LIPSCHITZ BANDITS WITH ARBITRARY FEEDBACK DELAYS

Anonymous authors
Paper under double-blind review

ABSTRACT

The Lipschitz bandit problem extends the traditional multi-armed bandit framework to continuous action spaces by assuming that the reward functions satisfy a Lipschitz condition. This work investigates Lipschitz bandits under arbitrary feedback delays, where reward signals are not received immediately upon taking an action but after an arbitrarily chosen delay. We consider both stochastic and adversarial reward settings, and propose an elimination-based algorithm and an EXP3-based algorithm, respectively. For both settings, our algorithms achieve a regret bound of $\tilde{O}(T^{\frac{d_z+1}{d_z+2}} + \sqrt{D})$ over a time horizon T with total delay D, where the definition of the zooming dimension d_z is different for distinct settings. Our bounds match existing delay-free regret guarantees for Lipschitz bandits and reveal an additional regret penalty of $\tilde{O}(\sqrt{D})$ incurred by feedback delays. To the best of our knowledge, this work provides the first regret guarantees for Lipschitz bandits with arbitrary delays and expands the theoretical frontier of online decision making with delayed feedback. We further provide numerical experiments demonstrating the robustness of our algorithms to feedback delays and adversarial reward sequences.

1 INTRODUCTION

The Lipschitz bandit framework (Agrawal, 1995; Kleinberg et al., 2008) extends the classical multi-armed bandit (MAB) formulation to continuous action spaces. In this setting, an agent sequentially selects actions from a metric space and receives stochastic rewards whose expectations vary smoothly according to a Lipschitz-continuous function. A standard approach for handling continuous action spaces is discretization, which converts continuous action selection into a finite-armed bandit problem (Kleinberg et al., 2019; Feng et al., 2022). This approach requires balancing a fundamental trade-off: a finer discretization reduces the bias of continuous approximation but increases the model complexity and sample regret of the finite-armed bandit problem.

In classical formulations, reward feedback is assumed to be immediate. However, in many real-world applications, feedback is delayed, forcing the agent to continue making decisions without access to the most recent observations. While bandit problems with delayed feedback have been extensively studied, their extension to Lipschitz bandits remains substantially underexplored. Recently, Liu et al. (2026) studied Lipschitz bandits with delayed feedback, but their analysis is restricted to stochastic delays. Therefore, in this work, we fill in this gap by investigating Lipschitz bandits under more general delay models, where delays can be arbitrarily chosen before the decision-making procedure starts, quantifying a worst-case effect of feedback delays.

1.1 LIPSCHITZ BANDITS WITH DELAYED FEEDBACK

Let (X, δ) be a compact metric space, where $\delta : X \times X \to \mathbb{R}_{\geq 0}$ denotes the metric on X. Let T be the number of rounds. For each round t, let $\mu_t : X \to [0, 1]$ be the reward function, which is assumed to be Lipschitz continuous, i.e., $|\mu_t(x) - \mu_t(y)| \leq \delta(x, y)$ for all $x, y \in X$. We call the tuple $(X, \delta, \{\mu_t\}_{t=1}^T)$ an instance of the Lipschitz bandit problem. In the standard stochastic reward settings, the reward function remains the same over time, i.e., $\mu_t \equiv \mu$ for all $t = 1, \ldots, T$. We also consider the "oblivious" adversarial reward settings, where the reward function μ_t is chosen arbitrarily before the bandit problem starts.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The interaction between the agent and environment proceeds as follows. At each round $t = 1, \ldots, T$, the agent selects an arm $x_t \in X$, and a reward r_t is generated after the arm selection. The reward r_t is drawn from a distribution that depends only on x_t and satisfies $\mathbb{E}[r_t] = \mu_t(x_t)$. The goal of the agent is to minimize the cumulative regret, which is defined as

$$R(T) = \max_{x^* \in X} \sum_{t=1}^{T} \mu_t(x^*) - \sum_{t=1}^{T} \mu_t(x_t).$$

In order to optimize the cumulative regret, the agent should use previous feedback to update its policy. However, in the delayed feedback setting, when a reward is generated, it may not be immediately observed by the agent. Instead, each reward r_t is observed after a delay of d_t rounds. Hence, at the beginning of round t, the agent observes the history of actions $\{x_i : i < t\}$ and the set of rewards $\{r_i : i + d_i < t\}$. A delay $d_t = 0$ indicates an immediate feedback. Without loss of generality, we assume that $t + d_t \leq T$, since any delay beyond $T - t$ has the same effect as $d_t = T - t$, in which case the reward at round t cannot be used to update the agent's policy. In this work, we assume that $\{d_t\}_{t=1}^{T}$ is *arbitrarily* determined before the interaction starts (i.e., oblivious delays), but the agent knows d_t only after the $(t + d_t)$-th round.

Another important concept for a Lipschitz bandit problem instance is the zooming dimension, whose definition is based on the covering number. A ball in (X, δ) with diameter $l \geq 0$ can be written as $B(x, l) := \{y \in X : \delta(x, y) < l/2\}$. Let $S \subseteq X$ be a subset of the action space, denote by $\text{cov}(S, l)$ the minimum number of balls with diameter l required to cover the set S. It is known that we can define the covering dimension of X using the covering number:

$$d := \inf\{n \geq 0 : \ \exists c > 0, \ \forall \epsilon > 0, \ \text{cov}(X, \epsilon) \leq c\epsilon^{-n}\}.$$

For the definition of zooming dimension, instead of covering the whole action space, we will specify a subset $X_\epsilon \subseteq X$ to be covered, and then compute the covering number. For a given multiplier $c > 0$, the corresponding c-zooming dimension is defined as

$$d_z = \inf\{n \geq 0 : \ \forall \epsilon \in (0, 1], \ \text{cov}(X_\epsilon, \epsilon) \leq c\epsilon^{-n}\}.$$

Typically, X_ϵ should include all actions that are difficult to distinguish from the optimal action x^*, and smaller ϵ means the actions in X_ϵ are more likely to be optimal. Intuitively, a smaller zooming-dimension d_z indicates a simpler problem instance. The explicit definition of zooming dimension is given in Sections 3 and 4 for the stochastic reward and adversarial reward settings, respectively.

Compared to the covering dimension, the zooming dimension does not control the scale of the whole action space X. Instead, it focuses only on a smaller region that is nearly optimal. Intuitively, the zooming dimension measures the difficulty of distinguishing the optimal arm from suboptimal arms. Note that the zooming dimension depends on the reward function μ, and may be significantly smaller than the dimension d of the whole space X.

In the absence of feedback delays, the state-of-the-art algorithms for Lipschitz bandits with stochastic rewards achieve a regret bound of $\tilde{O}(T^{\frac{d_z+1}{d_z+2}})$, where d_z denotes the zooming dimension (Kleinberg et al., 2019). Moreover, similar regret bounds are also achieved for adversarial rewards, where the only difference lies in the definition of zooming dimension. In this work, we advance the theory of Lipschitz bandits by proposing new algorithms for feedback delays and establishing regret bounds that explicitly depend on both T and the delay budget D.

Main Contributions. Our contributions are summarized as follows. We summarize our results and previous results in Table 1. To the best of our knowledge, we provide the first theoretical guarantees for Lipschitz bandits with arbitrary delays.

- We investigate an existing elimination-based algorithm in stochastic Lipschitz bandits, but with arbitrary instead of stochastic feedback delays. By developing a greedy-style analysis that characterizes the impact of worst-case delay patterns, we exhibit in Theorem 3.1 that the algorithm achieves an adaptive regret bound of $\tilde{O}(T^{\frac{d_z+1}{d_z+2}} + \sqrt{D})$ without requiring prior knowledge of either the total delay budget D or the zooming dimension d_z.
- We design the Delayed Lipschitz EXP3 (DLEXP3) algorithm for adversarial rewards with arbitrary feedback delays. By carefully selecting the zooming schedule and weight-update

procedure, we prove that, when the delay budget D and zooming dimension d_z are known, the Delayed Lipschitz EXP3 algorithm achieves $\tilde{O}(T^{\frac{d_z+1}{d_z+2}} + \sqrt{D})$ regret under a modified notion of zooming dimension.

- Our regret bounds recover the optimal rates of the corresponding delay-free problems in both stochastic and adversarial reward settings. Furthermore, our analysis demonstrates that arbitrary feedback delays contribute an additional penalty of order $\tilde{O}(\sqrt{D})$ to the regret, precisely characterizing the impact of delayed feedback on Lipschitz bandit problems.
- We conduct numerical experiments illustrating the robustness of DLPP and DLEXP3 to feedback delays, as well as the robustness of DLEXP3 to adversarial reward sequences.

	Reward	Delay	Regret
Kleinberg et al. (2008), Theorem 2.4	Stochastic	No Delay	$O\left(T^{\frac{d_z+1}{d_z+2}}(c\log T)^{\frac{1}{d_z+2}}\right)$ *
Podimata & Slivkins (2021), Theorem 2	Adversarial	No Delay	$O\left(T^{\frac{d_z+1}{d_z+2}} d^{1/2}(c\cdot C_{\text{dbl}})^{\frac{1}{d_z+2}}\log^5 T\right)$ †
Liu et al. (2026), Theorem 1	Stochastic	Bounded	$O\left(T^{\frac{d_z+1}{d_z+2}}(c\log T)^{\frac{1}{d_z+2}} + c\tau_{\max}\left(\frac{T}{\log T}\right)^{\frac{d_z}{d_z+2}}\right)$
Liu et al. (2026), Theorem 3	Stochastic	Stochastic	$O\left(\frac{1}{p}T^{\frac{d_z+1}{d_z+2}}(c\log T)^{\frac{1}{d_z+2}} + Q(p)\right)$ ‡
Theorem 3.1 (Ours)	Stochastic	**Arbitrary**	$O\left(T^{\frac{d_z+1}{d_z+2}}(c\log T)^{\frac{1}{d_z+2}} + \sqrt{D\log D}\right)$
Theorem 4.1 (Ours)	Adversarial	**Arbitrary**	$O\left(T^{\frac{d_z+1}{d_z+2}}\sqrt{d2^d}c^{\frac{1}{d_z+2}}\log T + \sqrt{dD\log T}\right)$

Table 1: Comparison with existing results for Lipschitz bandits and bandits with delayed feedback.

*d_z represents the zooming dimension with multiplier c (see Eq. (1)).
†The definition of d_z is different for adversarial bandits (see Eq. (3) and (4)).
‡$Q(\cdot)$ represents the quantile function of delay distribution.

2 Related Works

Lipschitz Bandits. Lipschitz bandits, first introduced by Agrawal (1995), extend the classical multi-armed bandit (MAB) framework to continuous action spaces. Kleinberg et al. (2008) further generalized this setting to arbitrary metric spaces. Building on these foundational results, subsequent works (Feng et al., 2022; Kleinberg et al., 2019) adapted techniques from finite-armed bandits to the Lipschitz setting, achieving provably efficient algorithms with near-optimal regret guarantees.

A variety of extensions to the standard Lipschitz bandit framework have also been studied. Podimata & Slivkins (2021) considered an oblivious adversary in a non-stationary environment, where the reward function evolves over time but is fixed in advance. Kang et al. (2023) investigated adversarial reward corruption and derive regret bounds that depend explicitly on the corruption budget. Nguyen et al. (2025) analyzed non-stationary settings, with regrets controlled by the number of significant distributional shifts. Other works also explored alternative reward assumptions and problem settings, including heavy-tailed rewards (Lu et al., 2019), as well as multi-agent and quantum extensions studied by Chakraborty et al. (2026) and Yi et al. (2026), respectively.

Most recently, Liu et al. (2026) studied Lipschitz bandits with stochastic delayed feedback, providing algorithms that handle both bounded and unbounded delays. In contrast to their work, we focus on a more general delay model, where delays may be arbitrary and potentially adversarial. In particular, we aim to derive regret bounds that explicitly depend on the total delay budget.

Bandits with Delayed Feedback. Multi-armed bandit problems with delayed observations have attracted widespread attention in the literature. Joulani et al. (2013) investigated general online

learning problems with delayed feedback, and then provide a modified upper confidence bound (UCB) algorithm for the stochastic MAB scenarios. Vernade et al. (2017) studied the stochastic MAB problem with delayed and censored rewards, while assuming a known delay distribution. Adversarial bandits with arbitrary feedback delays have also been actively studied. For example, Thune et al. (2019); Bistritz et al. (2019) provided typical results, where their regret bounds have an additive $O(\sqrt{D \log K})$ term compared to the delay-free results. Zimmert & Seldin (2020) further refined the regret bound so that their result is tighter when delays are highly unbalanced.

Van Der Hoeven & Cesa-Bianchi (2022) studied nonstochastic bandits where delays depends on both time and arm. Moreover, to further investigate bandit algorithms under severely limited feedback, Pike-Burke et al. (2018), Cesa-Bianchi et al. (2022) and Wang et al. (2021) studied bandits under delayed, anonymous and aggregated feedback. In addition, there have been other works considering various extensions, such as best-of-both-worlds algorithms (Masoudian et al., 2022) and linear bandits (Vernade et al., 2020).

3 Results for Stochastic Rewards

We first consider the stochastic reward setting with arbitrary oblivious feedback delays. In this section, we present the algorithm and establish a regret upper bound for the stochastic reward setting. We consider arbitrary delays that are chosen before the bandit interaction starts, but remain unknown to the agent until the corresponding feedback is received. Following (Liu et al., 2026), we assume that the reward is generated as $r_t = \mu(x_t) + \epsilon_t$, where $\{\epsilon_t\}_{t=1}^T$ are independent 1-sub-Gaussian random variables that satisfy $\Pr\{|\epsilon_t| > \xi\} \leq 2e^{-\xi^2/2}$ for all t. For simplicity of presentation, we specialize our discussion to the metric space $X = [0,1]^d$ equipped with the metric $\delta(x,y) = |x-y|_\infty$. Under this metric, each ball in (X,δ) is a d-dimensional cube, and the covering dimension of (X,δ) is exactly d. This setting is widely adopted in the literature, e.g., (Feng et al., 2022; Podimata & Slivkins, 2021), and can be generalized to arbitrary metric spaces by assuming access to a covering oracle (see Liu et al. 2026 and Section 6 in Podimata & Slivkins 2021 for details).

Before introducing the algorithm, we first define the *zooming tree* over $X = [0,1]^d$. Each node u in the zooming tree corresponds to an ℓ_∞-ball (i.e., a cube) in X, and we denote its diameter by $L(u)$. The root node u_{root} corresponds to the entire space X. Each node u is recursively partitioned into 2^d children, each having diameter $L(u)/2$, such that the children form an exact partition of u. Therefore, for any node u at depth $h(u)$, its diameter satisfies $L(u) = 2^{-h(u)}$. We denote the set of node u's children by $C(u)$.

We adopt the Delayed Lipschitz Phased Pruning algorithm, which was proposed by Liu et al. (2026) for stochastic delays, but here we investigate it under more general arbitrary delays. As illustrated in Algorithm 1, the algorithmic procedure proceeds in phases $m = 1, 2, \ldots$. In each phase, it maintains an "active" set U_m consisting of nodes at depth $m-1$ in the zooming tree. For every node $u \in U_m$, the algorithm selects an arbitrary point $x \in u$ and plays it (line 6). Whenever a delayed reward signal (u, r) is received, the algorithm updates the empirical mean reward $\hat{\mu}(u)$ and the number of observed rewards $v(u)$ (line 8-10). After all nodes in U_m have received sufficiently many observations, the algorithm prunes nodes with $\Omega(2^{-m})$ estimated optimality gap (line 17). Finally, the zoom-in step (line 19) then refines the discretization of the action space by partitioning each remaining node into smaller balls with half the diameter. In this case, the algorithm automatically adapts the discretization level according to the observed rewards, thereby balancing the trade-off between exploration and approximation error.

Though the algorithm itself adapts from existing works, we are now considering *arbitrary* feedback delays instead of stochastic delays, quantifying the *worst-case* effect of feedback delays. Moreover, though Algorithm 1 does not require knowledge of the total delay budget D, as we will show below, the algorithm achieves a regret bound that automatically adapts to the actual total delay D.

Regret Bound. For an arm $x \in X$, define its optimality gap as $\Delta(x) = \max_{x^* \in X} \mu(x^*) - \mu(x)$. Define the ϵ-optimal region as

$$X_\epsilon = \{x \in X : \Delta(x) \leq \epsilon\}.$$

The c-zooming dimension of the problem instance (X, δ, μ) is defined as

$$d_z = \inf\{n \geq 0 : \forall \epsilon \in (0,1], \text{cov}(X_\epsilon, \epsilon/16) \leq c\epsilon^{-n}\}, \tag{1}$$

Algorithm 1 Delayed Lipschitz Phased Pruning (Liu et al., 2026)

Require: $(X = [0,1]^d, \delta = \ell_\infty)$, T, and parameter $\delta > 0$.
Initialize timer $t = 1$, phase counter $m = 1$, and $U_m = \{u_{\text{root}}\}$.
while $t \le T$ **do**
  $U_m^+ \leftarrow U_m$; set $v(u) \leftarrow 0$, $\hat{\mu}(u) \leftarrow 0$ for all $u \in U_m$. // Initialize phase m.
  while $t \le T$ and $U_m^+ \neq \emptyset$ **do**
    for $u \in U_m^+$ **do**
      Sample $x \in u$ and pull the arm x.
      for any incoming reward (u', r) with $u' \in U_m^+$ **do**
        $\hat{\mu}(u') \leftarrow (\hat{\mu}(u') \cdot v(u') + y)/(v(u') + 1)$. // Update average rewards.
        $v(u') \leftarrow v(u') + 1$.
        Remove u' from U_m^+ if $v(u') \ge v_m = (2\ln T + \ln(2/\delta)) \cdot 2^{2m-1}$.
      end for
      $t \leftarrow t + 1$. Break if $t > T$.
    end for
  end while
  $\hat{\mu}_m^* \leftarrow \max_{u \in U_m} \hat{\mu}(u)$.
  for $u \in U_m$ **do** // Prune the nodes with high instantaneous regret.
    Remove u from U_m if $\hat{\mu}_m^* - \hat{\mu}(u) > 8 \cdot 2^{-m}$.
  end for
  $U_{m+1} \leftarrow \bigcup_{u \in U_m} C(u)$. // Zoom in the remaining nodes.
  $m \leftarrow m + 1$.
end while

where $\mathrm{cov}(X, \epsilon)$ denotes the minimum number of zooming-tree nodes with diameter at most ϵ required to cover X. The constant $1/16$ and restriction to tree-node coverings are introduced only for notational convenience and can be absorbed into the constant c. As shown later, the dependence of our regret bound on c is mild.

Our main theoretical result is summarized in the following theorem. We exhibit that Algorithm 1 achieves nearly the same regret rate as Lipschitz bandits without delays, with an additional penalty that depends on the total feedback delay.

Theorem 3.1. *With probability at least $1 - \delta$, the regret of Algorithm 1 satisfies*

$$R(T) \lesssim T^{\frac{d_z+1}{d_z+2}} \left(c \log \frac{T}{\delta} \right)^{\frac{1}{d_z+2}} + \sqrt{D \log D}.$$

Here $f \lesssim g$ denotes $f = O(g)$, $D \coloneqq \sum_{t=1}^{T} d_t$, and $d_z > 0$ denotes the c-zooming dimension of the problem instance (X, δ, μ).

The full proof of Theorem 3.1 is provided in Appendix B, where we refer to Appendix A for a brief introduction. Theorem 3.1 provides a high-probability bound for the achieved regret. If we take $\delta = 1/\mathrm{poly}(T)$, the $\frac{1}{\delta}$ factor can be omitted in the result. The regret bound in Theorem 3.1 consists of two components. The first term is of order $\tilde{O}(T^{\frac{d_z+1}{d_z+2}})$, which matches the optimal regret rate for Lipschitz bandits without delays established by Kleinberg et al. (2019). The second term, $O(\sqrt{D \log D})$, captures the additional regret incurred by arbitrary feedback delays.

While Algorithm 1 is identical to the one proposed by Liu et al. (2026), we develop a new analysis that establishes robustness against arbitrary delay sequences, whereas previous analyses are restricted to only stochastic delays. Moreover, for comparison, in finite-armed bandits, the additional regret caused by delays is typically bounded by $O(\sqrt{D \log K})$, where K denotes the number of arms (see, e.g., Thune et al. 2019; Zimmert & Seldin 2020). Our result recovers this dependence in the Lipschitz setting, since the number of active arms generated by the discretization procedure can be as large as T, yielding $\log K = O(\log T)$, while the maximum possible total delay satisfies $D = O(T^2)$.

4 Results for Adversarial Rewards

This section presents our algorithm and theoretical result for Lipschitz bandits with adversarial rewards and arbitrary delays. In this setting, both the reward functions $\{\mu_t\}_{t=1}^T$ and delays $\{d_t\}_{t=1}^T$ are arbitrarily chosen before the bandit interaction starts. For analytical convenience, we consider deterministic rewards, i.e., $r_t = \mu_t(x_t)$. As in Section 3, we specialize the metric space to $([0,1]^d, \ell_\infty)$ and adopt the same definition of the zooming tree.

Our algorithm is built upon the EXP3-based algorithm proposed by Podimata & Slivkins (2021), which extends the classical EXP3 framework (Auer et al., 2002) to Lipschitz bandits through adaptive discretization. Algorithm 2 provides the pseudo-code for Delayed Lipschitz EXP3 (DLEXP3) algorithm. Throughout the interaction, the algorithm maintains a set of nodes A_t in the zooming tree, referred to as the *active nodes* at round t. Following the notation in (Podimata & Slivkins, 2021), for any node v, we define $\mathrm{act}_t(v) = u$ if u is the active ancestor of v at round t.

We modify the EXP3-based zooming procedure to accommodate arbitrarily delayed feedback by carefully scheduling the weight updates and zooming operations. The algorithm takes η, β, and γ_t as parameters, whose values will be specified later. Each round consists of three major steps: action selection (lines 4–7), weight update (lines 8–13), and zooming (lines 14–24).

Algorithm 2 Delayed Lipschitz EXP3 (DLEXP3)

Require: $(X = [0,1]^d, \delta = \ell_\infty)$, T, and parameters $\eta, \beta, (\gamma_t)_{t=1}^T$.

1: Initialize counter $\tau = 0$ and active nodes $A_1 = \{u_{\text{root}}\}$.
2: Set $w_0(u_{\text{root}}) \leftarrow 1$, $\text{conf}_0(u_{\text{root}}) \leftarrow \frac{1}{\beta}$.
3: **for** $t = 1, 2, \ldots, T$ **do**
4: $\quad p_t(u) \leftarrow \frac{w_\tau(u)}{\sum_{v \in A_t} w_\tau(v)}$ for all $u \in A_t$. // Compute distribution p_t on A_t.
5: $\quad \pi_t(u) \leftarrow (1-\gamma_t)p_\tau(u) + \gamma_t/|A_t|$ for all $u \in A_t$. // Add uniform exploration.
6: $\quad$ Sample $u_t \in A_t$ according to probability π_t.
7: $\quad$ Sample $x_t \in u_t$ uniformly at random, and pull arm x_t.
8: $\quad$ **for** any incoming reward $(u_s, \mu_s(x_s))$ **do**
9: $\quad\quad$ **for** $v \in A_t$ **do** // Update weights using EXP3 rule.
10: $\quad\quad\quad w_{\tau+1}(v) \leftarrow w_\tau(v) \cdot \exp\left(\eta \left(\frac{\mu_s(x_s) 1\{\mathrm{act}_s(v) = u_s\} + (1+4\ln T)\beta}{\pi_s(\mathrm{act}_s(v))}\right)\right)$.
11: $\quad\quad$ **end for**
12: $\quad\quad \tau \leftarrow \tau + 1$.
13: $\quad$ **end for**
14: $\quad A_{t+1} \leftarrow A_t$.
15: $\quad$ **for** $v \in A_t$ **do**
16: $\quad\quad \text{conf}_t(v) \leftarrow \text{conf}_{t-1}(v) + \frac{\beta}{\pi_t(v)}$. // Calculate the total "confidence".
17: $\quad\quad$ **if** $\text{conf}_t(v) < t \cdot L(v)$ **then**
18: $\quad\quad\quad A_{t+1} \leftarrow (A_{t+1} \setminus \{v\}) \cup C(v)$. // Zoom in the node.
19: $\quad\quad\quad$ **for** $v' \in C(v)$ **do** // Distribute its weight to all children.
20: $\quad\quad\quad\quad w_\tau(v') \leftarrow w_\tau(v)/|C(v)|$.
21: $\quad\quad\quad\quad \text{conf}_t(v') \leftarrow \text{conf}_t(v)$.
22: $\quad\quad\quad$ **end for**
23: $\quad\quad$ **end if**
24: $\quad$ **end for**
25: **end for**

In the action selection step, the algorithm selects an active node u_t according to a distribution proportional to its weight and then samples an arbitrary arm $x_t \in u_t$ to play. Following (Podimata & Slivkins, 2021), for every active node $u \in A_t$, we construct the estimated reward

$$\hat{\mu}_t(u) := \frac{\mu(x_t) 1\{u = u_t\} + (1 + 4\ln T)\beta}{\pi_t(u)}.$$

This estimator consists of an inverse propensity score (IPS) estimator together with an additional confidence correction term. In the absence of delays, $\hat{\mu}_t(u)$ can be immediately used to update the weight of node u. However, under delayed feedback, the reward may arrive after u is zoomed in. If

node u remains active at that time, we directly update its weight. Otherwise, the update is transferred to each active descendant v of u at round $t + d_t$.

The remaining component of the algorithm is the zooming step. For each active node $u \in A_t$, we define its cumulative confidence as $\text{conf}_t(u) = \frac{1}{\beta} + \sum_{s=1}^{t} \frac{\beta}{\pi_s(\text{act}_s(u))}$. Intuitively, $\text{conf}_t(u)$ quantifies the accumulated estimation uncertainty associated with node u, and the algorithm refines the discretization whenever this uncertainty becomes smaller than the scale of the node. Furthermore, when $\beta < 1/2$, the sampling rule guarantees that $\frac{\beta}{\pi_t(u)} < L(u)$ for any node u that is zoomed in at round t. Unlike the weight update step, the zooming decision does not require waiting for delayed feedback and can therefore be performed immediately.

Remark. Define $\tilde{U}_t := \{s : s + d_s \le t\}$. Then, by the construction of the zooming tree and delayed update rule, after round t, we have the following useful representation of the weights:

$$w_\tau(u) = L(u)^d \exp\left(\eta \sum_{s \in \tilde{U}_t} \hat{\mu}_s(\text{act}_s(u))\right). \tag{2}$$

Regret Bound. We first define the adversarial gap of an arm $x \in X$ at round t as

$$\Delta_t(x) := \frac{1}{t} \sup_{x^* \in X} \sum_{s \in \tilde{U}_t} [\mu_t(x^*) - \mu_t(x)],$$

which serves as a counterpart of the optimality gap for adversarial rewards. The definition of $\Delta_t(x)$ considers only the rounds whose feedback has arrived by round t, selects the best fixed comparator arm x^* over these observable rounds, and measures the average cumulative optimality gap of arm x. In the absence of delays, this definition reduces to the one used in (Podimata & Slivkins, 2021), which have considered the adversarial Lipschitz bandit problem without feedback delay.

Using this notion of gap, we define the ϵ-optimal region as

$$X_\epsilon := \{x \in X : \exists t \ge \epsilon^{-2}/9,\ \Delta_t(x) \le O(\epsilon \ln T + \epsilon^2 \sqrt{D})\}, \tag{3}$$

where the constant hidden in the $O(\cdot)$ notation is specified in Lemma C.16. In other words, an arm x is considered ϵ-optimal if its adversarial gap is sufficiently small at some sufficiently large time t. Compared with the definition of adversarial gap in Podimata & Slivkins (2021), our definition allows an additional delay-dependent term $O(\epsilon^2\sqrt{D})$ in the optimality threshold. This relaxation accounts for the effect of delayed feedback in our setting. Moreover, when ϵ is small, the additional term decreases quadratically in ϵ.

Based on this definition, we introduce the adversarial c-zooming dimension with delayed feedback:

$$d_z := \inf\{n \ge 0 :\ \forall \epsilon \in (0, 1],\ \text{cov}(X_\epsilon, \epsilon) \le c\epsilon^{-n}\}, \tag{4}$$

where $\text{cov}(X_\epsilon, \epsilon)$ denotes the minimum number of zooming-tree nodes with diameter at most ϵ required to cover X_ϵ. As before, restricting the covering sets to nodes in the zooming tree is mainly for simplicity of presentation and affects only the constant c. The worst-case adversarial zooming dimension is $d_z \le d$.

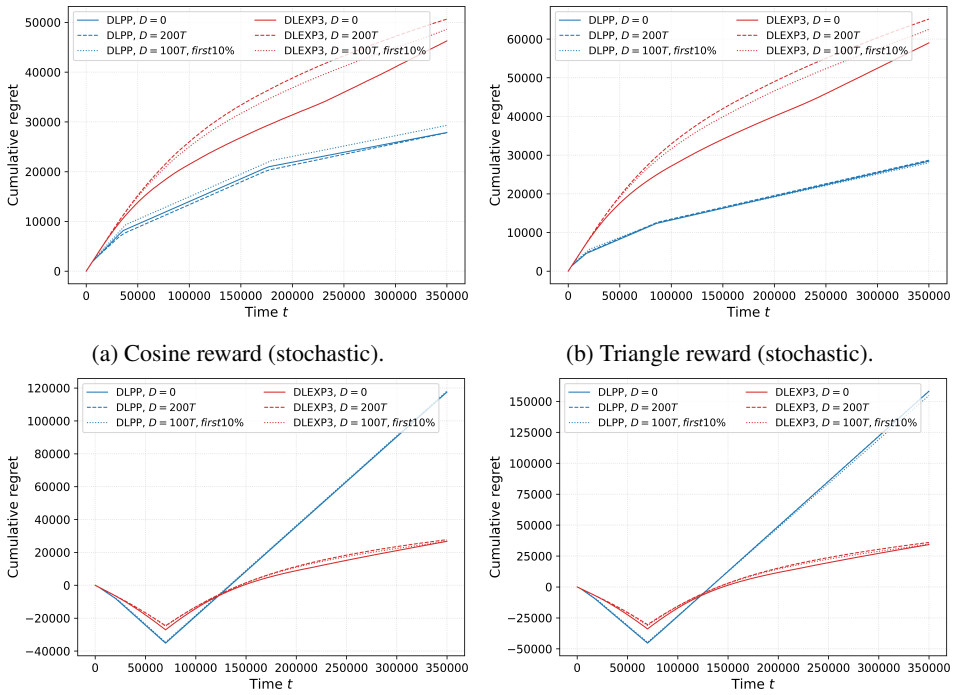
We now present the regret guarantee for Delayed Lipschitz EXP3 (Algorithm 2).

Theorem 4.1. *Assume that the agent has prior knowledge of the total delay budget $D = \sum_{t=1}^{T} d_t$ and the adversarial c-zooming dimension d_z. If we choose*

$$\eta = \beta = \Theta\left(\frac{\sqrt{d \ln T}}{\sqrt{2^d} T^{(d_z+1)/(d_z+2)} c^{\frac{1}{d_z+2}} \sqrt{\ln T} + \sqrt{D}}\right), \quad \gamma_t = (2 + 4 \ln T)\beta |A_t|,$$

then the regret of DLEXP3 algorithm is bounded as

$$\mathbb{E}[R(T)] \lesssim \sqrt{d 2^d} T^{\frac{d_z+1}{d_z+2}} c^{\frac{1}{d_z+2}} \log T + \sqrt{dD \log T}.$$



(a) Cosine reward (stochastic).

(b) Triangle reward (stochastic).

(c) Cosine reward (adversarial).

(d) Triangle reward (adversarial).

Figure 1: Comparison of DLPP with $\delta = 0.01$ and DLEXP3 under different feedback delays and reward schemes. In the stochastic setting, Gaussian noise with $\sigma = 0.4$ is added to the reward. In the adversarial setting, the reward function is flipped at $t = 0.2T$. Regret is evaluated with respect to a single global optimal arm and can therefore be negative in the adversarial setting.

The full proof is shown in Appendix C, and a sketched proof is provided in Appendix A. Theorem 4.1 shows that DLEXP3 achieves a regret bound of order $\tilde{O}(T^{\frac{d_z+1}{d_z+2}} + \sqrt{D})$ for adversarial Lipschitz bandits with delayed feedback. This rate has the same structure as the result in Theorem 3.1 under stochastic delays. The main difference lies in the definition of the zooming dimension. In both cases, the zooming dimension is upper bounded by the ambient dimension d in the worst case.

When there are no delays, the first term $\tilde{O}(T^{\frac{d_z+1}{d_z+2}})$ recovers the regret guarantee in Podimata & Slivkins (2021). Moreover, the additional delay penalty $\tilde{O}(\sqrt{D})$ matches the dependence on the total delay budget D in existing results for traditional finite-armed bandits with delayed feedback (Thune et al., 2019; Zimmert & Seldin, 2020). Different from the stochastic reward setting, here the combination of adversarial rewards and delayed feedback introduces nontrivial difficulty, and achieving the optimal regret guarantee in DLEXP3 requires the agent to know both the zooming dimension d_z and total delay budget D.

5 Experiments

In this section, we conduct numerical experiments to illustrate the empirical performance of Delayed Lipschitz Phased Pruning (DLPP, Algorithm 1) and Delayed Lipschitz EXP3 (DLEXP3, Algorithm 2). Specifically, we investigate (i) the robustness of both algorithms to feedback delays and (ii) the robustness of DLEXP3 to adversarial reward sequences.

We evaluate the algorithms on the metric space $(X = [0, 1], \delta(x, y) = |x - y|)$, corresponding to the one-dimensional special case of our framework. We consider two reward functions. The first is the cosine function $\mu(x) := \frac{1}{\pi} \cos(\pi \cdot x) + 0.5$, and the second is the triangular function $\mu(x) := 1 - |x - 0.1|$. Both functions are 1-Lipschitz and take values in $[0, 1]$, with a unique global maximizer.

In the stochastic setting, the observed reward is generated according to $r_t = r(x_t) + \sigma \cdot w_t$ for $w_t \sim \mathcal{N}(0,1)$ where $\{w_t\}_{t=1}^T$ are independent. Our analysis of DLPP assumes $\sigma = 1$. To accommodate a general noise scale σ, we modify the algorithm by replacing v_m with $v_{m,\sigma} := v_m \cdot \sigma^2$ in the algorithm. In the adversarial setting, we consider the reward sequence

$$r_t = \begin{cases} \mu(x_t), & t \le 0.2T, \\ \mu(1 - x_t), & t > 0.2T. \end{cases}$$

Thus, the adversary initially presents the original reward function and then flips it at time $0.2T$, causing the optimal arm to move to the opposite side of the domain. Under our regret definition, which compares against a single global optimal arm, the global optimal arms are $x = 1$ and $x = 0.9$ for the cosine and triangular reward functions, respectively.

To evaluate robustness to feedback delays, we consider three delay settings: (i) no delay, (ii) total delay $D = 200T$, and (iii) total delay $D = 100T$. For the setting (ii) with $D = 200T$, we sample $\{d_t\}_{t=1}^T$ uniformly from all nonnegative integer sequences satisfying $\sum_{t=1}^T d_t = D$. For the setting (iii), we instead sample $\{d_t\}_{t=1}^T$ uniformly from all nonnegative integer sequences satisfying $\sum_{t=1}^{0.1T} d_t = D$ and $d_t = 0$ for $t > 0.1T$ (i.e., only the first 10% rewards can be delayed)

For each reward function and each delay setting, we run each algorithm independently 10 times and report the average cumulative regret. The results are presented in Figure 1.

For stochastic rewards, both algorithms achieve sublinear regret across all three delay settings. The performance of DLEXP3 is more sensitive to feedback delays than that of DLPP. This is expected because DLEXP3 is designed to provide guarantees under worst-case delays and adversarial rewards, whereas the randomly generated delay schedules considered here are substantially less challenging than the worst-case instances captured by its theoretical analysis.

For adversarial rewards, DLEXP3 achieves sublinear regret and remains robust to the considered feedback delays. In contrast, DLPP suffers linear regret in this setting. This is expected because DLPP is designed for stochastic rewards and may concentrate its sampling on the initially optimal region before the reward function changes. After the change at $t = 0.2T$, it therefore cannot adapt to the new optimal arm.

Remark. In the stochastic setting, DLPP is in fact more sensitive to the noise scale σ. As σ increases, the algorithm requires more observations to reliably distinguish the optimal region from suboptimal regions, leading to slower pruning and consequently larger regret. In contrast, the performance of DLEXP3 is theoretically independent of σ, since the algorithm is designed to handle arbitrary reward sequences. Additional experiments examining the effect of the noise scale σ are provided in Appendix D.

6 Conclusion

This work studies Lipschitz bandits with arbitrary feedback delays, where the delay sequence is arbitrarily chosen before the interaction begins. We present an elimination-based algorithm and an EXP3-based algorithm for stochastic and adversarial rewards, respectively. Both algorithms recover the state-of-the-art regret rates of their corresponding delay-free settings. Moreover, feedback delays introduce an additional regret penalty of $O(\sqrt{D \log D})$ in the stochastic reward setting and $O(\sqrt{dD \log T})$ in the adversarial reward setting, both recovers the standard $O\left(\sqrt{D}\right)$ regret penalty up to logarithmic factors. These results shed light on the impact of delayed feedback on Lipschitz bandit problems. Finally, our numerical experiments illustrate the robustness of the proposed algorithms to feedback delays and demonstrate that DLEXP3 can successfully handle adversarial reward sequences.

In our current analysis for adversarial rewards, a fixed learning rate η is required to control the influence of delayed feedback, which further requires prior knowledge of total delay D and zooming dimension d_z. An interesting future direction is to develop adversarial bandit algorithms that adaptively achieve the optimal regret rate without prior knowledge of problem parameters.

AI USE STATEMENT

In this work, we used generative AI tools to polish writing, find related works and generate source codes for our experiments. We have not used generative AI tools for search ideation, drafting sections and proving mathematical claims. and generating synthetic datasets are not applicable to this work. We have checked the content polished by LLM to ensure accurate representation. Related works searched by generative AI have been verified before adding to the content. LLM-generated code have been debugged, modified, and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

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

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multi-armed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

Ilai Bistritz, Zhengyuan Zhou, Xi Chen, Nicholas Bambos, and Jose Blanchet. Online exp3 learning in adversarial bandits with delayed feedback. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Nicolo Cesa-Bianchi, Tommaso Cesari, Roberto Colomboni, Claudio Gentile, and Yishay Mansour. Nonstochastic bandits with composite anonymous feedback. *Journal of Machine Learning Research*, 23(277), 2022.

Sourav Chakraborty, Amit Kiran Rege, Claire Monteleoni, and Lijun Chen. Multi-agent lipschitz bandits, 2026.

Yasong Feng, Tianyu Wang, et al. Lipschitz bandits with batched feedback. *Advances in Neural Information Processing Systems*, 2022.

Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Online learning under delayed feedback. In *International conference on machine learning*, 2013.

Yue Kang, Cho-Jui Hsieh, and Thomas Chun Man Lee. Robust lipschitz bandits to adversarial corruptions. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 681–690, 2008.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Bandits and experts in metric spaces. *Journal of the ACM (JACM)*, 66(4):1–77, 2019.

Zhongxuan Liu, Yue Kang, and Thomas Lee. Lipschitz bandits with stochastic delayed feedback. In *The Fourteenth International Conference on Learning Representations*, 2026.

Shiyin Lu, Guanghui Wang, Yao Hu, and Lijun Zhang. Optimal algorithms for lipschitz bandits with heavy-tailed rewards. In *International Conference on Machine Learning*, pp. 4154–4163. PMLR, 2019.

Saeed Masoudian, Julian Zimmert, and Yevgeny Seldin. A best-of-both-worlds algorithm for bandits with delayed feedback. In *Advances in Neural Information Processing Systems*, 2022.

Nicolas Nguyen, Solenne Gaucher, and Claire Vernade. Non-stationary lipschitz bandits. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Ciara Pike-Burke, Shipra Agrawal, Csaba Szepesvari, and Steffen Grunewalder. Bandits with delayed, aggregated anonymous feedback. In *International Conference on Machine Learning*. PMLR, 2018.

Chara Podimata and Alex Slivkins. Adaptive discretization for adversarial lipschitz bandits. In *Conference on Learning Theory*, pp. 3788–3805. PMLR, 2021.

Tobias Sommer Thune, Nicolò Cesa-Bianchi, and Yevgeny Seldin. Nonstochastic multiarmed bandits with unrestricted delays. In *Advances in Neural Information Processing Systems*, 2019.

Dirk Van Der Hoeven and Nicolo Cesa-Bianchi. Nonstochastic bandits and experts with arm-dependent delays. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022.

Claire Vernade, Olivier Cappé, and Vianney Perchet. Stochastic bandit models for delayed conversions, 2017.

Claire Vernade, Alexandra Carpentier, Tor Lattimore, Giovanni Zappella, Beyza Ermis, and Michael Brückner. Linear bandits with stochastic delayed feedback. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Siwei Wang, Haoyun Wang, and Longbo Huang. Adaptive algorithms for multi-armed bandit with composite and anonymous feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 2021.

Bongsoo Yi, Yue Kang, and Yao Li. Quantum lipschitz bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 27844–27851, 2026.

Julian Zimmert and Yevgeny Seldin. An optimal algorithm for adversarial bandits with arbitrary delays. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, 2020.

A An overview of the Proofs

In this section, we briefly summarize the main technical approach for proving Theorem 3.1 and Theorem 4.1. For DLPP (Algorithm 3.1), we adapt a greedy-style analysis on an adversary to select the worst-case delay schedule, to calculate the maximum number of steps that the agent spends in each phase. For DLEXP3 (Algorithm 4.1), we follow the standard potential analysis of EXP3 algorithm, and carefully extract the delay-related terms from the total regret.

A.1 Stochastic Rewards: A Greedy-Style Analysis

In this section, we present the key steps in the proof of Theorem 3.1. Consider a fixed phase $m = 1, 2, \ldots$. By applying the concentration inequalities established in (Liu et al., 2026), we obtain that, with high probability, throughout phase m, every arm contained in any node $u \in U_m$ has optimality gap bounded by $O(2^{-m})$. Consequently, the cumulative regret can be decomposed according to the phases as

$$R(T) \leq \sum_{m \geq 1} O(2^{-m}) \times (\text{the total number of pulls in phase } m).$$

In the absence of feedback delays, every node $u \in U_m$ is sampled exactly v_m times, and hence the total number of pulls in phase m is $|U_m| v_m$. Using the definition of zooming dimension, this immediately yields a delay-free regret bound

$$R(T) \lesssim T^{\frac{d_z+1}{d_z+2}} \left(c \log \frac{T}{\delta} \right)^{\frac{1}{d_z+2}}.$$

However, under feedback delays, Algorithm 1 may sample a node for more than v_m times, since the feedback may not be observed. Let $v_m + a(u)$ denote the total number of pulls for node u during phase m. Then, the additional regret incurred during phase m can be bounded by

$$\sum_{u \in U_m} a(u) \times O(2^{-m}).$$

Therefore, the remaining task is to control the total number of additional pulls $A := \sum_{u \in U_m} a(u)$ in terms of the total delay budget D. Compared to (Liu et al., 2026), we are considering arbitrary instead of stochastic feedback delays, which require us to consider a worst-case scenario. Therefore,

we switch to the side of the "adversary" and consider the following equivalent inverse problem: given that Algorithm 1 suffers A additional pulls, what is the minimum total delay required to generate such an amount of over-sampling? By lower bounding this minimum delay as a function of A, we can then derive an upper bound on A in terms of the actual delay budget D.

For convenience, let $u_1, u_2, \ldots, u_{|U_m|}$ denote the order in which nodes are processed in the for loop, and define $a_i := a(u_i)$. Our analysis is based on the following two observations.

Observation I. For a fixed sequence $\{a_i\}$, the minimum total delay required to generate these additional pulls can be achieved by greedily postponing the feedback of the latest outstanding rewards. More specifically, suppose that node u_i is pulled $v_m + a_i$ times in total. To create exactly a_i additional pulls, the feedback corresponding to the v_m-th through $(v_m + a_i - 1)$-th pulls should be delayed until the round of the $(v_m + a_i)$-th pull.

Using Observation I and carefully summing the induced delays, the minimum total delay associated with a fixed sequence $\{a_i\}$ can be written as

$$D = \sum_{i=1}^{|U_m|} \sum_{s=1}^{a_{u_i}} s \cdot \left[\sum_{j=i}^{|U_m|} 1_{a_{u_j} \geq s-1} + \sum_{j=1}^{i-1} 1_{a_{u_j} \geq s} \right].$$

Observation II. The minimum delay is achieved when the values of a_i are sorted in non-decreasing order (see Appendix B for the complete proof.)

Using Observation II, without loss of generality, we assume that $a_1 \leq a_2 \leq \cdots \leq a_{|U_m|}$. Define $w_s = \sum_{i=1}^{|U_m|} 1\{a_i \geq s\}$, which represents the number of nodes with at least s additional pulls. Under this ordering, the above expression simplifies to $D = \sum_{s \geq 1} s \cdot w_s^2$. Clearly $A \leq D$, so we conclude with a Cauchy-Schwarz argument:

$$A = \sum_{s=1}^{D} w_s = \sum_{s=1}^{D} \sqrt{s} w_s \cdot \frac{1}{\sqrt{s}} \leq \sqrt{\sum_{s=1}^{D} s w_s^2 \cdot \sum_{s=1}^{D} \frac{1}{s}} = O\left(\sqrt{D \log D}\right).$$

A.2 Adversarial Rewards: Potential Analysis with Delay

In this section, we outline the main steps in the proof of Theorem 4.1. We first introduce several notations that will be used throughout the analysis. Let τ be the same as in Algorithm 2. Let A_τ denote the corresponding active set after processing τ feedback signals, and define $p_\tau(u) = \frac{w_\tau(u)}{\sum_{v \in A_\tau} w_\tau(u)}$ for $u \in A_\tau$. For each original round s, let $\tau_0(s)$ denote the value of τ immediately before selecting the action x_s, and let $\tau_1(s)$ denote the value of τ immediately before incorporating the feedback $\mu_s(x_s)$.

Our analysis follows the potential-based proof of EXP3 and the approach in (Podimata & Slivkins, 2021). We define the potential function $\Phi(\tau) = \ln\left(\sum_{u \in A_\tau} w_\tau(u)\right)$. Since the zooming step preserves the total weight, it does not change the potential function. Therefore, the potential changes only when Algorithm 2 incorporates a received feedback signal.

Suppose that $\mu_s(x_s)$ is the τ-th received feedback signal. By the weight update rule,

$$\begin{aligned}
\Phi(\tau) - \Phi(\tau - 1) &= \ln\left(\sum_{u \in A_\tau} p_{\tau-1}(u) \exp\left(\eta \hat{\mu}_s(\text{act}_s(u))\right) \right) \\
&\leq \ln\left(1 + \eta \sum_{u \in A_\tau} p_{\tau-1}(u) \hat{\mu}_s(\text{act}_s(u)) + \eta^2 \sum_{u \in A_\tau} p_{\tau-1}(u) \hat{\mu}_s^2(\text{act}_s(u)) \right) \\
&\leq \eta \sum_{u \in A_\tau} p_{\tau-1}(u) \hat{\mu}_s(\text{act}_s(u)) + \eta^2 \sum_{u \in A_\tau} p_{\tau-1}(u) \hat{\mu}_s^2(\text{act}_s(u)).
\end{aligned}$$

Here the choices of η, β, and γ_t guarantee that $\eta \hat{\mu}_s(\cdot) \leq 1$. Therefore, the second-order expansion follows from the inequality $e^x \leq 1 + x + x^2$ for $x \leq 1$, and the last step uses $\ln(1 + x) \leq x$.

Recall that the estimated reward is defined as $\hat{\mu}_s(v) = \frac{\mu_s(x_s)1\{u_s=v\}+(1+4\ln T)\beta}{\pi_s(v)}$. To extract the actual cumulative reward term $\sum_{s=1}^{T}\mu_s(x_s)$ from the estimated rewards, it is convenient to replace the distribution $p_{\tau_1(s)}$ appearing in the potential analysis by the distribution $p_{\tau_0(s)}$ used when selecting the action. These two distributions differ only because of delayed feedback. Moreover, when $\gamma_s \le 1/2$, the sampling rule gives $p_{\tau_0(s)}(u) \le \pi_s(u)/(1-\gamma_s) \le (1+2\gamma_s)\pi_s(u)$. Together with the condition $\eta\hat{\mu}_s(\cdot) \le 1$, this implies

$$\begin{aligned}\Phi(\tau)-\Phi(\tau-1) &\le 2\eta\langle\max\{0,p_{\tau_1(s)}-p_{\tau_0(s)}\},\hat{\mu}_s\rangle + \sum_{u\in A_s} p_{\tau_0(s)}(u)\left[\eta\hat{\mu}_s(u)+\eta^2\hat{\mu}_s^2(u)\right]\\ &\le 2\eta\langle\max\{0,p_{\tau_1(s)}-p_{\tau_0(s)}\},\hat{\mu}_s\rangle\\ &\quad+\eta\mu(x_s)+4\gamma_s\eta+O(\eta\beta\log T)\sum_{u\in A_s}\hat{\mu}_s(u).\end{aligned}$$

The formal definition of the inner product term is given in Lemma C.1. We omit the intermediate algebraic steps for brevity. On the other hand, a direct lower bound on the final potential implies

$$\Phi(T)-\Phi(0)=\ln\left(\sum_{u\in A_T} w_T(u)\right) \ge \ln w_T(u^*) = d\ln L(u^*)+\eta\sum_{s=1}^{T}\hat{\mu}_s(\text{act}_s(u^*))$$

for any node u^*. Combining the upper and lower bounds on the potential yields

$$\begin{aligned}\sum_{s=1}^{T}[\hat{\mu}_s(\text{act}_s(u^*))-\mu_s(x_s)] \le &\underbrace{-\frac{d\ln L(u^*)}{\eta}+4\sum_{s=1}^{T}\gamma_s+O(\beta\log T)\sum_{s=1}^{T}\sum_{u\in A_s}\hat{\mu}_s(u)}_{\text{Regret without delay}}\\ &\underbrace{+2\eta\sum_{s=1}^{T}\langle\max\{0,p_{\tau_1(s)}-p_{\tau_0(s)}\},\hat{\mu}_s\rangle}_{\text{Regret caused by delay}}.\end{aligned}$$

This inequality provides an upper bound on the "estimated regret" of Algorithm 2. It naturally decomposes the regret into two components: the standard EXP3-type regret term that is independent of delays, and an additional term that captures the effect of delayed feedback. To obtain the final regret bound, we control these two terms separately through the following four steps.

i) We first show that, for any fixed arm $x^* \in u^*$, the difference $\sum_{s=1}^{T}[\mu_s(x^*)-\hat{\mu}_s(\text{act}_s(u^*))]$ is sufficiently small with high probability. This allows us to convert the surrogate regret on the left-hand side of the previous inequality into the actual regret (see Lemma C.13).

ii) We then bound the delay-independent terms on the right-hand side using the final number of active nodes $|A_T|$ (see Lemma C.2).

iii) Next, we control the growth of the zooming tree by showing that with high probability, the number of activated nodes satisfies $O(2^d T^{\frac{d_z}{d_z+2}} C^{\frac{2}{d_z+2}})$. This result follows from the zooming rule, the definition of the adversarial zooming dimension, and a greedy counting argument (see Lemma C.17).

iv) Finally, we analyze the delay-dependent term. Taking expectation, we show that the total variation between the sampling distributions before and after receiving delayed feedback satisfies $\|\max\{0,p_{\tau_1(s)}-p_{\tau_0(s)}\}\|_1 \lesssim \eta[\tau_1(s)-\tau_0(s)]$. By summing over all rounds and carefully accounting for the contribution of each delay, we show that the regret caused by delays is bounded by $O(\eta D)$ (see Lemmas C.3 and C.4).

Combining these results, we obtain

$$R(T) \le \frac{O(d\log T)}{\eta}+\eta\cdot O(T|A_T|\ln T+D) \le \frac{O(d\log T)}{\eta}+\eta\cdot O(T^{\frac{2(d_z+1)}{d_z+2}}\ln T+D).$$

Balancing the learning rate η then gives the desired regret bound. Note that selecting the optimal value of η requires prior knowledge of both the total delay D and the zooming dimension d_z.

B Proof of Theorem 3.1

In this section, we provide proofs of Theorem 3.1.

B.1 Regret Bound under Clean Events

This part follows the analysis by Liu et al. (2026). First, we define the following clean event.

$$\mathcal{E} := \left\{ |\hat{\mu}(u) - \mu(x)| \le 4 \cdot 2^{-m}, \forall x \in u \in U_m, \forall 1 \le m \le m^*, \right\}. \tag{5}$$

Here, m^* denotes the number of completed phases (i.e., entered pruning step), and $\hat{\mu}(u)$ is the average of received rewards in the corresponding phase. Note that when u is removed U_m^+, this average is not updated anymore.

Then, if the event $\mathcal{E}$ holds, we claim that the optimal arm $x^* \in X$ is never pruned. We can prove by induction. Suppose it has not been pruned when phase m begins. Let $x^* \in u \in U_m$, $\hat{\mu}(u^*) = \max_{u \in U_m} \hat{\mu}(u)$, and pick any $x \in u^*$. Then

$$\hat{\mu}(u^*) - \hat{\mu}(u) \le 4 \cdot 2^{-m} + \mu(x) - \mu(x^*) + 4 \cdot 2^{-m} \le 8 \cdot 2^{-m},$$

and u is not pruned according to the pruning rule.

Moreover, under event $\mathcal{E}$, we also claim that the optimality gap $\Delta(x)$ for $x \in u \in U_m$ is at most $32 \cdot 2^{-m}$. This claim is trivially holds for $m = 1$. For $m > 1$, since the optimal arm x^* is not pruned during phase $m-1$, let $x^* \in u^* \in U_{m-1}$. Moreover, if $x \in u \in \mathcal{B}_{m-1}$ is not pruned, Hence,

$$\Delta(x) = \mu(x^*) - \mu(x) \le \hat{\mu}(u^*) - \hat{\mu}(u) + 8 \cdot 2^{1-m} \le 32 \cdot 2^{-m}.$$

With these claims, we can directly control the regret as follows:

$$\begin{aligned}
R(T) &= \sum_{m=1}^{m^*+1} \sum_{u \in U_m} \sum_{i=1}^{n(B)} \Delta(x_{B,i}) \\
&\le 16 \cdot 2^{-M} \cdot T + 32 \sum_{m=1}^{M} \sum_{u \in U_m} n(u) \cdot 2^{-m} \\
&\lesssim 2^{-M} \cdot T + \sum_{m=1}^{M} \left[|U_m| \cdot v_m \cdot 2^{-m} + 2^{-m} \sum_{u \in U_m} a_u \right] \\
&\lesssim 2^{-M} \cdot T + \sum_{m=1}^{M} \left[|U_m| \cdot (\log T + \log(1/\delta)) 2^m + 2^{-m} \sum_{u \in U_m} a_u \right],
\end{aligned}$$

where $1 \le M \le m^*$ is determined later, and a_u is the number of "additional plays" of the node u, that is, $a_u := n(u) - v_m$.

Note that for all $x \in u \in U_m$, its optimality gap is bounded by $32 \cdot 2^{-m} = 16L(u)$. By the definition of zooming dimension, we have

$$|U_m| \le c \cdot 2^{d_z(m-1)}.$$

So,

$$\begin{aligned}
R(T) &\lesssim 2^{-M} \cdot T + \sum_{m=1}^{M} [\log(T/\delta)] c \cdot 2^{(d_z+1)m} + \sum_{m=1}^{M} 2^{-m} \sum_{u \in U_m} a_B \\
&\lesssim 2^{-M} \cdot T + c \log(T/\delta) 2^{(d_z+1)M} + \sum_{m=1}^{M} 2^{-m} \sum_{u \in U_m} a_B.
\end{aligned}$$

Following Liu et al. (2026), by taking $M = \frac{\log \frac{T}{2^d c \log T}}{d_z+2}$, we have

$$R(T) \lesssim T^{\frac{d_z+1}{d_z+2}} \left(c \log \frac{T}{\delta} \right)^{\frac{1}{d_z+2}} + \sum_{m=1}^{M} 2^{-m} \sum_{B \in \mathcal{B}_m} a_B.$$

B.2 Proof of Theorem 3.1

According to Lemma B.1, the clean event $\mathcal{E}$ happens with probability at least $1-\delta$. So we only need to control $\sum_{u\in U_m} a_u$, given that $\sum_{t=1}^{T} d_t \le D$.

Consider an adversary that tries to maximize $\sum_{u\in U_m} a_u$. During each phase m, it is equivalent to consider the problem: fix $A = \sum_{u\in U_m} a_m$, and then minimize the total amount of delays needed to achieve it.

Let the nodes in U_m be ordered as $u_1, u_2, \dots, u_{|U_m|}$, which is the order that are played during the m-th phase. First, if we fix a_{u_i}, the optimal strategy for the adversary is to delay all of the v_m-th to $(v_m + a_{u_i} - 1)$-th observation to the step when we play u_i for the $(v_m + a_{u_i})$-th time. Then, with this strategy, given all a_{u_i}, the total delay used can be written as follows:

$$D = \sum_{i=1}^{|U_m|} \sum_{s=1}^{a_{u_i}} s \cdot \left[\sum_{j=i}^{|U_m|} 1_{a_{u_j} \ge s-1} + \sum_{j=1}^{i-1} 1_{a_{u_j} \ge s} \right].$$

To explain this formula, let $t_{i,s}$ be the step we play ball u_i for the $(v_m + s)$-th time, then $\sum_{j=i}^{|U_m|} 1_{a_{u_j}\ge s-1} + \sum_{j=1}^{i-1} 1_{a_{u_j}\ge s}$ is exactly $t_{i_s} - t_{i,s-1}$. It is then added to the delay for the s former plays in the ball u_i, increasing the total delay by $s \cdot (t_{i,s} - t_{i,s-1})$.

Now, we prove a claim: the optimal choice of $\{a_{u_i}\}$ satisfies $a_{u_i} \le a_{u_{i+1}}$ for $i = 1, 2, \dots, |U_m|-1$. To prove the claim, consider an arbitrary choice of $\{a_{u_i}\}$, if $p = a_{u_i} > a_{u_{i+1}} = q$, by interchanging a_{u_i} and $a_{u_{i+1}}$, the only changed part in the formulation of minimum total delay D is

$$\sum_{s=q+1}^{p} s \cdot 1_{a_{u_{i+1}} \ge s-1},$$

which is changed to

$$\sum_{s=q+1}^{p} s \cdot 1_{a_{u_i} \ge s}.$$

By comparing the indicators, we can observe that D is decreased by $(q+1)$. Therefore the claim holds.

With the above claim, we assume that the sequence $\{a_{u_i}\}$ is non-decreasing. Let $w_s := \sum_{i=1}^{|U_m|} 1_{a_{u_i}\ge s}$, that is, the "width" of the s-th layer, then $\sum_{s\ge 1} w_s = \sum_{u\in U_m} a_B = A$, and we can again rewrite D as

$$D = \sum_{s\ge 1} s \cdot w_s^2.$$

Clearly $A \le D$, so $w_s = 0$ for $s > D$. Using Cauchy-Schwarz inequality,

$$A = \sum_{s=1}^{D} \sqrt{s} w_s \frac{1}{\sqrt{s}} \le \sqrt{\sum_{s\ge 1} s w_s^2} \cdot \sqrt{\sum_{s=1}^{D} \frac{1}{s}} \le O(\sqrt{D \log D}).$$

Finally, plug it into the regret formula with $\sum_{m=1}^{M} r_m = O(1)$, we obtain

$$R(T) \lesssim T^{\frac{d_z+1}{d_z+2}} \left(c \log \frac{T}{\delta} \right)^{\frac{1}{d_z+2}} + \sqrt{D \log D}.$$

B.3 Auxiliary Lemmas

We adapt the following lemma from Liu et al. (2026).

Lemma B.1. *(Lemma 11 by Liu et al. (2026)) Let m^* be the last round that it is complete in Algorithm 1 (i.e., entered the pruning step). Define the following clean event*

$$\mathcal{E}' := \left\{ |\hat{\mu}_m(u) - \mu(x)| \le 2 \cdot 2^{-m} + \sqrt{\frac{4 \ln T + 2 \ln(2/\delta)}{v_m}}, \forall x \in u \in U_m, \forall 1 \le m \le m^* \right\},$$

where $\hat{\mu}_m(u)$ is calculated when u is removed from U_m^+. Then, it holds that $\Pr[\mathcal{E}'] \ge 1-\delta$.

C Proof of Theorem 4.1

For reader's convenience, we recall several main steps in the algorithm. First, the counter τ refers to how many reward signals are received. During each round, the agent selects a random node u according to the distribution

$$\pi_t(u) = (1-\gamma_t)p_t + \gamma_t/|A_t|,$$

where p_t is the distribution with probability mass proportional to the weight w_τ. When receive a feedback from round s, for all $u \in A_\tau$, update

$$\begin{aligned} w_{\tau+1}(u) &\leftarrow w_\tau(u) \cdot \exp\left(\eta \hat{\mu}_s(\mathrm{act}_s(u))\right) \\ &= w_\tau(u) \cdot \exp\left(\eta\left(\frac{\mu(x_s)1\{u_s = \mathrm{act}_s(u)\} + (1+4\ln T)\beta}{\pi_s(\mathrm{act}_s(u))}\right)\right). \end{aligned}$$

In addition, after each round, we calculate

$$\mathrm{conf}_t(u) = \frac{1}{\beta} + \sum_{s=1}^{t} \frac{\beta}{\pi_s(\mathrm{act}_s(u))},$$

and zoom in a node u if and only if $\mathrm{conf}_t(u) < t \cdot L(u)$.

Moreover, throughout this section, we assume that $t + d_t$ is at most T. Clearly, this assumption does not affect the performance of the algorithm.

C.1 Main Proof

First, we use the following lemma to control the "estimated" regret.

Lemma C.1. *Assume the parameter satisfies $\eta \le \beta \le 1$, $\gamma_t = (2+4\ln T)\beta|A_t| \le \frac{1}{2}$, then for any fixed active node u^* in the end,*

$$\begin{aligned} \sum_{s=1}^{T} \hat{\mu}_s(\mathrm{act}_s(u^*)) - \sum_{s=1}^{T}\mu(x_s) \le -\frac{d\ln L(u^*)}{\eta} + 4\sum_{s=1}^{T}\gamma_s + 4(1+2\ln T)\beta\sum_{s=1}^{T}\sum_{u\in A_s}\hat{\mu}_s(u) \\ + 2\sum_{s=1}^{T}\langle \max\{0, p_{\tau_1(s)} - p_{\tau_0(s)}\}, \hat{\mu}_s\rangle. \end{aligned}$$

Here, the "inner-product" term is defined by

$$\begin{aligned} &\langle \max\{0, p_{\tau_1(s)} - p_{\tau_0(s)}\}, \hat{\mu}_s\rangle \\ :=& \sum_{u\in A_{\tau_1(s)}} \max\left\{p_{\tau_1(s)}(u) - p_{\tau_0(s)}(act_s(u))\frac{L(u)^d}{L(act_s(u))^d}, 0\right\}\hat{\mu}_s(act_s(u)). \end{aligned}$$

Proof. Consider the potential function

$$\Phi(\tau) = \ln\left(\sum_{u\in A_\tau} w_\tau(u)\right) = \ln\left(\sum_{u\in A_\tau} L(u)^d \cdot \exp\left(\eta\sum_{\iota=1}^{\tau}\hat{\mu}_\tau(\mathrm{act}_\iota(u))\right)\right),$$

where for notational simplicity, we use $A_\tau = A_t$ if the τ-th observation is observed after round t.

First, for any node u^* that is active after the τ-th used observation

$$\begin{aligned} \Phi(\tau) - \Phi(0) &= \Phi(\tau) - 0 \\ &= \ln\left(\sum_{u\in A_\tau} w_\tau(u)\right) \\ &\ge \ln\left(w_\tau(u^*)\right) \\ &= d\ln L(u^*) + \eta\sum_{s=1}^{T}\hat{\mu}_s(\mathrm{act}_s(u^*)). \end{aligned}$$

On the other hand, we want to upper-bound the potential function. We have two steps to do during the process: updating weight using EXP3 rule, and zoom-in some of the nodes using the zoom-in rule. However, zooming-in the nodes does not change the potential function. So,

$$\begin{aligned}\Phi(\tau) - \Phi(\tau-1) &= \ln\left(\frac{\sum_{u\in A_{\tau-1}} w_{\tau-1}(u)\exp\left(\eta\hat{\mu}_s(\text{act}_s(u))\right)}{\sum_{u\in A_{\tau-1}} w_{\tau-1}(u)}\right)\\ &= \ln\left(\sum_{u\in A_{\tau-1}} p_{\tau-1}(u)\exp\left(\eta\hat{\mu}_s(\text{act}_s(u))\right)\right).\end{aligned}$$

Recall that

$$\eta\hat{\mu}_s(\text{act}_s(u)) \le \frac{1+(1+4\ln T)\beta}{\pi_s(\text{act}_s(u))} \le \frac{|A_s|(1+(1+4\ln T)\beta)\eta}{\gamma_s} \le 1.$$

So by the inequality $e^x \le 1+x+x^2$ for $x\le 1$ and the inequality $\ln(1+x)\le x$,

$$\begin{aligned}\Phi(\tau)-\Phi(\tau-1) &\le \ln\left(\sum_{u\in A_{\tau-1}} p_{\tau-1}(u)\left(1+\eta\hat{\mu}_s(\text{act}_s(u))+(\eta\hat{\mu}_s(\text{act}_s(u)))^2\right)\right)\\ &= \ln\left(1+\eta\sum_{u\in A_{\tau-1}} p_{\tau-1}(u)\hat{\mu}_s(\text{act}_s(u))+\eta^2\sum_{u\in A_{\tau-1}} p_{\tau-1}(u)\hat{\mu}_s^2(\text{act}_s(u))\right)\\ &\le \eta\sum_{u\in A_{\tau-1}} p_{\tau-1}(u)\hat{\mu}_s(\text{act}_s(u))+\eta^2\sum_{u\in A_{\tau-1}} p_{\tau-1}(u)\hat{\mu}_s^2(\text{act}_s(u))\\ &\le \Bigg[\sum_{u\in A_{\tau-1}} p_{\tau-1}(u)(\eta\hat{\mu}_s(\text{act}_s(u))+\eta^2\hat{\mu}_s^2(\text{act}_s(u)))\\ &\qquad -\sum_{u\in A_s} p_{\tau_0(s)}(u)(\eta\hat{\mu}_s(u)+\eta^2\hat{\mu}_s^2(u))\Bigg]\\ &\quad +\sum_{u\in A_s} p_{\tau_0(s)}(u)(\eta\hat{\mu}_s(u)+\eta^2\hat{\mu}_s^2(u))\\ &\le \underbrace{2\eta\sum_{u\in A_{\tau-1}}\left(p_{\tau-1}(u)-p_{\tau_0(s)}(\text{act}_s(u))\frac{L(u)^d}{L(\text{act}_s(u))^d}\right)\hat{\mu}_s(\text{act}_s(u))}_{\text{(a)}}\\ &\quad +\underbrace{\sum_{u\in A_s} p_{\tau_0(s)}(u)(\eta\hat{\mu}_s(u)+\eta^2\hat{\mu}_s^2(u))}_{\text{(b)}}.\end{aligned}$$

We now analyze (a) and (b). For part (a), we view it as an inner-product. Specifically, the first part compares the probability at timestep $\tau-1$ and the probability at timestep $\tau(s)$, when we zoom-in both the active set into $A_{\tau-1}$ for comparison. Note that the zoom-in step does not change (a). We write $\tau_0(s)$ be the value of τ before computing π_s, and $\tau_1(s)$ be the value of τ immediately before using the feedback $g_s(x_s)$. Then, we formally write

$$\text{(a)} \le 2\eta\langle\max\{0, p_{\tau_1(s)}-p_{\tau_0(s)}\}, \hat{\mu}_s\rangle.$$

Part (b) represents what would the potential change if there is no delay. We now analyze part (b). By that $\pi_s(u) = (1-\gamma_s)p_{\tau_0(s)}(u)+\gamma_s/|A_s|$ and the assumption $\gamma_s \le \frac{1}{2}$ for all s, $p_{\tau_0(s)}(u) \le$

$\frac{1}{1-\gamma_s}\pi_s(u) \le (1+2\gamma_s)\pi_s(u)$, so

$$\begin{aligned}
\text{(b)} &\le (1+2\gamma_s)\sum_{u\in A_s}\pi_s(u)(\eta\hat{\mu}_s(u)+\eta^2\hat{\mu}_s(u)) \\
&\le (1+2\gamma_s)\eta\sum_{u\in A_s}\pi_s(u)\frac{\mu(x_s)1\{u_s=u\}+(1+4\ln T)\beta}{\pi_s(u)} \\
&\quad + (1+2\gamma_s)\eta^2\sum_{u\in A_s}\left(\pi_s(u)\frac{\mu(x_s)1\{u_s=u\}+(1+4\ln T)\beta}{\pi_s(u)}\right)\hat{\mu}_s(u) \\
&\le (1+2\gamma_s)\eta\left(\mu(x_s)+(1+4\ln T)\beta|A_s|\right) \\
&\quad + 2\eta^2(1+(1+4\ln T)\beta)\sum_{u\in A_s}\hat{\mu}_s(u) \\
&\le \eta\mu(x_s)+2\gamma_s\eta\mu(x_s)+(1+2\gamma_s)\eta\gamma_s+\eta\beta\cdot 4(1+2\ln T)\sum_{u\in A_s}\hat{\mu}_s(u).
\end{aligned}$$

Finally, summing over τ, we have

$$\begin{aligned}
\Phi(T)-\Phi(0) &\le 2\eta\langle\max\{0,p_{\tau_1(s)}-p_{\tau_0(s)}\},\hat{\mu}_s\rangle \\
&\quad + \eta\sum_{s=1}^{T}\mu(x_s)+4\eta\sum_{s=1}^{T}\gamma_s+4(1+2\ln T)\eta\beta\sum_{s=1}^{T}\sum_{u\in A_s}\hat{\mu}_s(u).
\end{aligned}$$

Applying both upper and lower bounds of the potential, and dividing both sides by η, we get

$$\begin{aligned}
\sum_{s=1}^{T}\hat{\mu}_s(\text{act}_s(u^*))-\sum_{s=1}^{T}\mu(x_s) &\le -\frac{d\ln L(u^*)}{\eta}+4\sum_{s=1}^{T}\gamma_s+4(1+2\ln T)\beta\sum_{s=1}^{T}\sum_{u\in A_s}\hat{\mu}_s(u) \\
&\quad + 2\sum_{s=1}^{T}\langle\max\{0,p_{\tau_1(s)}-p_{\tau_0(s)}\},\hat{\mu}_s\rangle.
\end{aligned}$$

□

Using the above lemma, we can partition the "estimated" regret into several terms. The next lemma then apply concentration results to provide the true regret bound.

Lemma C.2. *Choose parameters so that $\eta=\beta\ge\frac{1}{T}$, $\gamma_t=(2+4\ln T)\beta|A_t|$. Then, with probability $1-O(T^{-2})$, the regret is bounded by*

$$R(T)\le\frac{2dh(u^*)}{\eta}+48\eta\ln T\sum_{s=1}^{T}|A_s|+24\ln^2 T+2\sum_{s=1}^{T}\langle\max\{0,p_{\tau_1(s)}-p_{\tau_0(s)}\},\hat{\mu}_s\rangle.$$

Proof. By Lemma C.1, C.13 and C.12, with probability $1 - O(T^{-2})$,

$$\begin{aligned}
\sum_{s=1}^{T} \mu_s(x^*) - \sum_{s=1}^{T} \mu_s(x_s) &\le \sum_{s=1}^{T} \mu_s(\mathrm{act}_s(u)) - \sum_{s=1}^{T} \mu_s(x_s) + \frac{6 \ln T}{\beta} \\
&\le -\frac{d \ln L(u^*)}{\eta} + 4 \sum_{s=1}^{T} \gamma_s + 4(1 + 2 \ln T)\eta \sum_{s=1}^{T} \sum_{u \in A_s} \hat{\mu}_s(u) \\
&\quad + 2 \sum_{s=1}^{T} \langle \max\{0, p_{\tau_1(s)} - p_{\tau_0(s)}\}, \hat{\mu}_s \rangle. \\
(\text{Use Lemma C.12}) &\le \frac{2dh(u^*)}{\eta} + 4 \sum_{s=1}^{T} \gamma_s \\
&\quad + 4(1 + 2 \ln T) \left[\beta \sum_{s=1}^{T} \sum_{u \in A_s} \mu_s(u) + 2 \ln T + \beta \sum_{s=1}^{T} |A_s| \right] \\
&\quad + 2 \sum_{s=1}^{T} \langle \max\{0, p_{\tau_1(s)} - p_{\tau_0(s)}\}, \hat{\mu}_s \rangle. \\
(\text{Use } \gamma_s = (2 + 4 \ln T)\eta) &= \frac{2dh(u^*)}{\eta} + 16(1 + 2 \ln T)\eta \sum_{s=1}^{T} |A_s| + 8(1 + 2 \ln T) \ln T \\
&\quad + 2 \sum_{s=1}^{T} \langle \max\{0, p_{\tau_1(s)} - p_{\tau_0(s)}\}, \hat{\mu}_s \rangle.
\end{aligned}$$

Then we obtain the result. □

Lemma C.3.

$$\mathbb{E}\left[\sum_{s=1}^{T} \| \max\left\{ p_{\tau_1(s)} - p_{\tau_0(s)}, 0 \right\} \|_1 \right] \le 8\eta \sum_{s=1}^{T} d_s.$$

Proof.

$$\| \max\left\{ p_{\tau_1(s)} - p_{\tau_0(s)}, 0 \right\} \|_1 \le \sum_{\tau=\tau_0(s)+1}^{\tau_1(s)} \| \max\{p_\tau - p_{\tau-1}, 0\} \|_1.$$

Now we focus on two adjacent steps, and note that zooming-in does not change the 1-norm. Assume that the τ-th observed reward comes from the t-th pull. Then, for each $u \in A_{\tau-1}$,

$$\begin{aligned}
p_\tau(u) - p_{\tau-1}(u) &= \frac{w_{\tau-1}(u) \cdot \exp\left(\eta \hat{\mu}_t(\mathrm{act}_t(u))\right)}{\sum_{v \in A_{\tau-1}} w_{\tau-1}(v) \cdot \exp\left(\eta \hat{\mu}_t(\mathrm{act}_t(v))\right)} - \frac{w_{\tau-1}(u)}{\sum_{v \in A_{\tau-1}} w_{\tau-1}(v)} \\
&\le \frac{w_{\tau-1}(u) \cdot \exp\left(\eta \hat{\mu}_t(\mathrm{act}_t(u))\right)}{\sum_{v \in A_{\tau-1}} w_{\tau-1}(v)} - \frac{w_{\tau-1}(u)}{\sum_{v \in A_{\tau-1}} w_{\tau-1}(v)} \\
&\le p_{\tau-1}(u) \cdot \left[\exp\left(\eta \hat{\mu}_t(\mathrm{act}_t(u))\right) - 1 \right].
\end{aligned}$$

Note that the definition of the parameters ensures that $\eta \hat{\mu}_t(\mathrm{act}_t(u)) \le 1$, so using $e^x \le 1 + 2x$ for $x \le 1$,

$$\begin{aligned}
p_\tau(u) - p_{\tau-1}(u) &\le 2p_{\tau-1}(u) \cdot \eta \left(\frac{g(x_t) 1\{u_t = \mathrm{act}_t(u)\}}{\pi_t(\mathrm{act}_t(u))} + \frac{(1 + 4 \ln T)\beta}{\pi_t(\mathrm{act}_t(u))} \right) \\
&\le 2p_{\tau-1}(u) \cdot \eta \left(\frac{1\{u_t = \mathrm{act}_t(u)\}}{\pi_t(\mathrm{act}_t(u))} + \frac{(1 + 4 \ln T)\beta |A_t|}{\gamma_t} \right) \\
&\le 2p_{\tau-1}(u) \cdot \eta \left(\frac{1\{u_t = \mathrm{act}_t(u)\}}{\pi_t(\mathrm{act}_t(u))} + 1 \right).
\end{aligned}$$

Note that u_t is independent of $\{p_0, p_1, \ldots, p_{\tau-1}\}$, since its feedback is never used until the τ-th update. Therefore, let $\mathcal{F}_\tau$ be the information filtration used for computing $\{p_0, \ldots, p_{\tau-1}\}$, we have

$$\begin{aligned}\mathbb{E}[\max\{p_\tau(u) - p_{\tau-1}(u), 0\}|\mathcal{F}_{\tau-1}] &\le 2p_{\tau-1}\eta\left(\frac{1}{\pi_t(\text{act}_t(u))}\mathbb{E}\left[1\{u_t = \text{act}_t(u)|\mathcal{F}_{\tau-1}\}\right] + 1\right)\\ &= 2p_{\tau-1}\eta\left(\frac{1}{\pi_t(\text{act}_t(u))}\mathbb{E}\left[1\{u_t = \text{act}_t(u)|\pi_t\}\right] + 1\right)\\ &= 2p_{\tau-1}\eta(1+1) = 4p_{\tau-1}\eta.\end{aligned}$$

Summing over u, we get

$$\mathbb{E}\left[\|\max\{p_\tau - p_{\tau-1}, 0\}\|_1|\mathcal{F}_{\tau-1}\right] \le 4\eta\sum_{u\in A_{\tau-1}} p_{\tau-1}(u) \le 4\eta.$$

This implies that

$$\mathbb{E}\left[\|\max\left\{p_{\tau_1(s)} - p_{\tau_0(s)}, 0\right\}\|_1\right] \le 4\eta(\tau_1(s) - \tau_0(s)).$$

We now analyze

$$\sum_{t=1}^{T}[\tau_1(t) - \tau_0(t)],$$

which is determined by the delays d_t. We apply similar approach as Lemma 4 of Bistritz et al. (2019). Recall that $U_t = \{s : s + d_s = t\}$ is the set of feedback received after t-th pull. Without loss of generality, we assume that if multiple feedback arrives at the same time, we perform the update in the order of their source timesteps (i.e., s). It is clear that changing the order does not affect the sum we are considering.

For $\tau_1(t) - \tau_0(t)$, there are two kinds of observations contributes to the difference. $s > t$ with $s + d_s < t + d_t$, i.e., $s > t$ but s is observed before observing t; and $s < t$ with $t \le s + d_s \le t + d_t$, i.e., $s < t$, not observed before t-th pull, but observed before observing t.

For the first part, it is clear that $s \in (t, t + d_t)$, so it is directly bounded by d_t. For the second part, we fix such s and consider how many t would satisfy such condition. It is then clear that $t \in (s, s + d_s]$, so for each s, it contributes to at most d_s different t's. Therefore,

$$\sum_{t=1}^{T}[\tau_1(t) - \tau_0(t)] \le 2\sum_{t=1}^{T} d_t.$$

The lemma is then obtained by combining the results. □

Lemma C.4.

$$\mathbb{E}\left[\sum_{s=1}^{T}\langle\max\{0, p_{\tau_1(s)} - p_{\tau_0(s)}\}, \hat{\mu}_s\rangle\right] \le 16\eta\sum_{s=1}^{T} d_s.$$

Proof. Note that given π_s, $p_{\tau_1(s)}$ is independent with x_s and $\hat{\mu}_s(x_s)$, since the value of x_s and $\hat{\mu}(x_s)$ is never used before updating the weight $w_{\tau_1(s)+1}$. Therefore,

$$\begin{aligned}\mathbb{E}\left[\langle\max\{0, p_{\tau_1(s)} - p_{\tau_0(s)}\}, \hat{\mu}_s\rangle|\pi_s\right] &= \langle\mathbb{E}\left[\max\{0, p_{\tau_1(s)} - p_{\tau_0(s)}\}|\pi_s\right], \mathbb{E}\left[\hat{\mu}_s|\pi_s\right]\rangle\\ &\le \|\mathbb{E}\left[\max\{0, p_{\tau_1(s)} - p_{\tau_0(s)}\}|\pi_s\right]\|_1 \cdot \|\mathbb{E}\left[\hat{\mu}_s|\pi_s\right]\|_\infty.\end{aligned}$$

However, since for any $u \in A_s$, $\hat{\mu}_s(u) \ge 0$, and

$$\begin{aligned}\mathbb{E}[\hat{\mu}_s(u)|\pi_s] &= \frac{1}{\pi_s(u)}\mathbb{E}[\mu(x_s)1\{u = u_s\}] + \frac{(1 + 4\ln T)\beta}{\pi_s(u)}\\ &\le \frac{\Pr\{u = u_s\}}{\pi_s(u)} + \frac{|A_s|(1 + 4\ln T)\beta}{\gamma_s}\\ &\le 1 + 1 = 2,\end{aligned}$$

we have $\|\mathbb{E}\left[\hat{\mu}_s|\pi_s\right]\|_\infty \le 2$. Therefore, taking sum over s and apply Lemma C.3, we have

$$\mathbb{E}\left[\sum_{s=1}^{T}\langle\max\{0, p_{\tau_1(s)} - p_{\tau_0(s)}\}, \hat{\mu}_s\rangle\right] \le 2\mathbb{E}\left[\sum_{s=1}^{T}\|\max\left\{p_{\tau_1(s)} - p_{\tau_0(s)}, 0\right\}\|_1\right] \le 16\eta\sum_{s=1}^{T} d_s.$$

□

We now provide the proof of theorem.

Proof of Theorem 4.1. Choose

$$\eta = \beta = \Theta\left(\frac{\sqrt{d\ln T}}{\sqrt{2^d}T^{(d_z+1)/(d_z+2)}c^{\frac{1}{d_z+2}}\sqrt{\ln T} + \sqrt{D}}\right),$$

and

$$\gamma_t = (2 + 4\ln T)\beta|A_t|.$$

By Lemma C.17, with probability $1 - O(T^{-2})$, $|A_T| = O\left(2^d T^{\frac{d_z}{d_z+2}} c^{\frac{2}{d_z+2}}\right)$, so $\gamma_t \le \frac{1}{2}$ can be satisfied. by Lemmas C.2 and C.6, with probability $1 - O(T^{-2})$,

$$R(T) \le \frac{2d\ln T}{\eta} + 48\eta\ln T \cdot T|A_T| + 24\ln^2 T + 2\sum_{s=1}^{T}\langle\max\{0, p_{\tau_1(s)} - p_{\tau_0(s)}\}, \hat{\mu}_s\rangle.$$

Apply the bound by C.17,

$$R(T) \le O\left(\sqrt{d2^d}T^{\frac{d_z+1}{d_z+2}}c^{\frac{1}{d_z+2}}\ln T\right) + 2\sum_{s=1}^{T}\langle\max\{0, p_{\tau_1(s)} - p_{\tau_0(s)}\}, \hat{\mu}_s\rangle.$$

Taking expectation and apply Lemma C.4, since the total regret is always upper-bounded by T,

$$\begin{aligned}
\mathbb{E}[R(T)] &\le O\left(\sqrt{d2^d}T^{\frac{d_z+1}{d_z+2}}c^{\frac{1}{d_z+2}}\ln T + \sqrt{dD\ln T}\right) + 2\sum_{s=1}^{T}\langle\max\{0, p_{\tau_1(s)} - p_{\tau_0(s)}\}, \hat{\mu}_s\rangle \\
&\le O\left(\sqrt{d2^d}T^{\frac{d_z+1}{d_z+2}}c^{\frac{1}{d_z+2}}\ln T + \sqrt{dD\ln T}\right) + O\left(\eta D\right) \\
&\le O\left(\sqrt{d2^d}T^{\frac{d_z+1}{d_z+2}}c^{\frac{1}{d_z+2}}\ln T + \sqrt{dD\ln T}\right).
\end{aligned}$$

□

C.2 Lemmas for Zooming Processes

The following lemmas are similar to those by Podimata & Slivkins (2021), where they are rewritten to fit our algorithm.

Lemma C.5. *Suppose a non-root node u is deactivated at round $t_d(u)$. Let v be its parent, then $t_d(u) \ge 2t_d(v) - 1$.*

Proof. The node v does not satisfy the zoom-in rule at round $t_d(v) - 1$, so

$$\sum_{t=1}^{t_d(v)-1}\frac{\beta}{\pi_t(\text{act}_t(v))} + \frac{1}{\beta} \ge (t_d(v) - 1)L(v) = (2t_d(v) - 2)L(u).$$

Note that $\text{conf}_t(u)$ is non-decreasing in t, we know that $\text{conf}_t(u) \ge (2t_d(v) - 2)L(u)$, meaning that $t_d(u) \ge 2t_d(v) - 1$. □

Lemma C.6. *The height of any active node is at most $\log_2\frac{1}{\beta}$. Specifically, as long as $\frac{1}{\beta} = O(T)$, the height is bounded by $O(\log T)$.*

Proof. When a node u zooms-in at round t, it does not satisfy the zoom-in rule at round $t-1$, so

$$\text{conf}_t(u) < tL(u), \quad \text{conf}_{t-1}(u) \geq (t-1)L(u).$$

Taking the difference,

$$\frac{\beta}{\pi_t(u)} < L(u) = 2^{-h(u)},$$

which means that

$$h(u) \leq \log_2 \left(\frac{\pi_t(u)}{\beta} \right) \leq \log_2 \frac{1}{\beta}.$$

□

Lemma C.7. *Suppose $\beta \geq 1/T$. For a node u active at round t,*

$$\sum_{s=1}^{t} L(\text{act}_s(u)) \leq 2tL(u) \log_2 T.$$

Proof. Let $v_0, v_1, \ldots, v_{h(u)}$ be the path from root to u, where $v_{h(u)} = u$. Then,

$$\begin{aligned}\sum_{s=1}^{t} L(\text{act}_s(u)) &= t_d(v_0)L(v_0) + \sum_{s=1}^{h(u)-1} [t_d(v_s) - t_d(v_{s-1})]L(v_s) + (t - t_d(v_{h(u)-1}))L(u) \\ &= \sum_{s=0}^{h(u)-1} t_d(v_s)[L(v_s) - L(v_{s+1})] + tL(u).\end{aligned}$$

By Lemma C.5,

$$t_d(v_s) \leq \frac{t_d(v_{s+1}) + 1}{2} \leq \cdots \leq 2^{s-h(s)+1} t_d(v_{h(s)-1}) + \sum_{r=1}^{s-h(s)+1} 2^{-r} \leq 2^{s-h(s)+1} t_d(v_{h(s)-1}) + 1.$$

So

$$\begin{aligned}\sum_{s=1}^{t} L(\text{act}_s(u)) &\leq \frac{1}{2} \sum_{s=0}^{h(u)-1} L(v_s)(2^{s-h(s)+1} t_d(v_{h(s)-1}) + 1) + tL(u) \\ &= \frac{1}{2} \sum_{s=0}^{h(u)-1} 2^{h(s)-s} L(u)(2^{s-h(s)+1} t_d(v_{h(s)-1}) + 1) + tL(u) \\ &\leq L(u) + tL(u) + \sum_{s=0}^{h(u)-1} L(u) t_d(v_{h(s)-1}) \\ &\leq L(u) + tL(u) + tL(u)h(u) \leq 2tL(u) \log_2 T\end{aligned}$$

□

Lemma C.8. *(Lemma C.4 by Podimata & Slivkins (2021)) Let $[t_a(u), t_d(u)]$ be the interval where u is active, and assume that $t_d(u) < T$, i.e., u satisfies the zoom-in rule in the end. Then,*

$$\sum_{t=t_a(u)}^{t_d(u)} \pi_t(u) \geq \frac{1}{9L(u)^2}.$$

Proof. Let v be the parent of u.

$$\begin{aligned}t_d(u)L(u) &> \sum_{t=1}^{t_d(u)} \frac{\beta}{\pi_t(\text{act}_t(u))} + \frac{1}{\beta} \\ &\geq \sum_{t=t_a(u)}^{t_d(u)} \frac{\beta}{\pi_t(u)}.\end{aligned}$$

Since the function $x \mapsto x^{-1}$ is convex, Jensen's inequality shows that $\frac{1}{n}\sum_{i=1}^{n} x_i^{-1} \geq \frac{n}{\sum_{i=1}^{n} x_i}$, so

$$\sum_{t=t_a(u)}^{t_d(u)} \frac{\beta}{\pi_t(u)} \geq \frac{\beta(t_d(u)-t_d(v))^2}{\sum_{t=t_a(u)}^{t_d(u)} \pi_t(u)} \geq \frac{\beta(t_d(u)-(t_d(u)+1)/2)^2}{\sum_{t=t_a(u)}^{t_d(u)} \pi_t(u)} \geq \frac{\beta t_d(u)^2}{9\sum_{t=t_a(u)}^{t_d(u)} \pi_t(u)},$$

where the second last inequality comes from Lemma C.5. Combining the inequalities,

$$\sum_{t=t_a(u)}^{t_d(u)} \pi_t(u) \geq \frac{\beta t_d(u)^2}{9 t_d(u) L(u)} \geq \frac{\beta t_d(u)}{9L(u)}.$$

Since $\text{conf}_t(u) \geq \frac{1}{\beta}$, the zoom-in rule ensures that $t_d(u)L(u) > \frac{1}{\beta}$, so

$$\sum_{t=t_a(u)}^{t_d(u)} \pi_t(u) \geq \frac{1}{9L(u)^2}.$$

□

C.3 Lemmas for Concentration

The first lemma comes from the results by Podimata & Slivkins (2021), which only depends on the probability π_t used when we are sampling arms. Denote the average return for node u as $\mu_t(u)$.

Lemma C.9. *(Lemma C.9 of Podimata & Slivkins (2021)) Fix a round t and a sequence of sets $A'_s \subseteq A_s$. With probability at least $1-2\delta$,*

$$\left|\beta\sum_{s=1}^{t}\sum_{u\in A'_s}\mu_s(u) - \beta\sum_{s=1}^{t}\sum_{u\in A'_s}\frac{1\{u=u_s\}\mu_s(x_s)}{\pi_s(u)}\right| \leq \beta^2\sum_{s=1}^{t}\sum_{u\in A'_s}\frac{1}{\pi_s(u)} + \ln(1/\delta).$$

Lemma C.10. *(Corollary C.1 of Podimata & Slivkins (2021)) Fix a round t and a node $u \in A_t$. With probability at least $1-2\delta$,*

$$\left|\sum_{s=1}^{t}\mu_s(\text{act}_s(u)) - \sum_{s=1}^{t}\frac{\mu_s(x_s)1\{u_s=\text{act}_s(u)\}}{\pi_s(\text{act}_s(u))}\right| \leq \beta\sum_{s=1}^{t}\frac{1}{\pi_s(\text{act}_s(u))} + \frac{\ln(1/\delta)}{\beta}.$$

Proof. Apply Lemma C.9 to singleton $A'_s = \{\text{act}_s(u)\}$ □

Lemma C.11. *Fix a round t and a node $u \in A_{t+1}$. Let $\tilde{U}_t = \{s : s+d_s \leq t\}$. Then with probability at least $1-2\delta$,*

$$\left|\sum_{s\in\tilde{U}_t}\mu_s(\text{act}_s(u)) - \sum_{s\in\tilde{U}_t}\frac{\mu_s(x_s)1\{u_s=\text{act}_s(u)\}}{\pi_s(\text{act}_s(u))}\right| \leq \beta\sum_{s\in\tilde{U}_t}\frac{1}{\pi_s(\text{act}_s(u))} + \frac{\ln(1/\delta)}{\beta}.$$

Proof. For $s \in \tilde{U}_t$, take $A'_s = \{\text{act}_s(u)\}$. Otherwise take $A'_s = \varnothing$. Then apply Lemma C.9. □

The next few lemmas follows Podimata & Slivkins (2021), but we write the proof explicitly to refine some constants.

Lemma C.12. *Fix a round t, then with probability at least $1-2\delta$,*

$$\beta\sum_{s=1}^{t}\sum_{u\in A_s}\hat{\mu}_s(u) - \beta\sum_{s=1}^{t}\sum_{u\in A_s}\mu_s(u) \leq \ln(1/\delta) + \beta\sum_{s=1}^{T}|A_s|.$$

Proof. By Lemma C.9, with probability at least $1-2\delta$,

$$\begin{aligned}
&\beta\sum_{s=1}^{t}\sum_{u\in A_s}\hat{\mu}_s(u)-\beta\sum_{s=1}^{t}\sum_{u\in A_s}\mu_s(u)\\
&\le \beta\sum_{s=1}^{t}\sum_{u\in A_s}\left[\frac{\mu_s(x_s)1\{u_t=u\}}{\pi_s(u)}-\mu_s(u)+\frac{(1+4\ln T)\beta}{\pi_s(u)}\right]\\
&\le \ln(1/\delta)+\beta\sum_{s=1}^{t}\sum_{u\in A_s}\frac{\beta+(1+4\ln T)\beta}{\pi_s(u)}\\
&\le \ln(1/\delta)+\beta\sum_{s=1}^{t}\sum_{u\in A_s}\frac{|A_s|(2+4\ln T)\beta}{\gamma_s}\\
&\le \ln(1/\delta)+\beta\sum_{s=1}^{t}|A_s|.
\end{aligned}$$

□

Lemma C.13. *Fix a round t and active node u. Let $x^*\in u$ be an arbitrary arm. With probability at least $1-2\delta$,*

$$\sum_{s=1}^{t}\hat{\mu}_s(\mathrm{act}_s(u))\ge\sum_{s=1}^{t}\mu_s(x^*)-\frac{4\ln T+\ln(1/\delta)}{\beta}.$$

Proof. By Lemma C.10, with probability $1-2\delta$,

$$\begin{aligned}
\sum_{s=1}^{t}\hat{\mu}_s(u)&=\sum_{s=1}^{t}\frac{\mu_s(x_s)1\{u_s=u\}}{\pi_s(u)}+\sum_{s=1}^{t}\frac{(1+4\ln T)\beta}{\pi_s(u)}\\
&\ge\sum_{s=1}^{t}\mu_s(\mathrm{act}_s(u))+\sum_{s=1}^{t}\frac{4\beta\ln T}{\pi_s(u)}-\frac{\ln(1/\delta)}{\beta}\\
&\ge\sum_{s=1}^{t}\mu_s(x^*)-\sum_{s=1}^{t}L(\mathrm{act}_s(u))+\sum_{s=1}^{t}\frac{4\beta\ln T}{\pi_s(u)}-\frac{\ln(1/\delta)}{\beta}.
\end{aligned}$$

By the zoom-in rule and Lemma C.7,

$$\sum_{s=1}^{t}\frac{4\beta\ln T}{\pi_s(\mathrm{act}_s(u))}-\sum_{s=1}^{t}L(\mathrm{act}_s(u))\ge 4(t-1)L(u)\ln T-\frac{4\ln T}{\beta}-2tL(u)\log_2 T\ge-\frac{4\ln T}{\beta}.$$

□

Lemma C.14. *Fix a round t and active node $u\in A_{t+1}$. Let $x^*\in u$ be an arbitrary arm. With probability at least $1-2\delta$,*

$$\sum_{s\in\tilde{U}_t}\hat{\mu}_s(\mathrm{act}_s(u))\ge\sum_{s\in\tilde{U}_t}\mu_s(x^*)-\frac{4\ln T+\ln(1/\delta)}{\beta}-(t-|\tilde{U}_t|).$$

Proof. Similar to the proof of Lemma C.13, by Lemma C.11

$$\begin{aligned}\sum_{s\in\tilde{U}_t}\hat{\mu}_s(u) &= \sum_{s\in\tilde{U}_t}\frac{\mu_s(x_s)1\{u_s=u\}}{\pi_s(u)} + \sum_{s\in\tilde{U}_t}\frac{(1+4\ln T)\beta}{\pi_s(u)}\\ &\geq \sum_{s\in\tilde{U}_t}\mu_s(\mathrm{act}_s(u)) + \sum_{s\in\tilde{U}_t}\frac{4\beta\ln T}{\pi_s(u)} - \frac{\ln(1/\delta)}{\beta}\\ &\geq \sum_{s\in\tilde{U}_t}\mu_s(x^*) - \sum_{s\in\tilde{U}_t}L(\mathrm{act}_s(u)) + \sum_{s\in\tilde{U}_t}\frac{4\beta\ln T}{\pi_s(u)} - \frac{\ln(1/\delta)}{\beta}\\ &\geq \sum_{s\in\tilde{U}_t}\mu_s(x^*) - \sum_{s=1}^{t}L(\mathrm{act}_s(u)) + \sum_{s=1}^{t}\frac{4\beta\ln T}{\pi_s(u)} - \frac{\ln(1/\delta)}{\beta} - (t-|\tilde{U}_t|)\\ &\geq \sum_{s\in\tilde{U}_t}\mu_s(x^*) - \frac{4\ln T+\ln(1/\delta)}{\beta} - (t-|\tilde{U}_t|).\end{aligned}$$

Here, the second last inequality comes from $\frac{4\beta\ln T}{\pi_s(u)} \leq \frac{4\beta|A_s|\ln T}{\gamma_s} \leq 1$. □

C.4 Bound on the Number of Activated Nodes

In this section, we control the number of activated nodes. We will provide a worst-case estimation through covering dimension and a high-probability through the modified adversarial zooming dimension. Recall $\tilde{U}_t := \{s : s + d_s \leq t\}$, that is, $\tilde{U}_t$ contains those rewards observable before t-th pull. Then,

$$w_{\tau_0(t)}(u) = L(u)^d \exp\left(\eta \sum_{s\in\tilde{O}_{t-1}}\hat{\mu}_s(\mathrm{act}_s(u))\right).$$

Lemma C.15. *If u is zoomed in after round t with $L(u) \leq \frac{1}{3+4\ln T}$, then for any active node u^*,*

$$\sum_{s\in\tilde{O}_{t-1}}[\hat{\mu}_s(\mathrm{act}_s(u^*)) - \hat{\mu}_s(\mathrm{act}_s(u))] \leq \frac{\ln\left(\frac{L(u)^d}{\beta L(u^*)^d}\right)}{\eta}.$$

Proof. Recall that when proving Lemma C.6, we show that $\frac{\beta}{\pi_t(u)} \leq L(u)$ must be satisfied. Then, by $\pi_t(u) = (1-\gamma_t)p_{\tau_0(t)}(u) + \frac{\gamma_t}{|A_t|}$,

$$p_t(u) = \frac{\pi_t(u) - \gamma_t/|A_t|}{1-\gamma_t} \geq \frac{\beta}{L(u)} - (2+4\ln T)\beta \geq \beta,$$

where the last inequality is by $L(u)^{-1} \geq 3 + 4\ln T$.

By the definition of $p_t(u)$, this implies that

$$\begin{aligned}w_{\tau_0(t)}(u) &= L(u)^d \exp\left(\eta \sum_{s\in\tilde{O}_{t-1}}\hat{\mu}_s(\mathrm{act}_s(u))\right)\\ &\geq \beta\sum_{v\in A_t} w_{\tau_0(t)}(v)\\ &\geq \beta L(u^*)^d \exp\left(\eta \sum_{s\in\tilde{O}_{t-1}}\hat{\mu}_s(\mathrm{act}_s(u^*))\right).\end{aligned}$$

Taking logarithm on both sides,

$$\sum_{s\in\tilde{O}_{t-1}}[\hat{\mu}_s(\mathrm{act}_s(u^*)) - \hat{\mu}_s(\mathrm{act}_s(u))] \leq \frac{\ln\left(\frac{L(u)^d}{\beta L(u^*)^d}\right)}{\eta}.$$

□

Lemma C.16. *Fix a node u with $L(u) \le \frac{1}{3+4\ln T}$ and suppose it is zoomed-in in round t. Then with probability at least $1 - O(T^{-3})$,*

$$\frac{1}{t}\sum_{s\in\tilde{O}_{t-1}} [\mu_s(x^*) - \mu_s(x)] \le O(L(u)\ln T + \sqrt{D}/t).$$

Proof. Let $x \in u$, and let $x^* \in u^*$ be an arbitrary arm. By Lemma C.11, with probability at least $1 - 2\delta$,

$$\begin{aligned}
&\sum_{s\in\tilde{O}_{t-1}} [\hat{\mu}_s(\text{act}_s(u)) - \mu_s(x)] \\
&\le \sum_{s\in\tilde{O}_{t-1}} [\mu_s(\text{act}_s(u)) - \mu_s(x)] + \sum_{s\in\tilde{O}_{t-1}} \frac{\beta + (1+4\ln T)\beta}{\pi_s(\text{act}_s(u))} + \frac{\ln(1/\delta)}{\beta} \\
&\le \sum_{s=1}^{t} L(\text{act}_s(u)) + \sum_{s=1}^{t} \frac{(2+4\ln T)\beta}{\pi_s(\text{act}_s(u))} + \frac{\ln(1/\delta)}{\beta}.
\end{aligned}$$

By Lemma C.7, $\sum_{s=1}^{t} L(\text{act}_s(u)) \le 4tL(u)\ln T$. Also by the zoom-in rule,

$$\sum_{s=1}^{t} \frac{\beta}{\pi_s(\text{act}_s(u))} \le tL(u).$$

Hence,

$$\sum_{s\in\tilde{O}_{t-1}} [\hat{\mu}_s(\text{act}_s(u)) - \mu_s(x)] \le 10tL(u)\ln T + \frac{\ln(1/\delta)}{\beta}.$$

Then, use Lemma C.14 and Lemma C.15,

$$\begin{aligned}
\sum_{s\in\tilde{O}_{t-1}} [\mu_s(x^*) - \mu_s(x)] \le &\sum_{s\in\tilde{O}_{t-1}} [\mu_s(x^*) - \hat{\mu}_s(\text{act}_s(u^*))] \\
&+ \sum_{s\in\tilde{O}_{t-1}} [\hat{\mu}_s(\text{act}_s(u^*) - \hat{\mu}_s(\text{act}_s(u))] \\
&+ \sum_{s\in\tilde{O}_{t-1}} [\hat{\mu}_s(\text{act}_s(u)) - \mu_s(x)] \\
&\le (t - 1 - |\tilde{O}_{t-1}|) \\
&\quad + \frac{2\ln(1/\delta) + 4\ln T}{\beta} + \frac{\ln\left(\frac{2L(u)^d}{\beta L(u^*)^d}\right)}{\eta} + 10tL(u)\ln T \\
&\le (t - 1 - |\tilde{O}_{t-1}|) \\
&\quad + \frac{2\ln(1/\delta) + 2d\ln T + \ln(2/\eta)}{\eta} + 10tL(u)\ln T.
\end{aligned}$$

The term $t - 1 - |\tilde{O}_{t-1}|$ represents the number of unobserved rewards, which is at most $O(\sqrt{D})$ since we require $d_s \ge t - 1 - s$ to make $s \notin \tilde{O}_{t-1}$. Note that $L(u) \ge \frac{1}{\beta}$ by Lemma C.6. If we take $\delta = T^{-3}$ and $\eta = \beta \ge \frac{1}{T}$, we have

$$\frac{1}{t}\sum_{s\in\tilde{O}_{t-1}} [\mu_s(x^*) - \mu_s(x)] \le O(L(u)\ln T + \sqrt{D}/t).$$

□

Lemma C.17. *With probability $1 - O(T^{-2})$, the total number of activated nodes are bounded by*

$$|A_T| \lesssim 2^d T^{\frac{d_z}{d_z+2}} c^{\frac{2}{d_z+2}},$$

where d_z is the adversarial c-zooming dimension.

Proof. We use the method by Podimata & Slivkins (2021) to prove this lemma. First, we show that in the worst case, the total number of zoomed-in nodes are at most $O(T)$. Note that for a node u, if we want u to zoom-in, according to Lemma C.8, we need to spend at least $\frac{1}{9L(u)^2}$ probability mass on it. Thus, in order to maximize the number of zoomed-in nodes, one should perform greedily and spend probability mass on large nodes. For a node with diameter l, the number of such nodes is at most l^{-d}. So if the layer of nodes with diameter l are all zoomed-in, l must satisfy

$$l^{-d} \cdot l^{-2} \leq 9T,$$

which indicates that $l^{-d} = O(T^{\frac{d}{d+2}}) = O(T)$. Therefore, the total number of zoomed-in nodes are at most $O(T)$.

Now, apply union bound and Lemma C.16, we know that with probability $1 - O(T^{-2})$, the inquality in Lemma C.16 holds for all u with $L(u) \leq \frac{1}{3+4\ln T}$. For nodes with $L(u) > \frac{1}{3+4\ln T}$, there are at most $O(\log^d T)$ such nodes, which is smaller than any $\Theta(T^\alpha)$ with $\alpha > 0$. Note that by Lemma C.8 again, $t_d(u) \geq \frac{1}{9L(u)^2}$, where $t_d(u)$ refers to the round when u is de-activated. In this case, Lemma C.16 implies that $\Delta_{t_d}(u) \leq O(L(u)\ln T + \sqrt{D}/t_d(u)) \leq O(L(u)\ln T + \sqrt{D}L(u)^2)$. So by our definition of the adversarial zooming dimension, for a diameter l, the number of u with $L(u) = l$ that zooms-in is at most $c \cdot l^{-d_z}$. Similarly to the greedy method, in order to maximize the number of zoomed-in nodes, one must perform the greedy method to fill large nodes first. Let H be the height of last layer that is completed in the greedy method, then it must satisfy

$$c\sum_{h=0}^{H}(2^{-h})^{-d_z} \cdot (2^{-h})^{-2} \leq 9T,$$

which implies that $c2^{(2+d_z)H} \leq T$, so the total number of activated nodes are at most

$$|A_T| \leq O(c2^{d_z H}) \leq O(c2^d \cdot (T/c)^{\frac{d_z}{d_z+2}}) = O\left(2^d T^{\frac{d_z}{d_z+2}} c^{\frac{2}{d_z+2}}\right).$$

□

D More details in the Experiments

In this section, we validate the performance of DLPP (Algorithm 1) and DLEXP3 (Algorithm 2) under stochastic reward. Here we want to illustrate how the performance is related to the noise scale σ. The results are shown in Figures 2 and 3. The DLPP algorithm is relatively more sensitive to the noise scale, since it requires more steps to distinguish less optimal arms from the best arm. In comparison, we can see that the performance of DLEXP3 algorithm is not affected by the noise scale, as it already assumes a worst-case reward sequence.

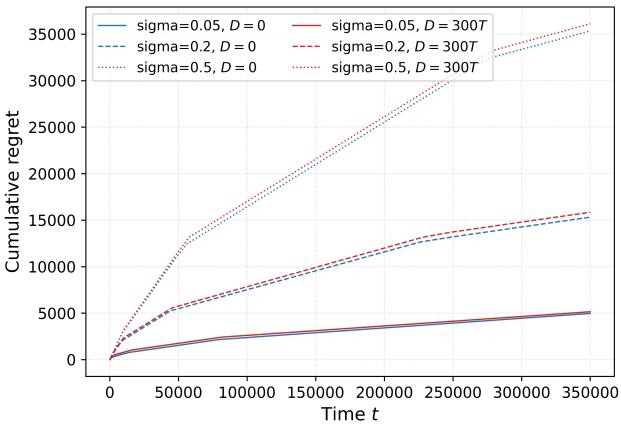


Figure 2: Performance of DLPP for different σ.

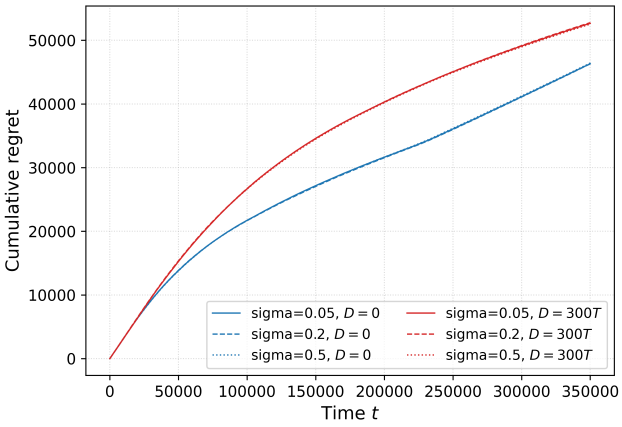


Figure 3: Performance of DLEXP3 for different σ.