. M. Kleinberg, and É. Tardos, "Polarization in geometric opinion dynamics," in *EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*, P. Biró, S. Chawla, and F. Echenique, Eds., ACM, 2021, pp. 499–519.

[51] S. Dean and J. Morgenstern, "Preference dynamics under personalized recommendations," in *EC '22: The 23rd ACM Conference on Economics and Computation, Boulder, CO, USA, July 11 - 15, 2022*, D. M. Pennock, I. Segal, and S. Seuken, Eds., ACM, 2022, pp. 795–816.

[52] M. Jagadeesan, N. Garg, and J. Steinhardt, *Supply-side equilibria in recommender systems*, 2022. arXiv: 2206.13489 [cs.GT].

[53] T. Zhou, J. Liu, C. Dong, and J. Deng, "Incentivized bandit learning with self-reinforcing user preferences," *CoRR*, vol. abs/2105.08869, 2021. arXiv: 2105.08869.

[54] Y. Yue and T. Joachims, "Interactively optimizing information retrieval systems as a dueling bandits problem," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09, Montreal, Quebec, Canada: Association for Computing Machinery, 2009, 1201–1208, ISBN: 9781605585161.

[55] Y. Yue, J. Broder, R. Kleinberg, and T. Joachims, "The k-armed dueling bandits problem," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1538–1556, 2012, JCSS Special Issue: Cloud Computing 2011.

[56] A. Agarwal, N. Johnson, and S. Agarwal, "Choice bandits," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 18 399–18 410.

[57] A. Rangi, M. Franceschetti, and L. Tran-Thanh, *Sequential choice bandits with feedback for personalizing users' experience*, 2021.

[58] J. C. Gittins, "Bandit processes and dynamic allocation indices," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 41, no. 2, pp. 148–164, 1979.

[59] H. Heidari, M. J. Kearns, and A. Roth, "Tight policy regret bounds for improving and decaying bandits," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, S. Kambhampati, Ed., IJCAI/AAAI Press, 2016, pp. 1562–1570.

[60] N. Levine, K. Crammer, and S. Mannor, "Rotting bandits," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon *et al.*, Eds., 2017, pp. 3074–3083.

[61] R. Kleinberg and N. Immorlica, "Recharging bandits," in *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, M. Thorup, Ed., IEEE Computer Society, 2018, pp. 309–319.

[62] V. Shah, J. H. Blanchet, and R. Johari, "Bandit learning with positive externalities," *CoRR*, vol. abs/1802.05693, 2018. arXiv: 1802.05693.

[63] L. Leqi, F. Kilinç-Karzan, Z. C. Lipton, and A. L. Montgomery, "Rebounding bandits for modeling satiation effects," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 4003–4014.

[64] P. Laforgue, G. Clerici, N. Cesa-Bianchi, and R. Gilad-Bachrach, "A last switch dependent analysis of satiation and seasonality in bandits," in *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, G. Camps-Valls, F. J. R. Ruiz, and I. Valera, Eds., ser. Proceedings of Machine Learning Research, vol. 151, PMLR, 2022, pp. 971–990.

[65] P. Awasthi, K. Bhatia, S. Gollapudi, and K. Kollias, "Congested bandits: Optimal routing via short-term resets," in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, Eds., ser. Proceedings of Machine Learning Research, vol. 162, PMLR, 2022, pp. 1078–1100.

[66] O. Papadigenopoulos, C. Caramanis, and S. Shakkottai, "Non-stationary bandits under recharging payoffs: Improved planning with sublinear regret," *CoRR*, vol. abs/2205.14790, 2022. arXiv: 2205.14790.

[67] J. Altschuler and K. Talwar, "Online learning over a finite action set with limited switching," in *Proceedings of the 31st Conference On Learning Theory*, S. Bubeck, V. Perchet, and

P. Rigollet, Eds., ser. Proceedings of Machine Learning Research, vol. 75, PMLR, 2018, pp. 1569–1573.

[68] S. Bubeck, B. Klartag, Y. T. Lee, Y. Li, and M. Sellke, "Chasing nested convex bodies nearly optimally," in *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2020, pp. 1496–1508. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611975994.91.

[69] J. C. Perdomo, T. Zrnic, C. Mendler-Dünner, and M. Hardt, "Performative prediction," *CoRR*, vol. abs/2002.06673, 2020. arXiv: 2002.06673.

[70] C. Mendler-Dünner, J. Perdomo, T. Zrnic, and M. Hardt, "Stochastic optimization for performative prediction," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 4929–4939.

[71] M. Jagadeesan, T. Zrnic, and C. Mendler-Dünner, "Regret minimization with performative feedback," *CoRR*, vol. abs/2202.00628, 2022. arXiv: 2202.00628.

[72] T. Zrnic, E. V. Mazumdar, S. S. Sastry, and M. I. Jordan, "Who leads and who follows in strategic classification?" In *NeurIPS*, 2021.

[73] M. Braverman, J. Mao, J. Schneider, and S. M. Weinberg, "Selling to a no-regret buyer," *CoRR*, vol. abs/1711.09176, 2017.

[74] Y. Deng, J. Schneider, and B. Sivan, "Strategizing against no-regret learners," *CoRR*, vol. abs/1909.13861, 2019.

[75] A. Roth, J. R. Ullman, and Z. S. Wu, "Watch and learn: Optimizing from revealed preferences feedback," *CoRR*, vol. abs/1504.01033, 2015. arXiv: 1504.01033.

[76] J. Dong, A. Roth, Z. Schutzman, B. Waggoner, and Z. S. Wu, "Strategic classification from revealed preferences," *CoRR*, vol. abs/1710.07887, 2017. arXiv: 1710.07887.

[77] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*, 1st. USA: Cambridge University Press, 2009, ISBN: 0521424267.

[78] A. Agarwal, S. Agarwal, S. Assadi, and S. Khanna, "Learning with limited rounds of adaptivity: Coin tossing, multi-armed bandits, and ranking from pairwise comparisons," in *Proceedings of the 2017 Conference on Learning Theory*, S. Kale and O. Shamir, Eds., ser. Proceedings of Machine Learning Research, vol. 65, PMLR, 2017, pp. 39–75.

[79] A. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," *CoRR*, vol. cs.LG/0408007, 2004.

[80] M. Hardt, N. Megiddo, C. H. Papadimitriou, and M. Wootters, "Strategic classification," *CoRR*, vol. abs/1506.06980, 2015. arXiv: 1506.06980.

[81] J. Dong, A. Roth, Z. Schutzman, B. Waggoner, and Z. S. Wu, "Strategic classification from revealed preferences," in *Proceedings of the 2018 ACM Conference on Economics and Computation*, ser. EC '18, Ithaca, NY, USA: Association for Computing Machinery, 2018, 55–70, ISBN: 9781450358293.

[82] M. Jagadeesan, T. Zrnic, and C. Mendler-Dünner, "Regret minimization with performative feedback," *CoRR*, vol. abs/2202.00628, 2022. arXiv: 2202.00628.

[83] C. Daskalakis and V. Syrgkanis, "Learning in auctions: Regret is hard, envy is easy," *CoRR*, vol. abs/1511.01411, 2015. arXiv: 1511.01411.

[84] T. Nedelec, C. Calauzenes, V. Perchet, and N. E. Karoui, "Robust stackelberg buyers in repeated auctions," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, S. Chiappa and R. Calandra, Eds., ser. Proceedings of Machine Learning Research, vol. 108, PMLR, 2020, pp. 1342–1351.

[85] A. Blum, N. Haghtalab, and A. D. Procaccia, "Learning optimal commitment to overcome insecurity," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27, Curran Associates, Inc., 2014.

[86] G. Alcantara-Jiménez and J. B. Clempner, "Repeated stackelberg security games: Learning with incomplete state information," *Reliability Engineering System Safety*, vol. 195, p. 106 695, 2020.

[87] B. Peng, W. Shen, P. Tang, and S. Zuo, "Learning optimal strategies to commit to," in *AAAI Conference on Artificial Intelligence*, 2019.

[88] N. Lauffer, M. Ghasemi, A. Hashemi, Y. Savas, and U. Topcu, *No-regret learning in dynamic stackelberg games*, 2022. arXiv: 2202.04786 [cs.GT].

[89] N. Collina, E. R. Arunachaleswaran, and M. Kearns, "Efficient stackelberg strategies for finitely repeated games," in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '23, London, United Kingdom: International Foundation for Autonomous Agents and Multiagent Systems, 2023, 643–651, ISBN: 9781450394321.

[90] T. Zrnic, E. Mazumdar, S. S. Sastry, and M. I. Jordan, "Who leads and who follows in strategic classification?" *CoRR*, vol. abs/2106.12529, 2021. arXiv: 2106.12529.

[91] G. Brown, S. Hod, and I. Kalemaj, *Performative prediction in a stateful world*, 2022. arXiv: 2011.03885 [cs.LG].

[92] Y. Deng, J. Schneider, and B. Sivan, *Strategizing against no-regret learners*, 2019. arXiv: 1909.13861 [cs.GT].

[93] R. Kumar, S. Dean, and R. D. Kleinberg, *Online convex optimization with unbounded memory*, 2022. arXiv: 2210.09903 [cs.LG].

[94] I. Anagnostides, I. Panageas, G. Farina, and T. Sandholm, *On the convergence of no-regret learning dynamics in time-varying games*, 2023. arXiv: 2301.11241 [cs.LG].

[95] N. Agarwal, B. Bullins, E. Hazan, S. M. Kakade, and K. Singh, *Online control with adversarial disturbances*, 2019. arXiv: 1902.08721 [cs.LG].

[96] M. Simchowitz, K. Singh, and E. Hazan, "Improper learning for non-stochastic control," *CoRR*, vol. abs/2001.09254, 2020. arXiv: 2001.09254.

[97] A. Cassel, A. Cohen, and T. Koren, *Efficient online linear control with stochastic convex costs and unknown dynamics*, 2022. arXiv: 2203.01170 [math.OC].

[98] E. Minasyan, P. Gradu, M. Simchowitz, and E. Hazan, *Online control of unknown time-varying dynamical systems*, 2022. arXiv: 2202.07890 [cs.LG].

[99] M. Aoki, "Local Controllability of a Decentralized Economic System1," *The Review of Economic Studies*, vol. 41, no. 1, pp. 51–63, Jan. 1974. eprint: https://academic.oup.com/restud/article-pdf/41/1/51/4402584/41-1-51.pdf.

[100] H. Kuhn and H.-W. Wohltmann, "Controllability of economic systems under alternative expectations hypotheses—the discrete case," *Computers Mathematics with Applications*, vol. 18, no. 6, pp. 617–628, 1989.

[101] M. Barbero-Liñán and B. Jakubczyk, *Second order conditions for optimality and local controllability of discrete-time systems*, 2013. arXiv: 1211.5784 [math.OC].

[102] U. Boscain, D. Cannarsa, V. Franceschi, and M. Sigalotti, *Local controllability does imply global controllability*, 2021. arXiv: 2110.06631 [math.OC].

[103] N. Agarwal, E. Hazan, and K. Singh, "Logarithmic regret for online control," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019.

[104] A. Cohen, A. Hassidim, T. Koren, N. Lazic, Y. Mansour, and K. Talwar, "Online linear quadratic control," *CoRR*, vol. abs/1806.07104, 2018. arXiv: 1806.07104.

[105] O. Anava, E. Hazan, and S. Mannor, *Online convex optimization against adversaries with memory and application to statistical arbitrage*, 2014. arXiv: 1302.6937 [cs.LG].

[106] X. Chen, E. Minasyan, J. D. Lee, and E. Hazan, *Provable regret bounds for deep online learning and control*, 2022. arXiv: 2110.07807 [cs.LG].

[107] S. Kakade, A. Krishnamurthy, K. Lowrey, M. Ohnishi, and W. Sun, *Information theoretic regret bounds for online nonlinear control*, 2020. arXiv: 2006.12466 [cs.LG].

[108] S. Lale, K. Azizzadenesheli, B. Hassibi, and A. Anandkumar, "Model learning predictive control in nonlinear dynamical systems," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 757–762.

[109] W. Luo, W. Sun, and A. Kapoor, *Sample-efficient safe learning for online nonlinear control with control barrier functions*, 2022. arXiv: 2207.14419 [cs.RO].

[110] P. Gradu, E. Hazan, and E. Minasyan, *Adaptive regret for control of time-varying dynamics*, 2022. arXiv: 2007.04393 [cs.LG].

[111] D. Muthirayan and P. P. Khargonekar, *Online learning robust control of nonlinear dynamical systems*, 2022. arXiv: 2106.04092 [eess.SY].

[112] T. Lykouris, M. Simchowitz, A. Slivkins, and W. Sun, "Corruption-robust exploration in episodic reinforcement learning," in *Proceedings of Thirty Fourth Conference on Learning Theory*, M. Belkin and S. Kpotufe, Eds., ser. Proceedings of Machine Learning Research, vol. 134, PMLR, 2021, pp. 3242–3245.

[113] X. Zhang, Y. Chen, J. Zhu, and W. Sun, "Corruption-robust offline reinforcement learning," *CoRR*, vol. abs/2106.06630, 2021. arXiv: 2106.06630.

[114] Y. Mansour, M. Mohri, J. Schneider, and B. Sivan, *Strategizing against learners in bayesian games*, 2022. arXiv: 2205.08562 [cs.LG].

[115] B. H. Zhang *et al.*, *Steering no-regret learners to optimal equilibria*, 2023. arXiv: 2306.05221 [cs.GT].

[116] J. D. Hartline, V. Syrgkanis, and É. Tardos, "No-regret learning in repeated bayesian games," *CoRR*, vol. abs/1507.00418, 2015. arXiv: 1507.00418.

[117] T. Zrnic, E. Mazumdar, S. Sastry, and M. Jordan, "Who leads and who follows in strategic classification?" In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, Curran Associates, Inc., 2021, pp. 15 257–15 269.

[118] G. Neu and J. Olkhovskaya, "Online learning in mdps with linear function approximation and bandit feedback.," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, Curran Associates, Inc., 2021, pp. 10 407–10 417.

[119] L. Shen, N. Ho-Nguyen, and F. Kılınç-Karzan, "An online convex optimization-based framework for convex bilevel optimization," *Mathematical Programming*, vol. 198, no. 2, pp. 1519–1582, Apr. 2023.

[120] L. Flokas, E.-V. Vlatakis-Gkaragkounis, and G. Piliouras, *Poincaré recurrence, cycles and spurious equilibria in gradient-descent-ascent for non-convex non-concave zero-sum games*, 2019. arXiv: 1910.13010 [math.OC].

[121] A. Agarwal and W. Brown, "Diversified recommendations for agents with adaptive preferences," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.

[122] S. Shalev-Shwartz and Y. Singer, "Online learning meets optimization in the dual," in *Proceedings of the 19th Annual Conference on Learning Theory*, ser. COLT'06, Pittsburgh, PA: Springer-Verlag, 2006, 423–437, ISBN: 3540352945.

[123] J. D. Abernethy, E. Hazan, and A. Rakhlin, "Competing in the dark: An efficient algorithm for bandit linear optimization," in *Annual Conference Computational Learning Theory*, 2008.

[124] M. Gasca and T. Sauer, "Polynomial interpolation in several variables," *Advances in Computational Mathematics*, vol. 12, no. 4, pp. 377–410, 2000.

[125] O. D. Kellogg, "On bounded polynomials in several variables," *Mathematische Zeitschrift*, vol. 27, no. 1, pp. 55–64, 1928.

[126] E. Price and Z. Song, "A robust sparse fourier transform in the continuous setting," in *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, V. Guruswami, Ed., IEEE Computer Society, 2015, pp. 583–600.

[127] E. Hazan, *Introduction to online convex optimization*, 2021. arXiv: 1909.05207 [cs.LG].

[128] S. Ahmadi, A. Blum, and K. Yang, *Fundamental bounds on online strategic classification*, 2023. arXiv: 2302.12355 [cs.LG].

[129] J. Miller, J. C. Perdomo, and T. Zrnic, "Outside the echo chamber: Optimizing the performative risk," *CoRR*, vol. abs/2102.08570, 2021. arXiv: 2102.08570.

[130] G. Piliouras and F.-Y. Yu, *Multi-agent performative prediction: From global stability and optimality to chaos*, 2022. arXiv: 2201.10483 [cs.LG].

[131] S. Dean and J. Morgenstern, *Preference dynamics under personalized recommendations*, 2022. arXiv: 2205.13026 [cs.LG].

[132] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani, "Adwords and generalized online matching," *J. ACM*, vol. 54, no. 5, 22–es, 2007.

[133] Y. Kanoria and H. Nazerzadeh, "Dynamic reserve prices for repeated auctions: Learning from bids," *CoRR*, vol. abs/2002.07331, 2020. arXiv: 2002.07331.

[134] N. Golrezaei, A. Javanmard, and V. S. Mirrokni, "Dynamic incentive-aware learning: Robust pricing in contextual auctions," *CoRR*, vol. abs/2002.11137, 2020. arXiv: 2002.11137.

[135] J. Morgenstern and T. Roughgarden, "Learning simple auctions," *CoRR*, vol. abs/1604.03171, 2016. arXiv: 1604.03171.

[136] Z. Feng, O. Schrijvers, and E. Sodomka, "Online learning for measuring incentive compatibility in ad auctions," *CoRR*, vol. abs/1901.06808, 2019. arXiv: 1901.06808.

[137] L. Jia, L. Tong, and Q. Zhao, *An online learning approach to dynamic pricing for demand response*, 2014. arXiv: 1404.1325 [math.OC].

[138] S. Agrawal, Y. Feng, and W. Tang, *Dynamic pricing and learning with bayesian persuasion*, 2023. arXiv: 2304.14385 [cs.GT].

[139] M. Mussi, G. Genalti, A. Nuara, F. Trovò, M. Restelli, and N. Gatti, *Dynamic pricing with volume discounts in online settings*, 2022. arXiv: 2211.09612 [cs.LG].

[140] T. Roughgarden, "Intrinsic robustness of the price of anarchy," *J. ACM*, vol. 62, no. 5, 2015.

[141] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," vol. 2, Apr. 2003.

Appendix A: Deferred Proofs from Chapter 2

A.1 Proof of Theorem 1

Proof. Let φ be the joint distribution over action pairs corresponding to Ψ. Let T denote the total number of steps we run the algorithm for; we will use $t \leq T$ as a changing step size. Suppose both player A and player B know φ[1]. We will define $\mathcal{L}_A^*(\Psi)$ and $\mathcal{L}_B^*(\Psi)$ in two phases: in the first phase, A and B trust their opponent and play according to deterministic sequences corresponding to approximations of φ. If either player violates the other's trust $o(T)$ times, then the player defects to playing $\mathcal{L}_A$ or $\mathcal{L}_B$ respectively forever after.

First we elaborate upon the trusting phase. Both players consider windows of length $\mathsf{Length}(t)$ which is monotonically increasing in t and also which grows sub-linearly in t. For concreteness, we pick a sub-linear monotonic increasing growth rate of $O(\sqrt{t})$ and describe how to implement the schedule of window lengths. We can keep track of a real-valued variable Z_t with $Z_1 = M \cdot N$, and after each window completes, update it by $Z_{t_{\text{next}}} = Z_t + \frac{1}{2\sqrt{t}}$ where t is the step at the end of the window. To get an integral window length, we define $\mathsf{Length}(t) := \lfloor Z_t \rfloor$. Thus in this case, the $\mathsf{Length}(t)$ grows as $O(\sqrt{t})$, satisfying both conditions. Both players then compute a weighting instantiated with pairs of pure strategies by assigning $c_i := \lfloor \mathsf{Length}(t) \cdot \varphi_i \rfloor$ example pairs (each of weight $1/\mathsf{Length}(t)$) to pure strategy pair $i \in [M \cdot N]$. This weighted distribution approximates φ given $\mathsf{Length}(t)$ samples. Note that the rounding approximation is feasible given only $\mathsf{Length}(t)$ samples since $\sum_{i=1}^{M \cdot N} c_i \leq \mathsf{Length}(t)$. These pure strategy pair samples are then lexicographically ordered. Then, both players act according to the pure strategies in order, thereby (over the window)

[1] φ can be communicated from Player A to Player B during a burn-in phase of length $> M \cdot N$, the dimension of the discrete joint distribution over pure player strategy pairs.

achieving an $(M \cdot N)/\mathsf{Length}(t)$ ℓ_1 approximation to φ:

$$\sum_{i=1}^{M\cdot N}\left|\varphi_i - \frac{c_i}{\mathsf{Length}(t)}\right| = \sum_{i=1}^{M\cdot N}\left|\varphi_i - \frac{\lfloor \mathsf{Length}(t)\cdot\varphi_i\rfloor}{\mathsf{Length}(t)}\right| \le \frac{M\cdot N}{\mathsf{Length}(t)}.$$

This process repeats for every window.

The distrustful phase occurs if one of the players does not follow the agreed-upon instructions T_{distrust} times, where T_{distrust} is taken to be $o(T)$. After this many violations, Player A defaults to playing L_A and likewise Player B defaults to playing L_B ever after.

We now show that this algorithm satisfies both conditions in the theorem statement. First, if both players use $\mathcal{L}_A^*(\Psi)$ and $\mathcal{L}_B^*(\Psi)$, the play converges to φ, the joint distribution of play corresponding to Ψ. This point is immediate to observe since $(M \cdot N)/\mathsf{Length}(t) \to 0$ as $t \to \infty$ as $\mathsf{Length}(t)$ is monotone increasing in t.

Now we prove that both players are no-Φ-regret with respect to any adversary. First we show no-Φ-regret for both players in the case where Player A plays $\mathcal{L}_A^*(\Psi)$ and Player B plays $\mathcal{L}_B^*(\Psi)$. Let $\hat{\varphi}_t$ be the approximation to φ implemented over the window corresponding to final step t, and suppose that $\left\|\varphi - \hat{\varphi}_t\right\|_1 < \varepsilon_t$. Recalling the proof of Theorem 1, for Player A (and analogously for Player B) we can bound

$$\begin{aligned}\left|\mathop{\mathbb{E}}_{(a,b)\sim\varphi}\left[u_A\left(a,b\right)\right] - \mathop{\mathbb{E}}_{(a,b)\sim\hat{\varphi}_t}\left[u_A\left(a,b\right)\right]\right| &= \left|\left(\varphi - \hat{\varphi}_t\right)^\top u_A\right| \\ &\le \|\,\varphi - \hat{\varphi}_t\|_1 \cdot \|u_A\|_2 \le \varepsilon_t \cdot C \cdot \sqrt{M\cdot N},\end{aligned}$$

where here we interpret $\varphi, \hat{\varphi}_t, u_A, u_B \in \mathbb{R}^{M\times N}$ as vectors over the space of all action pairs. Thus for this particular window, the overall gap from the expected reward for φ is $\varepsilon_t \cdot C \cdot \sqrt{M\cdot N}$.

Then we can similarly upper bound $\mathbb{E}_{(a,b)\sim\hat{\varphi}_t}\left[u_A\left(f_A(a),b\right)\right] \le \mathbb{E}_{(a,b)\sim\varphi}\left[u_A\left(f_A(a),b\right)\right] + \varepsilon_t \cdot$

$C \cdot M\sqrt{N}$ for any choice of $f_A \in \Phi_A$:

$$\begin{aligned}
(*) &= \left| \mathop{\mathbb{E}}_{(a,b)\sim\varphi} \left[u_A\left(f_A(a), b\right)\right] - \mathop{\mathbb{E}}_{(a,b)\sim\hat{\varphi}_t} \left[u_A\left(f_A(a), b\right)\right] \right| \\
&= \left| \sum_{k=1}^{M} \sum_{j=1}^{N} \left(\hat{\varphi}_t(k,j) - \varphi(k,j)\right) \cdot \sum_{i=1}^{M} f_A(a_k)_i \cdot u(\cdot, b_j) \right| \\
&\leq \| \varphi - \hat{\varphi}_t \|_1 \cdot \| \left[f_A(a_1)^\top u_A(\cdot, b_1), \cdots, f_A(a_M)^\top u_A(\cdot, b_N) \right] \|_2 \\
&\leq \varepsilon_t \cdot \sqrt{M \cdot N} \cdot \max_{k,j} \|f_A(a_k)\|_2 \cdot \|u_A(\cdot, b_j)\|_2 \\
&\leq \varepsilon_t \cdot \sqrt{M \cdot N} \cdot 1 \cdot \sqrt{M \cdot C^2} \\
&= \varepsilon_t \cdot M \cdot \sqrt{N} \cdot C.
\end{aligned}$$

Then recall that $\varepsilon_t \leq \frac{M \cdot N}{\mathsf{Length}(t)}$. Thus, overall, the average regret using due to the window is bounded by

$$\frac{1}{\mathsf{Length}(t)} \mathsf{Reg}_\Phi(\hat{\varphi}_t, t) \leq \frac{1}{\mathsf{Length}(t)} \mathsf{Reg}_\Phi(\varphi, t) + C_2 \cdot \frac{1}{\mathsf{Length}(t)},$$

where C_2 is another constant depending on C, M, N and where we use the shorthand $\mathsf{Reg}_\Phi(\cdot, t)$ to denote the Φ-regret over the window ending in step t. Now call $\hat{\varphi}$ the strategy where the joint distribution $\hat{\varphi}_t$ as previously defined gets played in each window t. Now we can bound the total Φ-regret for $\hat{\varphi}$ by the sum of the Φ-regrets for each window (maximizing $f_A \in \Phi_A$ over the steps in each window makes it more competitive than optimizing only one f_A over the whole length T sequence). Thus for total Φ-regret, we have:

$$\mathsf{Reg}_\Phi(\hat{\varphi}, T) \leq \quad \mathsf{Reg}_\Phi(\varphi, T) + \mathsf{NumWindows}(T) \cdot C_2 \quad \leq \quad \mathsf{Reg}_\Phi(\varphi, T) + o\left(T\right),$$

where

$$\mathsf{NumWindows}(T) := \min_{\sum_{t=1}^{k} \mathsf{Length}(t) \geq T} k.$$

The last step follows since $\mathsf{NumWindows}(T) \leq o(T)$, because $\mathsf{Length}(T) \leq o(T)$.

Since we already know that the strategy φ is no-Φ-regret and $\mathsf{Length}(T)$ is $o(T)$, we have proven that playing $\hat{\varphi}$ is no-Φ-regret in the case where Player A plays $\mathcal{L}_A^*(\Psi)$ and Player B plays $\mathcal{L}_B^*(\Psi)$.

The second case where the opposing player does not cooperate is easier: after at most $o(T)$ steps, the player switches to an algorithm $\mathcal{L}_A$ or $\mathcal{L}_B$ respectively which is no-Φ-regret and incurrs only $o(T)$ additional regret. Thus the theorem statement holds. □

A.2 Proof of Theorem 10

Proof. Our game consists of M actions $\mathcal{A}$ for the optimizer, and $N = 2M + \binom{M}{2}$ actions for the learner, which are divided into M *primary* actions $\mathcal{B}$, M *secondary* actions $\mathcal{S}$, and $\binom{M}{2}$ *safety* actions $\mathcal{Y}$.

If we restrict the learner to only playing primary actions, the game somewhat resembles a coordination game, where each pure strategy pair (a_j, b_j) is a Nash equilibrium. However, the set $\mathcal{B}$ is comprised of both *undominated* actions $\mathcal{B}_U$ and *dominated* actions $\mathcal{B}_D$, which are unknown to the optimizer, and where each $b_j \in \mathcal{B}_d$ is weakly dominated by the secondary action s_j. The optimizer receives reward 0 whenever the learner plays a secondary action, and so the challenge for the optimizer is to identify the pair (a_j, b_j) which maximizes $u_A(a_j, b_j)$, for $b_j \in \mathcal{B}_D$, which will be the Stackelberg equilibrium. Further, the safety actions y_{ij} essentially allow the learner to hedge between two actions; this does not pose substantial difficulty for the optimizer when the learner is no-swap-regret, yet creates an insurmountable barrier for learning the Stackelberg equilibrium in sub-exponential time against a mean-based learner.

An instance of a game $G \in \mathcal{G}$ is specified by the partition of $\mathcal{B}$ into $\mathcal{B}_U$ and $\mathcal{B}_D$. There is an action $s_j \in \mathcal{S}$ for each j, and for each pair (i, j) with $i < j$ there is an action $y_{ij} \in \mathcal{Y}$. The rewards for a game G are as follows. For any strategy pair, the optimizer's utility is given by:

- $u_A(a_j, b_j) = j/M$ for $b_j \in \mathcal{B}$;

- $u_A(a_i, b_j) = 0$ for $b_j \in \mathcal{B}$ and with $i \neq j$;
- $u_A(a_i, s_j) = 0$ for any $s_j \in \mathcal{S}$;
- $u_A(a_i, y_{jk}) = 0$ for any $y_{jk} \in \mathcal{Y}$;

and the learner's utility is given by:

- For $b_j \in \mathcal{B}_U$:
  - $u_B(a_j, b_j) = 1$;
  - $u_B(a_i, b_j) = 0$ for $i \neq j$;
- For $b_j \in \mathcal{B}_D$:
  - $u_B(a_i, b_j) = 0$ for any i;
- For $s_j \in \mathcal{S}$:
  - $u_B(a_j, s_j) = 1$ if $b_j \in \mathcal{B}_D$;
  - $u_B(a_j, s_j) = 0$ if $b_j \in \mathcal{B}_U$;
  - $u_B(a_i, s_j) = 0$ for $i \neq j$;
- For $y_{ij} \in \mathcal{Y}$:
  - $u_B(a_i, y_{ij}) = u_B(a_j, y_{ij}) = 2/3$;
  - $u_B(a_k, y_{ij}) = 0$ for $i, j \neq k$.

We assume that $\mathcal{B}_U$ is non-empty, and so there is some optimal pure Nash equilibrium (a_i^*, b_i^*) which yields a reward of i/M; it is simple to check that this is also the Stackelberg equilibrium.

Optimizing against no-swap learners. First, we give a method for matching the Stackelberg value against an arbitrary no-swap-regret learner, which corresponds to the pair (a_j, b_j) for the largest value j such that $b_j \in \mathcal{B}_U$. Consider a no-swap-regret learner which obtains a regret bound of $\tau = O(T^c)$ over T rounds. Let $\mathsf{SR}_t(b, b')$ for any learner actions b and b' denote the t-round cumulative swap regret between b and b', i.e. the total change in reward which would have occurred if b' was played instead for each of the first t rounds in which b was played. To model the behavior of an arbitrary no-swap-regret learner, we disallow the learner from taking any action which would increase $\mathsf{SR}_t(b, b')$ above τ, given the loss function for the current round, and otherwise allow the action to be chosen adversarially. While our model is deterministic for simplicity, it is straightforward to extend to the analysis to algorithms whose regret bounds hold in only expectation, e.g. by considering a distribution over values of τ in accordance with Markov's inequality (as no algorithm can have negative expected regret against arbitrary adversaries) and considering our expected regret to the Stackelberg value.

Our strategy for the optimizer is:

- For each $i \in [M]$, play a_i until either b_i or s_i is observed at least $t^* > \tau$ times;
- Return a_i^* for the largest i such that b_i is observed t^* times.

We show that this takes at most $O(T^c \cdot M^3)$ rounds. Once a_i^* is identified, we can commit to playing it indefinitely, at which point the learner must play b_i^* in all but at most $O(T^c \cdot \text{poly}(M))$ rounds, and so with $T = O(\text{poly}(M/\varepsilon))$ rounds we can increase the total fraction of rounds in which (a_i^*, b_i^*) is played to $1 - \varepsilon$, which yields the desired average reward bound.

The key to analyzing the runtime of our strategy is to consider the "buffer" in regret between any pair of actions before the threshold of τ is reached, which enables us to the bound the number of rounds in which instantaneously suboptimal actions are played. Note that prior the start of window i (where a_i is played), both b_i and s_i obtain reward 0 in each round, and as such cannot decrease their expected regret relative to any other action, as all rewards in the game are non-negative. Further, for any previous window j, both b_i and s_i incur regret of 1 with respect to either b_j or s_j, as well

as between the suboptimal and optimal action in window i, and thus cannot be observed more than τ times in the window. As such, observing b_i at least t^* times in window i indicates that $b_i \in \mathcal{B}_U$ (and likewise observing b_i at least t^* times indicates that $b_i \in \mathcal{B}_D$).

Any action $b \neq \mathsf{BR}(a_i)$ will incur positive swap regret with respect to $\mathsf{BR}(a_i)$, and cannot be played in window i once $\mathsf{SR}_t(b, \mathsf{BR}(a_i)) \geq \tau$. Each action begins with $\mathsf{SR}_1(b, \mathsf{BR}(a_i)) = 0$ at time $t = 1$; for each of the learner's actions, we consider the rate at which its buffer decays, as well as instances in which swap regret can decrease:

- Previously optimal $b \in \mathcal{B} \cup \mathcal{S} \setminus \mathsf{BR}(a_i)$: actions in $\mathcal{B} \cup \mathcal{S}$ can only accumulate negative swap regret with respect to $\mathsf{BR}(a_i)$ during rounds in which they were previously optimal; any previous optimum $b = \mathsf{BR}(a_j)$ for $j < i$ was played at most t^* times during window j, and so we have that $\mathsf{SR}_t(b, \mathsf{BR}(a_i)) \geq -t^*$.

- All $b \in \mathcal{B} \cup \mathcal{S} \setminus \mathsf{BR}(a_i)$: ignoring any previously accumulated regret buffer, each of these $2M - 1$ actions can be played at most τ rounds during window i before exhausting their initial buffer. Accounting for possible previous optima with $\mathsf{SR}_t(b, \mathsf{BR}(a_i)) < 0$, the number of rounds during window i in which some $b \in \mathcal{B} \cup \mathcal{S} \setminus \mathsf{BR}(a_i)$ is played is at most $Mt^* + (2M - 1)\tau$.

- Safety actions $y_{jk} \in \mathcal{Y}$: Suppose neither a_j or a_k have been played yet by the optimizer, including in the current window. As was the case for other actions which have never yielded positive instantaneous reward, y_{jk} can be played at most τ times before $\mathsf{SR}_t(y_{jk}, \mathsf{BR}(a_i)) \geq \tau$. If $j = i$, i.e. this is the first window in which y_{jk} obtains positive instantaneous reward, the per-round regret is $1/3$, and so at it can be played for most 3τ rounds. Further, y_{jk} a obtains a regret of $-2/3$ with respect to $\mathsf{BR}(a_k)$. If $k = i$ and the window for a_j has already been completed, y_{jk} can be played for at most 9τ rounds, as initially we have that $\mathsf{SR}_t(y_{jk}, \mathsf{BR}(a_i)) \geq -2\tau$, which again increases by $1/3$ per round. We then have that the total amount of rounds with safety actions played during window i is at most $(12M + M^2)\tau$, as there are fewer than M^2 total safety actions, and fewer than M in each of the latter cases.

This yields a per-window runtime across all actions of at most $Mt^* + (M^2 + 10M - 1)\tau$, which is $O(T^c \cdot M^3)$ across all windows, and so we obtain the desired result for optimizing against arbitrary no-swap-regret learners.

Optimizing against mean-based learners. Here, we show that there are mean-based no-regret algorithms for which exponentially many rounds are required for an optimizer to approximate the Stackelberg value against a learner. When considering horizons which are superpolynomial in the parameters of the game, it is most natural to consider algorithms with regret bounds which are non-trivial for smaller horizons, as well as an anytime variant of the mean-based property. We define an extension of the classical Multiplicative Weight Updates algorithm (MWU; see [31] for a survey), called Rounded Mean-Based Doubling, which inherits both properties in the anytime setting. We recall the algorithm statement and guarantees for RMBD.

Algorithm 10 Rounded Mean-Based Doubling (RMBD

Initialize and run MWU for $T_1 := 2$ rounds and n actions.
Let $T_2 := 2T_1$ and $i := 2$.
while $T_i \leq T$ **do**
  Initialize MWU for T_i rounds and n actions.
  Simulate running MWU for T_{i-1} rounds, using the average of the first T_{i-1} rewards each round.

  For T_{i-1} rounds, run MWU with action probabilities rounded to multiples of $4\gamma = \tilde{O}(T_i^{-1/2})$.
  Let $T_{i+1} = 2T_i$ and $i := i + 1$.
end while

Lemma 10. *When running* RMBD *for T rounds, the following hold at any round $t \leq T$:*

- RMBD *has cumulative regret $\tilde{O}(n\sqrt{t})$;*
- *If action j has the highest cumulative reward and $\sigma_{i,t} \leq \sigma_{j,t} - \tilde{O}(\sqrt{t})$, then action i is played with probability 0 at round t.*

Suppose a learner plays the action with highest cumulative reward at each round for $t_{\text{burn}} = \tilde{\Omega}(M^2)$ rounds, then plays RMBD thereafter for a total of T rounds. Note that this maintains the both properties of RMBD for all t. We show that at least $T = \exp(\Omega(M))$ rounds are required to

identify the Stackelberg strategy. The optimizer must check the learner's pure best response to each a_j for identification with certainty, and it is straightforward to construct a distribution in which any strategy which does not observe $\mathsf{BR}(a_j)$ for all j will have linear regret to Stack_A in expectation (e.g. where $\mathcal{B}_U$ contains one action chosen uniformly at random). The difficulty in exploration of the best responses comes from the safety actions, as a_j must have been played more frequently than any other action in order to not be dominated by some safety action. Let $\rho_{j,t}$ denote the number of rounds in which the optimizer has played a_j out of the first t. Observe that by construction of the game and the properties of RMBD, an primary or secondary action b_j or s_j in $\mathsf{BR}(a_j)$ will only be played with positive probability when:

$$\begin{aligned}\rho_{j,t} &\geq \frac{2}{3}(\rho_{j,t} + \rho_{k,t}) - \tilde{O}(\sqrt{t}) \\ &= 2\rho_{k,t} - \tilde{O}(\sqrt{t})\end{aligned}$$

for all k, which necessitates that $\rho_{j,t} \geq \frac{2t}{M} - \tilde{O}(\sqrt{t})$. Taking t_{burn} sufficiently large, we have that $\rho_{j,t} \geq \frac{3}{2}\rho_{k,t}$ for any $t \geq t_{\text{burn}}$ and all k. For any subsequent observation $\mathsf{BR}(a_k)$ at t', we must have that $\rho_{k,t'} \geq \frac{3}{2}\rho_{j,t}$, and so the number of rounds required to play an action before observing its best response grow at a rate of at least $(3/2)^M$, which completes the proof.

□

Appendix B: Deferred Proofs from Chapter 3

B.1 Analysis for Deferred Bandit Gradient

Proof. Equipped with the previous result, we can now prove the regret bound for Theorem 16. Let $r_t^* = \sum_{s=\max(t-H+1,1)}^{t} \frac{r_s \otimes (y_s \oslash x_s)}{H}$, where $\otimes$ and $\oslash$ denote elementwise multiplication and division, respectively, and let $\hat{r}_t = \sum_{s=\max(t-H+1,1)}^{t} \frac{r_s}{H}$. Further, let $x_\epsilon^* = \Pi_{\mathcal{K}_T,\epsilon}[x^*]$. Observe that the following hold for every t:

$$\frac{r_{t,i} \cdot y_{t,i}}{x_{t,i}} = r_{t,i}\left(1 + \frac{\xi_{t,i}}{x_{t,i}}\right);$$

$$\mathbb{E}[r_t^* - \hat{r}_t] = \frac{1}{H}\sum_{s=t-H+1}^{t}((r_s \otimes \xi_s) \oslash x_s); \tag{B.1}$$

$$\mathbb{E}\left[\widetilde{\nabla}_t\right] = r_t^*. \tag{B.2}$$

Observe that by the constraints on each $\mathcal{K}_{t,\epsilon}$, each x_t can be expressed as $(1-\epsilon)x + \epsilon \mathbf{u}_n$ for some $x \in \mathcal{K}\, t \subseteq \Delta(n)$, and so we will always have $x_{t,i} \geq \frac{\epsilon}{n}$. To bound the squared norms of $\widetilde{\nabla}_t$ in order apply Lemma 4, consider the maximizing case where $x_t = \frac{\epsilon}{n}$ and $y_t = \frac{2\epsilon}{n}$ in all but one element, and where $r_{t,i} = 1$ for all rewards; $\mathbb{E}\left[\left\|\widetilde{\nabla}_t\right\|^2\right]$ is increasing whenever probability mass in x_t is transferred from an arm $x_{t,i}$ to $x_{t,j} > x_{t,i}$, and thus we can obtain a bound in terms the expectation of a squared binomial random variable X with $H = \frac{n}{\epsilon}$ trials, where each trial has value at most 1 with probability $\frac{2\epsilon(n-1)}{n}$ (if any of the $n-1$ are sampled), and value $\frac{1}{H}$ otherwise. This yields:

$$\begin{aligned}\mathbb{E}\left[\left\|\widetilde{\nabla}_t\right\|^2\right] &\leq H\left(\frac{2\epsilon(n-1)}{n}\right)\left(1 - \frac{2\epsilon(n-1)}{n}\right) + \left(\frac{2(H-1)\epsilon(n-1)}{n} + 1\right)^2 \\ &\leq 2(n-1) + (2n-1)^2 \\ &\leq 4n^2. \end{aligned} \tag{B.3}$$

Over all T, for any fixed $x^* \in \mathcal{K}_T$ we have:

$$\sum_{t=1}^{T} r_t^\top x^* - \hat{r}_t^\top x^* \leq \frac{H}{2} = \frac{n}{2\epsilon}, \tag{B.4}$$

as only fractional rewards from the last H rounds are omitted from being counted appropriately in $\sum_t \hat{r}_t$. We now analyze the regret of our algorithm with respect to the sequence $\{\hat{r}_t\}$. For $x^* \in \mathcal{K}_T$ we have:

$$\begin{aligned}
\sum_{t=1}^{T} \hat{r}_t^\top x^* - \sum_{t=1}^{T} \mathbb{E}\left[\hat{r}_t^\top y_t\right] &\leq \sum_{t=1}^{T} \hat{r}_t^\top x_\epsilon^* - \sum_{t=1}^{T} \mathbb{E}\left[\hat{r}_t^\top x_t\right] + \sqrt{n}\epsilon T && \text{(each } r_t \text{ is } \tfrac{\sqrt{n}}{2}\text{-Lipschitz)} \\
&\leq \sum_{t=1}^{T} r_t^{*\top} x_\epsilon^* - \sum_{t=1}^{T} r_t^{*\top} x_t + \sqrt{n}\epsilon T + \sum_{t=1}^{T}\sum_{i=1}^{n} \frac{\left|\xi_{t,i}\right|}{x_{t,i}} && \text{(by (B.1))} \\
&\leq \mathbb{E}\left[\mathsf{Reg}_{COGD}(\widetilde{\nabla}_1, \ldots, \widetilde{\nabla}_T)\right] + \sqrt{n}\epsilon T + \sum_{t=1}^{T}\sum_{i=1}^{n} \frac{\left|\xi_{t,i}\right|}{x_{t,i}} && \text{(B.5)} \\
&\leq \frac{\eta}{2}\sum_{t=1}^{T} \mathbb{E}\left[\left\|\widetilde{\nabla}_t\right\|^2\right] + \frac{\sqrt{2}}{\eta} + \sqrt{n}\epsilon T + \sum_{t=1}^{T}\sum_{i=1}^{n} \frac{\left|\xi_{t,i}\right|}{x_{t,i}}. && \text{(by Lemma 4)}
\end{aligned}$$

Line (B.5) holds by observing that our algorithm is equivalent to running Contracting Online Gradient Descent to the sequence $\{\widetilde{\nabla}_t\}$, where $\mathbb{E}\left[\sum_{t=1} \widetilde{\nabla}_t\right] = \sum_{t=1} r_t^*$ by (B.2). As such, by (B.3) and (B.4) we have that

$$\sum_{t=1}^{T} r_t^\top x^* - \sum_{t=1}^{T} \mathbb{E}\left[r_t^\top y_t\right] \leq 2\eta n^2 T + \frac{\sqrt{2}}{\eta} + 3\sqrt{n}\epsilon T + \frac{n}{2\epsilon} + \sum_{t=1}^{T}\sum_{i=1}^{n} \frac{\left|\xi_{t,i}\right|}{x_{t,i}}.$$

□

B.2 Proof of Lemma 2

Proof. Let the menu time μ_i for each item be given by

$$\mu_i := \frac{k \cdot \frac{x_i}{f_i(v)}}{\sum_{j=1}^{n} \frac{x_j}{f_j(v)}}.$$

It is straightforward to see that $\sum_i \mu_i = k$. Intuitively, menu time corresponds roughly to the relative frequency with which an item must be included in the menu, where an item with $\mu_i = 1$ must always be included in the menu; the amount of menu time "charged" for a menu is inversely proportional to the sum of item scores in the menu, which enables an "apples to apples" comparison between resulting selection probabilities.

We first show that any $x \in \mathrm{IRD}(v, M)$ results in μ_i at most 1 per item. For any $x \in \mathrm{IRD}(v, M)$, consider an arbitrary convex combination of the menu-conditional item distributions given by items' scores $f_i(v)$, with the probability of each menu given by p_K. Allocate "menu time units" μ_K to each menu K in proportion with $p_K / \sum_{i \in K} f_i(v)$, such that $\sum_K \mu_K = k$, and further let $\mu_{K,i} = \mu_K / k$ for each $i \in K$. Observe that selection probability of an item i is given by

$$\begin{aligned} x_i &= \sum_{K: i \in K} p_K \cdot \frac{f_i(v)}{\sum_{j \in K} f_j(v)} \\ &= \frac{1}{Z} \sum_{K: i \in K} \frac{\mu_K}{k} \cdot f_i(v) \\ &= \frac{f_i(v)}{Z} \sum_{K: i \in K} \mu_{K,i} \end{aligned}$$

where Z is a normalizing constant such that $\sum_K \mu_K = k$, and so we have that $\sum_K \mu_{K,i} \leq 1$ as each μ_K is positive. Further, we have that

$$\begin{aligned} \sum_{K: i \in K} \mu_{K,i} &= Z \cdot \frac{x_i}{f_i(v)} \\ &= \frac{x_i}{f_i(v)} \cdot \frac{k}{\sum_{j=1}^{n} \frac{x_j}{f_j(v)}} \end{aligned}$$

upon solving for Z such that $\sum_K \mu_K = k$, which gives us that

$$\sum_{K: i \in K} \mu_{K,i} = \mu_i$$

and yields $\mu_i \leq 1$ for each item.

Next, we describe an algorithm $\texttt{MenuDist}(v, x, M)$ for constructing a menu distribution z which yields $\mathbb{E}_{K\sim z}[p(K, v)] = x$ for any x and v satisfying $\max_i \mu_i \leq 1$, constructively showing that $x \in \text{IRD}(v, M)$. Given x, v, and M which satisfy $\max_i \mu_i \leq 1$, we construct a menu distribution iteratively over $H \leq n$ *stages*. Let $\mu_i^1 = \mu_i$ be the initial remaining menu time for each item. At each stage $h \geq 1$ we will decrement the remaining time to $\mu_i^{h+1} \leq \mu_i^h$ for each item and track its change $\Delta_i^h = \mu_i^h - \mu_i^{h+1}$, with $\Delta^h = \sum_i \Delta_i^h$. Further, with μ_j^h as the kth highest remaining value, let $k_+^h < k$ be the number of items i with $\mu_i^h > \mu_j^h$, and $k_*^h \geq 1$ be the number of items i with $\mu_i^h = \mu_j^h$. At each stage, we maintain the following invariants:

1. $k_*^{h+1} \geq k_*^h + 1$ if $k_*^h < n$;
2. $\Delta_i^h = \frac{\Delta^h}{k}$ if $\mu_i^h > \mu_j^h$ (where μ_j^h is the kth highest value);
3. $\Delta_i^h = \frac{(k-k_+^h)\Delta^h}{k_*^h \cdot k}$ if $\mu_i^h = \mu_j^h$;
4. $\Delta_i^h = 0$ if $\mu_i^h < \mu_j^h$; and
5. $\Delta_i^h = \mu_i^h$ if $k_*^h = n$.

Intuitively, we are decreasing μ_i of each item with strictly larger μ_i^h than the kth highest at identical rates, corresponding to inclusion in the menu at that stage with probability 1, and breaking ties among the kth highest with fractional inclusion in the remaining spots. We decrease until the number of such tied items increases by at least 1 (or deplete all remaining menu time if all items are tied), and then continue to the next stage. Observe that when $\max_i \mu_i \leq 1$, this results in $\mu_i^{H+1} = 0$ for all items; if an item has $\mu_i^h > \mu_j^h$ (only occurring when $k_+^h > 0$), its remaining menu time decreases at a $1/k$ fraction of the total rate of decrease, and no μ_i^h will ever drop below 0 as we break ties among the highest remaining, and the total amount of depleted menu time is at most k. Once $k_+^h = 0$ we only ever decrease μ_i^h for items tied for the highest remaining value, and thus successfully satisfy $\mu_i^{H+1} = 0$ while maintaining our invariants.

We now show how to construct a menu distribution z from the quantities Δ_i^h which yields $p(z, v) = \mathbb{E}_{K\sim z}[p(K, v)] = x$. For a stage h, we construct a (multi)set of menus $S^h = \{K_s^h : s \in [|S^h|]\}$ with size $|S_h| = k_*^h$, where we can assume each $K_s^h \in S^h$ is distinct without loss of generality (e.g. by marking duplicates with some auxiliary notation). Let $J^h = \{i : \mu_i^h = \mu_j^h\}$ be the set of items tied for kth highest remaining; construct S^h iteratively over k_*^h steps by adding a menu K_s^h which includes all items with $\mu_i^h > \mu_j^h$, and $k - k_+^h$ items from J^h which belong to the fewest menus in S^h thus far, breaking ties arbitrarily. There are a total of $(k - k_+^h)k_*^h$ inclusions of some item in J^h to some menu in S^h, which results in each item $i \in J^h$ being included in exactly $k - k_+^h$ menus in S^h (as $(k - k_+^h)k_*^h$ is divisible by $k - k_+^h$), and the uniform distribution over S^h then satisfies

$$\Pr_{K_s^h\sim\text{Unif}(S^h)}\left[i \in K_s^h\right] = \begin{cases} 1 & \mu_i^h > \mu_j^h \\ \frac{k-k_+^h}{k_*^h} & \mu_i^h = \mu_j^h \\ 0 & \mu_i^h < \mu_j^h \end{cases}.$$

Let the menu distribution z^h be the distribution over S^h given by

$$\Pr_{K\sim z_h}\left[K = K_s^h\right] = \frac{\sum_{i\in K} f_i(v)}{\sum_{s\in|S^h|}\sum_{j\in K_s^h} f_j(v)},$$

which yields

$$\begin{aligned}\Pr_{K\sim z^h}\left[\text{Agent chooses i}\right] &= \sum_{K\in S^h : i\in K} \frac{f_i(v)}{\sum_{q\in K} f_q(v)} \cdot \frac{\sum_{q\in K} f_q(v)}{\sum_{s\in|S^h|}\sum_{j\in K_s^h} f_j(v)} \\ &= \frac{k\Delta_i^h}{\Delta^h} \cdot \frac{f_i(v)}{\sum_{s\in|S^h|}\sum_{j\in K_s^h} f_j(v)} \\ &\triangleq \frac{k\Delta_i^h}{\Delta^h} \cdot \frac{f_i(v)}{Z^h}\end{aligned}$$

as the probability of an agent choosing i conditional on being shown a menu sampled from z^h. Defining $Z = \sum_{h=1}^H \frac{Z^h\Delta^h}{k}$, let z be the mixture over distributions z^h with mass $\frac{Z^h\Delta^h}{Zk}$ on each.

Sampling a menu $K \sim z$ yields

$$\begin{aligned}
\Pr_{K\sim z}[\text{Agent chooses i}] &= \sum_{h=1}^{H} \frac{Z^h \Delta^h}{Zk} \cdot \Pr_{K\sim z^h}[\text{Agent chooses i}] \\
&= \sum_{h=1}^{H} \frac{Z^h \Delta^h}{Zk} \cdot \frac{k\Delta_i^h}{\Delta^h} \cdot \frac{f_i(v)}{Z^h} \\
&= f_i(v) \cdot \frac{\mu_i}{Z} && (\textstyle\sum_h \Delta_i^h = \mu_i) \\
&= x_i, && (\mu_i \propto x_i / f_i(v))
\end{aligned}$$

where $Z = \sum_{j=1}^{n} k \frac{x_j}{f_j(v)}$ then holds by proportionality as x and $p(z, v)$ are both probability distributions over $[n]$. By construction, it follows that $x \in \text{IRD}(v, M)$.

The algorithm `MenuDist`(v, x, M) which implements this construction can be run in poly(n) time, as the quantities Δ_i^h are computed over $H \leq n$ rounds each requiring $O(n)$ computation (after an initial sort of descending μ_i values), and each set S^h contains $k_*^h \leq n$ menus, each of which can be constructed in $O(k)$ time (by adding elements from J^h in a round-robin fashion) while maintaining the quantities necessary to compute the final normalizing constants. Sampling can be implemented efficiently as well, e.g. by sampling from a uniform distribution and thresholding appropriately. □

B.3 Analysis for Locally Learnable Preference Models

Each proof gives a learning algorithm which operates in a ball around the uniform vector, which is contained in EIRD(M) whenever $\lambda \geq \frac{k^2}{n}$.

B.3.1 Proof of Univariate Polynomial Local Learnability

Proof. Query the uniform vector v_U where each $v_i = \frac{1}{n}$. Let $Z = \frac{\sqrt{nd/6}}{\alpha}$. Consider three sets each of $d/2$ memory vectors where the items with indices satisfying $i \mod 3 = z$ each have memory values $\frac{1}{n} + \frac{j}{Z}$, items satisfying $i \mod 3 = z + 1$ have values $\frac{1}{n} - \frac{j}{Z}$, and the remainder have $\frac{1}{n}$ (for $z \in \{0, 1, 2\}$, and for $1 \leq j \leq \frac{d}{2}$). All such vectors lie in V_α, as $2n/3 \cdot (d/(2Z))^2 \leq \alpha^2$. Query

each of the $3d/2$ vectors. For each query, let R_v be the sum of all scores of the items held at $\frac{1}{n}$, divided by the sum of those same items' scores in the uniform query. Divide all scores by R_v. Let R_v^* be the be the corresponding ratio of these sums of scores under $\{f_i\}$; each sum is within $[\frac{\lambda}{3}, 1]$ at each vector, and the sums of observed scores have additive error at most $n\beta/3$. As such, R_v has additive error at most $\frac{2n\beta}{\lambda}$ from R_v^*. This gives us estimates for $d+1$ points of $\hat{f}_i(x_j) = \hat{y}_j$ for each polynomial, up to some universal scaling factor. We can express this d-degree polynomial $\hat{f}_i$ via Lagrange interpolation:

$$L_{d,j}(x) = \prod_{k \neq j}^{d} \frac{x - x_k}{x_j - x_k};$$
$$\hat{f}_i(x) = \sum_{j=0}^{d} \hat{y}_j L_{n,j}.$$

Note that $\sum_i \hat{f}_i(v_U) = 1$ as the scores coincide exactly with our query results at the uniform vector. To analyze the representation error, let $\{f_i^*\}$ be the set of true polynomials f_i rescaled to sum to 1 at the uniform vector; this involves dividing by a factor $S \in [n\lambda, n]$, and produces identical scores at every point. Consider the difference $\left|\hat{y}_j - y_j^*\right|$ for each $y_j^* = f_i^*(x_j)$. The query error for $\hat{y}_j$ prior to rescaling is at most β; rescaling by R_v^* would increase this to at most $3\beta/\lambda$, which is amplified to at most

$$\left|\hat{y}_j - y_j^*\right| \leq \frac{3\beta}{\lambda} + \frac{2n\beta}{\lambda} \leq \frac{3n\beta}{\lambda}$$

as each query score is at most 1 (and our setting is trivial for $n \leq 2$). The magnitude of each of the $d+1$ Lagrange terms can be bounded by:

$$\begin{aligned}\left|L_{d,j}(x)\right| &\leq \prod_{j=1}^{d/2} \frac{Z^2}{j^2} \\ &\leq \frac{Z^d}{((d/2)!)^2}\end{aligned}$$

for any $x \in [0, 1]$, and so for any function $\hat{f}_i(x)$ we can bound its distance from $f_i^*(x)$ by:

$$\begin{aligned}\left|f_i^*(x) - \hat{f}_i(x)\right| &= (d+1) \cdot \frac{3n\beta Z^d}{\lambda((d/2)!)^2} \\ &\leq \frac{(d+1)3n\beta Z^d}{\lambda 2^{d/2}}.\end{aligned}$$

This holds simultaneously for each $\hat{f}_i$ which, using the fact that the true ratio is at least λ/n and the per-function bound applies to each denominator term, gives us a total bound on the score estimates we generate:

$$\begin{aligned}\left|\frac{\hat{f}_i(x)}{\sum_{j=1}^{x} \hat{f}_j(x)} - \frac{f_i(x)}{\sum_{j=1} f_j(x)}\right| &\leq \left(1 + \frac{(d+1)3n\beta Z^d}{\lambda 2^{d/2}}\right) \cdot \frac{(d+1)3n^3\beta Z^d}{\lambda^2 2^{d/2}} \\ &\leq \frac{7n^3 d\beta Z^d}{\lambda^2 2^{d/2}} \\ &\leq \frac{3 \cdot (6nd)^{d/2+2}\beta}{\alpha^d \lambda^2 2^{d/2}} \\ &= \frac{(3nd)^{d/2+2}\beta}{\alpha^d \lambda^2}.\end{aligned}$$

Taking $\beta \leq \frac{\epsilon \alpha^d \lambda^2}{(3nd)^{d/2+2}}$ gives us an absolute error of at most ϵ per item score, satisfying a Euclidean bound of ϵ from any true score vector $M(w)/M_w^*$ for our hypothesis $\hat{M}(v) = \{\hat{f}_i(v_i) : i \in [n]\}$. □

B.3.2 Proofs of Multivariate Polynomial Local Learnability

Recall that the two classes of multivariate polynomial models we consider are *bounded-degree multilinear polynomial* preference models $\mathcal{M}_{BMLP}$, where:

- for each i, $M(v)_i = f_i(v)$, where f_i is a degree-d multilinear (i.e. linear in each item) polynomial which takes values in $[\lambda, 1]$ over $\Delta(n)$ for some constant $\lambda > 0$,

and the class of *bounded-degree normalized multivariate polynomial* preference models $\mathcal{M}_{BNMP}$, where:

- for each i, $M(v)_i = f_i(v)$, where f_i is a degree-d polynomial which takes values in $[\lambda, 1]$ over $\Delta(n)$ for some constant $\lambda > 0$, where $\sum_i f_i(v) = C$ for some constant C.

We prove local learnability results for each case.

Lemma 11. *$\mathcal{M}_{BMLP}$ is $O(n^d)$-locally learnable by an algorithm $\mathcal{A}_{BMLP}$ with $\beta \leq O(\frac{\epsilon^2}{\text{poly}(n(d/\alpha)^d)})$.*

Proof. Consider the set of polynomials where each v_n term is reparameterized as $1 - \sum_{i=1}^{n-1} v_i$, then translated so that the uniform vector appears at the origin (i.e. with $x_n = -\sum_{i=1}^{n-1} x_i$). Our approach will be to learn a representation of each polynomial normalized their sum, which is unique up to a universal scaling factor. Let f_i^* be the representation of f_i in this translation. Consider the $N = \sum_{j=0}^{d} \binom{n-1}{j}$-dimensional basis $\mathcal{B}$ where each variable in a vector x corresponds to a monomial of at most d variables in v, each with degree 1, with the domain constrained to ensure mutual consistency between monomials, e.g.:

$$\mathcal{B} = \{1, v_1, \ldots, v_{n-1}, v_1 v_2, \ldots, \prod_{j=n-d}^{n-1} v_j\}.$$

Observe that each f_i^* is a linear function in this basis. Let $q_i(x) = M(v)_i / M_v^*$ denote the normalized score for item i at v, where v translates to x in the new basis. For we any x we have:

$$\frac{f_i^*(x)}{\sum_{j=1}^{n} f_j^*(x)} = q_i(x),$$

and let $\hat{q}_i(x)$ denote the analogous perturbed query result, both of which sum to 1 over each i. We are done if we can estimate the vector $q(x)$ up to distance ϵ for any x.

With $f_i^*(x) = \langle a, x \rangle + a_0$ and $\sum_{i=1}^{n} f_i^*(x) = \langle b, x \rangle + b_0$, our strategy will be to estimate the ratio of each coefficient with b_0, for each f_i^*, in increasing order of degree. While our parameterization does not include item n, we will explicitly estimate b separate from each a, which we can then use to estimate $f_n^*(x) = \langle b, x \rangle + b_0 - \sum_{i=1}^{n-1} f_i^*(x)$. For a monomial m of degree j, we can estimate its coefficient for all f_i^* simultaneously by moving the values for variables it contains simultaneously from the $\mathbf{0}$ vector, and viewing the restriction to its subset monomials as a univariate polynomial of

degree j. We will use a single query to the $\mathbf{0}$ vector, and $2j+1$ additional queries for each degree-j monomial (which can be used for learning that monomial's coefficient in all f_i^* simultaneously), resulting in a total query count of:

$$1 + \sum_{j=1}^{d} (2j+1) \cdot \binom{n-1}{j} = 1 + \sum_{j=1}^{d} (2j+1) \frac{(n-1)!}{j!(n-j-1)!}$$
$$= O(n^d).$$

Querying $\mathbf{0}$ gives us an estimate for each additive term:

$$\frac{\hat{a}_0^i}{b_0} = \hat{q}_i(\mathbf{0})$$

which sum to 1 over all items (and we will take $\hat{b}_0 = 1$). We now describe our strategy for computing higher-order coefficients in terms of lower-order coefficients under the assumption of *exact* queries, after which we conduct error propagation analysis. For a monomial m of degree j, let $x_{(h,m)}$ be the point where $x_{(h,m),i} = hZ$ if an item i belongs to m and 0 otherwise, with higher degree terms satisfying the basis constraints (i.e. $(hZ)^3$ for a degree-3 subset of m, and $(hZ)^j$ for m), which also results in the term for a monomial containing any item not in m being set to zero. Query $x_{(h,m)}$ for $2j+1$ distinct values h in $\{\pm 1, \ldots, \pm(j+1)\}$. For $Z = \alpha/(2d(d+1))$ all queries lie in the α-ball, as the ℓ_1 norm of the positive coefficients, as well as the negative offset for item n, are both bounded by $\alpha/2$ in the original simplex basis. Suppose all coefficients up to degree $j-1$ are known. The result of such a query (with $z = hZ$) is equivalent to:

$$q(x_{(h,m)}) = \frac{a_m z^j + f_a(z)}{b_m z^j + f_b(z)}$$

where f_a and f_b are $(j-1)$-degree univariate polynomials, where each coefficient of some degree $k \leq j-1$ is expressed by summing the coefficients for degree-k monomials which are

subsets of m, for a and b respectively. Rearranging, we have:

$$a_m = q_i(x_{(h,m)}) \cdot b_m + \frac{q(x_{(h,m)}) \cdot f_b(z) - f_a(z)}{z^j}.$$

This gives us a linear relationship between a_m and b_m in terms of known quantities after just one query where $z \neq 0$. Suppose we could make *exact* queries; if we observe two distinct linear relationships, we can solve for a_m and b_m. If each query gives us the same linear relationship, i.e. $q_i(x_{(h,m)}) = q_i(x_{(h',m)})$ for every query pair (h, h'), then equality also holds for each of the $(q_i(x_{(h,m)}) \cdot f_b(z) - f_a(z))/z^j$ terms. If the latter term is truly a constant function c:

$$\frac{q_i(x_{(h,m)}) \cdot f_b(z) - f_a(z)}{z^j} = c$$

then we also have:

$$(a_m z^j + f_a(z)) \cdot f_b(z) - (b_m z^j + f_b(z)) \cdot f_a(z) = c z^j (b_m z^j + f_b(z)).$$

Each side is a polynomial with degree at most $2j$, and thus cannot agree on $2j + 1$ points unless equality holds. However, if equality does hold, we have that either $c = 0$ or $b_m = 0$, as the left side has degree at most $2j - 1$, and both z^j and $b_m z^j + f_b(z)$ are bounded away from 0 for any $z \neq 0$. If $c \neq 0$, then we have that $b_m = 0$ and $a_m = c$. If $c = 0$, then we have

$$a_m z^j f_a(z) f_b(z) - b_m z^j f_a(z) f_b(z) = 0,$$

which implies $a_m = b_m$, as $f_a(z) f_b(z)$ cannot be equal to 0 everywhere due to each a_0^i and b_0 being positive. Our answer to $q(x_{(h,m)})$ will be bounded above 0 and below 1, allowing for us to solve for both a_m and b_m as

$$a_m = b_m = \frac{q_i(x_{(h,m)}) \cdot f_b(z) - f_a(z)}{(1 - q_i(x_{(h,m)}))z^j}.$$

To summarize, if given exact query answers for $2j + 1$ distinct points, we must be in one of the following cases:

- We observe at least two distinct linear relationships between a_m and b_m from differing query answers;
- We observe a non-zero constant $\frac{q_i(x_{(h,m)}) \cdot f_b(z) - f_a(z)}{z^j} = c$ for each query, and have $a_m = c$;
- We observe $\frac{q_i(x_{(h,m)}) \cdot f_b(z) - f_a(z)}{z^j} = 0$ for each query, and can solve for $a_m = b_m$.

To begin our error analysis for perturbed queries, we first show a bound on the size of the coefficients for a polynomial which is bounded over a range.

Lemma 12. *Each degree-d' coefficient of f_i^* is at most ${d'}^{2d'}$.*

Proof. First note that the constant coefficient and the coefficient for each linear term have magnitude at most 1, as the function is bounded in $[\lambda, 1]$ over the domain (which includes $\mathbf{0}$). For a degree-d' monomial m, consider the univariate polynomial corresponding to moving each of its variables in synchrony while holding the remaining variables at 0, whose degree-d' coefficient is equal to a_m. Consider the Lagrange polynomial representation of this polynomial

$$L_{d',j}(x) = \prod_{k \neq j}^{d'} \frac{x - x_k}{x_j - x_k};$$
$$\hat{f}_i(x) = \sum_{j=0}^{d'} \hat{y}_j L_{n,j}.$$

for $d' + 1$ evenly spaced points in the range $[-1/n, 1/d' - 1/n]$, which are all feasible under the simplex constraints (corresponding to $v_i \in [0, 1/d']$ in the original basis, for each $i \in m$). Each pair of points is separated by a distance of at least $1/({d'}^2)$, and so the leading coefficient of each Lagrange term is at most ${d'}^{2(d'-1)}$. Each $\hat{y}_j$ is in $[\lambda, 1]$ and so we have

$$a_m \leq (d' + 1){d'}^{2(d'-1)}$$
$$\leq {d'}^{2d'}$$

for each $d' > 1$. □

As we estimate coefficients for monomials of increasing degree, we will maintain the invariant that each degree-j coefficient of a and b is estimated up to additive error ϵ_j, with respect to the normalization where $b_0 = 1$. Immediately we have $\epsilon_0 = \beta$ for the estimates $\hat{a}_0$ from our query to the $\mathbf{0}$ vector. We will also let β_j denote the error of a polynomial $\hat{f}_a$ restricted to terms for subsets of a j-degree monomial m

For a monomial m, suppose we receive 2 queries $\hat{q}_i(x_{(h,m)})$ and $\hat{q}_i(x_{(h',m)})$ for some h and h' where

$$\left|\hat{q}_i(x_{(h,m)}) - \hat{q}_i(x_{(h',m)})\right| \geq F_j$$

for some quantity F_j. Then we have:

$$\begin{aligned}
\hat{a}_m &= \hat{q}_i(x_{(h,m)})\hat{b}_m + \frac{\hat{q}_i(x_{(h,m)}) \cdot \hat{f}_b(hZ) - \hat{f}_a(hZ)}{(hZ)^j} \\
&= \hat{q}_i(x_{(h',m)})\hat{b}_m + \frac{\hat{q}_i(x_{(h,'m)}) \cdot \hat{f}_b(h'Z) - \hat{f}_a(h'Z)}{(h'Z)^j} \\
\hat{b}_m &= \frac{\hat{a}_m}{\hat{q}_i(x_{(h,m)})} + \frac{\frac{\hat{f}_a(hZ)}{\hat{q}_i(x_{(h,m)})} - \hat{f}_b(hZ)}{(hZ)^j}; \\
&= \frac{\hat{a}_m}{\hat{q}_i(x_{(h',m)})} + \frac{\frac{\hat{f}_a(h'Z)}{\hat{q}_i(x_{(h',m)})} - \hat{f}_b(h'Z)}{(h'Z)^j}; \\
\frac{\hat{a}_m}{\hat{q}_i(x_{(h',m)})} - \frac{\hat{a}_m}{\hat{q}_i(x_{(h,m)})} &= \frac{\frac{\hat{f}_a(hZ)}{\hat{q}_i(x_{(h,m)})} - \hat{f}_b(hZ)}{(hZ)^j} - \frac{\frac{\hat{f}_a(h'Z)}{\hat{q}_i(x_{(h',m)})} - \hat{f}_b(h'Z)}{(h'Z)^j}; \\
\hat{a}_m &= \frac{\frac{\hat{q}_i(x_{(h',m)})\hat{f}_a(hZ)}{\hat{q}_i(x_{(h,m)})} - \hat{q}_i(x_{(h',m)})\hat{f}_b(hZ)}{\left(1 - \frac{\hat{q}_i(x_{(h',m)})}{\hat{q}_i(x_{(h,m)})}\right) \cdot (hZ)^j} - \frac{\hat{f}_a(h'Z) - \frac{\hat{f}_b(h'Z)}{\hat{q}_i(x_{(h',m)})}}{\left(1 - \frac{\hat{q}_i(x_{(h',m)})}{\hat{q}_i(x_{(h,m)})}\right) \cdot (h'Z)^j}; \\
\hat{b}_m &= \frac{\frac{\hat{q}_i(x_{(h,m)}) \cdot \hat{f}_b(hZ) - \hat{f}_a(hZ)}{(hZ)^j} - \frac{\hat{q}_i(x_{(h,'m)}) \cdot \hat{f}_b(h'Z) - \hat{f}_a(h'Z)}{(h'Z)^j}}{\hat{q}_i(x_{(h',m)}) - \hat{q}_i(x_{(h',m)})};
\end{aligned}$$

where $\hat{f}_a$ and $\hat{f}_b$ are the univariate polynomials from summing the lower-order coefficient estimates

for each degree up to $j-1$. The additive error to each $\hat{f}_a(hZ)$ and $\hat{f}_a(hZ)$ can be bounded by:

$$\beta + \sum_{k=1}^{j-1} \binom{n}{k} (hZ)^k k^{2k} \epsilon_k = \beta + \sum_{k=1}^{j-1} \binom{n-1}{k} (k^2 hZ)^k \epsilon_k.$$

Further, the magnitude of each $\hat{f}_a(hZ)$ and $\hat{f}_b(hZ)$ is at most $1 + \sum_{k=1}^{j-1} \binom{n-1}{k} (k^2 hZ)^k$. We can bound the error of other terms as follows:

- Each $\hat{q}_i(x_{(h',m)}) - \hat{q}_i(x_{(h',m)})$ has magnitude at least F_j and at most 1, and additive error at most 2β;
- Each $\hat{q}_i(x_{(h',m)})$ has value at least $\frac{\lambda}{n}$ and at most 1, and additive error at most β;
- Each $\frac{\hat{q}_i(x_{(h',m)})}{\hat{q}_i(x_{(h,m)})}$ term is either greater than $\frac{1}{1-F_j}$ or at most $1 - F_j$; the true ratio between the numerator and denominator is at least λ/n most n/λ, with additive error up to β in both.
- Each $1 - \frac{\hat{q}_i(x_{(h',m)})}{\hat{q}_i(x_{(h,m)})}$ term, is either greater than F_j or at most $1 - \frac{1}{1-F_j}$;
- Each $(hZ)^j$ has magnitude at least Z^j;

The error in the numerator of $\hat{a}_m$, and the fractional terms in the numerator of $\hat{b}_m$ is dominated by multiplying the functions of $\hat{q}_i$ with the polynomials themselves. As such, we can bound the error to a_m and b_m by ϵ_j if we have that:

$$\begin{aligned}
\epsilon_j &\geq O\left(\frac{n\beta}{\lambda F_j Z^j} \cdot \left(1 + \sum_{k=1}^{j-1} \binom{n-1}{k} (k^2 hZ)^k\right)\right) \\
&= O\left(\frac{n\beta}{\lambda F_j Z^j} \cdot \left(1 + \sum_{k=1}^{j-1} \binom{n-1}{k} (h\alpha)^k\right)\right) \\
&= O\left(\frac{nd^{2j}\beta}{\lambda \alpha^j F_j}\right)
\end{aligned}$$

for any $\alpha < 1/(nd)$. Now suppose all pairs of query answers we see are separated by less than F_j;

the additive error to each estimate of the quantity

$$\hat{c}_{(h,m)} = \frac{\hat{q}_i(x_{(h,m)}) \cdot \hat{f_b}(z) - f_a(z)}{(hZ)^j}$$

is $\mathcal{E}_j = O\left(\frac{\beta}{Z^j} \cdot \left(1 + \sum_{k=1}^{j-1} \binom{n}{k} (k^2 hZ)^k\right)\right) = O\left(\beta \cdot nd^{2j}/\alpha^j\right)$. If each such quantity has value at most $\mathcal{E}_j$, we assume this quantity is zero and solve for $a_m = b_m$. If some are larger, we must be in the case where $\hat{b}_m \approx 0$ and so we set $a_m = \hat{c}_{(h,m)}$ for any query result. By taking each $F_j = O(\sqrt{\beta}\,\mathrm{poly}(n, d^j, 1/\alpha^j))$ we can obtain a bound of $\epsilon_j = O(\sqrt{\beta}\,\mathrm{poly}(n, d^j, 1/\alpha^j))$ to each coefficient regardless of which case we are in; after summing the error contribution across coefficients and accounting for renormalization, recalling that $\lambda = \Omega(1/n)$, we obtain a bound of ϵ on score vector errors (for any desired norm) provided that $\epsilon \geq \sqrt{\beta}\,\mathrm{poly}(n, d^d, 1/\alpha^d)$. □

Next, we prove the local learnability result for normalized multivariate polynomials.

Lemma 13. *$\mathcal{M}_{BMNP}$ is $O(n^d)$-locally learnable by an algorithm $\mathcal{A}_{BMNP}$ with $\beta \leq \frac{\epsilon}{\alpha^d F(n,d)}$, where $F(n, d)$ is some function depending only on n and d which is finite for all $n, d \in \mathbb{Z}$.*

Proof. Our approach will be to construct a set of $O(n^d)$ queries which results in a data matrix which is nonsingular in the space of d-degree multivariate polynomials, solve for the coefficients of each f_i as a linear function over this basis, and show that the basis is sufficiently well-conditioned such that our approximation error is bounded.

Consider the set of polynomials where each v_n term is reparameterized as $1 - \sum_{i=1}^{n-1} v_i$, then translated so that the uniform vector appears at the origin (i.e. with $v_n = -\sum_{i=1}^{n-1} v_i$). Our approach will be to learn a representation of each polynomial directly, as they are already normalized to sum to a constant (which must be in the range $[1, n]$). Let f_i^* be the representation of f_i in this translation. Let $\mathcal{B}$ be the $N = \sum_{j=0}^{d} (n-1)^j$-dimensional basis where each variable in a vector x corresponds to a monomial of variables in v with degree at most d, with the domain constrained to ensure mutual consistency between monomials, e.g.:

$$\mathcal{B} = \{1, v_1, \ldots, v_{n-1}, v_1^2, v_1 v_2, \ldots, v_{n-1}^d\}.$$

Observe that f_i^* is a linear function in this basis, with $f_i^*(x) = \langle a, x \rangle$ and $\sum_{i=1}^n f_i^*(x) = \langle b, x \rangle$ for any x represented in $\mathcal{B}$.

There is a large literature on constructing explicit query sets for multivariate polynomial interpolation, which ensure that the resulting data matrix is nonsingular; see [124] for an overview. The set must have at least N points to ensure uniqueness of interpolation, and this is sufficient when points are appropriately chosen. Let S^* be any such set such that each point $\|w\|_1 \leq 1/2$ for each $w \in S^*$, and let $C_{n,d}$ be the ℓ_∞ condition number of the resulting matrix Y (which will be positive due to nonsingularity) given by:

$$Y = \begin{bmatrix} y_1^{(1)} & \cdots & y_N^{(1)} \\ \vdots & & \vdots \\ y_1^{(j)} & \cdots & y_N^{(j)} \\ \vdots & & \vdots \\ y_1^{(N)} & \cdots & y_N^{(N)} \end{bmatrix}$$

where $y^{(j)}$ is the representation of $s^{(j)}$ in the basis $\mathcal{B}$. We show that for any α, we can construct a matrix X from a query set S^α of size N where $\|v\|_1 \leq \alpha/2$ for each $v \in S^\alpha$. For each $s^{(j)}$, let $v^{(j)} = \alpha s^{(j)}$, which results in $\|v\|_1 \leq \alpha/2$ for the parameterization over $n-1$ items, and so radius of α holds when including all n items. This results in a matrix X given by

$$X = \begin{bmatrix} x_1^{(1)} & \cdots & x_N^{(1)} \\ \vdots & & \vdots \\ x_1^{(j)} & \cdots & x_N^{(j)} \\ \vdots & & \vdots \\ x_1^{(N)} & \cdots & x_N^{(N)} \end{bmatrix}$$

We then have

$$X = YD,$$

where D is a diagonal matrix with the jth diagonal entry ν_j equal to α^{d_j}, where d_j is degree of the jth monomial in $\mathcal{B}$, as our scaling by α is amplified for each column in correspondence with the associated degree; the values of D will range from α^d to 1. We can then bound the condition number of X as:

$$\begin{aligned}
\text{cond}(X) &= \text{cond}(YD) \\
&= \|YD\| \left\|(DY)^{-1}\right\| \\
&\leq \|Y\| \, \|D\| \left\|D^{-1}\right\| \left\|Y^{-1}\right\| \\
&= \text{cond}(Y) \cdot \text{cond}(D) \\
&\leq C_{n,d} \frac{\max_j \nu_j}{\min_j \nu_k} \\
&= \frac{C_{n,d}}{\alpha^d}.
\end{aligned}$$

Let q denote the vector of exact answers to each query in x from f_i, equal to $a\dot{x}$ and let $\hat{q}$ be the answers we observe for item i from querying each x. As X is nonstationary, we have that $Xa = q$, and by standard results in perturbation theory for linear systems, for $\hat{a}$ such that $X\hat{a} = \hat{q}$ we have that:

$$\begin{aligned}
\frac{\|\hat{a} - a\|}{\|a\|} &\leq \text{cond}(X) \frac{\|\hat{q} - q\|}{\|q\|} \\
&\leq \frac{\beta n C_{n,d}}{k^2 \alpha^d}
\end{aligned}$$

as each entry in q is at least $\lambda \geq k^2/n$. Further note that the maximum coefficient of a degree-d multivariate polynomial which takes maximum value 1 over the unit ball (and hence the simplex) can be shown to be bounded by a finite function of n and d (see [125]); when accounting for this factor in relative error across all terms and items, as well as the condition number, we have that for $\beta \leq \frac{\epsilon}{\alpha^d F(n,d)}$ for some function $F(n, d)$, the scores generated by the functions $\hat{f}_i$ using our estimated coefficients $\hat{a}$ result in score vector estimates bounded by ϵ. □

B.3.3 Proof of SFR Local Learnability

We now prove that functions with local sparse Fourier transformation are locally learnable. Recall that a function $f(x)$ has a ℓ-sparse Fourier transform if it can be written as

$$f(x) = \sum_{i=1}^{\ell} \xi_i e^{2\pi \mathbf{i} \eta_i x},$$

where η_i is the i-th frequency, ξ_i is the corresponding magnitude, and $\mathbf{i} = \sqrt{-1}$.

We will use the following result about learning sparse Fourier transforms [126].

Theorem 29 ([126])**.** *Consider any function $f(x) : \mathbb{R} \to \mathbb{R}$ of the form*

$$f(x) = f^*(x) + g(x),$$

where $f^*(x) = \sum_{i=1}^{\ell} \xi_i e^{2\pi \mathbf{i} \eta_i x}$ with frequencies $\eta_i \in [-F, F]$ and frequency separation $\hat{\alpha} = \min_{i \neq j} |\eta_i - \eta_j|$, and $g(x)$ is the arbitrary noise function. For some parameter $\delta > 0$, we define the noise-level *over an interval $I = [a, b] \subseteq \mathbb{R}$ as*

$$\mathcal{N}^2 = \frac{1}{|I|} \int_I |g(x)|^2 dx + \delta \sum_{i=1}^{\ell} |\xi_i|^2 .$$

There exists an algorithm that takes samples from the interval I with length $|I| > O(\frac{\log(\ell/\delta)}{\hat{\alpha}})$ and returns a set of ℓ pairs $\{(\xi_i', \eta_i')\}$ such that for any $|\xi_i| = \Omega(\mathcal{N})$ we have for an appropriate permutation of the indices

$$|\eta_i - \eta_i'| = O\Big(\frac{\mathcal{N}}{|I||\xi_i|}\Big), \qquad |\xi_i - \xi_i'| = O(\mathcal{N}), \forall i \in [\ell] .$$

The algorithm takes $O(\ell \log(F|I|) \log(\frac{\ell}{\delta}) \log(\ell))$ samples and $O(\ell \log(F|I|) \log(\frac{F|I|}{\delta}) \log(\ell))$ and succeeds with probability at least $1 - 1/k^c$ for any arbitrary constant c.

Furthermore, the algorithm used in the above theorem uses samples of the form $x_0, x_0 + \sigma \cdots x_0 +$

$\ell \log(\ell/\delta)\sigma$ for randomly chosen x_0 and $\sigma = O(|I|/\ell \log(\ell/\delta))$.

We will use the above theorem to learn the sparse Fourier representation of the preference model. Recall that for a memory vector v and item $i \in [n]$, $M(v)_i = f_i(v_i)$.

Proof. Let v_{unif} denote the uniform memory vector. We will learn each function f_i separately. Fix $i \in [n]$. We will set the interval I to be $[1/n - Z, 1/n + Z]$ for some sufficiently small $\frac{\log(\ell/\delta)}{\hat{\alpha}} \leq Z \leq \alpha/2$ where $\hat{\alpha}$ is the frequency separation, where $\alpha = \tilde{\Omega}(1/\hat{\alpha})$ so that Z is defined. Let $S = \{x_j\}_{j=1}^{\tilde{O}(\ell)}$ for $x_j \in [-Z, Z]$ be a set of points such that the Fourier learning algorithm queries $1/n + x$ for each $x \in S$. For each point $x \in S$, we define the memory vector $v^x = v_{\text{unif}} + xe_i - xe_j$ where j is a fixed randomly chosen other index. All such vectors lie in V_α, as $2(\alpha/2)^2 \leq \alpha^2$. We query all vectors v^x for $x \in S$, along with v_{unif}. Recall that $\hat{s}_v$ is the empirical score vector at a memory vector v. For each vector v, let R_v be the sum of all scores of all the $n-2$ items held at $\frac{1}{n}$, divided by the sum of those same items' scores in the uniform vector v_{unif}. For each vector v^x we multiply the score $\hat{s}_{v^x,i}$ of item i by R_{v^x} to obtain a noisy sample of $f_i(1/n + x)$. For $i \in \widetilde{O}(\ell)$, let the i-th sample be denoted by $\hat{y}_i$ and the true value $f_i(1/n + x_i)$ be denoted by y_i. We then pass all these samples to the Fourier learning algorithm in Theorem 29 in order to get an estimate $\hat{f}$ of f.

We now analyze the error in the samples. Let R_v^* be the corresponding ratio of these sums of scores under $\{f_i\}$; each sum is within $[\frac{\lambda}{3}, 1]$ at each vector, and the sums of observed scores have additive error at most $2n\beta$. As such, R_v has additive error at most $\frac{2n\beta}{\lambda}$ from R_v^*. For each vector v^x we have that $\hat{s}_{v^x,i}/(\sum_j \hat{s}_{v^x,j})$ is within a β error from $s_{v^x,i}/(\sum_j s_{v^x,j})$. Hence, the total error in each sample is bounded as:

$$|\hat{y}_i - y_i| \leq \frac{7n\beta}{\lambda}.$$

Using this we can bound the total noise term by $\mathcal{N} = 8n\beta/\lambda$ using our choice of $\delta = (\beta n)/(\lambda \sum_{i=1}^{\ell} |\xi_i|)$. The algorithm will return a set of $\{(\hat{\eta}_i, \hat{\xi}_i)\}$ such that

$$|\eta_i - \eta_i'| = O\Big(\frac{1}{\alpha}\Big), \qquad |\xi_i - \xi_i'| = O(\frac{\beta n}{\lambda}), \forall i \in [\ell].$$

So for function $\hat{f}_i(x)$ we can bound its distance from $f_i(x)$ by:

$$\begin{aligned}\left|f_i(x) - \hat{f}_i(x)\right| &= \left|\sum_{i=1}^{\ell} \xi_i e^{2\pi \mathbf{i} \eta_i x} - \sum_{i \in [\ell]} \hat{\xi}_i e^{2\pi \mathbf{i} \hat{\eta}_i x}\right| \\ &\leq \sum_{i \in [\ell]} \left|\xi_i e^{2\pi \mathbf{i} \eta_i x} - \hat{\xi}_i e^{2\pi \mathbf{i} \hat{\eta}_i x}\right| \\ &\leq \sum_{i \in [\ell]} \left|\xi_i - \hat{\xi}_i\right| |\eta_i - \hat{\eta}_i| \\ &\leq O(\frac{\ell n \beta}{\lambda \alpha}),\end{aligned}$$

since we normalize the above estimates to get a score estimate, the total bound on the score estimates can be bounded as:

$$\left|\frac{\hat{f}_i(x)}{\sum_{j=1}^{x} \hat{f}_j(x)} - \frac{f_i(x)}{\sum_{j=1} f_j(x)}\right| \leq O(\frac{\ell \beta n}{\alpha \lambda}).$$

Taking $\beta \leq \frac{\epsilon \lambda \alpha}{\sqrt{n} \ell}$ gives us an error of at most $\epsilon \sqrt{n}$, satisfying a Euclidean bound of ϵ from any true score vector $M(w)/M^*_w$ for our hypothesis model $\hat{M}(v) = \{\hat{f}_i(v_i) : i \in [n]\}$. □

B.4 Analysis for Algorithm 6: Targeting EIRD for Smooth Preferences

B.4.1 Effective Memory Horizons

Here we give some results for bounding the change in memory as rounds progress.

Lemma 14 (Bounding Memory Drift)**.** *For any $\gamma \in (0, 1)$, $g \in (0, \gamma]$, $t \geq 1$, and $w \geq 1$ such that $g^w \geq 1 - 2\beta$ and $g^{t+w-1} \leq \frac{1}{2}$, we have that $d_{TV}(v_t, v_{t+w})$ is at most β.*

Proof. We can express the memory vector v_{t+w} as

$$\begin{aligned} v_{t+w} &= \frac{\sum_{s=t}^{t+w-1} \gamma^{t+w-s-1} \cdot i_s + \sum_{s=1}^{t-1} \gamma^{t+w-s-1} \cdot i_s}{\sum_{s=1}^{t+w-1} \gamma^{t+w-s-1}} \\ &= \frac{\sum_{s=t}^{t+w-1} \gamma^{t+w-s-1} \cdot i_s}{\sum_{s=1}^{t+w-1} \gamma^{t+w-s-1}} + v_t \cdot \left(1 - \frac{\sum_{s=t}^{t+w-1} \gamma^{t+w-s-1}}{\sum_{s=1}^{t+w-1} \gamma^{t+w-s-1}}\right) \end{aligned}$$

which then yields

$$\begin{aligned} d_{TV}(v_t, v_{t+w}) &\leq \frac{\sum_{s=0}^{w-1} \gamma^s}{\sum_{s=0}^{t+w-2} \gamma^s} \\ &= \frac{1-\gamma^w}{1-\gamma^{t+w-1}} \\ &\leq \frac{1-g^w}{1-g^{t+w-1}} \qquad \text{(B.6)} \\ &\leq \beta. \end{aligned}$$

Step B.6 follows from the fact that

$$\begin{aligned} \frac{\partial}{\partial \gamma}\left(\frac{1-\gamma^w}{1-\gamma^{t+w-1}}\right) &= \frac{-w\gamma^{w-1}(1-\gamma^{t+w-1}) + (1-\gamma^w)(t+w-1)\gamma^{t+w-2}}{(1-\gamma^{t+w-1})^2} \\ &= \frac{\gamma^{w-1}\left((t+w-1)\gamma^{t-1} - (t-1)\gamma^{t+w-1} - w\right)}{(1-\gamma^{t+w-1})^2} \\ &\leq 0, \end{aligned}$$

as the function $(t+w-1)\gamma^{t-1} - (t-1)\gamma^{t+w-1} - w$ is increasing over $\gamma \in [0,1]$ from $-w$ to 0 (and thus the derivative for (B.6) is negative at any $\gamma \in (0,1)$). □

We can use this to obtain an upper limit on w in terms of c such that this bound holds.

Lemma 15. *For $\gamma \geq 1 - 1/T^c$, $t \geq T^c$, and $w \leq \beta \cdot T^c$, we have that $d_{TV}(v_t, v_{t+w}) \leq \beta$.*

Proof. Let $g = 1 - \frac{1}{T^c}$, where we have that $g^{(T^c)} \leq \frac{1}{e} \leq \frac{1}{2}$. Further, we have:

$$\begin{aligned}
\log\left(\frac{1}{1-2\beta}\right) &\geq 2\beta \\
&\geq \frac{2w}{T^c} \\
&\geq w \log\left(\frac{1}{1-\frac{1}{T^c}}\right) && (\text{ for } \tfrac{1}{T^c} \leq \tfrac{1}{2}) \\
&= \log\left(\frac{1}{g^w}\right).
\end{aligned}$$

As $\log(1/x)$ is decreasing in x we have that $g^w \geq 1 - 2\beta$, which yields the result via Lemma 14. □

B.4.2 Main Result for Algorithm 6

Here we restate Algorithm 6 and and give a proof of its guarantees.

Algorithm 11 (Targeting EIRD for Smooth Models).

Let $c^* = \min(c, 3/4)$, $\epsilon = \tilde{O}(nL^{1/4}\lambda^{-3/2}T^{-c/4})$, $Q = \tilde{O}(\frac{n^2}{\lambda^4\epsilon^2})$, and $\eta = (nT)^{-1/2}$.
Initialize $q = 0$, $v_0 = v^* = \mathbf{u}_n$, $F_i = \frac{C}{n}$ for $i \in [n]$, $M^* = \{F_i\}$
Initialize DBG for ϵ, η.
while $t \leq T$ **do**
  if $t < T^{c^*}$ **then**
    Show arbitrary menu K_t to agent
  else if $t \geq T^{c^*}$ and either $\|v_t - v^*\|_1 \geq \frac{\lambda\epsilon}{2nL}$ or $q = 0$ **then**
    for $b \in \{0, \ldots, \lceil \frac{n-1}{k-1} \rceil\}$ **do**
      Show agent menu $K_b = \{1\} \cup \{b(k-1)+2, \ldots, (b+1)(k+1)+1\}$ for Q rounds
      Let $\hat{F}_i = (\#\text{ times } i_t = i)\,/(\#\text{ times } i_t = 1)$ within the Q rounds, for $i \in K_b$
    end for
    Set $F_i = \frac{C \cdot \hat{F}_i}{\sum_{j=1}^n \hat{F}_j}$ for each $i \in [n]$, $M^* = \{F_i\}$
    Set $v^* = v^t$, increment q by $\lceil \frac{n-1}{k-1} \rceil \cdot Q$, set $\mathcal{K}_{t-q+1} = \mathcal{K}_{t-q} \cap \mathrm{IRD}(v^*, M^*)$ for DBG
  else
    Get x_{t-q} from DBG
    Let $z_t = \texttt{MenuDist}(v_t, x_{t-q}, M^*)$, sample menu $K_t \sim z_t$
    Show K_t to agent, update DBG with observed i_t and $r_t(i_t)$
  end if
  Set $v_{t+1} = U(v_t, i_t, t)$, for each round counted by q if necessary
end while

Theorem 30. *For an agent with a (λ, L)-smooth preference model M for $\lambda \geq k/n$, and γ-discounted memory for $\gamma \geq 1 - \frac{1}{T^c}$ and $c \in (0, 1]$, Algorithm 6 obtains regret bounded by*

$$\max_{x^* \in \text{EIRD}(M)} \sum_{t=1}^{T} r_t^\top x^* - \mathbb{E}\left[\sum_{t=1}^{T} r_t(i_t)\right] = \tilde{O}\left((n/\lambda)^{3/2} L^{1/4} \cdot T^{1-c/4}\right).$$

Proof. In each round, outside of those spent updating our estimates of preference (counted by q), we receive a target distribution x_t from DBG, and we construct a menu distribution z_t which aims to induce a choice distribution x_t on behalf of the agent. Recall that we assume γ is known, and so we can exactly track the agent's memory vector v_t across rounds; we note that this result can be extended to the case where only a lower bound on γ is known by checking the condition on $\|v_t - v^*\|_1$ across all possible values of γ in each round. Observe that if our preference estimates F_i were to always exactly track the agent's true preferences $f_i(v_t)$, and yield exact representations of $\text{IRD}(v_t)$ in each round, then we would have perturbations $\xi_t = \mathbf{0}$ to each target distribution x_t by the guarantee of Lemma 2, and a decision set $\mathcal{K}_T$ which contains EIRD. As such, our regret would be immediately bounded by

$$\max_{x^* \in \text{EIRD}(M)} \sum_{t=1}^{T} r_t^\top x^* - \mathbb{E}\left[\sum_{t=1}^{T} r_t(i_t)\right] \leq \mathsf{Reg}_{\text{EIRD}}(\text{DBG}; T, \epsilon, \eta) + T^{1-c/4} + q = \tilde{O}(T^{1-c/4} + q)$$

as $T^{c^*} \leq T^{1-c/4}$ for any $c \in (0, 1]$ (ignoring non-T terms). Here, note that EIRD is non-empty and contains $\mathbf{u}_n$ by Lemma [1], as $\lambda \geq \frac{k}{n}$. The remainder of our analysis will focus on showing that the perturbations ξ_t from preference estimate imprecision remain small, and that the estimation time q does not grow too large.

Case 1: $c^* = c$. For any $c \leq 3/4$, the agent's memory is "saturated" by the time we conclude showing arbitrary menus and proceed to our alternation between learning and optimization. Let ϵ satisfy

$$\epsilon \geq 8nL^{1/4}\lambda^{-3/2}\log(2T/\delta)^{1/4}T^{-c/4}$$

with $\delta = 1/T$. Observe that the following hold:

$$\frac{\lambda^2\epsilon}{8nkL} \cdot T^c \geq \frac{64n^2(n-1)\log(2T/\delta)}{(k-1)\lambda^4\epsilon^2} \qquad (\epsilon^3 \geq \epsilon^4 \geq 512n^4L\lambda^{-6}\log(2T/\delta)T^{-c})$$
$$\frac{\lambda\epsilon^2}{4nL} \cdot T^c \geq \frac{64n^2(n-1)\log(2T/\delta)}{(k-1)\lambda^4\epsilon^2} \qquad (\epsilon^4 \geq 256n^4k^{-1}L\lambda^{-5}\log(2T/\delta) \cdot T^{-c})$$

By Lemma 15, in each of the $Q \cdot \lceil \frac{n-1}{k-1} \rceil$ prior to updating score estimates F_i, we have

$$\left| \frac{f_i(v^*)}{\sum_{j \in K_b} f_i(v^*)} - \frac{f_i(v_t)}{\sum_{j \in K_b} f_i(v_t)} \right| \leq \frac{\lambda^2\epsilon}{8n},$$

by (λ, L)-smoothness for $\{f_i\}$, as $Q \cdot \lceil \frac{n-1}{k-1} \rceil \leq \frac{\lambda^2\epsilon}{8nkL} \cdot T^c$. By a Hoeffding bound, we then have

$$\begin{aligned} \Pr\left[\left| (\#\text{ times } i \in K_b \text{ chosen }) - \frac{Qf_i(v^*)}{\sum_{j \in K_b} f_i(v^*)} \right| > \frac{Q\lambda^2\epsilon}{4n} \right] &\leq 2\exp\left(-\frac{Q^2\lambda^4\epsilon^2}{64n^2Q} \right) \\ &\leq \frac{\delta}{T}, \end{aligned}$$

as $Q \geq \frac{64n^2\log(2T/\delta)}{\lambda^4\epsilon^2}$. If this holds for all i across each K_b (including each instance for $i = 1$), we then have

$$\left| \hat{F}_i - \frac{f_i(v^*)}{f_1(v^*)} \right| \leq \frac{\lambda\epsilon}{2n},$$

as $f_1(v^*) \geq \lambda$. Observe that the normalization to $\{F_i\}$ is equivalent to rescaling the empirical frequencies observed for each K_b such that their scores agree on $i = 1$ and the largest score across all items is at most 1, which will not increase relative error for any item. As such, we have

$$|F_i - f_i(v^*)| \leq \frac{\lambda\epsilon}{2n}$$

as well. Further, in any subsequent round where $\|v_t - v^*\|_1 \leq \frac{\lambda\epsilon}{2nL}$ we have

$$\begin{aligned}|F_i - f_i(v_t)| &\leq |F_i - f_i(v^*)| + |f_i(v_t) - f_i(v^*)| \\ &\leq \frac{\lambda\epsilon}{n},\end{aligned}$$

as each f_i is L-Lipschitz with respect to the ℓ_1 norm, and so for all rounds t where DBG is played, we have $|F_i - f_i(v_t)| \leq \frac{\lambda\epsilon}{n}$ for each i with probability $1 - \delta$, which contributes at most 1 to our expected regret if $\delta = 1/T$. By Lemma 15, this holds for at least $\frac{\lambda\epsilon}{4nL} \cdot T^c$ subsequent rounds. We now show that this yields a bound of $|\xi_i| \leq \frac{\epsilon}{n} x_i$ in each round for DBG.

Lemma 16. *Suppose that $|F_i - f_i(v_t)| \leq \frac{\lambda\epsilon}{n}$ for each i. Then, for any menu distribution z which realizes a choice distribution x for scores $\{F_i\}$, the choice distribution y for scores $\{f_i(v_t)\}$ satisfies $|x_i - y_i| \leq \frac{\epsilon}{n} x_i$.*

Proof. Let $y_i = \frac{f_i(v_t)}{F_i} x_i$ for each i. Observe that $(x, \{F_i\})$ and $(y, \{f_i(v_t)\})$ yield the same menu time values:

$$\frac{k \cdot \frac{x_i}{F_i}}{\sum_{j=1}^n \frac{x_j}{F_j}} = \frac{k \cdot \frac{y_i}{f_i(v_t)}}{\sum_{j=1}^n \frac{y_j}{f_j(v_t)}}.$$

By the first construction in Lemma 2, this implies that any menu distribution z realizing x under $\{F_i\}$ satisfies:

$$\frac{x_i}{F_i} = \sum_{K \in z : i \in K} \frac{z_K}{\sum_{j \in K} F_j}$$

and so the same distribution will satisfy

$$\begin{aligned}\sum_{K \in z : i \in K} \frac{z_K}{\sum_{j \in K} f_i(v_t)} &= \frac{\frac{f_i(v_t)}{F_i} x_i}{f_i(v_t)} \\ &= \frac{y_i}{f_i(v_t)}\end{aligned}$$

under $\{f_i(v_t)\}$, yielding a choice distribution of y. As such, we have that

$$\begin{aligned}|x_i - y_i| &\leq \left|\frac{F_i - f_i(v_t)}{F_i}\right| x_i \\ &\leq \frac{\epsilon}{n} x_i.\end{aligned}$$

□

Given a set of scores $M^* = \{F_i\}$, the set of feasible distributions can be expressed via linear constraints as

$$\text{IRD}(M^*) = \{x \in \Delta(n) : \frac{kx_i}{F_i} \leq \sum_{j=1}^{n} \frac{kx_j}{F_j}\}.$$

To ensure that we never remove points which belong to EIRD from our target set due to imprecision in IRD estimates, we can relax each target set by $\frac{\epsilon}{n}$ along each dimension:

$$\text{IRD}_\epsilon(M^*) = \left\{\left[\left(1 - \frac{\epsilon}{n}\right)x, \left(1 - \frac{\epsilon}{n}\right)x\right] \cap \Delta(n) : x \in \text{IRD}(M^*)\right\}.$$

However, such points will not be chosen by by our algorithm anyway, due to the ϵ-contraction from $\mathcal{K}_t$ to $\mathcal{K}_{t,\epsilon}$ in DBG. By the bound on ϵ, the total time spent updating our estimates F_i (counted by q) is at most ϵT, as at least

$$\frac{\lambda\epsilon}{4nL} T^c \geq \frac{64n^2(n-1)\log(2T/\delta)}{(k-1)\lambda^4\epsilon^3}$$

rounds elapse between each learning stage of length $Q \cdot \lceil \frac{n-1}{k-1} \rceil = \frac{64n^2(n-1)\log(2T/\delta)}{(k-1)\lambda^4\epsilon^2}$. As such, our total expected regret can be bounded by

$$\max_{x^* \in \text{EIRD}(M)} \sum_{t=1}^{T} r_t^\top x^* - \mathbb{E}\left[\sum_{t=1}^{T} r_t(i_t)\right] \leq \tilde{O}\left((n/\lambda)^{3/2} L^{1/4} T^{-c/4}\right)$$

assuming worst-case reward during each of the first T^c stages as well as the $q \leq \epsilon T$ stages spent

estimating F_i.

Case 2: $c > c^*$. When $c > c^*$ our analysis proceeds similarly, with the exception we can no longer uniformly bound the number of rounds in which $\|v_t - v_1\| \leq \frac{\lambda\epsilon}{2nL}$ between updates to F_i. However, we show that the amortized learning time is similar and that q can still be bounded by $\tilde{O}(T^{1-c/4})$. Let $c^*(t) = \min(c, \log(t)/\log(T))$ be the parameterization of c^* satisfying $t = T^{c^*(t)}$ for $c^*(t) \leq c$; note that we may still apply Lemma 15 according to $c^*(t)$ rather than c. At any $t \geq T^{3/4}$, $c^*(t) \geq 3/4$ suffices to bound the requisite change in v_t which occurs during the $Q \cdot \lceil \frac{n-1}{k-1} \rceil =$ learning rounds as we did in Case 1, as

$$\frac{\lambda^2\epsilon}{8nkL} \cdot T^{c^*(t)} \geq \frac{64n^2(n-1)\log(2T/\delta)}{(k-1)\lambda^4\epsilon^2}. \qquad (\epsilon^3 \geq 512n^4L\lambda^{-6}\log(2T/\delta)T^{-3/4})$$

For any range $t = [2^h, 2^{h+1}]$ for $h \geq \log(T^{3/4})$ and $h \leq \log(T^c)$, our bound on v_t holds for at least

$$\frac{\lambda\epsilon}{4nL} \cdot T^{c^*(t)} \geq \frac{\lambda\epsilon}{4nL} \cdot 2^h$$

rounds, and so at most $\frac{4nL}{\lambda\epsilon}$ learning stages occur during this window. For $h \geq \log(T^c)$ we can apply the same bound as in Case 1. Across all $h \leq \log(T)$ we have that

$$\begin{aligned} q &\leq \frac{4nL}{\lambda\epsilon} \cdot \frac{64n^2(n-1)\log(2T/\delta)}{(k-1)\lambda^4\epsilon^2}\log(T) \\ &\leq \epsilon T\log(T), \end{aligned}$$

yielding a total regret bound matching that in Case 1 up to a $\log(T)$ factor. □

B.5 Analysis for Algorithm 7: Targeting $\Delta^{\Phi}(n)$ for Scale-Bounded Preferences

Theorem 31. *9[Scale-Bounded Discounted Regret Bound] For any agent with a preference model M which is (σ, λ)-scale-bounded and $(\frac{\lambda}{\sigma}, L)$-smooth with $\sigma \leq \sqrt{n/(2k)}$, and with γ-discounted*

memory for $\gamma = 1 - \frac{1}{T^c}$ for $c \in (0,1)$, Algorithm 7 obtains regret

$$\max_{x^* \in \Delta^\phi(n)} \sum_{t=1}^{T} r_t^\top x^* - \mathbb{E}\left[\sum_{t=1}^{T} r_t(i_t)\right] = \tilde{O}\left((n^4 L\left(T^{1-c/2} + T^{1/2+c/2}\right)\right)$$

with respect to the ϕ-smoothed simplex, for $\phi = 4k\lambda\sigma^2$.

Algorithm 12 (Targeting Δ^ϕ for Scale-Bounded Preferences).

Let $\epsilon = \tilde{O}(n^4 L \cdot \max(T^{-c/2}, T^{c/2-1/2}))$, $S = \tilde{O}(n^{3/2}T^c)$, $\eta = \tilde{O}(n^{-1/2}L \cdot T^{c/2-1/2})$.
`- burn-in -`
for $t = 1$ to T^c **do**
  Show agent menu $K = \{1, \dots, k\}$
end for
while $\max_i \left|v_{t,i} - \frac{1}{n}\right| \geq \frac{\epsilon}{4n^2 L\sigma}$ **do**
  Show agent k items with smallest $v_{t,i}$, choosing randomly among ties up to T^{-c}
end while
`- initial learning -`
for T^c rounds **do**
  Let $F_{t,i} = \sigma^{-1}((1-\lambda)v_{t,i} + \lambda)$ if $v_{t,i} < \frac{1}{n}$, else $F_{t,i} = \sigma((1-\lambda)v_{t,i} + \lambda)$
  Let $z_t = \texttt{MenuDist}(v_t, \mathbf{u}_n, \{F_{t,i}\})$, show agent $K_t \sim z_t$
end for
Set $F_i = \sum_t \mathbf{1}(i_t = i) \cdot F_{t,i}/(\frac{1}{n}T^c)) \cdot C(\sum_{j=1}^n F_{t,j})^{-1}$, set $v^* = v_t$
`- optimization -`
Initialize OGD over $\Delta_\epsilon^\phi(n)$ for T/S rounds with η, set $x_1 = v^* := v_t$.
for $s = 1$ to T/S **do**
  Receive x_s from OGD
  for S rounds do **do**
    if $\|v_t - v^*\|_1 \geq \epsilon/(2nL)$ **then**
      Let $\tilde{v} = v_t$
      Show agent $K_t \sim z = \texttt{MenuDist}(v_t, \tilde{v}, \{F_i\})$ for T^c/L^2 rounds
      Set $F_i = \sum_t \mathbf{1}(i_t = i) \cdot F_i/(\tilde{v}_i T^c/L^2) \cdot C(\sum_{j=1}^n F_j)^{-1}$, set $v^* = v_t$
    end if
    Show agent $K_t \sim z_t = \texttt{MenuDist}(v^*, x_s, \{F_i\})$
  end for
  Set $\tilde{\nabla}_s = \sum_{h=t-S+1}^{t} e_{i_h} r_h(i_h)/(x_{h,i}S)$, update OGD
end for

Burn-in. Here we analyze the behavior of the algorithm in the initial $\tilde{O}(T^c)$ rounds, and show that each $v_{t,i}$ approaches $\frac{1}{n}$ with high probability. At $t = T^c$, the memory vector v_t is entirely concentrated on k items, and at least $n - k$ others have $v_{t,i} = 0$. We show that by showing the

k items with the smallest $v_{t,i}$ during each of the next $\tilde{O}(T^c)$ rounds, we reach a memory vector $v_t \in \left[\frac{1}{n} \pm \frac{\epsilon}{4n^2 L\sigma}\right]^n$ with high probability. Suppose $v_{t,i}$ at least T^{-c} below that for all but at most $k-1$ items, and each of the original $n-k$ smallest items have values $v_{t,j}$ within $\lambda/2$. Then, the probability that i is chosen in round t is at least

$$\begin{aligned} y_{t,i} &\geq \frac{\lambda}{k\sigma^2(\lambda + (1-\lambda)(\lambda/2))} \\ &\geq \frac{4}{3n}. \end{aligned}$$

As such, $v_{t,i}$ approaches $\frac{1}{n}$ faster than the average $v_{t,j}$ until it is no longer among the smallest k. For $\lambda > T^{-c/2}$ the probability of $v_{t,i}$ falling more than $\lambda/2$ (or any constant) below any of the other $n-k$ decays exponentially, and thus the expected value of each $v_{t,i}$ at $2T^c$ is $\frac{1}{n} + O(T^{-c})$ after enough rounds to decrease the memory weight of the first T^c rounds to $O(T^{-c})$, which occurs after $O(T^c \log(T^c))$ rounds following the analysis of Lemma [...14]. By Azuma's inequality, considering the martingale tracking the deviation between $v_{t,i}$ and its conditional expectation, this holds with high probability as well, up to $\frac{1}{n} + \tilde{O}(T^{-c/2}) = \pm\frac{\epsilon}{4n^2 L\sigma}$. The event that this fails to hold within $\tilde{O}(T^c)$ rounds contributes $O(1)$ to the total expected regret. Throughout the proof, we hide log terms and some constants inside $\tilde{O}$ notation.

Initial learning. Here we leverage the scale-bounded preference structure to obtain efficient estimators for scores near the current memory vector. With $v_t \in \left[\frac{1-\alpha}{n}, \frac{1+\alpha}{n}\right]^n$, for any $\lambda > 0$ and sufficiently small α we have

$$\frac{f_i(v_t)}{f_j(v_t)} \leq \frac{n(\lambda + (1-\lambda)\frac{1+\alpha}{n})}{2k(\lambda + (1-\lambda)\frac{1-\alpha}{n})} \leq \frac{n}{3k}$$

and so for any distribution $x \in \left[\frac{1}{2n}, \frac{3}{2n}\right]^n$ uniform distribution we have menu times

$$\mu_i \leq \frac{k \cdot 3n/2}{n \cdot 3k/2} \leq 1$$

for each i according to the true scores $f_i(v_t)$. Our aim here is to learn accurate estimates of each of these scores. Observe that each of our proposed set of scores $\{F_{t,i}\}$ satisfies the scale-bounded conditions, and contains $\mathbf{u}_n \in \text{IRD}(v_t, \{F_{t,i}\})$; as before, we can again take $v_t \in \left[\frac{1}{n} \pm \frac{\epsilon}{4n^2 L\sigma}\right]^n$ to hold with high probability over each of the T^c rounds, as every $v_{t,i}$ moves closer to $\frac{1}{n}$ in expectation every round (up to T^{-c}), and thus the martingale tracking maximum deviation of the memory vector from expectation in any round under this process is bounded by $\tilde{O}(T^{c/2})$. Given this, we can obtain an unbiased estimator of each $f_i(v_t)$; we initially assume that $\sum_i f_i(v_t) = \sum_i F_{t,i}$ for each trial, and will subsequently correct for this in our sample aggregation by renormalizing such that $\sum_i F_{t,i} = C$.

Lemma 17. *At any t, if sampling from a menu distribution which generates a choice distribution x_t at v_t according to scores $\{F_{t,i}\}$, with $\left|x_{t,i} - v_{t,i}\right| \leq \lambda$, then an unbiased estimator of each true preference score $f_i(v_t)$ is given by*

$$\mathbb{E}\left[\frac{F_{t,i}}{x_{t,i}} \cdot \mathbf{1}(i \text{ is chosen})\right] = f_i(v_t),$$

with range bounded by $\frac{n}{2k\sigma}$ if $x_t \in \Delta^\phi(n)$ and $F_{t,i}$ satisfies scale-bounded constraints for x_t at v_t.

Proof. Recalling Lemma 16, we have that the expected choice distribution y_t satisfies

$$y_{t,i} = \frac{f_i(v_t)x_{t,i}}{F_{t,i}},$$

and rearranging gives us the estimator, as both $x_{t,i}$ and $F_{t,i}$ here are fixed. We can bound the range using the properties of scale-bounded functions and conditions for ϕ:

$$\begin{aligned}
\frac{F_{t,i}}{x_{t,i}} &\leq \frac{\sigma(\lambda + (1-\lambda)(x_i + \lambda))}{x_i} \\
&\leq \frac{\sigma(\lambda n + (1-\lambda)4k\lambda\sigma^2)}{4k\lambda\sigma^2} && (x_i \geq \phi/n \geq 4k\lambda\sigma^2/n) \\
&\leq \frac{n}{2k\sigma}.
\end{aligned}$$

□

By the scale-bounded condition, the quantities $\sum_i f_i(v_t)$ and $\sum_i F_{t,i}$ are within a factor of σ^2, and so renormalizing to C can only increase each estimator's range to $\frac{n\sigma}{2k}$. Normalizing the assumed sum of each trial $\sum_i F_{t,i}$ to C, as we do when aggregating to estimate F_i, yields a sum of random variables, each of whose mean is $f_i(\mathbf{u}_n)$. Applying Azuma's inequality to the sequence of trials for $T^c \geq \tilde{O}(n^8 L^2/\epsilon^2) \geq \tilde{O}(n^6\sigma^4/(k\epsilon)^2)$ suffices to yield $|f_i(v^*) - F_i| \leq \frac{\epsilon}{4n^2\sigma}$; additionally, given the Lipschitz condition on f_i and that $v^* \in \left[\frac{1}{n} \pm \frac{\epsilon}{4n^2 L\sigma}\right]^n$, we also have that $|f_i(\mathbf{u}_n) - F_i| \leq \frac{\epsilon}{2n^2\sigma}$ as well. In the following rounds, this will allow us to accurately target any distribution in $\mathrm{IRD}(v^*)$ whenever $\|v^* - v_t\| \leq \frac{\epsilon}{2n^2 L\sigma}$.

Lemma 18. *Suppose we have estimates F_i which satisfy the scale-bounded constraints for x_t at v_t, with $\left|x_{t,i} - v_{t,i}\right| \leq \lambda$, and further that $|f_i(v_t) - F_i| \leq \frac{\epsilon}{\sigma n^2}$ for each i. Then, the generated distribution y_t when targeting x_t according to $\{F_{t,i}\}$ satisfies $\|y_t - x_t\|_1 \leq \frac{\epsilon}{n}$.*

Proof. The generated distribution is given by

$$y_i = x_i + \frac{f_i(v_t) - F_i}{F_i} x_i$$

and so

$$|y_i - x_i| \leq \frac{\epsilon}{\sigma n} \cdot \frac{x_i}{F_i} \leq \frac{\epsilon}{n}.$$

□

Online gradient descent. We treat each batch of S rounds as a single step for OGD (with x_s as the point chosen by OGD), and show that each of the following invariants is maintained across every step with high probability:

1. We complete each learning stage with estimates F_i satisfying $|f_i(v_t) - F_i| \leq \frac{\epsilon}{2\sigma n^2}$;
2. $|f_i(v_t) - F_i| \leq \frac{\epsilon}{\sigma n^2}$ holds in every round where x_s is targeted;
3. Each gradient estimate satisfies $\left\|\tilde{\nabla}_t - r_s\right\|_2 \leq \frac{4\epsilon}{n}$, $\left\|\tilde{\nabla}_t\right\|_2 \leq 2\sqrt{n}$, and $\mathbb{E}\left[\left\|\tilde{\nabla}_s\right\|^2\right] \leq O(n^2)$.

4. We begin each step with $\|v_t - x_s\|_1 \leq \frac{4\epsilon}{n}$;

5. We complete each step with $\|v_t - x_s\|_1 \leq \frac{2\epsilon}{n}$;

6. The expected choice distribution x_t in each round of the step satisfies $\|x_t - x_s\|_1 \leq \frac{4\epsilon}{n}$ and $x_t \in \Delta^{\phi}(n)$.

(1.) This holds whenever $\|v^* - v_t\| \leq \frac{\epsilon}{2n^2L\sigma}$, provided that each update results in $|f_i(v^*) - F_i| \leq \frac{\epsilon}{2n^2\sigma}$ at the time of completion. The latter follows along the lines of the initial learning stage; while we may initially have accuracy of only $\frac{\epsilon}{n^2\sigma}$ accuracy, the learning occurs within $\tilde{O}(n^8/\epsilon^2) \leq T^c/L^2$ rounds to a target accuracy of $\frac{\epsilon}{4\sigma n^2}$ for $\tilde{v}$, and so the total drift throughout learning can be bounded by $\frac{\epsilon}{4n^2L\sigma}$ as the fraction of memory in which the drift applies is bounded by Lemma 15 (assuming a sufficiently large constant lower bound for L, without loss of generality). This further implies the desired accuracy for F_i at at the updated v^*.

(2.) This holds given that we re-learn whenever $\|v^* - v_t\| \geq \frac{\epsilon}{2n^2L\sigma}$.

(3.) Each x_s lies within $\Delta^{\phi}_{\epsilon}(n)$, and any total drift of the memory vector outside of $\Delta^{\phi}_{\epsilon}(n)$ can be bounded by $\frac{\epsilon}{2n} \cdot v_t$, and so our target distribution always lies within $\Delta^{\phi}(n)$, as well as $\mathrm{IRD}(v_t, M)$ for $\lambda\phi \gg \epsilon$ by Lemma 9. Given (6.), we will have that $\mathbb{E}\left[\left\|\tilde{\nabla}_s - r_s\right\|\right] \leq \frac{4\epsilon}{\sqrt{n}}$, similarly to as in the analysis of DBG, where $r_s = \sum_S r_t/S$. Further, with each $x_t \geq \frac{\phi}{n} \geq \frac{\epsilon}{n}$, we will also have $\mathbb{E}\left[\left\|\tilde{\nabla}_s\right\|^2\right] \leq O(n^2)$ as we had for DBG as well, with a norm bound of $2\sqrt{n}$ holding with high probability.

(4.) This holds as $\left\|\tilde{\nabla}_s\right\|_2 \eta \leq 2\sqrt{n}\eta \leq \frac{\epsilon}{2n}$.

(5.) When $\|v_t - x_s\|_2 \leq \frac{4\epsilon}{n}$ but $\|v_t - x_s\|_2 \geq \frac{2\epsilon}{n}$, our time spent targeting x_s (up to $\frac{\epsilon}{n}$, by Lemma 18) is sufficient to decrease the distance by at least $\frac{3\epsilon}{8n^2L\sigma}$ with high probability before drifting more than $\frac{\epsilon}{2n^2L\sigma}$ from v^*, and following the potential drift of $\frac{\epsilon}{4n^2L\sigma}$ during re-learning each F_i, we remain

closer by $\frac{\epsilon}{8n^2L\sigma}$ with high probability. Thus, at most $16n\sigma$ stages are needed to reach within $\frac{2\epsilon}{n}$ from x_s, which holds $S = \tilde{O}(n^{3/2}T^c)$ and $\tilde{O}(T^c/L^2)$ total between each update to v^*.

(6.) This follows from the accuracy guarantees of each learning stage as well as the drift bounds applied to each $\tilde{v}$.

Regret bound. Given each of these, we can analyze our regret in accordance with the bounds for Online Gradient Descent, as well as the error resulting from the above approximations to an exact execution of OGD. This gives us a total regret bound for our algorithm of:

$$\begin{aligned}
\mathsf{Reg}_{\Delta^\phi(n)} = \quad & \underbrace{S\left(\eta \cdot \sum_{s=1}^{T/S} \left\|\tilde{\nabla}_s\right\|^2 + \frac{\sqrt{2}}{\eta}\right)}_{\text{OGD regret over } \Delta_\epsilon^\phi(n),\, T/S \text{ steps}} + \underbrace{\sum_{s=1}^{T/S}\sum_{h=t}^{t+S-1} \sqrt{n}\left(\left\|r_s - \tilde{\nabla}_s\right\| + \|x_h - x_s\|\right)}_{\text{gradient and reward error}} \\
& + \underbrace{\left(\max_{\Delta^\phi(n)\times\Delta_\epsilon^\phi(n)} \sum_t r_t^\top x^* - \sum_t r_t^\top x\right)}_{\text{target set imprecision vs. } \Delta_n^\phi} + \underbrace{\tilde{O}\left(T^c\right)}_{\text{burn-in and initial learning}} \\
= \; & \tilde{O}\left(n^2T^{1/2+c/2} + n^{7/2}L(T^{1/2+c/2} + T^{1-c/2}) + n^4LT^{1-c/2} + T^c\right).
\end{aligned}$$

Appendix C: Deferred Proofs from Chapter 4

C.1 Follow the Regularized Leader

Here we state the FTRL algorithm and several of its key properties; see e.g. [127] for proofs of Propositions 14 and 15.

Algorithm 13 Follow the Regularized Leader (FTRL)

Choose a time horizon T, step size η, and γ-strongly convex regularizer $\psi : \mathcal{Y} \to \mathbb{R}$
Let $y_1 = \operatorname{argmin}_{y \in \mathcal{Y}} \psi(y)$
for $t = 1$ to T **do**
  Play y_t and observe loss $f_t(y_t)$
  Set $\nabla_t = \nabla f_t(y_t)$
  Set $y_{t+1} = \operatorname{argmin}_{y \in \mathcal{Y}} \left(\eta \cdot \sum_{s=1}^{t} \nabla_s^\top y + \psi(y) \right)$
end for

Proposition 14. *For a γ-strongly convex regularizer $\psi : \mathcal{Y} \to \mathbb{R}$ where $|\psi(y) - \psi(y')| \le G$ for all $y, y' \in \mathcal{Y}$, and for convex L-Lipschitz losses $f_1, \ldots, f_T$, the regret of* FTRL *is bounded by*

$$\mathsf{Reg}_T(\text{FTRL}) \le \eta \frac{TL^2}{\gamma} + \frac{G}{\eta}.$$

Proposition 15. *Any pair of points y_t and y_{t+1} chosen by* FTRL *satisfies $\|y_{t+1} - y_t\| \le \eta \frac{L}{\gamma}$.*

C.2 Algorithms for Adversarial Disturbances

C.2.1 NestedOCO-BD and Proofs for Theorem 22

We show that it is possible simulate NestedOCO over the undisturbed states $\hat{y}_t$ under the assumption that the dynamics are in $\alpha\rho$-locally controllable for some $\alpha \in (0, 1)$ while retaining sufficient range in the feasible region around y_t to correct for the disturbance w_{t-1} from the previous

round. Here, the oracle call for computing x_t in each round is updated to consider the true state y_{t-1}.

Algorithm 14 OEN-FTRL with Adversarial Disturbances (NestedOCO-BD).

Initialize NestedOCO for T rounds over $(\mathcal{X}, \mathcal{Y}, D)$ for $\alpha\rho$-locally controllable dynamics
for $t = 1$ to T **do**
  Let $\hat{y}_t$ be the target state chosen by NestedOCO
  Use $\texttt{Oracle}(y_{t-1}, \hat{y}_t)$ to compute $x_t = \operatorname{argmin}_{x \in \mathcal{X}} \|D(x, y_{t-1}) - \hat{y}_t\|^2$
  Play action x_t.
  Observe disturbed state $y_t = \hat{y}_t + w_t$ and loss $f_t(y_t)$.
  Update NestedOCO with state $\hat{y}_t$ and loss $f_t(\hat{y}_t)$.
end for

Theorem 22 follows directly from Theorems 32, 33, and 34. Intuitively, when the per-round disturbance magnitude is at most $\frac{\rho - \alpha\rho}{1+\rho} \cdot \pi\left(D(x_t, y_{t-1})\right)$, one can calibrate NestedOCO for the case of $\alpha\rho$-locally controllable dynamics and maintain sufficient "slack" to correct for the previous round's disturbance in every round. When disturbances exceed $\frac{\rho}{1+\rho} \cdot \pi\left(D(x_t, y_{t-1})\right)$, an adversary can continually push the state towards the boundary of $\mathcal{Y}$, which may require vanishing disturbance magnitude as rounds progress due to the limited range promised by local controllability near the boundary.

Theorem 32. *For a ρ-locally controllable instance $(\mathcal{X}, \mathcal{Y}, D)$ with convex losses $f_t : \mathcal{Y} \to \mathbb{R}$ and adversarial disturbances w_t where $\|w_t\| \le \frac{\rho - \alpha\rho}{1+\rho} \cdot \pi\left(D(x_t, y_{t-1})\right)$ and $\sum_{t=1}^{T} \|w_t\| \le E$, the regret of* NestedOCO-BD *with respect to the reward of any state is bounded by*

$$\mathsf{Reg}_T(\text{NestedOCO-BD}) \le O\left(\sqrt{T \cdot (\alpha\rho)^{-1}} + E\right),$$

with T queries made to an oracle for non-convex optimization.

Proof. We show by induction that each call to $\texttt{Oracle}(y_{t-1}, \hat{y}_t)$ yields a feasible action x_t satisfying $\hat{y}_t = D(x_t, y_{t-1})$. This is immediate for $t = 1$, and suppose this holds up to some round $t - 1$, where we have that $y_{t-1} = \hat{y}_{t-1} + w_{t-1}$. Given that NestedOCO selects actions under $\alpha\rho$-local

controllability, we can bound

$$\|\hat{y}_t - \hat{y}_{t-1}\| \leq \alpha\rho \cdot \pi(\hat{y}_{t-1}).$$

Further, the magnitude of the disturbance w_{t-1} is bounded by

$$\|w_{t-1}\| \leq \frac{\rho - \alpha\rho}{1+\rho} \cdot \pi(\hat{y}_{t-1}),$$

yielding that

$$\begin{aligned}
\|\hat{y}_t - y_{t-1}\| &\leq \|\hat{y}_t - \hat{y}_{t-1} - w_{t-1}\| \\
&\leq \left(\alpha\rho + \frac{\rho - \alpha\rho}{1+\rho}\right) \cdot \pi(\hat{y}_{t-1}). && (y_{t-1} = w_{t-1} + \hat{y}_{t-1})
\end{aligned}$$

As such, we have that

$$\begin{aligned}
\rho \cdot \pi(y_{t-1}) &\geq \rho\left(1 - \frac{\rho - \alpha\rho}{1+\rho}\right) \cdot \pi(\hat{y}_{t-1}) \\
&= \rho\left(\alpha + \frac{1-\alpha}{1+\rho}\right) \cdot \pi(\hat{y}_{t-1}),
\end{aligned}$$

and so by ρ-local controllability some feasible action x_t exists, as $\hat{y}_t$ lies in $\mathcal{B}_{\rho\cdot\pi(y_{t-1})}$. The regret bound for NestedOCO holds over the states $\hat{y}_t$, and so we can bound the total regret of NestedOCO-BD with respect to any $y^* \in \mathcal{Y}$ as:

$$\begin{aligned}
\sum_{t=1}^{T} f_t(y_t) - f_t(y^*) &\leq \sum_{t=1}^{T} f_t(\hat{y}_t) - f_t(y^*) + L\|y_t - \hat{y}_t\| \\
&\leq \mathsf{Reg}_T(\text{OEN-FTRL}) + L\sum_{t=1}^{T}\|w_t\| && \text{(Thm. 21)} \\
&\leq 2\sqrt{\frac{(1+\frac{R}{r\alpha\rho})TGL^2}{\gamma}} + LE.
\end{aligned}$$

□

We show that the dependence on E is tight up to the constant. Note that we we can obtain regret $O(\sqrt{T \cdot (\alpha\rho)^{-1}}) + LE$ in the following instance via NestedOCO-BD.

Theorem 33. *Suppose for any $\alpha > 0$ and $\rho \in (0, 1]$ an adversary can choose w_t with $\|w_t\| \leq \frac{\rho-\alpha\rho}{1+\rho} \cdot \pi\left(D(x_t, y_{t-1})\right)$, where $\sum_{t=1}^{T} \|w_t\| = E$ for any E. There is a ρ-locally controllable instance $(\mathcal{X}, \mathcal{Y}, D)$ with L-Lipschitz convex losses f_t such that any algorithm $\mathcal{A}$ obtains regret $\mathsf{Reg}_T(\mathcal{A}) \geq \max(LE, \frac{\rho-\alpha\rho}{1+\rho} TL)$.*

Proof. Consider any norm $\|\cdot\|$ over $\mathbb{R}^n$. Let $\mathcal{Y}$ be the unit ball $B_1(\mathbf{0})$, and let each $f_t(y_t) = L\|y_t\|$. Consider any action space $\mathcal{X}$ and dynamics D where ρ-local controllability exactly characterizes the range of D, i.e. for any y and y', there is some x such that $D(x, y) = y'$ if and only if $y' \in \mathcal{B}_{\rho \cdot \pi(y)}(x, y)$.

First, note that $\pi(y) = 1 - \|y\|$ for any $y \in \mathcal{Y}$. In each round t , suppose an algorithm plays an action x_t at state y_{t-1} which yields an target undisturbed update $\hat{y} = D(x_t, y_{t-1})$. The adversary can then choose any w_t satisfying $\|w_t\| \leq \frac{\rho-\alpha\rho}{1+\rho} \cdot (1 - \|\hat{y}_t\|)$; suppose each w_t is given by

$$w_t = \hat{y}_t \cdot \frac{\frac{\rho-\alpha\rho}{1+\rho} \cdot (1 - \|\hat{y}_t\|)}{\|\hat{y}_t\|}$$

if $\hat{y}_t$ is non-zero, and an arbitrary vector w_t with $\|w_t\| = \frac{\rho-\alpha\rho}{1+\rho}$ if $\hat{y}_t = \mathbf{0}$. This satisfies the disturbance norm bound, and further yields $y_t = \hat{y}_t + w_t$, where for non-zero $\hat{y}$ we have

$$y_t = \hat{y}_t \cdot \left(1 + \frac{\frac{\rho-\alpha\rho}{1+\rho} \cdot (1 - \|\hat{y}_t\|)}{\|\hat{y}_t\|}\right)$$

and thus for any $\hat{y}$,

$$\begin{aligned} \|y_t\| &\geq \|\hat{y}_t\| + \frac{\rho - \alpha\rho}{1+\rho} \cdot (1 - \|\hat{y}_t\|) \\ &\geq \frac{\rho - \alpha\rho}{1+\rho}, \end{aligned}$$

yielding a loss $f_t(y_t) \geq L \cdot \frac{\rho-\alpha\rho}{1+\rho}$ at a disturbance cost of $\|w_t\| = \frac{\rho-\alpha\rho}{1+\rho}(1 - \|\hat{y}_t\|)$. Assuming the

adversary continues this strategy in each round until any disturbance budget $E = \sum_{t=1}^{T} \|w_t\|$ is exhausted, this yields a regret for any algorithm of at least

$$\mathsf{Reg}_T(\mathcal{A}) \geq \min\left(LE, \frac{\rho - \alpha\rho}{1+\rho} TL\right),$$

as $y^* = \mathbf{0}$ obtains total loss 0. □

The disturbance upper bound is indeed necessary for ρ-locally controllable dynamics. We show a sharp threshold effect at $\frac{\rho}{1+\rho} \cdot \pi(D(x_t, y_{t-1}))$, wherein an adversary who is allowed to exceed this limit by any amount can force an algorithm to incur linear regret even with only a constant budget. Note that for any $\rho \in (0, 1]$ and $\alpha < 0$, there is some $\beta \in [0, 1)$ such that $\frac{\rho - \alpha\rho}{1+\rho} \geq \frac{\rho}{1+\beta\rho}$.

Theorem 34. *Suppose an adversary can choose any state disturbances w_t with $\|w_t\| \leq \frac{\rho}{1+\beta\rho} \cdot \pi\left(D(x_t, y_{t-1})\right)$, for any $\rho \in (0, 1]$ and any $\beta \in [0, 1)$. Then, there is a ρ-locally controllable instance $(\mathcal{X}, \mathcal{Y}, D)$ with convex losses f_t such that any algorithm $\mathcal{A}$ obtains regret $\mathsf{Reg}_T(\mathcal{A}) = \Theta(T)$ even if $\sum_{t=1}^{T} \|w_t\| = O(1)$.*

Proof. Consider any instance $(\mathcal{X}, \mathcal{Y}, D)$ where ρ-local controllability exactly characterizes the range of D, i.e. for any y and y', there is some x such that $D(x, y) = y'$ if and only if $y' \in \mathcal{B}_{\rho \cdot \pi(y)}(x, y)$.

Let $d_t = \pi(y_t)$ for each round. Beginning at any round t, suppose the adversary observes an action x_t which yields an update $\hat{y}_t = D(x_t, y_{t-1})$. Let $z_t = \operatorname{argmin}_{y \in \mathrm{bd}(\mathcal{Y})} \|y - \hat{y}_t\|$, and suppose the adversary chooses the disturbance:

$$w_t = \operatorname*{argmin}_{w : \|w\| \leq \frac{\rho}{1+\beta\rho} \cdot \pi(\hat{y}_t)} \|\hat{y}_t + w_t - z_t\| .$$

This forces y_t closer to the boundary at each round, regardless of the choice of x_t:

$$\begin{aligned} d_t &= \left(1 - \frac{\rho}{1+\beta\rho}\right) \cdot \pi(\hat{y}_t) \\ &\leq \left(1 + \rho - \frac{\rho}{1+\beta\rho} - \frac{\rho^2}{1+\beta\rho}\right) d_{t-1} && (\pi(\hat{y}_t) \leq (1+\rho)d_{t-1}) \\ &\leq \frac{1 + \beta\rho + \beta\rho^2 - \rho^2}{1+\beta\rho} d_{t-1} \\ &\leq \left(1 - \frac{(1-\beta)\rho^2}{1+\beta\rho}\right) d_{t-1}, \end{aligned}$$

where $\pi(\hat{y}_t) \leq (1+\rho)d_{t-1}$ holds by our assumption on $D(x, y)$. Assuming the adversary applies a disturbance w_t selected as above in each round $t \leq T$, we have that

$$d_t \leq \left(1 - \frac{(1-\beta)\rho^2}{1+\beta\rho}\right)^t \cdot d_0,$$

where the magnitude of each disturbance is bounded by

$$\begin{aligned} \|w_t\| &\leq \frac{\rho + \rho^2}{1+\beta\rho} d_{t-1} \\ &\leq \frac{\rho + \rho^2}{1+\beta\rho} \left(1 - \frac{(1-\beta)\rho^2}{1+\beta\rho}\right)^{t-1} \cdot d_0, \end{aligned}$$

where we take the initial state distance to the boundary $d_0 = \pi(y_0)$ to be a constant bounded away from zero. This yields that the sum of disturbance magnitudes $E = \sum_{t=1}^{T} \|w_t\|$ is at most:

$$\begin{aligned} \sum_{t=1}^{T} \|w_t\| &\leq d_0 \frac{\rho + \rho^2}{1+\beta\rho} \cdot \sum_{t=1}^{T} \left(1 - \frac{(1-\beta)\rho^2}{1+\beta\rho}\right)^{t-1} \\ &\leq d_0 \cdot \frac{\rho + \rho^2}{(1-\beta)\rho^2} \\ &= O(1). \end{aligned}$$

Now suppose that the loss at each round is given by $f_t(y_t) = \|y_t - y_0\|$. Then, our regret with

respect to y_0 is at least:

$$\begin{aligned}
\sum_{t=1}^{T} f_t(y_t) - f_t(y_0) &\leq \sum_{t=1}^{T} d_0 - d_t \\
&\leq d_0 \left(T - \sum_{t=1}^{T} \frac{(1-\beta)\rho^2}{1+\beta\rho} \right) \\
&\leq d_0 \left(T - \frac{1 - \frac{(1-\beta)\rho^2}{1+\beta\rho}}{\frac{(1-\beta)\rho^2}{1+\beta\rho}} \right) \\
&\leq d_0 \left(T - \frac{1+\beta\rho}{(1-\beta)\rho^2} \right) \\
&= \Theta(T).
\end{aligned}$$

□

Together, the previous three theorems yield Theorem 22.

C.2.2 NestedOCO-UD and Proofs for Theorem 23

We can remove the bound on the maximum disturbance for strongly locally controllable instances, as the feasible update sets do not vanish at the boundary of $\mathcal{Y}$. Recall that an instance $(\mathcal{X}, \mathcal{Y}, D)$ satisfies strong ρ-local controllability for $\rho > 0$ if, for any $y \in \mathcal{Y}$ and $y^* \in \mathcal{B}_\rho(y) \cap \mathcal{Y}$, there is some x such that $D(x, y) = y^*$. We assume without loss of generality that $\rho \leq 2R$, where R is the radius of $\mathcal{Y}$.

Intuitively, our algorithm tracks the target state which would be chosen by FTRL in the absence of all disturbances (by recording the loss counterfactual loss rather than the one truly experienced), and always seeks to minimize distance to that state.

Theorem 35. *For a strongly ρ-locally controllable instance $(\mathcal{X}, \mathcal{Y}, D)$ with convex losses $f_t : \mathcal{Y} \to \mathbb{R}$ and adversarial disturbances w_t where $\sum_{t=1}^{T} \|w_t\| \leq E$, the regret of* NestedOCO-UD *is*

Algorithm 15 NestedOCO with Unbounded Disturbances (NestedOCO-UD).

Initialize FTRL for T rounds over $\mathcal{Y}$ with step size $\eta = \sqrt{\frac{G\gamma}{TL^2}}$.
for $t = 1$ to T **do**
  Let $\hat{y}_t$ be the target state chosen by FTRL.
  Use $\texttt{Oracle}(y_{t-1}, \hat{y}_t)$ to compute $x_t = \operatorname{argmin}_{x \in \mathcal{X}} \|D(x, y_{t-1}) - \hat{y}_t\|^2$.
  Play action x_t.
  Observe disturbed state $y_t = D(x_t, y_{t-1}) + w_t$ and loss $f_t(y_t)$.
  Update FTRL with state $\hat{y}_t$ and loss $f_t(\hat{y}_t)$.
end for

bounded by

$$\mathsf{Reg}_T(\text{NestedOCO-UD}) \leq O\left(\sqrt{T} + E \cdot \rho^{-1}\right)$$

with respect to the reward of any state, with T queries made to an oracle for non-convex optimization.

Proof. We begin by bounding the total state error $\sum_{t=1}^{t} \|y_t - \hat{y}_t\|$ across rounds. First, note that for any fixed $\rho > 0$, and any desired $\alpha \in (0, 1)$, we have that $\eta\frac{L}{\gamma} \leq \rho\alpha$ for sufficiently large T, as $\eta\frac{L}{\gamma} = \sqrt{\frac{G}{T\gamma}}$; we assume this holds for any given choice of α, and so we have that $\|\hat{y}_{t+1} - \hat{y}_t\| \leq \rho\alpha$ by Proposition 15. For a total disturbance budget E, we separately consider disturbances w_t depending on whether or not the accumulated disturbance error up to w_t is driven to 0 in the next round. Define W_+ and W_- as:

$$W_+ = \{w_t : D(x_{t+1}, y_t) \neq \hat{y}_{t+1}\}$$

and

$$W_- = \{w_t : D(x_{t+1}, y_t) = \hat{y}_{t+1}\}$$

with $E_+ = \sum_{w_t \in W_+} \|w_t\|$ and $E_- = \sum_{w_t \in W_-} \|w_t\|$. First, observe that at each round t corresponding to $w_t \in W_-$, given that $\|\hat{y}_{t+1} - y_t\| \leq \rho$ we have that $\|w_t\| = \|y_t - \hat{y}_t\| \leq (1+\alpha)\rho$, as $\|\hat{y}_{t+1} - \hat{y}_t\| \leq \alpha\rho$.

As such, we have that

$$\begin{aligned}\sum_{t:w_t \in W_-} f_t(y_t) - f_t(\hat{y}_t) &\leq \sum_{t:w_t \in W_-} L \left\| y_t - \hat{y}_t \right\| \\ &\leq (1+\alpha) L E_-.\end{aligned}$$

Next, consider any $w_t \in W_+$. As our instance is strongly ρ-locally controllable, we must have that $\|\hat{y}_{t+1} - y_t\| > \rho$, as otherwise there would some feasible action x_{t+1} which would be selected that would yield $w_t \in W_-$. Since $\|\hat{y}_{t+1} - \hat{y}_t\| \leq \alpha\rho$, it then must be the case that $\|w_t\| = \|y_t - \hat{y}_t\| > (1-\alpha)\rho$, and so we can bound the number of disturbances in W_+ as:

$$|W_+| \leq \frac{E_+}{(1-\alpha)\rho}.$$

Assuming a maximal distance $\|\hat{y}_t - y_t\| = 2R$ for each round t corresponding to some $w_t \in W_+$, this yields

$$\begin{aligned}\sum_{t:w_t \in W_+} f_t(y_t) - f_t(\hat{y}_t) &\leq \sum_{t:w_t \in W_+} L \left\| y_t - \hat{y}_t \right\| \\ &\leq \frac{2LRE_+}{(1-\alpha)\rho}\end{aligned}$$

We can assume α is small enough to yield $\frac{2R}{\rho} \geq (1+\alpha) \cdot (1-\alpha)$, and so we have

$$\sum_{t=1}^{T} f_t(y_t) - f_t(\hat{y}_t) \leq \frac{2LRE}{(1-\alpha)\rho}.$$

The regret bound for FTRL holds over the states $\hat{y}_t$, and so we can bound the total regret of

NestedOCO-BD with respect to any $y^* \in \mathcal{Y}$ as:

$$\begin{aligned}\sum_{t=1}^{T} f_t(y_t) - f_t(y^*) &\leq \sum_{t=1}^{T} f_t(\hat{y}_t) - f_t(y^*) + \sum_{t=1}^{T} f_t(y_t) - f_t(\hat{y}_t) \\ &\leq \eta \frac{TL^2}{\gamma} + \frac{G}{\eta} + \frac{2LRE}{(1-\alpha)\rho} \qquad \text{(Prop. 14)} \\ &\leq 2\sqrt{\frac{TGL^2}{\gamma}} + \frac{2LRE}{(1-\alpha)\rho}.\end{aligned}$$

□

Theorem 36 (Regret Lower Bound for Unconstrained Adversarial Disturbances)**.** *Suppose an adversary can choose any state disturbances w_t with $\sum_{t=1}^{T} \|w_t\| = E$. For any $\rho \in (0, 1]$, there is a strongly ρ-locally controllable instance $(\mathcal{X}, \mathcal{Y}, D)$ with convex losses f_t such that any algorithm $\mathcal{A}$ obtains regret $\mathsf{Reg}_T(\mathcal{A}) = \min(\frac{2LRE}{\rho}, 2TLR)$.*

Proof. Let $\mathcal{Y} = [-R, R]$ for any $R > 0$ and let $f_t(y_t) = -Ly_t + LR$ for each y. Suppose strong ρ-local controllability exactly characterizes the range of D, i.e. for any $y, y' \in \mathcal{Y}$ there is some x such that $D(x, y) = y'$ if and only if $|y - y'| \leq \rho$. Consider an adversary who chooses disturbances w_t in each round such that $y_t = -R$ until their disturbance budget E is exhausted. This requires a disturbance of magnitude at most $R + \rho$ for w_1, as we assume $y_0 = 0$, and at most ρ in subsequent rounds, and thus the adversary can force any algorithm to remain at $y_t = -R$ for $(E - R)\rho^{-1}$ rounds.

As such, any algorithm must incur loss of at least $2LR(E - R)\rho^{-1}$ across these rounds, and further must incur average loss LR over the subsequent $2R\rho^{-1}$ rounds (if T is not yet reached), for an additional loss of $2LR^2\rho^{-1}$, as they can only decrease per-round loss by $L\rho$ given the restriction on the range of D. As the optimal state $y^* = R$ obtains loss 0, the total regret is at least:

$$\sum_{t=1}^{T} f_t(y_t) - f_t(y^*) \geq \min\left(\frac{2LRE}{\rho}, 2TLR\right).$$

□

Together, the previous two theorems yield Theorem 23. Note that for both algorithms it

remains computationally efficient to optimize over action-linear dynamics, as the constraint that $D(x, y_{t-1}) \in \mathcal{Y}$ can be encoded as a convex contraint over $\mathcal{X}$.

C.3 Background and Proofs for Section 4.4.1: Performative Prediction

C.3.1 Background

Introduced by [69], the Performative Prediction problem captures settings in which the data distribution for which a classifier is deployed may shift as a function of the classifier itself, notably including strategic classification [80] as well as problems related to reinforcement learning and causal inference. While a number of extensions of strategic classification to online settings have been considered [81, 117, 128], the bulk of the literature on performative prediction considers settings with a fixed loss function and distribution "update map" [69, 129, 82, 70, 130, 91], where the update map may sometimes depend on the current distribution (as in the Stateful Performative Prediction setting of [91]). For the *location-scale* family of update maps introduced by [129] (and additionally explored by [82] from a regret minimization perspective), which yields a convex "performative risk" objective function, a formulation of Online Performative Prediction is given by [93] as an application of online convex optimization with unbounded memory, in which the classification loss function may change over time and the distribution updates may occur gradually.

Here, we generalize the problem formulation of [93] to also accommodate notions of statefulness similar to that in [91]. In particular, the instances we consider will resemble location-scale maps when restricting attention only the performatively stable classifiers for each distribution, yet the update effect of a non-stable classifier may be distribution-dependent and nonlinear, provided that the update map satisfies local controllability (viewing classifiers as actions and distributions as states) and mild regularity properties (e.g. invertibility and Lipschitz conditions).

C.3.2 Model

In the setting of Online Performative Prediction we consider, as formulated by [93], in each round $t \in [T]$ we deploy some classifier x_t, and observe samples from some distribution p_t, which

may change dynamically as a function of the history of interactions. Here, we take $\mathcal{X} \subseteq \mathbb{R}^n$ as our space of classifiers, e.g. representing weight vectors for regression, which we assume is bounded and convex. The initial data distribution is given by some distribution p_0 over $\mathbb{R}^n$. In each round, upon deploying a classifier x_t, the distribution is updated according to

$$p_t = (1 - \theta)p_{t-1} + \theta\, \mathcal{D}(x_t, y_{t-1}),$$

for $\theta \in (0, 1]$, where $\mathcal{D}(x_t, y_{t-1})$ is the distribution *update map* taking as input our classifier x_t and some representation of the *state* $y \in \mathcal{Y}$, where we assume $\mathcal{Y} \subseteq \mathbb{R}^n$ is convex, contains $\mathcal{B}_r(\mathbf{0})$, is bounded with radius R, and that $y_0 = 0$. We make the following assumptions on $\mathcal{D}$.

Assumption 3. *We assume the distribution update map $\mathcal{D}(x, y)$ operates as follows:*

- $\mathcal{D}(x, y) = A(x, y) + \xi$, *with* $A : \mathcal{X} \times \mathcal{Y} \to \mathcal{Y}$,
- ξ *is a random variable in* $\mathbb{R}^n$ *with mean* μ *and covariance* Σ,
- $A(x, y)$ *satisfies* ρ*-local controllability and has an inverse action mapping* $X(y, y^*)$ *where*

$$A(X(y, y^*), y) = y^*,$$

  defined over feasible pairs, which is L_y*-Lipschitz in* y *(when feasibility of* y^* *holds), and*

- *There is a linear invertible function* $s : \mathcal{X} \to \mathcal{Y}$ *such that* $A(x, y) = s(x)$ *if* $y = s(x)$, *where* $s^{-1} : \mathcal{Y} \to \mathcal{X}$ *is S-Lipschitz.*

Further, $A(x, y)$ *is known and* ξ *can be sampled freely.*

The inverse action mapping assumption simply enforces that classifiers need not change drastically to have the same update effect under small changes to the state. The final assumption imposes a linear structure over *performatively stable* classifiers (i.e. classifiers for which the resulting distribution will remain fixed under $\mathcal{D}$, as formulated by [69]), but we note that the distribution may update

in an arbitrarily nonlinear fashion (subject to the other conditions) when x_t is not a performatively stable classifier for the distribution induced by the previous state y_{t-1}. The ability to accommodate a state component is reminiscent of prior work involving notions of statefulness in performative prediction such as [91]. Our setting generalizes that of [93], in which the map A is taken to be a fixed matrix. For any nonsingular matrix A there is immediately a linear map $s(x) = A^{-1}x$, and local controllability can be defined in terms of the largest and smallest absolute eigenvalues of A (as a special case of our Example 1 with a fixed matrix). We view the nonsingularity assumption (and invertibility in the more general case) as fairly mild, as it amounts to assuming that the distribution map can depend on all parameters of classifier without any necessary (linear) dependency structure imposed, and that no two classifiers are equivalent only to the population but not the optimizer (as otherwise one could simply reduce dimensionality of $\mathcal{X}$). However, even in the case where A is singular, we note that this issue is resolvable augmenting the state representation y_t to incorporate the choice of free classifier parameters which affect loss but not distribution updates (e.g. by adding a vector w_t to y_t which is orthogonal to the range of A and linear in x_t). We assume invertibility here for simplicity, and we take $\mathcal{Y}$ to be simply be given by the range of s over $\mathcal{X}$. At each round t, some scoring function $f_t(x, z)$ is chosen adversarially, and our loss is then given by

$$\tilde{f}_t(x_t, p_t) = \mathop{\mathbb{E}}_{z\sim p_t}[f_t(x_t, z)].$$

We assume each f_t is convex and L_z-Lipschitz in both x and z, and that $p_0 = y_0 + \xi$. We measure our regret with respect to the best performatively stable classifier, i.e. the loss of any classifier as if were held constant indefinitely as the distribution updates. We define our regret as follows:

$$\mathsf{Reg}_T(\mathcal{A}) = \max_{x^*} \sum_{t=1}^{T} \tilde{f}_t(x_t, p_t) - \tilde{f}_t(x^*, \mathcal{D}(x^*, s(x^*)))$$

Here, the role of $s(x^*)$ captures the convergence of the distribution to a stable point, resulting from taking the limit of the distribution update rule as t grows large.

As in many of the applications we consider, here our loss is determined both by our action (the

classifier) and the state (in terms of the distribution). Our approach for casting Online Performative Prediction as an instance of online nonlinear control in our framework will be to define appropriate surrogate convex losses which depend only on the state, over which we run NestedOCO. Here, these will correspond to losses only over the updated distribution component $\mathcal{D}(x_t, y_{t-1})$, which we show closely track our true incurred loss.

C.3.3 Analysis

For each round t, define the surrogate loss $f_t^*(y)$ as:

$$f_t^*(y) = \mathop{\mathbb{E}}_{z \sim y_t + \xi} \left[f_t(s^{-1}(y), z) \right].$$

Lemma 19. *Each $f_t^*(y)$ is convex and $(1+S)L_z$-Lipschitz in y.*

Proof. Consider any individual sample $v \sim \xi$. We can then view $g(y) = (s^{-1}(y), y+v)$ as a vector-valued function which is $(1+S^*)$-Lipschitz. The function $f_t(g(y))$ is a L_z-Lipschitz and convex function of this linear function of y, and thus $f_t(s^{-1}(y), y+v)$ is convex and $(1+S^*)L_z$-Lipschitz in y. The function $f_t^*(y)$ is an average of such functions, taken over the expectation of ξ, and thus is convex and $(1+S^*)L_z$-Lipschitz in y as well. □

Observe that $f_t^*(y) = \tilde{f}_t(s^{-1}(y), \mathcal{D}(s^{-1}(y), y)$. We will run NestedOCO for these losses over the ρ-locally controllable instance $(\mathcal{X}, \mathcal{Y}, A)$, where we can track the current state $y_t = A(x_t, y_{t-1})$ at each step as a function of our past actions given knowledge of A, and can compute gradients of $f_t^*(y_t)$ to arbitrary desired precision by sampling from ξ. This will yield the regret bound from Theorem 21 with respect to the surrogate losses, and the key challenge will be to analyze our error between the true and surrogate losses.

Lemma 20. *For any round t we have that*

$$\tilde{f}_t(x_t, p_t) - f_t^*(y_t) \leq (1-\theta)^h M + \frac{\eta L_z (1+S)}{\gamma} \cdot \left(L_y + \frac{1-\theta}{\theta} \right)$$

Proof. For any $h < t$, the loss of x_t over the distribution $y_{t-h}+\xi = \mathcal{D}(x_{t-h}, y_{t-h-1})$ can be expressed as

$$\hat{f}_t(x_t, y_{t-h}) = \mathop{\mathbb{E}}_{z\sim\xi+y_{t-h}} [f_t(x_t, z)] ,$$

which is convex and L_z-Lipschitz in both parameters when taking the expectation over ξ. For round t in isolation, using the inverse action mapping bound and the bound on $\|y_t - y_{t-1}\|$ from Proposition 15 we have that

$$\begin{aligned}
\hat{f}_t(x_t, y_t) - f_t^*(y_t) &= \hat{f}_t(x_t, y_t) - \hat{f}_t(s^{-1}(y_t), y_t) \\
&= \hat{f}_t(X(y_{t-1}, y_t), y_t) - \hat{f}_t(X(y_t, y_t), y_t) \\
&\leq \frac{\eta L_y L_z}{\gamma},
\end{aligned}$$

and further for previous states that

$$\hat{f}_t(x_t, y_{t-h}) - f_t^*(y_t) = (L_y + h)\frac{\eta L_z(1+S)}{\gamma}.$$

We can decompose the distribution p_t into updates from past rounds as

$$p_t = (1-\theta)^t p_0 + \sum_{h=0}^{t-1} \theta(1-\theta)^h \, \mathcal{D}(x_{t-h}, y_{t-h-1})$$

which then yields a loss discrepancy of at most

$$\begin{aligned}
\tilde{f}_t(x_t, p_t) - f_t^*(y_t) &\leq (1-\theta)^t f_t(x_t, p_0) + \frac{\eta L_z(1+S)}{\gamma}\left(\sum_{h=0}^{t-1} \theta(1-\theta)^h (L_y + h)\right) \\
&\leq \frac{\eta L_z(1+S)}{\gamma} \cdot \left(L_y + \frac{1-\theta}{\theta} + (1-\theta)^t\right)
\end{aligned}$$

between the true and surrogate loss for round t. □

We can now bound the cumulative regret of NestedOCO for the problem.

Theorem 37. *For any $\theta > 0$, when Assumption 3 holds for the distribution update rule, Online Performative Prediction can be cast as a ρ-locally controllable instance of online control with nonlinear dynamics, for which* NestedOCO *obtains regret*

$$\mathsf{Reg}_T(\textsc{NestedOCO}) \leq 2\sqrt{\frac{(1 + L_y + \frac{R}{r\rho} + \frac{2-\theta}{\theta})TGL_z^2(1+S)^2}{\gamma}}$$

with respect to the best performatively stable classifier classifier.

Proof. Combining the previous results with Theorem 21, we have that for any $x^* \in \mathcal{X}$ our regret is at most

$$\begin{aligned}
\sum_{t=1}^{T} \tilde{f}_t(x_t, p_t) - \tilde{f}_t(\mathcal{D}(x^*, s(x^*))) &\leq \sum_{t=1}^{T} \hat{f}_t(y_t) - \tilde{f}_t(x^*, \mathcal{D}(x^*, s(x^*))) + \sum_{t=1}^{T} \tilde{f}_t(x_t, p_t) - f_t^*(y_t) \\
&\leq \eta \left(1 + L_y + \frac{2-\theta}{\theta} + \frac{R}{r\rho}\right) \frac{TL_z(1+S)}{\gamma} + \frac{G}{\eta} \\
&= 2\sqrt{\frac{(1 + L_y + \frac{R}{r\rho} + \frac{2-\theta}{\theta})TGL_z^2(1+S)^2}{\gamma}}
\end{aligned}$$

upon setting $\eta = \sqrt{\frac{G\gamma}{(1+L_y+\frac{R}{r\rho}+\frac{2-\theta}{\theta})TL_z^2(1+S)^2}}$. □

Theorem 24 follows directly from Theorem 37. For Online Performative Prediction, in the full generality of the setting considered, the per-round optimization problem may not be convex, in which case we make use of the non-convex optimization oracle access for NestedOCO. However, in each of the following applications we show that the action selection step can indeed be implemented efficiently without imposing additional restrictions on the dynamics.

C.4 Background and Proofs for Section 4.4.2: Adaptive Recommendations

C.4.1 Background

Motivated by problems involving preference dynamics and feedback loops in recommendation systems (see e.g.[34]), a number of recent works [49, 50, 131, 52, 121, 33] have explored models of

repeated recommendation where given to an agent whose preferences or opinions evolve over time. Several of these models [49, 131, 52] consider population-level effects for settings where a single recommendation is given each round and consumers (or producers) update their behavior according to linear dynamics. Nonlinear preference dynamics with *menus* of recommendations for a single agent are considered in [121, 33], where the aims to minimize regret for adversarial losses over the agent's choices. The Adaptive Recommendations formulation of [121] somewhat resembles the "Dueling Bandits" setting of [55], where $k > 1$ actions are chosen in each round, yet where preferences can now evolve dynamically as a function of the history rather than remaining fixed. Whereas [121, 33] study a bandit formulation of the problem with unknown preference dynamics, here we consider a full-feedback model with known dynamics, allowing for relaxed structural assumptions (on the agent's "memory horizon" and "preference scoring functions") at the cost of stronger informational assumptions, while maintaining the overall dynamics of the problem.

C.4.2 Model

Here, we are tasked with repeatedly recommending menus of content to an agent. Out of a universe of n elements (e.g. video channels, clothing items), we show a subset of size k (denoted K_t) to the agent in each round, for T total rounds. The agent chooses one item $i \in K_t$ from the menu, according to a distribution in terms of their *preferences*, which are a function of their selection history. Conditioned on being shown a menu K_t, the agent's choice distribution has positive mass only on the k items $i \in K_t$. The agent's representation of their selection history is given by their *memory vector* $v_t \in \Delta(n)$, and choices are determined by their *preference scoring functions* $s_i : \Delta(n) \to [\lambda, 1]$ for each i, which map the agent's memory vector to relative preference scores for each item. The menu we show to the agent may be chosen from some distribution $x_t \in \Delta(\binom{n}{k})$, and for each $K_t \in [\binom{n}{k}]$ the agent's menu-conditional distribution $p_t(\cdot; K_t, v_{t-1}) \in \Delta(n)$ is proportional to the scores $s_i(v_t)$ for items in K_t, given as

$$p_t(i; K_t, v_{t-1}) = \frac{s_i(v_{t-1})}{\sum_{j \in K_t} s_j(v_{t-1})}$$

for each $i \in K_t$, with $p_t(j; K_t, v_{t-1}) = 0$ for $j \notin K_t$. The joint item choice distribution, considering both random selection of a menu K_t according to x_t, and the agent's choice from K_t, is given by

$$p_t(\cdot; x_t, v_{t-1}) = \sum_{K_t \in \binom{n}{k}} x_t(K_t) \cdot p_t(\cdot; K_t, v_{t-1})$$

which we may denote simply by the vector $p_t \in \Delta(n)$, or as a function $p_t(x_t)$. In contrast to prior work, here we consider a deterministic variant of the problem as an illustration of the flexibility of our framework for online nonlinear control. In particular, we assume that the agent's memory vector v_t updates according to its expectation over p_t as

$$v_t = (1 - \theta_t)v_{t-1} + \theta_t p_t,$$

where $\theta_t \in [\theta, 1]$ is the per-round update speed, and we assume that the agent's scoring functions s_i are known. We receive convex and L-Lipschitz losses $f_t(p_t)$ in each round in terms of the agent's choices, over which we aim to minimize regret with respect to some distribution set $\mathcal{Y} \subseteq \Delta(n)$.

The prior work [121, 33] has considered two particular subsets of $\Delta(n)$ as regret benchmarks. We show that both can be cast as locally controllable instances of online control, and further, we make use of local controllability to give a general characterization of convex sets $\mathcal{Y} \subseteq \Delta(n)$ over which sublinear regret is attainable. We recall some key definitions and results from [121, 33].

Definition 20 (Instantaneously Realizable Distributions)**.** *The set of instantaneously realizable distributions at a memory vector $v \in \Delta(n)$ is given by*

$$\mathrm{IRD}(v) = \mathrm{convhull}\left\{p(\cdot; K, v) : K \in \left[\binom{n}{k}\right]\right\}.$$

Each such set $\mathrm{IRD}(v_{t-1})$ corresponds to the feasible distributions p_t, given the agent's scoring functions and memory v_{t-1}. It is shown by [33] that each IRD sets can be directly characterized in terms of the ratios between target frequencies and scores.

Proposition 16 (Menu Times for IRD [33])**.** *Given a memory vector $v \in \Delta(n)$ and target distribution*

$p \in \Delta(n)$, let the menu time μ_i for item i be given by

$$\mu_i = \frac{k \cdot \frac{p(i)}{s_i(v)}}{\sum_{j=1}^{n} \frac{p(j)}{s_j(v)}},$$

where $\sum_{i=1}^{n} \mu_i = k$. Then, $p \in \mathrm{IRD}(v)$ if and only if $\mu_i \leq 1$ for each $i \in [n]$.

We recall the prior benchmark sets considered, and the corresponding assumptions which yield feasibility of regret minimization. We state informal analogues of the prior results as translated to our setting, which we then show formally below.

Definition 21 (Everywhere Instantaneously Realizable Distributions)**.** *The set of everywhere instantaneously realizable distributions is given by*

$$\mathrm{EIRD} = \bigcap_{v \in \Delta(n)} \mathrm{IRD}(v).$$

Proposition 17 (Corollary of [121])**.** *If $\lambda \geq \frac{k}{n} + \frac{k}{n(n-1)}$, then* EIRD *is non-empty, and there is a $o(T)$ regret algorithm with respect to any distribution $p \in$* EIRD*.*

Distributions $p_t \in$ EIRD are always feasible regardless of v_{t-1} by an appropriate choice of x_t, but EIRD may be quite small in relation to $\Delta(n)$. Under stronger assumptions for each s_i, a potentially much larger set becomes feasible as a regret benchmark.

Definition 22 (ϕ-Smoothed Simplex)**.** *The ϕ-smoothed simplex $\Delta^{\phi}(n)$ for $\phi \in [0, 1]$ is given by*

$$\Delta^{\phi}(n) = \{(1 - \phi)v + \phi \mathbf{u}_n : v \in \Delta(n)\}$$

Definition 23 (Scale-Bounded Functions)**.** *A scoring function $s_i : \Delta(n) \to [\frac{\lambda}{\sigma}, 1]$ is said to be (σ, λ)-scale-bounded for $\sigma > 1$ and $\lambda > 0$ if, for all $v \in \Delta(n)$, we have that*

$$\sigma^{-1}((1 - \lambda)v_i + \lambda) \leq s_i(v) \leq \sigma((1 - \lambda)v_i + \lambda).$$

For such functions, each score $s_i(v)$ cannot be too far from item i's weight in memory, and it is shown that $\mathrm{IRD}(v)$ contains a ball around v for each $v \in \Delta^{\phi}(n)$, for an appropriate choice of ϕ.

Proposition 18 (Corollary of [33])**.** *If each s_i is (σ, λ)-scale-bounded, then there is a $o(T)$ regret algorithm with respect to any distribution $p \in \Delta^{\phi}(n)$, for $\phi = \Theta(k\lambda\sigma^2)$.*

We extend these results to general convex benchmark sets $\mathcal{Y} \subseteq \Delta(n)$, where we can characterize the feasibility of regret minimization via local controllability using the menu times μ_i. When ρ-local controllability holds over a set $\mathcal{Y}$, we can minimize regret via NestedOCO using surrogate losses $f_t^*(v_t)$, which closely track our true losses $f_t(p_t)$.

C.4.3 Analysis

We make use of the menu time quantities μ_i for a memory vector v and target distribution p to translate our notion of local controllability to the Adaptive Recommendations setting. Let $\mathcal{Y}$ be any convex subset of $\Delta(n)$, let $\mathcal{X} = \Delta(\binom{n}{k})$, where the dynamics $D_t(x_t, v_{t-1})$ are given by

$$D_t(x_t, v_{t-1}) = (1 - \theta_t)v_{t-1} + \theta_t p_t(x_t).$$

Note that $D_t(x_t, v_{t-1})$ is action-linear in x_t, and thus we can solve for x_t efficiently (in terms of $\dim(\mathcal{X}) = O(n^k)$); further, there is a construction given in [33] for removing exponential dependence on k when computing menu distributions. We consider $\mathcal{Y}$ as an $(n-1)$-dimensional subset of $\mathbb{R}^n$, where we define the the ball $\mathcal{B}_\rho(v)$ of radius ρ around a point $v \in \mathcal{Y}$ as:

$$\mathcal{B}_\rho(v) = \{p \in \Delta(n) : \|p - v\| \leq \rho\}.$$

Theorem 38. *An instance of Adaptive Recommendations $(\mathcal{X}, \mathcal{Y}, D)$ satisfies $\rho\theta$-local controllability if, for any $v \in \mathcal{Y}$ and $p \in \mathcal{B}_{\rho \cdot \pi(v)}$, we have that*

$$\frac{(k-1)p(i)}{s_i(v)} \leq \sum_{j \neq i}^{n} \frac{p(j)}{s_j(v)}$$

for every $i \in [n]$.

This follows immediately from Proposition 17 and the definition of local controllability, which can analogously extend to strong local controllability. We can use this formulation to unify the feasibility analysis for each of the previously considered sets.

Lemma 21. *For* $\lambda \geq \frac{k-1}{n-1} + \epsilon$ *and* $\epsilon \geq 0$*, the* EIRD *set contains a ball of radius* $\rho = \Theta(\frac{\epsilon}{nk+\epsilon})$ *around* $\mathbf{u}_n$*, and any instance* $(\mathcal{X}, \text{EIRD}, D)$ *satisfies* θ*-local controllability.*

Proof. For any $v \in \Delta(n)$, $i \in [n]$, and $p \in \mathcal{B}_\rho(\mathbf{u}_n)$ we have $p(i) \leq \frac{1}{n} + \frac{\rho\sqrt{2}}{2}$ and $s_i(v) \geq \frac{k-1}{n-1} + \epsilon$, yielding that

$$\frac{(k-1)p(j)}{s_j(v)} \leq \frac{1 + \frac{\rho n\sqrt{2}}{2}}{\frac{n}{n-1} + \frac{\epsilon n}{k-1}},$$

and over all items $j \neq i$ (with $s_j(v) \leq 1$) we have

$$\sum_{j \neq i}^{n} \frac{p(j)}{s_j(v)} \geq 1 - \frac{1}{n} - \frac{\rho\sqrt{2}}{2}.$$

Observe that the bounds for each term are equalized at $\frac{n-1}{n}$ when $\rho = \epsilon = 0$, and so $\mathbf{u}_n \in$ EIRD whenever $\lambda \geq \frac{k-1}{n-1}$. We can specify $\epsilon(\rho)$ in terms of ρ to maintain equality, and thus inclusion of $p \in$ EIRD. Taking $\epsilon(\rho)$ in terms of ρ as

$$\begin{aligned}
\epsilon(\rho) &= \frac{\rho n(k-1)}{\frac{2(n-1)}{\sqrt{2}n} - \rho} \\
&= \frac{\frac{\rho n(k-1)\sqrt{2}}{2}}{\left(1 - \frac{1}{n} - \frac{\rho\sqrt{2}}{2}\right)} \\
&= (k-1)\left(\frac{\frac{1}{n} + \frac{\rho\sqrt{2}}{2}}{1 - \frac{1}{n} - \frac{\rho\sqrt{2}}{2}} - \frac{1}{n-1}\right)
\end{aligned}$$

gives us that

$$\frac{1}{n-1} + \frac{\epsilon(\rho)}{k-1} \geq \frac{\frac{1}{n} + \frac{\rho\sqrt{2}}{2}}{1 - \frac{1}{n} - \frac{\rho\sqrt{2}}{2}}$$

for $\rho \geq 0$, and so we maintain that $p \in \text{EIRD}$. Inverting, we have

$$\rho(\epsilon) = \frac{\epsilon \frac{2(n-1)}{\sqrt{2}n}}{n(k-1) + \epsilon}$$

as the radius of a ball around $\mathbf{u}_n$ contained in EIRD. To see that EIRD is θ-locally controllable, consider any v_{t-1} and v^* in EIRD where $v^* \in \mathcal{B}_{\pi(v_{t-1})}(v_{t-1})$, and let $v_t = (1-\theta_t)v_{t-1} + \theta_t v^*$. By playing an action distribution x_t which induces $p_t(x_t) = v^*$, the memory vector is then updated to v_t. This is feasible for any $v_t \in \mathcal{B}_{\theta \cdot \pi(v_{t-1})}(v_{t-1})$, as each corresponds to some $v^* \in \mathcal{B}_{\pi(v_{t-1})}(v_{t-1})$. □

We remark that for the EIRD set, if losses are given over p_t rather than v_t, one can define dynamics which directly consider the state to simply be the induced distribution p_t in each round, which satisfies strong local controllability with any $p_t \in \text{EIRD}$ feasible at each round; in general, we consider dynamics to view the memory vector as the state, as the feasible updates p_t are a function of v_t. Such is the case for the ϕ-smoothed simplex, for which we can state an analogous local controllability result.

Lemma 22. *If each s_i is (σ, λ)-scale-bounded, then any instance $(\mathcal{X}, \Delta^\phi(n), D)$ over the ϕ-smoothed simplex for $\phi = \Theta(k\lambda\sigma^2)$ satisfies $\Omega(\theta\lambda\phi)$-local controllability.*

Proof. The following lemma from [33] shows that a ball of distributions around any memory vector $v \in \Delta^\phi(n)$ is feasible under $\text{IRD}(v)$.

Lemma 23 (IRD for Scale-Bounded Preferences [33])**.** *Let each s_i be (σ, λ)-scale-bounded with $\sigma \leq \sqrt{4(n-1)/k}$, and let $v \in \Delta^\phi(n)$ be a vector in the ϕ-smoothed simplex, for $\phi \geq \Theta k\lambda\sigma^2$. Then, $p \in \text{IRD}(v)$ for any vector $p \in \mathcal{B}_{\lambda\phi}(v) \cap \Delta^\phi(n)$.*

Let $d = \min(\lambda\phi, \pi(v_{t-1})) \leq \lambda\phi\pi(v_{t-1})$ for any v_{t-1} in $\Delta^\phi(n)$. Any $v^* \in \mathcal{B}_d(v_{t-1})$ then

is contained in $\text{IRD}(v_{t-1})$, and so playing x_t such that $p_t(x_t) = v^*$ yields an update to $v_t = (1-\theta_t)v_{t-1} + \theta v^*$, which is feasible for any $v_t \in \mathcal{B}_{d\theta}(v_{t-1})$, and so $\Omega(\theta\lambda\phi)$-local controllability holds. □

For any such set $\mathcal{Y}$ which yields locally controllable dynamics for the instance $(\mathcal{X}, \mathcal{Y}, D)$, we can minimize regret over $\mathcal{Y}$ via NestedOCO, where we optimize with respect to the surrogate losses $f_t^*(v_t)$. Note that for our regret benchmark of the best per-round instantaneously distribution in $\mathcal{Y}$, any fixed vector v^* which is instantaneously targeted across all rounds yields an item distribution $p_t = v^*$ in each round, and so $f_t^*(v^*) = f_t(p^*)$. We assume that y_0 is bounded inside $\mathcal{Y}$ (which typically will hold for $y_0 = \mathbf{u}_n$).

Theorem 39. *For any ρ-locally controllable instance $(\mathcal{X}, \mathcal{Y}, D)$ of Adaptive Recommendations with update speed $\theta > 0$, running* NestedOCO *over the surrogate losses $f_t^*(v_t)$ yields regret*

$$\mathsf{Reg}_T(\text{NestedOCO}) \leq 2\sqrt{\frac{(2 + \frac{R}{r\rho} + \frac{1}{\theta})TGL^2}{\gamma}}$$

with respect to the true losses $f_t(p_t)$ over $\mathcal{Y}$.

Proof. Beyond applying the regret bound for NestedOCO from Theorem 21, the key step here is

to bound surrogate loss errors as:

$$\begin{aligned}
\sum_{t=1}^{T} f_t(p_t) - f_t(v^*) &\leq \sum_{t=1}^{T} f_t^*(v_t) - f_t(v^*) + \sum_{t=1}^{T} f_t(v_t) - f_t(p_t) \\
&\leq \eta\left(1 + \frac{R}{r\rho}\right)\frac{TL^2}{\gamma} + \frac{G}{\eta} + \sum_{t=1}^{T} f_t(v_t) - f_t\left(\frac{v_t - (1-\theta_t)v_{t-1}}{\theta_t}\right) \\
&\leq \eta\left(1 + \frac{R}{r\rho}\right)\frac{TL^2}{\gamma} + \frac{G}{\eta} + \sum_{t=1}^{T} f_t(v_t) - f_t\left(v_{t-1} + \frac{v_t - v_{t-1}}{\theta_t}\right) \\
&\leq \eta\left(1 + \frac{R}{r\rho}\right)\frac{TL^2}{\gamma} + \frac{G}{\eta} + L\left(1 + \frac{1}{\theta}\right)\sum_{t=1}^{T} \|v_t - v_{t-1}\| \\
&\leq \eta\left(2 + \frac{R}{r\rho} + \frac{1}{\theta}\right)\frac{TL^2}{\gamma} + \frac{G}{\eta} \\
&= 2\sqrt{\frac{(2 + \frac{R}{r\rho} + \frac{1}{\theta})TGL^2}{\gamma}}
\end{aligned}$$

upon setting $\eta = \sqrt{\frac{G\gamma}{(2+\frac{R}{r\rho}+\frac{1}{\theta})TL^2}}$, which yields the theorem. □

Theorems 25 and 26 follow from Theorem 39, as well as from Lemmas 21 and 22, respectively.

C.5 Background and Proofs for Section 4.4.3: Adaptive Pricing

C.5.1 Background

While there is a large literature on designing online mechanisms for pricing discrete goods via auctions [132, 133, 134, 135, 136, 73], there is comparatively little work related to online pricing problems for real-valued goods. Most work for such problems to date requires strong assumptions on valuation functions, often either assuming linearity [137] or additivity [138], or requiring approximability via discretization [139]. Here, we introduce a novel formulation for an Adaptive Pricing problem which builds on the myopic-demand fixed-cost setting of [75], which we extend to accommodate adversarial *consumption rates* for the agent (which affect demand, as a function of the agent's *reserves*) as well as adversarial production costs. As in [75], our setting can accommodate general convex (increasing) production cost functions and concave (increasing)

valuations for the agent, provided that valuations additionally are homogeneous; to our knowledge, this encompasses a much wider class of valuations and costs than considered by any prior work on no-regret dynamic pricing for real-valued goods.

C.5.2 Model

In each round t, an agent (the *consumer*) begins with goods reserves $y_{t-1} \in \mathbb{R}^n_{\geq 0}$ (with $y_0 = \mathbf{0}$), then consumes an adversarially chosen fraction $\theta_t \in [\theta, 1]$ of each good simultaneously (e.g. corresponding to their rate of manufacturing downstream items, using the goods as components), updating their reserves to $(1 - \theta_t)y_{t-1}$. We (the *producer*) show the consumer some vector $p_t \in \mathbb{R}^n_+$ of per-unit prices for each good, and the consumer purchases some bundle of goods x_t. The consumer's valuation function for reserves of goods is given by $v : \mathbb{R}^n_+ \to \mathbb{R}_+$, and their selection of $x_t = x^*(p_t, \theta_t, y_{t-1})$ is given by

$$x^*(p_t, \theta_t, y_{t-1}) = \operatorname*{argmax}_{x \in \mathbb{R}^n_+} v(x + (1 - \theta_t)y_{t-1}) - \langle p_t, x \rangle.$$

We later discuss behavior of x^* when the argmax is undefined; it will suffice for us to only consider price vectors for which it is defined. This updates the consumer's reserves to $y_t = x_t + (1 - \theta_t)y_{t-1}$. Upon seeing the consumer's purchased bundle x_t, we receive their payment $\langle p_t, x_t \rangle$ minus our production cost $c_t(x_t) : \mathbb{R}^n_+ \to \mathbb{R}_+$, where c_t is adversarially chosen. Our utility is then given by

$$f_t(p_t, x_t) = \langle p_t, x_t \rangle - c_t(x_t).$$

We make the following assumptions on production costs c_t and the consumer's valuation v.

Assumption 4 (Production Costs)**.** *We assume that for each c_t, the following hold over $\mathbb{R}^n_+$:*

- *c_t is non-negative, convex, and L_c-Lipschitz,*
- *$\lim_{\epsilon \to 0} c_t(\epsilon \cdot \mathbf{1}) \leq C_0$ for some $C_0 \geq 0$, and*
- *$c_t(x) \geq \phi \|x\| + C_0$ for some $\phi > 0$.*

Further, each c_t is revealed prior to setting prices p_{t+1}.

Assumption 5 (Consumer Valuations)**.** *We assume that the following hold over some set $\mathcal{Y} \subseteq \mathbb{R}^n_+$:*

- *v is non-negative, continuous, and differentiable,*
- *v is strictly concave and increasing,*
- *v is (λ, β)-Hölder continuous for some $\lambda \geq 1$ and $\beta \in (0, 1]$, i.e.*

$$|v(y) - v(y')| \leq \lambda \|y - y'\|^\beta,$$

and

- *v is homogeneous of degree k for some $k \in (0, 1)$, i.e. $v(by) = b^k v(y)$ for any $b > 0$.*

Further, v is known to the producer.

Given the concavity assumption, we note that it is without loss of generality to assume that $k \in (0, 1)$ for the homogeneity parameter. There are several well-studied valuation families which satisfy these properties for an appropriate set $\mathcal{Y}$; see [75] for proofs of each example.

Example 3 (Constant Elasticity of Substitution (CES))**.** *Valuations of the form*

$$v(y) = \left(\sum_{i=1}^n \alpha_i y_i^\kappa \right)^\beta,$$

with each $\alpha_i, \kappa, \beta > 0$ and $\kappa, \beta\kappa < 1$, are Hölder continuous, differentiable, strictly concave, non-decreasing, and homogeneous over a convex set in $\mathbb{R}^n_+$.

Example 4 (Cobb-Douglas)**.** *Valuations of the form*

$$v(y) = \prod_{i=1}^n y_i^{\alpha_i},$$

with $\alpha_i > 0$ and $\sum_{i=1}^n \alpha_i < 1$ are Hölder continuous, differentiable, strictly concave, non-decreasing, and homogeneous over a convex set in $\mathbb{R}^n_+$.

We initially assume that Assumption 5 holds over all of $\mathbb{R}^n_+$, but will restrict our attention to the set $\mathcal{Y} \subseteq \mathbb{R}^n_+$ of bundles where $v(y) \geq \phi \|y\|$ for each $y \in \mathcal{Y}$, and we note that our results can be extended to arbitrary downward-closed convex sets (where $by \in \mathcal{Y}$ for any $y \in \mathcal{Y}$ and $b \in (0, 1]$). In Section C.5.3 we that show Assumptions 4 and 5 yield several important properties which enable optimization via our framework. We show a unique mapping between price vectors and bundle purchases (for any fixed reserves and consumption rate), that restricting attention to $\mathcal{Y}$ is justified under rationality constraints, and that $\mathcal{Y}$ is convex.

Further, there is some price vector which yields a reserve update to any $y_t \in \mathcal{Y}$ in a neighborhood around y_{t-1}, yielding local controllability. Crucially, we show that there are concave surrogate rewards $f^*_t(y_t)$ which will closely track our true rewards $f_t(p_t, x_t)$, leveraging the following property of homogeneous functions.

Proposition 19 (Euler's Theorem for Homogeneous Functions)**.** *A continuous and differentiable function $v : \mathcal{Y} \to \mathbb{R}_+$ is homogeneous of degree k if and only if*

$$\langle \nabla v(y), y \rangle = k \cdot v(y).$$

We run NestedOCO directly over these concave surrogate rewards (by inverting the sign of each), where each p_t can be computed efficiently in terms of y_{t-1} and θ_t, and we show that the surrogate reward distance from our true rewards is bounded. While our rewards will not be Lipschitz over $\mathcal{Y}$ in general, we show that appropriately calibrating our step size yields sublinear regret with dependence on the Hölder continuity parameters. We measure our regret with respect to the set of *stable reserve policies*, i.e. pricing policies where y_t remains constant.

Definition 24 (Regret for Stable Reserve Policies)**.** *Let $\mathcal{P}_\mathcal{Y} = \{P_y : y \in \mathcal{Y}\}$ be the set of stable reserve policies, where for any y_{t-1} and θ_t satisfying $(1 - \theta_t)y_{t-1} \leq y^*$, playing prices computed by a policy $p_t = P^*_y(y_{t-1}, \theta)$ yields*

$$(1 - \theta_t)y_{t-1} + x^*(p_t, \theta_t, y_{t-1}) = y^*.$$

It is straightforward to see that any $P_y^* \in \mathcal{P}_{\mathcal{Y}}$ maintains the invariant that $y_t = y^*$, provided that some such p_t is always feasible.

C.5.3 Analysis

We show a series of results establishing the key conditions allowing us to formulate this problem as a locally controllable instance of online nonlinear control. We first show that any positive bundle is the unique optimal purchase for some positive price vector.

Lemma 24. *For any reserves $y_{t-1} \in \mathbb{R}^n_{\geq 0}$, consumption rate $\theta_t \in [\theta, 1]$, and vector $y_t \in \mathbb{R}^n_+$ where $y_t > (1-\theta_t)y_{t-1}$ elementwise, the bundle $x_t = y_t - (1-\theta_t)y_{t-1}$ is the unique solution to*

$$x_t = x^*(p_t, \theta_t, y_{t-1})$$

for prices $p_t = \nabla v(y_t)$.

Proof. Recall that the consumer's bundle choice is given by

$$x^*(p_t, \theta_t, y_{t-1}) = \operatorname*{argmax}_{x \in \mathbb{R}^n_+} v(x + (1-\theta_t)y_{t-1}) - \langle p_t, x \rangle.$$

Note that $v((1-\theta_t)y_{t-1} + x) - \langle p_t, x \rangle$ is strictly concave in x for any $x \in \mathbb{R}^n_+$, as the gradients

$$\nabla_x v((1-\theta_t)y_{t+1} + x) = \nabla_{y_t} v(y_t)$$

are preserved at each point $y_t = (1-\theta_t)y_{t+1} + x$, and subtracting the linear function $\langle x, p_t \rangle$ does not affect strict concavity. We also have that $p_t \in \mathbb{R}^n_+$ for prices $p_t = \nabla v(y_t)$, as v is strictly concave and non-decreasing. This yields that $v((1-\theta_t)y_{t-1} + x) - \langle p_t, x \rangle$ has a unique global maximum at $x_t = y_t - (1-\theta_t)y_{t-1}$, as $\nabla_x(v((1-\theta_t)y_{t+1} + x) - \langle p_t, x \rangle) = \mathbf{0}$. □

As such, the argmax for $x^*(p_t, \theta_t, y_{t-1})$ is unique whenever $p_t = \nabla v(y)$ for some $y \in \mathbb{R}^n_+$. We let $p^*(x_t; y_{t-1}, \theta_t) = \nabla v((1-\theta_t)y_{t-1} + x_t)$ denote this price vector which induces a purchase of x_t. For

any other price vector p, the maximizing bundle x_t either approaches a point on the boundary of $\mathbb{R}^n_+$, or grows unboundedly. We restrict our attention to bundles contained in $\mathbb{R}^n_+$, and show that the issue of unboundedness is resolved by rationality considerations for the producer. We characterize the per-round rewards of stable reserve policies as concave functions of $y \in \mathbb{R}^n_+$, and show that the optimal such policy corresponds to some state $y^* \in \mathcal{Y}$, where $\mathcal{Y}$ is convex and bounded.

Lemma 25. *The round-t reward of a stable reserve policy P_y corresponding to any $y \in \mathbb{R}^n_+$ is given by a strictly concave function*

$$f_t(P_y) = \theta_t k \cdot v(y) - c_t(\theta_t y).$$

Proof. We first note that we can maintain $y_t = y$ in every round by Lemma 24, as $y_0 = \mathbf{0}$ and $(1 - \theta_t)y < y$. As such, a bundle $x_t = \theta_t y$ is purchased in each round at prices $\nabla v(y)$, and our reward is given by

$$\begin{aligned} f_t(P_y) &= f_t(p^*(\theta_t y; y, \theta_t), \theta_t y) \\ &= \langle \nabla v(y), \theta_t y \rangle - c_t(\theta_t y) \\ &= \theta_t k \cdot v(y) - c_t(\theta_t y), \end{aligned}$$

where the final step follows from Proposition 19 for homogeneous functions. The function $\theta_t k \cdot v(y)$ is strictly concave, which is preserved upon subtracting the convex function $c_t(\theta_t y)$. □

Lemma 26. *The set $\mathcal{Y} = \{y \in \mathbb{R}^n_+ : v(y) \geq \phi \|y\|\}$ is convex.*

Proof. Consider any two points $y, y' \in \mathcal{Y}$, and let $y'' = ay + (1 - a)y'$ for any $a \in [0, 1]$. Recall

that $y^* \in \mathbb{R}^n_+$ belongs to $\mathcal{Y}$ if and only if $v(y^*) \geq \phi \|y^*\|$. By concavity of v, we have that

$$\begin{aligned} v(y'') &= v(ay + (1-a)y') \\ &\geq av(y) + (1-a)v(y') \\ &\geq \phi \|ay\| + \phi \|(1-a)y'\| \\ &\geq \phi \|ay + (1-a)y'\| \\ &= \phi \|y''\| \end{aligned}$$

and so $y'' \in \mathcal{Y}$, yielding convexity of $\mathcal{Y}$. □

Lemma 27. *For any $z \in \mathbb{R}^n_+$ where $z \notin \mathcal{Y}$, there is some $y \in \mathcal{Y}$ such that $f_t(P_y) \geq f_t(P_z)$ for any θ_t and c_t.*

Proof. Consider some $z \notin \mathcal{Y}$ such that $v(z) = \psi \|z\|$, for $\psi < \phi$, and let $y = \left(\frac{\psi}{\phi}\right)^{1/k} z$. By homogeneity of v, we have that $v(y) = \frac{\phi}{\psi} v(z) = \phi \|z\|$, and so $y \in \mathcal{Y}$ as $\|z\| > \|y\|$. For any round with costs c_t and consumption rate θ_t we then have that:

$$\begin{aligned} f_t(P_y) - f_t(P_z) &= \theta_t k \left(v(y) - v(z)\right) - c_t(\theta_t y) + c_t(\theta_t z) \\ &= \theta_t k \left(\frac{\psi}{\phi} - 1\right) \psi \|z\| - c_t(\theta_t y) + c_t(\theta_t z) && \text{(homogeneity of } v\text{)} \\ &\geq \theta_t k \left(\frac{\psi}{\phi} - 1\right) \psi \|z\| + \theta_t \phi \|z - y\| && \text{(lower bound and convexity of } c_t\text{)} \\ &\geq \theta_t k \left(\frac{\psi}{\phi} - 1\right) \psi \|z\| + \theta_t \left(1 - \left(\frac{\psi}{\phi}\right)^{1/k}\right) \phi \|z\| \\ &\geq \theta_t \left(1 - \frac{\psi}{\phi}\right) \phi \|z\| - \theta_t \left(1 - \frac{\psi}{\phi}\right) \psi \|z\| && (k, \tfrac{\psi}{\phi} < 1) \\ &> 0. && (\phi > \psi) \end{aligned}$$

□

Thus the optimal P_y for any cost and consumption sequence corresponds to some $y \in \mathcal{Y}$. We can also bound the radius of $\mathcal{Y}$.

Lemma 28. *Let* $V = \max_{y \in \mathbb{R}^n_+ : \|y\|=1} v(y)$. *Then, for every* $y \in \mathcal{Y}$ *we have that*

$$\|y\| \leq \left(\frac{V}{\phi}\right)^{\frac{1}{1-k}}.$$

Proof. Let $y^* = \operatorname{argmax}_{y:\|y\|=1} v(y)$, where we have $v(y^*) = V$. Consider the vector by^* for any $b > 0$. By homogeneity of v, we have that

$$\begin{aligned} v(by^*) &= b^k v(y^*) \\ &= b^k V. \end{aligned}$$

For any $b > \left(\frac{V}{\phi}\right)^{\frac{1}{1-k}}$ we have that

$$\begin{aligned} v(by^*) &= \frac{b}{b^{1-k}} \cdot V \\ &\leq b\phi, \end{aligned}$$

where $\|by^*\| > b$ and thus $by^* \notin \mathcal{Y}$. This holds for all vectors with norm b, as any such vector z will have at most $b^k V$ by homogeneity, which yields the result. □

The previous result also implies that $by \in \mathcal{Y}$ for any $b < 1$ and $y \in \mathcal{Y}$. We assume that $V > \phi$, which is without loss of generality as we may otherwise take ϕ to be smaller artificially; we assume ϕ is small enough to ensure that $\mathcal{Y}$ contains a ball $\mathcal{B}_1(y_1)$ of radius 1 around some $y_1 \in \mathcal{Y}$, and we let $R = \left(\frac{V}{\phi}\right)^{\frac{1}{1-k}}$. We consider the dynamics to be given by

$$D_t(p_t, y_{t-1}) = (1 - \theta_t)y_{t-1} + x^*(p_t, \theta_t, y_{t-1}).$$

We let $\mathcal{Z} = \mathbb{R}^n_+$ denote our action space of price vectors; while dynamics here are not action-linear, we can still compute our desired action $p_t = \nabla v(y_t)$ efficiently, as we assume we have knowledge of v. While the dynamics depend on θ_t, our choice of action p_t depends only on the target update

y_t to the consumer's reserves, by Lemma 24. Further, upon observing x_t, we can solve for θ_t as

$$\theta_t = 1 - \frac{y_t - x_t}{y_{t-1}}$$

for purposes of representing our surrogate losses, which are given by

$$f_t^*(y_t) = \theta_t k \cdot v(y) - c_t(\theta_t y).$$

We now show that the dynamics satisfy local controllability.

Lemma 29 (Local Controllability)**.** *The instance* $(\mathcal{Z}, \mathcal{Y}, D_t)$ *satisfies* θ*-local controllability for each round* t*.*

Proof. We show that θ-local controllability holds over all of $\mathbb{R}_+^n$, which implies θ-local controllability over $\mathcal{Y}$ as each distance $\pi(y_{t-1})$ while the feasible update region remains the same. By Lemma 24, any update where $y_t \geq (1 - \theta_t)y_{t-1}$ elementwise is feasible. Each $\pi(y_{t-1})$ over $\mathbb{R}_+^n$ is simply the minimum element of y_t, which we denote here by m. Each element of y_{t-1} is decreased by at least θm, and so any y_t in the ℓ_∞ ball of radius $\theta m = \theta\pi(y_{t-1})$, and thus the ℓ_2 ball of radius $\theta\pi(y_{t-1})$, is feasible. □

We are now ready to analyse the regret of NestedOCO for the problem. The remaining key issues to resolve will be the errors between our true and surrogate rewards f_t and f_t^*, as well as the lack of Lipschitz continuity for our rewards. We will make use of more general formulations of the guarantees of FTRL, (see e.g. [127]).

Proposition 20. *For a* γ*-strongly convex regularizer* $\psi : \mathcal{Y} \to \mathbb{R}$ *where* $|\psi(y) - \psi(y')| \leq G$ *for all* $y, y' \in \mathcal{Y}$*, and for convex losses* $f_1, \ldots, f_T$*, the regret of* FTRL *is bounded by*

$$\mathsf{Reg}_T(\text{FTRL}) \leq \sum_{t=1}^{T} (g_t(y_t) - g_t(y_{t+1})) + \frac{G}{\eta},$$

where $g_t(y) = \langle \nabla_t f_t(y_t), y \rangle$ *and* $g_t(y_t) - g_t(y_{t+1}) \geq \frac{\gamma}{\eta} \|y_{t+1} - y_t\|^2$.

We show that this implies a regret bound for (λ, β)-Hölder continuous convex losses, recovering the λ-Lipschitz bounds when $\beta = 1$.

Theorem 40. *For (λ, β)-Hölder continuous convex losses,* FTRL *with obtains regret bounded by*

$$\mathsf{Reg}_T(\text{FTRL}) \leq T\lambda \left(\frac{\eta\lambda}{\gamma}\right)^{\beta/(2-\beta)} + \frac{G}{\eta}$$

and chooses points which satisfy $\|y_{t+1} - y_t\| \leq \left(\frac{\eta\lambda}{\gamma}\right)^{1/(2-\beta)}$ in each round.

Proof. For (λ, β)-Hölder continuous convex losses f_t, we have that

$$\begin{aligned} g_t(y_t) - g_t(y_{t+1}) &= \langle \nabla_t f_t(y_t), y_t - y_{t+1} \rangle \\ &= \langle \nabla_t f_t(y_t), (2y_t - y_{t+1}) - y_t \rangle \\ &\leq f_t(2y_t - y_{t+1}) - f_t(y_t) \end{aligned}$$

by convexity of f_t, where $\|(2y_t - y_{t+1}) - y_t\| = \|y_t - y_{t+1}\|$, and so

$$g_t(y_t) - g_t(y_{t+1}) \leq \lambda \|y_t - y_{t+1}\|^\beta$$

by Hölder continuity. Combining with the lower bound on $g_t(y_t) - g_t(y_{t+1})$ from Proposition 20 gives us that

$$\frac{\gamma}{\eta} \|y_{t+1} - y_t\|^2 \leq g_t(y_t) - g_t(y_{t+1}) \leq \lambda \|y_t - y_{t+1}\|^\beta$$

and thus

$$g_t(y_t) - g_t(y_{t+1}) \leq \lambda \left(\frac{\eta\lambda}{\gamma}\right)^{\beta/(2-\beta)},$$

yielding a regret bound of

$$\mathsf{Reg}_T(\text{FTRL}) \le T\lambda\left(\frac{\eta\lambda}{\gamma}\right)^{\beta/(2-\beta)} + \frac{G}{\eta}$$

with per-round distance at most $\|y_{t+1} - y_t\| \le \left(\frac{\eta\lambda}{\gamma}\right)^{1/(2-\beta)}$. □

We note that the concave surrogate rewards $f_t^*(y_t)$ are a sum of a $(k\lambda, \beta)$-Hölder continuous function and a $(L_c, 1)$-Hölder continuous (i.e. Lipschitz) function; we assume that each function is (L, β)-Hölder continuous with $L = k\lambda + L_c$, which is sufficient for for large enough T as we will have $\|y_t - y_{t-1}\| \le 1$ and thus $\|y_t - y_{t-1}\| \le \|y_t - y_{t-1}\|^\beta$. We use a similar analysis to bound the error between true and surrogate rewards, yielding our regret bound for NestedOCO.

Theorem 41. *The regret of* NestedOCO *with respect to the stable reserve policies $\mathcal{P}_{\mathcal{Y}}$ is bounded by*

$$\mathsf{Reg}_T(\text{NestedOCO}) \le 2L\left(\frac{G}{\gamma}\right)^{\beta/2}\left(T\left(3 + \left(\frac{R}{\theta}\right)^\beta\right)\right)^{(2-\beta)/2}.$$

Proof. We reparameterize to treat the bundle y_1 where $\mathcal{B}_1(y_1) \subseteq \mathcal{Y}$ as the origin, and assume the choice of regularizer has y_1 as its minimum. By Theorem 21, for any step size and $\delta > 0$ such that $\|y_t - y_{t-1}\| \le \delta\theta$, running NestedOCO for the θ-locally controllable instance $(\mathcal{Z}, \mathcal{Y}, D)$ over the surrogate rewards f_t^*, with inradius 1 and radius R, obtains

$$\begin{aligned}
\sum_{t=1}^T f_t^*(y^*) - \sum_{t=1}^T f_t^*(y_t) &\le TL(\delta R)^\beta + TL\left(\frac{\eta L}{\gamma}\right)^{\beta/(2-\beta)} + \frac{G}{\eta} \\
&\le TL\left(1 + \left(\frac{R}{\theta}\right)^\beta\right)\left(\frac{\eta L}{\gamma}\right)^{\beta/(2-\beta)} + \frac{G}{\eta} \\
&\le 2L\left(\frac{G}{\gamma}\right)^{\beta/2}\left(T\left(1 + \left(\frac{R}{\theta}\right)^\beta\right)\right)^{(2-\beta)/2} \\
&\triangleq \mathsf{Reg}_T(f^*)
\end{aligned}$$

for any $y^* \in \mathcal{Y}$, upon setting $\delta = \frac{1}{\theta}\left(\frac{\eta\lambda}{\gamma}\right)^{1/(2-\beta)}$ and $\eta = \left(\frac{G}{KT}\right)^{(2-\beta)/2}$, where

$$K^* = L\left(1 + \left(\frac{R}{\theta}\right)^{\beta}\right)\left(\frac{L}{\gamma}\right)^{\beta/(2-\beta)}.$$

Note that the surrogate rewards exactly track the true rewards when a stable reserve policy P_{y^*} is played, and so our regret with respect to the best stable reserve policy P_{y^*} is at most

$$\begin{aligned}
\sum_{t=1}^{T} f_t(P_{y^*}) - \sum_{t=1}^{T} f_t(y_t) &\leq \mathsf{Reg}_T(f^*) + \sum_{t=1}^{T} f_t^*(y_t) - f_t(p_t, x_t) \\
&\leq \mathsf{Reg}_T(f^*) + \sum_{t=1}^{T} \langle \nabla v(y_t), \theta y_t - x_t \rangle - c_t(\theta y_t) + c_t(x_t) \\
&\leq \mathsf{Reg}_T(f^*) + \sum_{t=1}^{T} (1-\theta_t)\left(\langle \nabla v(y_t), y_{t-1} - y_t \rangle + L\left\|y_t - y_{t-1}\right\|\right) \\
&\qquad\qquad (x_t = (1-\theta_t) y_{t-1}) \\
&\leq \mathsf{Reg}_T(f^*) + \sum_{t=1}^{T} \left(\langle \nabla v(y_t), y_t - (2y_t - y_{t-1}) \rangle + L\left\|y_t - y_{t-1}\right\|\right) \\
&\leq \mathsf{Reg}_T(f^*) + \sum_{t=1}^{T} v(y_t) - v(2y_t - y_{t-1}) + L\left\|y_t - y_{t-1}\right\| \\
&\qquad\qquad \text{(concavity of } v) \\
&\leq \mathsf{Reg}_T(f^*) + \sum_{t=1}^{T} 2L\left\|y_t - y_{t-1}\right\|^{\beta} \qquad \text{(Hölder, } \|y_t - y_{t-1}\| \leq 1) \\
&\leq \mathsf{Reg}_T(f^*) + 2TL\left(\frac{\eta L}{\gamma}\right)^{\beta/(2-\beta)} \\
&\leq 2L\left(\frac{G}{\gamma}\right)^{\beta/2}\left(T\left(3 + \left(\frac{R}{\theta}\right)^{\beta}\right)\right)^{(2-\beta)/2}
\end{aligned}$$

upon updating K^* to K as

$$K = L\left(3 + \left(\frac{R}{\theta}\right)^{\beta}\right)\left(\frac{L}{\gamma}\right)^{\beta/(2-\beta)},$$

which yields the theorem. □

Theorem 27 follows directly from Theorem 41.

C.6 Background and Proofs for Section 4.4.4: Steering Learners

C.6.1 Background

While much of the literature related to no-regret learning in general-sum games considers either rates of convergence to (coarse) correlated equilibria [13, 11] or welfare guarantees for such equilibria [140, 14], a recent line of work [73, 92, 114] has considered the question of *optimizing* one's reward when playing against a no-regret learner. A target benchmark which has emerged for this problem is the value of the *Stackelberg* equilibrium of a game (the optimal mixed strategy to "commit to", assuming an opponent best responds), which was shown by attainable by [92] against any no-regret algorithm and optimal in many cases (e.g. for no-swap learners), both up to $o(T)$ terms, and further which may yield higher reward for the optimizer than (coarse) correlated equilibria.

We show a class of instances for which the problem for optimizing reward against a learner playing according to gradient descent can be formulated as a locally controllable instance of online nonlinear control with adversarial perturbations and surrogate state-based losses. The simplest non-trivial instances we consider are those where the optimizer's reward is a function only of the learner's actions (i.e. all rows of their reward matrix are identical), and the optimization problem amounts to *steering* the learner to a desired strategy via one's choice of actions. Additionally, we allow the game matrices to change over time, which has not been substantially considered in prior work to our knowledge. We require that the learner's matrices do not change too quickly (which we model as adversarial disturbances to dynamics), and the optimizer's matrices can change arbitrarily provided that they remain close to *some* row-identical matrix (which we model as imprecision in our surrogate loss function).

C.6.2 Model

Here we are tasked with playing a sequence of bimatrix games against a no-regret learning opponent, where the game matrices may change adversarially in each round. We assume the

following properties hold for the adversarial sequence of games.

Assumption 6. *For a sequence* $\{(A_t, B_t) : t \in [T]\}$ *of* $m \times n$ *bimatrix games, with* $m > n$*:*

- *Each entry of* A_t *and* B_t *lies in* $[-\frac{L}{2\sqrt{n}}, \frac{L}{2\sqrt{n}}]$
- *the convex hull of the of the rows of each* B_t *contains the unit ball in* $\mathbb{R}^n$*,*
- $\left\|xA_t - xA_t^*\right\| \leq \delta_t$ *for any* $x \in \Delta(m)$*, where each row of* A_t^* *is identical, and*
- $\|xB_t - xB_{t-1}\| \leq \epsilon_t$ *for any* $x \in \Delta(m)$*.*

Each game (A_t, B_t) is revealed after Players A and B commit to their respective strategies $x_t \in \Delta(m)$ and $y_t \in \Delta(n)$. Observe that due to the first property, for any $z \in \mathcal{B}_1(\mathbf{0})$, there is some $x \in \Delta(m)$ such that $xB = z$. By the second property, we have that $xA_t^* = x'A_t^*$ for any $x, x' \in \Delta(m)$.

We recall the Online Gradient Descent algorithm with convex losses ℓ_t from [141].

Algorithm 16 Online Gradient Descent (OGD)

Input: Convex set $\mathcal{Y} \subseteq \mathbb{R}^n$, initial point $y_1 \in \mathcal{Y}$, and step sizes $\theta_1, \ldots, \theta_T$.
for $t = 1$ to T **do**
  Play y_t and observe loss $\ell_t(y_t)$.
  Set $\nabla_t = \nabla \ell_t(y_t)$.
  Set $y_{t+1} = \Pi_{\mathcal{Y}}(y_t - \theta_t \nabla_t) = \operatorname{argmin}_{y \in \mathcal{Y}} \|y_t - \theta_t \nabla_t - y\|$.
end for

Proposition 21 ([141])**.** *For differentiable convex losses* $\ell_t : \mathcal{Y} \to \mathbb{R}$*, with* $\theta_{t+1} \leq \theta_t$ *for each* $t \leq T$*, then for all* $y^* \in \mathcal{Y}$ *the regret of* OGD *is bounded by*

$$\sum_{t=1}^{T} \ell_t(y_t) - \ell_t(y^*) \leq \frac{2R_B^2}{\theta_T} + \sum_{t=1}^{T} \frac{\theta_t}{2} \|\nabla_t\|^2,$$

where R_B *is the radius of* $\mathcal{Y}$*. If* $\|\nabla_t\| \leq G_B$ *and* $\theta_t = \frac{2R_B}{G_B\sqrt{T}}$ *for all* $t \leq T$*, we have that*

$$\sum_{t=1}^{T} \ell_t(y_t) - \ell_t(y^*) \leq 2R_B G_B \sqrt{T}.$$

We assume that Player B plays according to OPGD in our setup, with $y_1 = \mathbf{u}_n$ and $\theta = \frac{R_B}{G_B\sqrt{T}}$. At each round t, we (Player A) choose some mixed strategy $x_t \in \Delta(n)$, and Player B plays some mixed strategy $y_t \in \Delta(n)$. Utilities for each player are given by the game (A_t, B_t) as

$$u_t^A(x_t, y_t) = x_t A_t y_t;$$
$$u_t^B(x_t, y_t) = x_t B_t y_t.$$

Note that the loss gradient $-\nabla u_t^B(x_t, y_t)$ each round for Player B (for negative utilities) is given by

$$\nabla_t = -x_t B,$$

and so their mixed strategy is updated at each round according to

$$y_t = \Pi_{\Delta(n)}\left(y_{t-1} + \theta(x_{t-1}B_{t-1})\right).$$

Our utility is given by $x_t A_t y_t = \mathbf{u}_n A_t^* y_t + x_t(A_t - A_t^*)y_t$, as x_t does not affect rewards from A_t^*. We benchmark the regret of an algorithm $\mathcal{A}$ against the optimal profile $(x, y) \in \Delta(m) \times \Delta(n)$:

$$\mathsf{Reg}_T(\mathcal{A}) = \max_{(x,y)\in\Delta(m)\times\Delta(n)} \sum_{t=1}^{T} xA_t y - x_t A_t y_t.$$

Note that the per-round average utility for the maximizing (x, y) is at least as high as that obtained by the Stackelberg equilibrium of the average game $\left(\sum_t \frac{A_t}{T}, \sum_t \frac{B_t}{T}\right)$, as for this objective one can choose both players' strategies without restriction. We remark that finding the Stackelberg equilibrium for any fixed game (A_t^*, B_t) in our setting, where A_t^* has identical rows, is straightforward: it suffices to optimize over $[n]$, as any fixed action $j \in [n]$ is a best response to some $x \in \Delta(m)$ by our assumption on the rows of B_t, and as our rewards are only a function of Player B's strategy y. However, we are not aware of any prior work which enables competing with the average-game Stackelberg value against a learning opponent when games arrive online.

C.6.3 Analysis

We first show that the problem can be formulated via known, strongly θ-locally controllable dynamics with adversarial disturbances. As B_t changes slowly between rounds, we can run NestedOCO-UD with disturbances representing the error resulting from assuming that B_t does not change from B_{t-1}.

Lemma 30. *Given the knowledge available prior to selecting x_t, updates for y_{t+1} can be expressed via known action-linear dynamics $(\mathcal{X}, \mathcal{Y}, D_t)$ which satisfy strong θ-local controllability, and with adversarial disturbances w_t satisfying $\sum_{t=1}^{T} \|w_t\| \leq \theta \sum_{t=1}^{T} \epsilon_t$.*

Proof. First, note that we can compute Player B's current strategy y_t, as it is a function only of games and strategies up to round $t-1$, all of which are observable. Given the update rule for OGD, we can formulate the dynamics $D_t(x_t, y_t)$ update as

$$\begin{aligned} D_t(x_t, y_t) &= \Pi_{\Delta(n)} \left(y_t + \theta(x_t B_t)\right) \\ &= \Pi_{\Delta(n)} \left(y_t + \theta(x_t B_{t-1}) + \theta(x_t(B_t - B_{t-1}))\right) \\ &= \Pi_{\Delta(n)} \left(y_t + \theta(x_t B_{t-1})\right) + w_t \end{aligned}$$

where w_t represents the error from assuming $B_t = B_{t-1}$. by standard properties of Euclidean projection, and the change bound on B_t, we have that $\|w_t\| \leq \|\theta(x_t(B_t - B_{t-1}))\| \leq \theta\epsilon_t$. Further, the update is action-linear (up to projection, prior to w_t).

To see that D_t satisfies strong θ-local controllability, we recall that the convex hull of the rows of B_{t-1} contain the unit ball, and so for any y^* in $\mathcal{B}_\theta(y_t) \cap \Delta(n)$ there is some $x_t \in \Delta(m)$ such that $\theta(x_t B_{t-1}) = y^* - y_t$. □

At round each round t, our loss is given by $f_t(x_t, y_t) = -x_t A_t y_t$. There are two barriers to running our algorithm. First, the update for y_t is determined by x_{t-1} and not x_t, yet we do not see A_{t-1} prior to selecting x_{t-1}, which would be required to take the appropriate step following f_{t-1}. Second, the loss depends on x_t in addition to y_t. To address both issues, we instead run NestedOCO-UD with

surrogate losses $\tilde{f}_t(\tilde{y}_t) = -\mathbf{u}_n A_{t-1} y_t$, with action rounds relabeled to account for the fact that x_{t-1} influences the step for y_t (which does not change the behavior of the algorithm). We set $A_0 = \mathbf{0}_{m,n}$.

Theorem 42. *Repeated play against an opponent using* OGD *with step size $\theta = \Theta(T^{-1/2})$ in a sequence of games (A_t, B_t) satisfying Assumption 6 can be cast as an instance of online control with strongly θ-locally controllable dynamics, for which the regret of* NestedOCO-UD *is at most*

$$\mathsf{Reg}_T(\textsc{NestedOCO-UD}) \le O\left(\sqrt{T} + \sum_{t=1}^{T}(\delta_t + \epsilon_t)\right),$$

with efficient per-round computation.

Proof. We first analyze regret with respect to the surrogate losses $\tilde{f}_t(y_t)$. To run NestedOCO-UD for $\alpha > 0$, it suffices to calibrate the step size for the internal FTRL instance such that $\eta\frac{L}{\gamma} \le \theta\alpha$. Given that rewards are bounded in $[-\frac{L}{2\sqrt{n}}, \frac{L}{2\sqrt{n}}]$, we have that each $x_t B_t y_t$ is $\frac{L}{\sqrt{n}}$-Lipschitz for the ℓ_1 norm, and thus L-Lipschitz for the ℓ_2 norm, so we can take $G_B = L$. Further, the ℓ_2 radius of $\Delta(n)$ is $R_B = \sqrt{2}/2$, and so we have that

$$\theta = \sqrt{\frac{2}{L^2 T}}.$$

Then, for a strongly θ-locally controllable instance with total perturbation bound $\sum_{t=1}^{T} \|w_t\| \le E$, we obtain the regret bound

$$\mathsf{Reg}_T(\textsc{NestedOCO-UD}) \le \eta\frac{TL^2}{\gamma} + \frac{G}{\eta} + \frac{2LRE}{(1-\alpha)\theta} \qquad \text{(Thm. 35)}$$

for any

$$\eta \le \min\left(\sqrt{\frac{G\gamma}{L^2 T}}, \alpha\sqrt{\frac{2}{T}}\right).$$

By Lemma 30, we can efficiently run NestedOCO-UD over the surrogate losses $\tilde{f}_t$ and bound

regret with respect to any $y^* \in \mathcal{Y}$ as:

$$\sum_{t=1}^{T} \tilde{f}_t(y_t) - \tilde{f}_t(y^*) \leq \eta \frac{TL^2}{\gamma} + \frac{G}{\eta} + \frac{\sqrt{2}L \cdot \sum_{t=1}^{T} \epsilon_t}{1-\alpha}.$$

Further, we can bound the error from the surrogate losses as

$$\begin{aligned}
\sum_{t=1}^{T} f_t(x_t, y_t) - \tilde{f}_t(y_t) &= \sum_{t=1}^{T} f_t(x_t, y_t) - f_{t-1}(\mathbf{u}_n, y_t) \\
&\leq \frac{L}{2\sqrt{n}} + \sum_{t=1}^{T-1} f_t(x_t, y_t) - f_t(\mathbf{u}_n, y_{t+1}) \quad (f_0(\mathbf{u}_n, y_1) = 0,\ f_T(x_T, y_T) \leq \tfrac{L}{2\sqrt{n}}) \\
&\leq \frac{L}{2\sqrt{n}} + \eta \frac{TL^2}{\gamma} + \sum_{t=1}^{T-1} x_t(A_t - A_t^*) y_t \qquad \text{(Prop. 15)} \\
&\leq \frac{L}{2\sqrt{n}} + \eta \frac{TL^2}{\gamma} + \sum_{t=1}^{T} \delta_t, \qquad \text{(Assumption 6, Cauchy-Schwarz)}
\end{aligned}$$

and likewise, for any $(x^*, y^*) \in \Delta(m) \times \Delta(n)$ we can bound

$$\begin{aligned}
\sum_{t=1}^{T} \tilde{f}_t(y^*) - f_t(x^*, y^*) &\leq -f_T(x^*, y^*) - \sum_{t=1}^{T-1} x^*(A_t - A_t^*) y^* \\
&\leq \frac{L}{2\sqrt{n}} + \sum_{t=1}^{T} \delta_t.
\end{aligned}$$

Combining the previous results, we have that for any $(x^*, y^*) \in \Delta(m) \times \Delta(n)$, the regret of NestedOCO-UD with respect to the true losses is bounded by

$$\begin{aligned}
\sum_{t=1}^{T} f_t(x_t, y_t) - f_t(x^*, y^*) &\leq \sum_{t=1}^{T} \tilde{f}_t(\tilde{y}_t) - \tilde{f}_t(y^*) + \sum_{t=1}^{T} f_t(x_t, y_t) - \tilde{f}_t(y_t) + \sum_{t=1}^{T} \tilde{f}_t(y^*) - f_t(x^*, y^*) \\
&\leq \eta \frac{2TL^2}{\gamma} + \frac{G}{\eta} + \frac{L}{\sqrt{n}} + 2\sum_{t=1}^{T} \delta_t + \frac{\sqrt{2}L \cdot \sum_{t=1}^{T} \epsilon_t}{1-\alpha} \\
&\leq 3 \cdot \max\left(\sqrt{\frac{TGL^2}{\gamma}}, \sqrt{\frac{T}{2\alpha^2}} \right) + \frac{L}{\sqrt{n}} + 2\sum_{t=1}^{T} \delta_t + \frac{\sqrt{2}L \cdot \sum_{t=1}^{T} \epsilon_t}{1-\alpha}
\end{aligned}$$

for any $\alpha \in (0, 1)$, which yields the theorem; Theorem 28 then follows directly. □