Adaptive Discretization for Adversarial Lipschitz Bandits

Chara Podimata PODIMATA@G.HARVARD.EDU
Harvard University

Aleksandrs Slivkins SLIVKINS@MICROSOFT.COM
Microsoft Research New York City

Editors: Mikhail Belkin and Samory Kpotufe

Abstract

Lipschitz bandits is a prominent version of multi-armed bandits that studies large, structured action spaces such as the $[0, 1]$ interval, where similar actions are guaranteed to have similar rewards. A central theme here is the adaptive discretization of the action space, which gradually "zooms in" on the more promising regions thereof. The goal is to take advantage of "nicer" problem instances, while retaining near-optimal worst-case performance. While the stochastic version of the problem is well-understood, the general version with adversarial rewards is not. We provide the first algorithm for adaptive discretization in the adversarial version, and derive instance-dependent regret bounds. In particular, we recover the worst-case optimal regret bound for the adversarial version, and the instance-dependent regret bound for the stochastic version.

A version with full proofs (and additional results) appears at `arxiv.org/abs/2006.12367v2`.

Keywords: Lipschitz bandits, adaptive discretization

1. Introduction

Multi-armed bandits is a simple yet powerful model for decision-making under uncertainty, extensively studied since 1950ies and exposed in several books, *e.g.,* (Bubeck and Cesa-Bianchi, 2012; Slivkins, 2019; Lattimore and Szepesvári, 2020). In a basic version, the algorithm repeatedly chooses actions (a.k.a. arms) from a fixed action space, and observes their rewards. Only rewards from the chosen actions are revealed, leading to the exploration-exploitation tradeoff.

We focus on *Lipschitz bandits*, a prominent version that studies large, structured action spaces such as the $[0, 1]$ interval. Similar actions are guaranteed to have similar rewards, as per Lipschitz-continuity or a similar condition. In applications, actions can correspond to items with feature vectors, such as products, documents or webpages; or to offered prices for buying, selling or hiring; or to different tunings of a complex system such as a datacenter or an ad auction.

A key theme here is *adaptive discretization* of the action space which gradually "zooms in" on the more promising regions thereof (Kleinberg et al., 2008, 2019; Bubeck et al., 2008, 2011a; Slivkins et al., 2013; Slivkins, 2014; Munos, 2011; Slivkins, 2011; Valko et al., 2013a; Minsker, 2013; Bull, 2015; Ho et al., 2016; Grill et al., 2015). This approach takes advantage of "nice" problem instances – ones in which near-optimal arms are confined to a relatively small region of the action space – while retaining near-optimal worst-case performance. The point of departure for all this work is *uniform discretization* (Kleinberg and Leighton, 2003; Kleinberg, 2004; Kleinberg et al., 2008, 2019; Bubeck et al., 2008, 2011a), a simple algorithm which discretizes the action space uniformly and obtains worst-case optimal regret bounds.

All prior work on Lipschitz bandits concerns the *stochastic* version, in which the rewards of each action are drawn from the same, albeit unknown, distribution in each round. In contrast, *adversarial bandits* allow the rewards to be adversarially chosen. This version is also widely studied in the literature (starting from Auer et al., 2002b), and tends to be much more challenging. The adversarial version of Lipschitz bandits is not understood beyond uniform discretization.

We provide the first algorithm for adaptive discretization in adversarial Lipshitz bandits, and derive instance-dependent regret bounds.[1] Our regret bounds are optimal in the worst case, and improve dramatically when the near-optimal arms comprise a small region of the action space. In particular, we recover the instance-dependent regret bound for the stochastic version of the problem (Kleinberg et al., 2008, 2019; Bubeck et al., 2008, 2011a).

Problem Statement: Adversarial Lipschitz Bandits. We are given a set $\mathcal{A}$ of actions (a.k.a. *arms*), the time horizon T, and a metric space $(\mathcal{A}, \mathcal{D})$, also called the *action space*. The adversary chooses randomized reward functions $g_1, \ldots, g_T : \mathcal{A} \to [0,1]$. In each round t, the algorithm chooses an arm $x_t \in \mathcal{A}$ and observes reward $g_t(x_t) \in [0,1]$ and nothing else. We focus on the *oblivious adversary*: all reward functions are chosen before round 1. The adversary is restricted in that the expected rewards $\mathbb{E}[g_t(\cdot)]$ satisfy the Lipschitz condition:[2]

$$\mathbb{E}\left[\, g_t(x) - g_t(y) \,\right] \leq \mathcal{D}(x,y) \quad \forall x, y \in \mathcal{A},\, t \in [T]. \tag{1.1}$$

The algorithm's goal is to minimize *regret*, a standard performance measure in multi-armed bandits:

$$R(T) := \sup_{x\in\mathcal{A}} \; \textstyle\sum_{t\in[T]} g_t(x) - g_t(x_t). \tag{1.2}$$

A problem instance consists of action space $(\mathcal{A}, \mathcal{D})$ and reward functions $g_1, \ldots, g_T$. The stochastic version of the problem (*stochastic rewards*) posits that each g_t is drawn independently from some fixed but unknown distribution $\mathcal{G}$. A problem instance is then the tuple $(\mathcal{A}, \mathcal{D}, \mathcal{G}, T)$.

The canonical examples are a d-dimensional unit cube $(\mathcal{A}, \mathcal{D}) = ([0,1]^d, \ell_p)$, $p \geq 1$ (where $\ell_p(x,y) = \|x-y\|_p$ is the p-norm), and the *exponential tree metric*, where $\mathcal{A}$ is a leaf set of a rooted infinite tree, and the distance between any two leaves is exponential in the height of their least common ancestor. Our results are equally meaningful for large but finite action sets.

Our results are most naturally stated without an explicit Lipschitz constant L. The latter is implicitly "baked into" the metric $\mathcal{D}$, *e.g.,* if the action set is $[0,1]$ one can take $\mathcal{D}(x,y) = L\,|x-y|$. However, we investigate the dependence on L in corollaries. Absent L, one can take $\mathcal{D} \leq 1$ w.l.o.g.

Our Results. We present ADVERSARIALZOOMING, an algorithm for adaptive discretization of the action space. Our main result is a regret bound of the form

$$\mathbb{E}[R(T)] \leq \widetilde{\mathcal{O}}(\, T^{(z+1)/(z+2)} \,), \tag{1.3}$$

where $z = \texttt{AdvZoomDim} \geq 0$ is a new quantity called the *adversarial zooming dimension*.[3] This quantity, determined by the problem instance, measures how wide-spread the near-optimal arms are in the action space. In fact, we achieve this regret bound with high probability.

[1] That is, regret bounds which depend on the properties of the problem instance that are not known initially.

[2] The expectation in (1.1) is over the randomness in the reward functions. While adversarial bandits are often defined with deterministic reward functions, it is also common to allow randomness therein, *e.g.,* to include stochastic bandits as a special case. The said randomness is essential to include *stochastic* Lipschitz bandits as a special case. Indeed, for stochastic rewards, (1.1) specializes to the Lipschitz condition from prior work on stochastic Lipschitz bandits. A stronger Lipschitz condition $g_t(x) - g_t(y) \leq \mathcal{D}(x,y)$ is unreasonable for many applications; *e.g.,* if the rewards correspond to user's clicks or other discrete signals, we can only assume Lipschitzness "on average".

[3] As usual, the $\widetilde{\mathcal{O}}(\cdot)$ and $\widetilde{\Omega}(\cdot)$ notation hides $\mathrm{polylog}(T)$ factors.

The meaning of this result is best seen via corollaries:

- We recover the optimal *worst-case* regret bounds for the adversarial version. Prior work (Kleinberg, 2004; Kleinberg et al., 2008, 2019; Bubeck et al., 2008, 2011a) obtains Eq. (1.3) for the d-dimensional unit cube, and more generally Eq. (1.3) with $z = \texttt{CovDim}$, the covering dimension of the action space. The latter bound is the best possible for any given action space. We recover it in the sense that $\texttt{AdvZoomDim} \leq \texttt{CovDim}$. Moreover, we match the worst-case optimal regret $\widetilde{\mathcal{O}}(\sqrt{KT})$ for instances with $K < \infty$ arms and any metric space.
- We recover the optimal *instance-dependent* regret bound from prior work on the stochastic version (Kleinberg et al., 2008, 2019; Bubeck et al., 2008, 2011a). This bound is Eq. (1.3) with $z = \texttt{ZoomDim}$, an instance-dependent quantity called the *zooming dimension*, and it is the best possible for any given action space and any given value of ZoomDim (Slivkins, 2014). ZoomDim can be anywhere between 0 and CovDim, depending on the problem instance. We prove that, essentially, $\texttt{AdvZoomDim} = \texttt{ZoomDim}$ for stochastic rewards.
- Our regret bound can similarly improve over the worst case even for adversarial rewards. In particular, we may have $\texttt{AdvZoomDim} = 0$ for arbitrarily large CovDim, even if the reward functions change substantially. Then we obtain $\widetilde{\mathcal{O}}(\sqrt{T})$ regret, as if there were only two arms.

Adaptive discretization algorithms from prior work (Kleinberg et al., 2008, 2019; Bubeck et al., 2008, 2011a) do not extend to the adversarial version. For example, specializing to K-armed bandits with uniform metric $\mathcal{D} \equiv 1$, these algorithms reduce to a standard algorithm for stochastic bandits (UCB1, Auer et al., 2002a), which fails badly for many simple instances of adversarial rewards.

Adversarial Zooming Dimension. The new notion of AdvZoomDim can be defined in a common framework with CovDim and ZoomDim from prior work. All three notions are determined by the problem instance, and talk about *set covers* in the action space. Each notion specifies particular subset(s) of arms to be covered, denoted $\mathcal{A}_\varepsilon \subset \mathcal{A}$, $\varepsilon > 0$, and counts how many "small" subsets are needed to cover each $\mathcal{A}_\varepsilon$. For a parameter $\gamma > 0$ called the *multiplier*, the respective "dimension" is

$$\inf \left\{ d \geq 0 : \ \mathcal{A}_\varepsilon \text{ can be covered with } \gamma \cdot \varepsilon^{-d} \text{ sets of diameter at most } \varepsilon, \quad \forall \varepsilon > 0 \right\}. \tag{1.4}$$

Generally, a small "dimension" quantifies the simplicity of a problem instance.

The covering dimension CovDim has $\mathcal{A}_\varepsilon \equiv \mathcal{A}$. The intuition comes from the d-dimensional cube, for which $\texttt{CovDim} = d$. [4] Thus, we are looking for the covering property enjoyed by the unit cube. Note that CovDim is determined by the action space alone, and is therefore known to the algorithm.

Both ZoomDim and AdvZoomDim are about covering near-optimal arms. Each subset $\mathcal{A}_\varepsilon$ comprises all arms that are, in some sense, within ε from being optimal. These subsets may be easier to cover compared to $\mathcal{A}$; this may reduce (1.4) compared to the worst case of CovDim.

The zooming dimension ZoomDim is only defined for stochastic rewards. It focuses on the standard notion of *stochastic gap* of an arm x compared to the best arm: $\texttt{Gap}(x) := \max_{y \in \mathcal{A}} \mathbb{E}[g_t(y)] - \mathbb{E}[g_t(x)]$. Each subset $\mathcal{A}_\varepsilon$ is defined as the set of all arms $x \in \mathcal{A}$ with $\texttt{Gap}(x) \leq O(\varepsilon)$.

AdvZoomDim extends ZoomDim as follows. The *adversarial gap* of a given arm x measures this arm's suboptimality compared to the best arm on a given time-interval $[0, t]$. Specifically,

$$\texttt{AdvGap}_t(x) := \tfrac{1}{t} \max_{y \in \mathcal{A}} \textstyle\sum_{\tau \in [t]} g_\tau(y) - g_\tau(x). \tag{1.5}$$

[4] More formally, the covering dimension of $([0, 1]^d, \ \ell_p)$, $p \geq 1$ is d, with multiplier $\gamma = \text{poly}(d, p)$.

Given $\varepsilon > 0$, an arm x is called inclusively ε-optimal if $\texttt{AdvGap}_t(x) < \mathcal{O}(\varepsilon \ln^{3/2} T)$ for some end-time $t > \Omega(\varepsilon^{-2})$; the precise definition is spelled out in Eq. (3.1). In words, we include all arms whose adversarial gap is sufficiently small at some point in time. It suffices to restrict our attention to a *representative set* of arms $\mathcal{A}_{\texttt{repr}} \subset \mathcal{A}$ with $|\mathcal{A}_{\texttt{repr}}| \leq O(T^{1+\texttt{CovDim}})$, specified in the analysis.[5] Thus, the subset $\mathcal{A}_\varepsilon$ is defined as the set of all arms $x \in \mathcal{A}_{\texttt{repr}}$ that are inclusively ε-optimal.

By construction, $\texttt{AdvZoomDim} \leq \texttt{CovDim}$ for any given multiplier $\gamma > 0$. For stochastic rewards, $\texttt{AdvZoomDim}$ coincides with $\texttt{ZoomDim}$ up to a polylog $\big(T,\, |\mathcal{A}_{\texttt{repr}}|\big)$ multiplicative change in γ.

The definition of $\texttt{AdvZoomDim}$ is quite flexible. First, we achieve the stated regret bound for all $\gamma > 0$ at once, with a multiplicative $\gamma^{1/(z+2)}$ dependence thereon. Second, we could relax (1.4) to hold only for ε smaller than some threshold θ; the regret bound increases by $+\widetilde{\mathcal{O}}(\sqrt{T}\,\theta^{-\texttt{CovDim}})$.

Examples. We provide a flexible family of examples with small $\texttt{AdvZoomDim}$. Fix an arbitrary action space $(\mathcal{A}, \mathcal{D})$ and time horizon T. Consider M problem instances with stochastic rewards, each with $\texttt{ZoomDim} \leq d$. Construct an instance with adversarial rewards, where each round is assigned in advance to one of these stochastic instances. This assignment can be completely arbitrary: *e.g.,* the stochastic instances can appear consecutively in "phases" of arbitrary duration, or they can be interleaved in an arbitrary way. Then $\texttt{AdvZoomDim} \leq d$ for constant M, d under some assumptions.

In particular, we allow arbitrary disjoint subsets $S_1\,,\;\dots\;,S_M \subset \mathcal{A}$ such that each stochastic instance $i \in [M]$ can behave arbitrarily on S_i as long as the spread between the largest and smallest mean rewards exceeds a constant. All arms outside S_i receive the same "baseline" mean reward, which does not exceed the mean rewards inside S_i. The analysis of this example is somewhat non-trivial, and separate from the main regret bound (1.3).

Challenges and Techniques. We build on the high-level idea of *zooming* from prior work on the stochastic version (Kleinberg et al., 2008, 2019; Bubeck et al., 2008, 2011a), but provide a very different implementation of this idea. We maintain a partition of the action space into "active regions", and refine this partition adaptively over time. We "zoom in" on a given region by partitioning it into several "children" of half the diameter; we do it only if the sampling uncertainty goes below the region's diameter. In each round, we select an active region according to (a variant of) a standard algorithm for bandits with a fixed action set, and then sample an arm from the selected region according to a fixed, data-independent rule. The standard algorithm we use is $\texttt{EXP3.P}$ (Auer et al., 2002b); prior work on stochastic rewards used $\texttt{UCB1}$ (Auer et al., 2002a).

Adversarial rewards bring about several challenges compared to the stochastic version. First, the technique in $\texttt{EXP3.P}$ does not easily extend to variable number of arms, or to increasing the action set by "zooming" (whereas the technique in $\texttt{UCB1}$ does, for stochastic rewards). Second, the sampling uncertainty is not directly related to the total probability mass allocated to a given region. In contrast, this relation is straightforward and crucial for the stochastic version. Third, the adversarial gap is much more difficult to work with. Indeed, the analysis for stochastic rewards relies on two crucial but easy steps — bounding the gap for regions with small sampling uncertainty, and bounding the "total damage" inflicted by all small-gap arms — which no longer work for adversarial rewards.

These challenges prompt substantial complications in the algorithm and the analysis. For example, to incorporate "children" into the multiplicative weights analysis, we split the latter into two steps: first we update the weights, then we add the children. To enable the second step, we partition the parent's weight equally among the children. Effectively, we endow each child with a copy of the parent's data, and we need to argue that the latter is eventually diluted by the child's own data.

[5] Essentially, $\mathcal{A}_{\texttt{repr}}$ contains a uniform discretization for scale $^1/_T$ and also the local optima for such discretization.

Another example: to argue that we only "zoom in" if the parent has small adversarial gap, we need to enhance the "zoom-in rule": in addition to the "aggregate" rule (the sampling uncertainty must be sufficiently small), we need the "instantaneous" one: the current sampling probability must be sufficiently large, and it needs to be formulated in just the right way. Then, we need to be much more careful about deriving the "zooming invariant", a crucial property of the partition enforced by the "zoom-in rule". In turn, this derivation necessitates the algorithm's parameters to change from round to round, which further complicates the multiplicative weights analysis.

An important part of our contribution is formalizing what we mean by "nice" problem instances, and boiling the analysis down to an easily interpretable notion such as $\texttt{AdvZoomDim}$.

Remarks. We obtain an *anytime* version, with similar regret bounds for all time horizons T at once, using the standard *doubling trick*: in each phase $i \in \mathbb{N}$, we restart the algorithm with time horizon $T = 2^i$. The only change is that the definition of $\texttt{AdvZoomDim}$ redefines $\mathcal{A}_\varepsilon$ to be the set of all arms that are inclusively ε-optimal within some phase.

Our regret bound depends sublinearly on the *doubling constant* $C_{\texttt{dbl}}$: the smallest $C \in \mathbb{N}$ such that any ball can be covered with C sets of at most half the diameter. Note that $C_{\texttt{dbl}} = 2^d$ for a d-dimensional unit cube, or any subset thereof. The doubling constant has been widely used in theoretical computer science, e.g., see (Kleinberg et al., 2009) for references.

Related Work. Lipschitz bandits are introduced in (Agrawal, 1995) for action space $[0, 1]$, and optimally solved in the worst case via uniform discretization in (Kleinberg, 2004; Kleinberg et al., 2008, 2019; Bubeck et al., 2008, 2011a). Adaptive discretization is introduced in (Kleinberg et al., 2008, 2019; Bubeck et al., 2008, 2011a), and subsequently extended to contextual bandits (Slivkins, 2014), ranked bandits (Slivkins et al., 2013), and contract design for crowdsourcing (Ho et al., 2016). (The terms "zooming algorithm/dimension" trace back to Kleinberg et al. (2008, 2019).) Kleinberg et al. (2008, 2019) consider regret rates with instance-dependent constant (*e.g.,* $\log(t)$ for finitely many arms), and build on adaptive discretization to characterize worst-case optimal regret rates for any given metric space. Pre-dating the work on adaptive discretization, Kocsis and Szepesvari (2006); Pandey et al. (2007); Munos and Coquelin (2007) allow a "taxonomy" on arms without any numerical information (and without any non-trivial regret bounds).

Several papers recover adaptive discretization guarantees under mitigated Lipschitz conditions: when Lipschitzness only holds near the best arm $x^\star$ or when one of the two arms is $x^\star$ (Kleinberg et al., 2008, 2019; Bubeck et al., 2008, 2011a); when the algorithm is only given a taxonomy of arms, but not the metric (Slivkins, 2011; Bull, 2015); when the actions correspond to contracts offered to workers, and no Lipschitzness is assumed (Ho et al., 2016), and when expected rewards are Hölder-smooth with an unknown exponent (Locatelli and Carpentier, 2018).

In other work on mitigating Lipschitzness, Bubeck et al. (2011b) recover the optimal worst-case bound with unknown Lipschitz constant. Munos (2011); Valko et al. (2013a); Grill et al. (2015) consider adaptive discretization in the "pure exploration" version, and allow for a parameterized class of metrics with unknown parameter. Krishnamurthy et al. (2020) posit a weaker, "smoothed" benchmark and recover adaptive discretization-like regret bounds without any Lipschitz assumptions.

All work discussed above assumes stochastic rewards. Adaptive discretization is extended to expected rewards with bounded change over time (Slivkins, 2014), and to a version with ergodicity and mixing assumptions (Azar et al., 2014). For Lipschitz bandits with adversarial rewards, the uniform discretization approach easily extends (Kleinberg, 2004), and *nothing else is known*.[6]

[6] We note that Maillard and Munos (2010) achieve $O(\sqrt{T})$ regret for the full-feedback version.

While Lipschitz bandits only capture "local" similarity between arms, other structural models such as convex, *e.g.,* (Flaxman et al., 2005; Agarwal et al., 2010, 2011; Saha and Tewari, 2011; Bubeck et al., 2015; Bubeck and Eldan, 2016; Bubeck et al., 2017) (resp., linear, *e.g.,* (Dani et al., 2008; Abbasi-Yadkori et al., 2011; Abernethy et al., 2008)) bandits allow for *long-range inferences*: by observing some arms, an algorithm learns something about other arms that are far away. This is why $\widetilde{\mathcal{O}}(\sqrt{T})$ regret rates are achievable in adversarial convex (resp., linear) bandits and to extensions thereof (*e.g.,* (Valko et al., 2013b, 2014)), via different techniques.

Organization. Our algorithm is presented in Section 2. Sections 3 and 4 state our results and outline the regret analysis. The examples are spelled out in Appendix 5. While the results in Section 3 are stated in full generality, we present the algorithm and the analysis for the special case of d-dimensional unit cube for ease of exposition. The extension to arbitrary metric spaces requires a careful decomposition of the action space, but no new ideas otherwise; it is outlined in Appendix 6. All omitted proofs can be found in the full version.

2. Our Algorithm: Adversarial Zooming

For ease of presentation, we develop the algorithm for the special case of d-dimensional unit cube, $(\mathcal{A}, \mathcal{D}) = ([0,1]^d,\ \ell_\infty)$. Our algorithm partitions the action space into axis-parallel hypercubes. More specifically, we consider a rooted directed tree, called the *zooming tree*, whose nodes correspond to axis-parallel hypercubes in the action space. The root is $\mathcal{A}$, and each node u has 2^d children that correspond to its quadrants. For notation, $\mathcal{U}$ is the set of all tree nodes, $\mathcal{C}(u)$ is the set of all children of node u, and $L(u) = \max_{x,y\in u} \mathcal{D}(x,y)$ is its diameter in the metric space; w.l.o.g. $L(\cdot) \le 1$.

On a high level, the algorithm operates as follows. We maintain a set $A_t \subset \mathcal{U}$ of tree nodes in each round t, called *active nodes*, which partition the action space. We start with a single active node, the root. After each round t, we may choose some node(s) u to "zoom in" on according to the *zoom-in rule*, in which case we de-activate u and activate its children. We denote this decision with $z_t(u) = \mathbb{1}\{\text{zoom in on } u \text{ at round } t\}$. In each round, we choose an active node U_t according to the *selection rule*. Then, we choose a representative arm $x_t = \texttt{repr}_t(U_t) \in U_t$ to play in this round. The latter choice can depend on t, but not on the algorithm's observations; the choice could be randomized, *e.g.,* we could choose uniformly at random from U_t.

The main novelty of our algorithm is in the zoom-in rule. However, presenting it requires some scaffolding: we need to present the rest of the algorithm first. The selection rule builds on EXP3 (Auer et al., 2002b), a standard algorithm for adversarial bandits. We focus on EXP3.P, a variant that uses "optimistic" reward estimates, the inverse propensity score (IPS) plus a "confidence term" (see Eq. (2.1)). This is because we need a similar "confidence term" from the zooming rule to "play nicely" with the EXP3 machinery. If we never zoomed in *and* used $\eta = \eta_t$ for multiplicative updates in each round, then our algorithm would essentially coincide with EXP3.P. Specifically, we maintain weights $w_{t,\eta}(u)$ for each active node u and round t, and update them multiplicatively, as per Eq. (2.2). In each round t, we define a probability distribution p_t on the active nodes, proportional to the weights w_{t,η_t}. We sample from this distribution, mixing in some low-probability uniform exploration.

We are ready to present the pseudocode (Algorithm 1). The algorithm has parameters $\beta_t, \gamma_t, \eta_t \in (0, 1/2]$ for each round t; we fix them later in the analysis as a function of t and $|A_t|$. Their meaning is that β_t drives the "confidence term", γ_t is the probability of uniform exploration, and η_t parameterizes the multiplicative update. To handle the changing parameters η_t, we use a trick from (Bubeck, 2010; Bubeck and Cesa-Bianchi, 2012): conceptually, we maintain the weights $w_{t,\eta}$ for all values of η

Algorithm 1: ADVERSARIALZOOMING

```
Parameters: $\beta_t, \gamma_t, \eta_t \in (0, 1/2]$    for each round $t$.
Variables: active nodes $A_t \subset \mathcal{U}$, weights $w_{t,\eta} : \mathcal{U} \to (0, \infty]$   $\forall$ round $t$, $\eta \in (0, 1/2]$
Initialization: $w_1(\cdot) = 1$ and $A_1 = \{\texttt{root}\}$ and $\beta_1 = \gamma_1 = \eta_1 = 1/2$.
for $t = 1, \ldots, T$ do
    $p_t \leftarrow \{$ distribution $p_t$ over $A_t$, proportional to weights $w_{t,\eta_t}$ $\}$.
    Add uniform exploration: distribution $\pi_t(\cdot) \leftarrow (\, 1 - \gamma_t \,)\; p_t(\cdot) + \gamma_t / |A_t|$ over $A_t$.
    Select a node $U_t \sim \pi_t(\cdot)$, and then its representative: $x_t = \texttt{repr}(U_t)$.      // selection rule
    Observe the reward $g_t(x_t) \in [0, 1]$.
    for $u \in A_t$ do
        $\widehat{g}_t(u) = \frac{g_t(x_t) \cdot \mathbb{1}\{u = U_t\}}{\pi_t(u)} + \frac{(1 + 4 \log T)\, \beta_t}{\pi_t(u)}$      // IPS + "conf term"      (2.1)
        $w_{t+1,\,\eta}(u) = w_{t,\eta}(u) \cdot \exp(\, \eta \cdot \widehat{g}_t(u)\,)\,,\ \forall \eta \in (0, 1/2]$      // MW update      (2.2)
        if $z_t(u) = 1$ then                                            // zoom-in rule
            $A_{t+1} \leftarrow A_t \cup \mathcal{C}(u) \setminus \{u\}$        // activate children of u, deactivate u
            $w_{t+1}(v) = w_{t+1}(u) / |\mathcal{C}(u)|$   for all $v \in \mathcal{C}(u)$.     // split the weight
```

simultaneously, and plug in $\eta = \eta_t$ only when we compute distribution p_t. Explicitly maintaining all these weights is cleaner and mathematically well-defined, so this is what our pseudocode does.

For the subsequent developments, we need to carefully account for the ancestors of the currently active nodes. Suppose node u is active in round t, and we are interested in some earlier round $s \leq t$. Exactly one ancestor of u in the zooming tree has been active then; we call it the *active ancestor* of u and denote $\texttt{act}_s(u)$. If u itself was active in round s, we write $\texttt{act}_s(u) = u$.

For computational efficiency, we do not explicitly perform the multiplicative update (2.2). Instead, we recompute the weights $w_{t,\,\eta_t}$ from scratch in each round t, using the following characterization:[7]

Lemma 1 *Let $\mathcal{C}_{\texttt{prod}}(u) = \prod_v |\mathcal{C}(v)|$, where v ranges over all ancestors of node u in the zooming tree (not including u itself). Then for all nodes $u \in A_t$, rounds t, and parameter values $\eta \in (0, 1/2]$,*

$$w_{t+1,\,\eta}(u) = \mathcal{C}^{-1}_{\texttt{prod}}(u) \cdot \exp\left(\eta \textstyle\sum_{\tau \in [t]} \widehat{g}_\tau(\texttt{act}_\tau(u))\right). \tag{2.3}$$

Remarks. We make no restriction on how many nodes u can be “zoomed-in” in any given round. However, our analysis implies that we cannot immediately zoom in on any “newborn children”.

When we zoom in on a given node, we split its weight equally among its children. Maintaining the total weight allows the multiplicative weights analysis to go through, and the equal split allows us to conveniently represent the weights in Lemma 1 (which is essential in the multiplicative weights analysis, too). An undesirable consequence is that we effectively endow each child with a copy of the parent's data; we deal with it in the analysis via Eq. (4.4).

The meaning of the confidence term in Eq. (2.1) is as follows. Define the *total confidence term*

$$\texttt{conf}^{\texttt{tot}}_t(u) := 1/\beta_t + \textstyle\sum_{\tau \in [t]}\ \beta_\tau / \pi_\tau(\texttt{act}_\tau(u)). \tag{2.4}$$

[7] We also use Lemma 1 in several places in the analysis.

Essentially, we upper-bound the cumulative gain from node u up to time t using

$$\texttt{conf}^{\texttt{tot}}_t(u) + \textstyle\sum_{\tau\in[t]} \texttt{IPS}_t(\texttt{act}_t(u)), \quad \text{where} \quad \texttt{IPS}_t(u) := g_t(x_t)\cdot \mathbb{1}\left\{u = U_t\right\}/\pi_t(u). \tag{2.5}$$

The $+4\log T$ term in Eq. (2.1) is needed to account for the ancestors later in the analysis; it would be redundant if there were no zooming and the active node set were fixed throughout.

The Zoom-In Rule. Intuitively, we want to zoom in on a given node u when its per-round sampling uncertainty gets smaller than its diameter $L(u)$, in which case exploration at the level of u is no longer productive. A natural way to express this is $\texttt{conf}^{\texttt{tot}}_t(u) \le t\cdot L(u)$, which we call the *aggregate* zoom-in rule. However, it does not suffice: we also need an *instantaneous* version which asserts that the current sampling probability is large enough. Making this precise is somewhat subtle. Essentially, we lower-bound $\texttt{conf}^{\texttt{tot}}_t(u)$ as a sum of "instantaneous confidence terms"

$$\texttt{conf}^{\texttt{inst}}_\tau(u) := \widetilde{\beta}_\tau + \beta_\tau/\pi_\tau(\texttt{act}_\tau(u)), \quad \tau\in[t], \tag{2.6}$$

where $\widetilde{\beta}_\tau \in (0, 1/2]$ are new parameters. We require each such term to be at most $L(u)$. In fact, we require a stronger upper bound $e^{L(u)}-1$, which plugs in nicely into the multiplicitative weights argument, and implies an upper bound of $L(u)$. Thus, the zoom-in rule is as follows:

$$z_t(u) := \mathbb{1}\left\{ \texttt{conf}^{\texttt{inst}}_t(u) \le e^{L(u)}-1 \right\}\cdot \mathbb{1}\left\{ \texttt{conf}^{\texttt{tot}}_t(u) \le t\cdot L(u) \right\} \tag{2.7}$$

Parameters $\widetilde{\beta}_\tau$ must be well-defined for all $\tau\in[0,T]$ and satisfy the following, for any rounds $t<t'$:

$$\left\{\ \widetilde{\beta}_\tau \text{ decreases in } \tau\ \right\} \text{ and } \left\{\ \widetilde{\beta}_t \ge \beta_t\ \right\} \text{ and } \textstyle\int_t^{t'} \widetilde{\beta}_\tau\, \mathrm{d}\tau \le \frac{1}{\beta_{t'}} - \frac{1}{\beta_t}. \tag{2.8}$$

We cannot obtain the third condition of Eq. (2.8) with equality because parameters β_t and $\widetilde{\beta}_t$ depend on $|A_t|$, and the latter is not related to t with a closed form solution.

3. Our Results

Running Time. The per-round running time of the algorithm is $\widetilde{\mathcal{O}}\left(T^{d/(d+2)}\right)$, where $d = \texttt{CovDim}$. Indeed, given Lemma 1, in each round t of the algorithm we only need to compute the weight $w_{t,\eta}(\cdot)$ for all active nodes and one specific $\eta=\eta_t$. This takes only $O(1)$ time per node (since we can maintain the total estimated reward $\sum_{\tau\in[t]} \widehat{g}_\tau(\texttt{act}_\tau(u))$ separately). So, the per-round running time is $O(|A_T|)$, which is at most $\widetilde{\mathcal{O}}\left(T^{d/(d+2)}\right)$, as we prove in Lemma **??**.

Regret Bounds. Our regret bounds are broken into three steps. First, we state the "raw" regret bound in terms of the algorithm's parameters, with explicit assumptions thereon. Second, we tune the parameters and derive the "intermediate" regret bound of the form $\widetilde{\mathcal{O}}(\sqrt{T\,|A_T|})$. Third, we derive the "final" regret bound, upper-bounding $|A_T|$ in terms of $\texttt{AdvZoomDim}$. For ease of presentation, we use failure probability $\delta = T^{-2}$; for any known $\delta>0$, regret scales as $\log 1/\delta$. The covering dimension is denoted d, for some constant multiplier $\gamma_0>0$ (we omit the $\log(\gamma_0)$ dependence). The precise definition of an inclusively ε-optimal arm in the definition of $\texttt{AdvZoomDim}$ is that

$$\texttt{AdvGap}_t(\cdot) < 30\,\varepsilon\,\ln(T)\,\sqrt{d\,\ln\left(C_{\texttt{dbl}}\cdot T\right)} \quad \text{for some end-time } t > \varepsilon^{-2}/9. \tag{3.1}$$

Theorem 2 *Assume the sequences $\{\eta_t\}$ and $\{\beta_t\}$ are decreasing in t, and satisfy*

$$\eta_t \leq \beta_t \leq \gamma_t/|A_t| \quad \textit{and} \quad \eta_t\,(\,1+\beta_t(1+4\log T)\,) \leq \gamma_t/|A_t|. \tag{3.2}$$

With probability at least $1-T^{-2}$, ADVERSARIALZOOMING *satisfies*

$$R(T) \leq \mathcal{O}(\ln T)\left(\sqrt{d\,T} + \frac{1}{\beta_T} + \frac{\ln(\,C_{\texttt{dbl}}\cdot|A_T|\,)}{\eta_T} + \textstyle\sum_{t\in[T]}\beta_t + \gamma_t\,\ln T\right) \tag{3.3}$$

$$\leq \mathcal{O}\left(\sqrt{T\,|A_T|}\right)\cdot\ln^2(T)\,\sqrt{d\,\ln(\,T\,|A_T|\,)\ln(\,C_{\texttt{dbl}}\,|A_T|\,)} \quad \textit{(tuning the parameters)} \tag{3.4}$$

$$\leq \mathcal{O}\left(T^{\frac{z+1}{z+2}}\right)\cdot\left(d^{1/2}\,(\,\gamma\,C_{\texttt{dbl}}\,)^{1/(z+2)}\ln^5 T\right), \tag{3.5}$$

where $d=$ $\texttt{CovDim}$ and $z=$ $\texttt{AdvZoomDim}$ with multiplier $\gamma>0$. The parameters in (3.4) are:

$$\beta_t = \widetilde{\beta}_t = \eta_t = \sqrt{2\ln(\,|A_t|\cdot T^3\,)\ln(\,C_{\texttt{dbl}}\cdot|A_t|\,)}\;\;/\;\;\sqrt{t\,|A_t|\;d\cdot\ln^2 T},$$
$$\gamma_t = (2+4\log T)\;|A_t|\cdot\beta_t. \tag{3.6}$$

Remark 3 *We can relax the definition of* $\texttt{AdvZoomDim}$ *so that (1.4) needs to hold only for scales ε smaller than some threshold θ. Then we obtain the regret bound in (3.5) plus $\widetilde{\mathcal{O}}\left(\sqrt{T}\,\theta^{-\texttt{CovDim}}\right)$.*

Special cases. First, we argue that for stochastic rewards $\texttt{AdvZoomDim}$ coincides with the zooming dimension $\texttt{ZoomDim}$ from prior work, up to a small change in the multiplier γ. (We specify the latter by putting it in the subscript.) The key is to relate each arm's stochastic gap to its adversarial gap.

Lemma 4 *Consider an instance of Lipschitz bandits with stochastic rewards. For any $\gamma>0$, with probability at least $1-{}^{1}\!/\!_{T}$ it holds that:*

$$\texttt{ZoomDim}_{\gamma\cdot f} \leq \texttt{AdvZoomDim}_{\gamma\cdot f} \leq \texttt{ZoomDim}_{\gamma}, \quad \textit{where } f = \left(O(\text{poly}(d)\;\ln^3 T)\right)^{\log(C_{\texttt{dbl}})-\texttt{ZoomDim}_{\gamma}}.$$

This lemma holds for any representative set $\mathcal{A}_{\texttt{repr}}$. Then the base in factor f scales with $\ln(|\mathcal{A}_{\texttt{repr}}|)$.

Second, for problem instances with $K<\infty$ arms, we recover the standard $\widetilde{\mathcal{O}}(\sqrt{KT})$ regret bound by observing that any problem instance has $\texttt{AdvZoomDim} = 0$ with multiplier $\gamma=K$ and $\mathcal{A}_{\texttt{repr}} = [K]$. ADVERSARIALZOOMING satisfies $R(T) \leq \mathcal{O}(\,\sqrt{KT}\cdot\sqrt{C_{\texttt{dbl}}}\cdot\ln^5 T\,)$ w.h.p.

Third, we analyze the dependence on the Lipschitz constant. Fix a problem instance, and multiply the metric by some $L>1$. The Lipschitz condition (1.1) still holds, and the definition of $\texttt{AdvZoomDim}$ implies that regret scales as $L^{z/(z+2)}$. This is optimal in the worst case by prior work.[8]

Corollary 5 *Fix a problem instance and a multiplier $\gamma>0$, and let $R_\gamma(T)$ denote the right-hand side of (3.5). Consider a modified problem instance with metric $\mathcal{D}' = L\cdot\mathcal{D}$, for some $L\geq 1$. Then* ADVERSARIALZOOMING *satisfies $R(T) \leq L^{z/(z+2)}\cdot R_\gamma(T)$, with probability at least $1-T^{-2}$.*

[8] For a formal statement, consider the unit cube with $\mathcal{A} = [0,1]^d$ and metric $D(x,y) = L\cdot\|x-y\|_p$, for some constants $d\in\mathbb{N}$ and $p\geq 1$. Then the worst-case optimal regret rate is $\tilde{\mathcal{O}}\left(L^{d/(d+2)}\cdot T^{(d+1)/(d+2)}\right)$. The proof for $d=1$ can be found, *e.g.,* in Ch. 4.1 of Slivkins (2019); the proof for $d>1$ can be derived similarly.

4. Regret Analysis (Outline)

We outline the key steps and the proof structure; the lengthy details are in the next section. For ease of presentation, we focus on the d-dimensional unit cube $(\mathcal{A},\mathcal{D}) = ([0,1]^d,\, \ell_\infty)$.

We start with some formalities. First, we posit a *representative arm* $\texttt{repr}_t(u) \in u$ for each tree node u and each round t, so that $x_t = \texttt{repr}_t(U_t)$. W.l.o.g., all representative arms are chosen before round 1. Thus, we can endow u with rewards $g_t(u) := g_t(\texttt{repr}_t(u))$. Second, let $x^\star_S(u) \in \arg\max_{x\in u} \sum_{t\in S} g_t(x)$ be the best arm in u over the set S of rounds (ties broken arbitrarily). Let $x^\star_S = x^\star_S(\mathcal{A})$ be the best arm over S. Let $u^\star_t$ be the active node at round t which contains $x^\star_{[t]}$.

The representative set $\mathcal{A}_{\texttt{repr}} \subset \mathcal{A}$ (used in the definition of $\texttt{AdvZoomDim}$) consists of arms $\texttt{repr}(u)$, $x^\star_{[t]}(u)$ for all tree nodes of height at most $1+\log T$ and all rounds t. Only these arms are invoked by the algorithm or the analysis. This enables us to transition to deterministic rewards that satisfy a certain "per-realization" Lipschitz property (Eq. (**??**) in the appendix).

Part I: Properties of the Zoom-In Rule. This part depends on the zoom-in rule, but not on the selection rule, *i.e.,* it works no matter how distribution π_t is chosen. First, the zoom-in rule ensures that all active nodes satisfy the following property, called the *zooming invariant*:

$$\texttt{conf}^{\texttt{tot}}_t(u) \geq (t-1)\cdot L(u) \quad \text{if node } u \text{ is active in round } t \tag{4.1}$$

It is proved by induction on t, using the fact that when a node does *not* get zoomed-in, this is because either instantaneous or the aggregate zoom-in rule does not apply.

Let us characterize the *lifespan* of node u: the time interval $[\tau_0(u),\, \tau_1(u)]$ during which the node is active. We lower-bound the deactivation time, using the instantaneous zoom-in rule:

$$\text{node } u \text{ is zoomed-in} \quad \Rightarrow \quad \tau_1(u) \geq 1/L(u). \tag{4.2}$$

It follows that only nodes of diameter $L(\cdot) \geq {}^1\!/_{2T}$ can be activated. Next, we show that a node's deactivation time is (approx.) at least twice as the parent's:

$$\text{node } u \text{ is zoomed-in} \quad \Rightarrow \quad \tau_1(u) \geq 2\,\tau_1(\texttt{parent}(u)) - 2. \tag{4.3}$$

We use this to argue that a node's own datapoints eventually drown out those inherited from the parent when the node was activated. Specifically:

$$\text{node } u \text{ is active at time } t \quad \Rightarrow \quad \tfrac{1}{t}\,\textstyle\sum_{\tau\in[t]}\, L(\texttt{act}_\tau(u)) \leq 4\,\log(T)\cdot L(u). \tag{4.4}$$

Next, we prove that the total probability mass spent on a zoomed-in node must be large:

$$\text{node } u \text{ is zoomed-in} \quad \Rightarrow \quad \mathcal{M}(u) := \textstyle\sum_{\tau=\tau_0(u)}^{\tau_1(u)} \pi_\tau(u) \geq \frac{1}{9\,L^2(u)} \tag{4.5}$$

This statement is essential for bounding the number of active nodes in Part IV. To prove it, we apply both the zooming invariant (4.1) and the (aggregate) zooming rule. Finally, the instantaneous zoom-in rule implies that the zoomed-in node is chosen with large probability:

$$\text{node } u \text{ is zoomed-in at round } t \quad \Rightarrow \quad \pi_t(u)/\pi_t(u^\star_t) \geq \beta_t^2/e^{L(u)}. \tag{4.6}$$

Part II: Multiplicative Weights. This part depends on the selection rule, but not on the zooming rule: it works regardless of how $z_t(u)$ is defined. We analyze the following potential function: $\Phi_t(\eta) = \left(\frac{1}{|A_t|}\sum_{u\in A_t} w_{t+1,\eta}(u)\right)^{1/\eta}$, where $w_{t+1,\eta}(u)$ is given by (2.3), with $\Phi_0(\cdot) = 1$.

We upper- and the lower-bound the telescoping product

$$Q := \ln\left(\frac{\Phi_T(\eta_T)}{\Phi_0(\eta_0)}\right) = \ln\left(\prod_{t=1}^{T} \frac{\Phi_t(\eta_t)}{\Phi_{t-1}(\eta_{t-1})}\right) = \sum_{t\in[T]} Q_t, \text{ where } Q_t = \ln\left(\frac{\Phi_t(\eta_t)}{\Phi_{t-1}(\eta_{t-1})}\right).$$

We lower-bound Q in terms of the "best node" $u^\star_T$, accounting for the ancestors via $\mathcal{C}_{\texttt{prod}}(\cdot)$:

$$Q \geq \textstyle\sum_{t\in[T]} \widehat{g}_t(\texttt{act}_t(u^\star_T)) - \ln\left(|A_T| \cdot \mathcal{C}_{\texttt{prod}}(u^\star_T)\right)/\eta_T. \tag{4.7}$$

For the upper bound, we focus on the Q_t terms. We transition from potential $\Phi_{t-1}(\eta_t)$ to $\Phi_t(\eta_t)$ in two steps: first, the weights of all currently active nodes get updated, and then we zoom-in on the appropriate nodes. The former is handled using standard techniques, and the latter relies on the fact that the weights are preserved. We obtain:

$$Q \leq \textstyle\sum_{t\in[T]} g_t(x_t) + \sum_{t\in[T]} O(\ln T)\ \left(\gamma_t + \beta_t \sum_{u\in A_t} \widehat{g}_t(u)\right). \tag{4.8}$$

Part III: from Estimated to Realized Rewards. We argue about realized rewards, with probability (sat) at least $1 - {}^1\!/T$. We bring in two more pieces of the overall puzzle: a Lipschitz property and a concentration bound for IPS estimators. If node u is active at time t, then

$$\textstyle\sum_{\tau\in[t]} g_\tau(x^\star_{[t]}(u)) - \sum_{\tau\in[t]} L(\texttt{act}_\tau(u)) - 4\sqrt{td}\,\ln T \leq \sum_{\tau\in[t]} g_\tau(\texttt{act}_\tau(u)). \tag{4.9}$$

(We only use Lipschitzness through (4.9).) For any subsets $A'_\tau \subseteq A_\tau$, $\tau \in [T]$ it holds that:

$$\left|\textstyle\sum_{\tau\in[t],\ u\in A'_\tau} g_\tau(u) - \texttt{IPS}_\tau(u)\right| \leq O(\ln T)/\beta_t + \sum_{\tau\in[t],\ u\in A'_\tau} \beta_\tau/\pi_\tau(u). \tag{4.10}$$

The analysis of EXP3.P derives a special case of (4.10) with $A'_\tau = \{u\}$ in all rounds τ. The stronger version relies on *negative association* between random variables $\widehat{g}_\tau(u), u \in A'_\tau$.

Putting these two properties together, we relate estimates $\widehat{g}_t(u)$ with the actual gains $g_t(u)$. First, we argue that we do not *over*-estimate by too much: fixing round t,

$$\textstyle\sum_{\tau\in[t],\ u\in A'_\tau} \beta_\tau\left(\widehat{g}_\tau(u) - g_\tau(u)\right) \leq O(\ln T)\left(1 + \sum_{\tau\in[t]} \beta_\tau\, |A'_\tau|\right). \tag{4.11}$$

This holds for any subsets $A'_\tau \subset A_\tau$, $\tau \in [t]$ which only contain ancestors of the nodes in A'_t.

Second, we need a stronger version for a singleton node u, one with $L(u)$ on the right-hand side. If node u is zoomed-in in round t, then for each arm $y \in u$ we have:

$$\textstyle\sum_{\tau\in[t]} \widehat{g}_\tau(\texttt{act}_\tau(u)) - g_\tau(y) \leq O\left(L(u)\cdot t\,\ln(T) + \sqrt{t\,d}\,\ln T + (\ln T)/\beta_t\right). \tag{4.12}$$

Third, we argue that the estimates $\widehat{g}_t(u)$ form an approximate upper bound. We only need this property for singleton nodes: for each node u which is active at round t, we have

$$\textstyle\sum_{\tau\in[t]} \widehat{g}_\tau(\texttt{act}_\tau(u)) - g_\tau\left(x^\star_{[t]}(u)\right) \geq -O\left(\sqrt{t\,d}\,\ln T + (\ln T)/\beta_t\right). \tag{4.13}$$

To prove (4.13), we also use the zooming invariant (4.1) and the bound (4.4) on inherited diameters.

Using these lemmas in conjunction with the upper/lower bounds of Q we can derive the "raw" regret bound (3.3), and subsequently the "tuned" version (3.4).

Part IV: the Final Regret Bound. We bound $|A_T|$ to derive the final regret bound in Theorem 2. First, use the probability mass bound (4.5) to bound $|A_T|$ in the worst case. We use an "adversarial activation" argument: given the rewards, what would an adversary do to activate as many nodes as possible, if it were only constrained by (4.5)? The adversary would go through the nodes in the order of decreasing diameter $L(\cdot)$, and activate them until the total probability mass exceeds T. The number of active nodes with diameter $L(u) \in [\varepsilon, 2\varepsilon]$, denoted $N^{\texttt{act}}(\varepsilon)$, is bounded via `CovDim`.

Second, we bound $\texttt{AdvGap}_t(\cdot)$. Plugging probabilities π_t into (4.6), bound the "estimated gap",

$$\textstyle\sum_{\tau\in[t]} \widehat{g}_\tau(\texttt{act}_\tau(u_t^\star)) - \sum_{\tau\in[t]} \widehat{g}_\tau(\texttt{act}_\tau(u)) \leq \ln\left(\frac{9\mathcal{C}_{\texttt{prod}}(u_t^\star)}{\mathcal{C}_{\texttt{prod}}(u)\cdot\beta_t^2}\right)/\eta_t. \tag{4.14}$$

for a node u which is zoomed-in at round t. To translate this to the actual $\texttt{AdvGap}_t(\cdot)$, we bring in the machinery from Part III and the worst-case bound on $|A_T|$ derived above.

$$\texttt{AdvGap}_t(\texttt{repr}(u)) \leq L(u)\cdot\mathcal{O}\left(\ln(T)\sqrt{d\ln\left(C_{\texttt{dbl}}\cdot T\right)}\right). \tag{4.15}$$

We can now upper-bound $N^{\texttt{act}}(\varepsilon)$ via `AdvZoomDim` rather than `CovDim`. With this, we run another "adversarial activation" argument to upper-bound $|A_T|$ in terms of `AdvZoomDim`.

5. `AdvZoomDim` Examples

We provide a flexible "template" for examples with small `AdvZoomDim`. We instantiate this template for some concrete examples, which apply generically to adversarial Lipschitz bandits.

Theorem 6 *Fix action space $(\mathcal{A},\mathcal{D})$ and time horizon T. Let d be the covering dimension.* [9]

Consider problem instances $\mathcal{I}_1, \ldots, \mathcal{I}_M$ with stochastic rewards, for some M. Suppose each $\mathcal{I}_i$ has a constant zooming dimension z, with some fixed multiplier $\gamma > 0$. Construct the combined instance*: an instance with adversarial rewards, where each round is assigned in advance (but otherwise arbitrarily) to one of these stochastic instances.*

Then `AdvZoomDim` $\leq z$ *with probability at least $1 - 1/T$, with multiplier*

$$\gamma' = \gamma\cdot\left(\mathcal{O}\left(M\ln(T)\sqrt{d\ln\left(C_{\texttt{dbl}}\cdot T\right)\cdot\ln\left(T\cdot|\mathcal{A}_{\texttt{repr}}|\right)}\right)\right)^{\log(C_{\texttt{dbl}})-z}.$$

The representative set $\mathcal{A}_{\texttt{repr}} \subset \mathcal{A}$ (needed to specify `AdvZoomDim`*) can be arbitrary.*

This holds under the following assumptions on problem instances $\mathcal{I}_i$:

- *There are disjoint subsets $S_1, \ldots, S_M \subset \mathcal{A}$ such that each stochastic instance $\mathcal{I}_i$, $i \in [M]$ assigns the same "baseline" mean reward b_i to all arms in $\cup_{j\neq i} S_j$, mean rewards* at least *b_i to all arms inside S_i, and mean rewards* at most *b_i to all arms in $\mathcal{A} \setminus \sup_{j\in[M]} S_j$.*
- *For each stochastic instance $\mathcal{I}_i$, $i \in [M]$, the difference between the largest mean reward and b_i (called the* spread*) is at least $1/3$.*

[9] As before, the covering dimension is with some constant multiplier $\gamma_0 > 0$, and we suppress the logarithmic dependence on γ_0.

We emphasize the generality of this theorem. First, the assignment of rounds in the combined instance to the stochastic instances $\mathcal{I}_1, \ldots, \mathcal{I}_M$ can be completely arbitrary: *e.g.,* the stochastic instances can appear consecutively in "phases" of arbitrary duration, or they can be interleaved in an arbitrary way. Second, the subsets $S_1, \ldots, S_M \subset \mathcal{A}$ can be arbitrary. Third, each stochastic instance $\mathcal{I}_i$, $i \in [M]$ can behave arbitrarily on S_i, as long as $\mu_i^\star - b_i' \geq 1/3$, where $\mu_i^\star$ and b_i' are, resp., the largest and the smallest rewards on S_i. The baseline reward can be any $b_i \leq b_i'$, and outside $\cup_{j\in[M]} S_j$ one can have any mean rewards that are smaller than b_i.

Now we can take examples from stochastic Lipschitz bandits and convert them to (rather general) examples for adversarial Lipschitz bandits. Rather than attempt a compehensive survey of examples for the stochastic case, we focus on two concrete examples that we adapt from (Kleinberg et al., 2019): concave rewards and "distance to the target". For both examples, we posit action space $\mathcal{A} = [0,1]$ and distances $\mathcal{D}(x,y) = |x-y|$. Note that the covering dimension is $d = 1$. The expected reward of each arm x in a given stochastic instance i is denoted $\mu_i(x)$. In both examples, $\mu_i(\cdot)$ will have a single peak, denoted $x_i^\star \in \mathcal{A}$, and the baseline reward satisfies $\mu_i(x_i^*) - b_i \geq 1/3$.

- *Concave rewards:* For each instance i, $\mu_i(x)$ is a strongly concave function on S_i, in the sense that $\mu_i''(x)$ exists and $\mu_i''(x) < \varepsilon$ for some $\varepsilon > 0$. Then the zooming dimension is $z = 1/2 < d = 1$, with appropriately chosen multiplier $\gamma > 0$. [10]
- *"Distance to target":* For each instance i, $\mu_i(x) = \min\left(0, \mu_i(x_i^\star) - \mathcal{D}(x, x_i^\star)\right)$ for all arms $x \in S_i$. Then the zooming dimension is in fact $z = 0$.

6. Extension to Arbitrary Metric Spaces

In this appendix, we sketch out an extension to arbitrary metric spaces. The main change is that the zooming tree is replaced with a more detailed decomposition of the action space. Similar decompositions have been implicit in all prior work on adaptive discretization, starting from (Kleinberg et al., 2019; Bubeck et al., 2011a). No substantial changes in the algorithm or analysis are needed.

Preliminaries. Fix subset $S \subset \mathcal{A}$ and $\varepsilon > 0$. The diameter of S is $\sup_{x,y\in S'} \mathcal{D}(x,y)$. An *$\varepsilon$-covering* of S is a collection of subsets $S' \subset \mathcal{A}$ of diameter at most ε whose union covers S. The ε-covering number of S, denoted $\mathcal{N}_\varepsilon(S)$, is the smallest cardinality of an ε-covering. Note that the covering property in (1.4) can be restated as $\inf\left\{ d \geq 0 : \; \mathcal{N}_\varepsilon(\mathcal{A}_\varepsilon) \leq \gamma \cdot \varepsilon^{-d}, \quad \forall \varepsilon > 0 \right\}$.

A *greedy ε-covering* of S is an ε-covering constructed by the following "greedy" algorithm: while there is a point $x \in S$ which is not yet covered, add the closed ball $B(x, \varepsilon/2)$ to the covering. Thus, this ε-covering consists of closed balls of radius $\varepsilon/2$ whose centers are at distance more than $\varepsilon/2$.

A *rooted* directed acyclic graph (DAG) is a DAG with a single source node, called the *root*. For each node u, the distance from the root is called the *height* of u and denoted $h(u)$. The subset of nodes reachable from u (including u itself) is called the *sub-DAG* of u. For an edge (u,v), we say that u is a *parent* and v is a *child* relative to one another. The set of all children of u is denoted $\mathcal{C}(u)$.

Metric Space Decomposition. Our decomposition is a rooted DAG, called *Zooming DAG*, whose nodes correspond to balls in the metric space.

[10] A somewhat subtle point: an algorithm tailored to concave-rewards instances can achieve $\tilde{\mathcal{O}}(\sqrt{T})$ regret, *e.g.,* via uniform discretization (Kleinberg and Leighton, 2003). However, this algorithm would not be optimal in the worst case: it would only achieve regret $\tilde{\mathcal{O}}(T^{3/4})$ whereas the worst-case optimal regret rate is $\tilde{\mathcal{O}}(T^{2/3})$.

Definition 7 (zooming DAG) *A* zooming DAG *is a rooted DAG of infinite height. Each node u corresponds to a closed ball $B(u)$ in the action space, with radius $r(u) = 2^{-h(u)}$ and center $x(u) \in \mathcal{A}$. These objects are called, respectively, the* action-ball*, the* action-radius*, and the* action-center *of u. The following properties are enforced:*

(a) each node u is covered by the children: $B(u) \subset \cup_{v\in\mathcal{C}(u)} B(v)$.

(b) each node u overlaps with each child v: $B(u) \cap B(v) \neq \emptyset$.

(c) for any two nodes of the same action-radius r, their action-centers are at distance $> r$.

The action-span *of u is the union of all action-balls in the sub-DAG of u.*

Several implications are worth spelling out:

- the nodes with a given action-radius r cover the action space (by property (a)), and there are at most $\mathcal{N}_r(\mathcal{A})$ of them (by property (c)). Recall that $\mathcal{N}_r(\mathcal{A}) \leq \gamma \cdot r^{-d}$, where d is the covering dimension with multiplier γ.
- each node u has at most $\mathcal{N}_{r(u)/2}(B(u)) \leq C_{\texttt{dbl}}$ children (by properties (b,c)), and its action-span lies within distance $3\, r(u)$ from its action-center (by property (b)).

A zooming DAG exists, and can be constructed as follows. The nodes with a given action-radius r are constructed as a greedy $(2r)$-cover of the action space. The children of each node u are all nodes of action-radius $r(u)/2$ whose action-balls overlap with $B(u)$.

Our algorithm only needs nodes of height up to $O(\log T)$. We assume that some "zooming DAG", denoted `ZoomDAG`, is fixed and known to the algorithm.

Note that a given node in `ZoomDAG` may have multiple parents. Our algorithm adaptively constructs subsets of `ZoomDAG` that are directed trees. Hence a definition:

Definition 8 (zooming tree) *A subgraph of* `ZoomDAG` *is called a* zooming tree *if it is a finite directed tree rooted at the root of* `ZoomDAG`*. The* ancestor path *of node u is the path from the root to u.*

For a d-dimensional unit cube, `ZoomDAG` can be defined as a zooming tree, as per Section 2.

Changes in the Algorithm. When zooming in on a given node u, it activates all children of u in `ZoomDAG` that are not already active (whereas the version in Section 2 activates all children of u). The representative arms $\texttt{repr}_t(u)$ are chosen from the action-ball of u.

Changes in the Analysis. We account for the fact that the action-span of each node u lies within $3\, r(u)$ of its action-center (previously it was just $r(u)$). This constant 3 is propagated throughout.

Acknowledgments

The authors are thankful to the anonymous reviewers for feedback that helped improve the paper.

Most of the research was done while CP was an intern at Microsoft Research NYC. CP is supported in part by the National Science Foundation under grant CCF-1718549 and the Harvard Data Science Initiative.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, 2011. 6

Jacob D. Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, 2008. 6

Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, pages 28–40. Citeseer, 2010. 6

Alekh Agarwal, Dean P Foster, Daniel J Hsu, Sham M Kakade, and Alexander Rakhlin. Stochastic convex optimization with bandit feedback. *Advances in Neural Information Processing Systems*, 24:1035–1043, 2011. 6

Rajeev Agrawal. The continuum-armed bandit problem. *SIAM J. Control and Optimization*, 33(6): 1926–1951, 1995. 5

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002a. 3, 4

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multi-armed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002b. Preliminary version in *36th IEEE FOCS*, 1995. 2, 4, 6

Mohammad Gheshlaghi Azar, Alessandro Lazaric, and Emma Brunskill. Online stochastic optimization under correlated bandit feedback. In *31th Intl. Conf. on Machine Learning (ICML)*, pages 1557–1565, 2014. 5

Sébastien Bubeck. *Bandits Games and Clustering Foundations*. PhD thesis, Univ. Lille 1, 2010. 6

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning*, 5(1): 1–122, 2012. Published with *Now Publishers* (Boston, MA, USA). Also available at `https://arxiv.org/abs/1204.5721`. 1, 6

Sébastien Bubeck and Ronen Eldan. Multi-scale exploration of convex functions and bandit convex optimization. In *Conference on Learning Theory*. PMLR, 2016. 6

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. Online optimization in x-armed bandits. In *21st Advances in Neural Information Processing Systems (NIPS)*, pages 201–208, 2008. 1, 2, 3, 4, 5

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvari. Online Optimization in X-Armed Bandits. *J. of Machine Learning Research (JMLR)*, 12:1587–1627, 2011a. Preliminary version in *NIPS 2008*. 1, 2, 3, 4, 5, 13

Sébastien Bubeck, Gilles Stoltz, and Jia Yuan Yu. Lipschitz bandits without the lipschitz constant. In *22nd Intl. Conf. on Algorithmic Learning Theory (ALT)*, pages 144–158, 2011b. 5

Sébastien Bubeck, Ofer Dekel, Tomer Koren, and Yuval Peres. Bandit convex optimization:\sqrtt regret in one dimension. In *Conference on Learning Theory*. PMLR, 2015. 6

Sébastien Bubeck, Yin Tat Lee, and Ronen Eldan. Kernel-based methods for bandit convex optimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 72–85, 2017. 6

Adam Bull. Adaptive-treed bandits. *Bernoulli J. of Statistics*, 21(4):2289–2307, 2015. 1, 5

Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, 2008. 6

Abraham Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*. SIAM, 2005. 6

Jean-Bastien Grill, Michal Valko, and Rémi Munos. Black-box optimization of noisy functions with unknown smoothness. In *28th Advances in Neural Information Processing Systems (NIPS)*, 2015. 1, 5

Chien-Ju Ho, Aleksandrs Slivkins, and Jennifer Wortman Vaughan. Adaptive contract design for crowdsourcing markets: Bandit algorithms for repeated principal-agent problems. *J. of Artificial Intelligence Research*, 55:317–359, 2016. Preliminary version appeared in *ACM EC 2014*. 1, 5

Jon Kleinberg, Aleksandrs Slivkins, and Tom Wexler. Triangulation and embedding using small sets of beacons. *J. of the ACM*, 56(6), September 2009. Subsumes conference papers in *IEEE FOCS 2004* and *ACM-SIAM SODA 2005*. 5

Robert Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *18th Advances in Neural Information Processing Systems (NIPS)*, 2004. 1, 3, 5

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *40th ACM Symp. on Theory of Computing (STOC)*, pages 681–690, 2008. 1, 2, 3, 4, 5

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Bandits and experts in metric spaces. *J. of the ACM*, 66(4):30:1–30:77, May 2019. Merged and revised version of conference papers in *ACM STOC 2008* and *ACM-SIAM SODA 2010*. Also available at `http://arxiv.org/abs/1312.1277`. 1, 2, 3, 4, 5, 13

Robert D. Kleinberg and Frank T. Leighton. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 594–605, 2003. 1, 13

Levente Kocsis and Csaba Szepesvari. Bandit Based Monte-Carlo Planning. In *17th European Conf. on Machine Learning (ECML)*, pages 282–293, 2006. 5

Akshay Krishnamurthy, John Langford, Aleksandrs Slivkins, and Chicheng Zhang. Contextual bandits with continuous actions: Smoothing, zooming, and adapting. *J. of Machine Learning Research (JMLR)*, 27(137):1–45, 2020. Preliminary version at *COLT 2019*. 5

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, Cambridge, UK, 2020. Versions available at `https://banditalgs.com/` since 2018. 1

Andrea Locatelli and Alexandra Carpentier. Adaptivity to smoothness in x-armed bandits. In *31th Conf. on Learning Theory (COLT)*, pages 1463–1492, 2018. 5

Odalric-Ambrym Maillard and Rémi Munos. Online Learning in Adversarial Lipschitz Environments. In *European Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 305–320, 2010. 5

Stanislav Minsker. Estimation of extreme values and associated level sets of a regression function via selective sampling. In *26th Conf. on Learning Theory (COLT)*, pages 105–121, 2013. 1

Rémi Munos. Optimistic optimization of a deterministic function without the knowledge of its smoothness. In *25th Advances in Neural Information Processing Systems (NIPS)*, pages 783–791, 2011. 1, 5

Rémi Munos and Pierre-Arnaud Coquelin. Bandit algorithms for tree search. In *23rd Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2007. 5

Sandeep Pandey, Deepak Agarwal, Deepayan Chakrabarti, and Vanja Josifovski. Bandits for Taxonomies: A Model-based Approach. In *SIAM Intl. Conf. on Data Mining (SDM)*, 2007. 5

Ankan Saha and Ambuj Tewari. Improved regret guarantees for online smooth convex optimization with bandit feedback. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011. 6

Aleksandrs Slivkins. Multi-armed bandits on implicit metric spaces. In *25th Advances in Neural Information Processing Systems (NIPS)*, 2011. 1, 5

Aleksandrs Slivkins. Contextual bandits with similarity information. *J. of Machine Learning Research (JMLR)*, 15(1):2533–2568, 2014. Preliminary version in *COLT 2011*. 1, 3, 5

Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, November 2019. Published with *Now Publishers* (Boston, MA, USA). Also available at `https://arxiv.org/abs/1904.07272`. 1, 9

Aleksandrs Slivkins, Filip Radlinski, and Sreenivas Gollapudi. Ranked bandits in metric spaces: Learning optimally diverse rankings over large document collections. *J. of Machine Learning Research (JMLR)*, 14(Feb):399–436, 2013. Preliminary version in *27th ICML*, 2010. 1, 5

Michal Valko, Alexandra Carpentier, and Rémi Munos. Stochastic simultaneous optimistic optimization. In *30th Intl. Conf. on Machine Learning (ICML)*, pages 19–27, 2013a. 1, 5

Michal Valko, Nathaniel Korda, Rémi Munos, Ilias N. Flaounas, and Nello Cristianini. Finite-time analysis of kernelised contextual bandits. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI 2013, Bellevue, WA, USA, August 11-15, 2013*. AUAI Press, 2013b. 6

Michal Valko, Rémi Munos, Branislav Kveton, and Tomáš Kocák. Spectral bandits for smooth graph functions. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*. JMLR.org, 2014. 6