Long-Memory AutoRegressive Bandits

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

The Autoregressive Fractionally Integrated (ARFIMA) processes naturally occur in the context of real-world scenarios that exhibit long memory properties. The construct of ARFIMA process allows us to easily formulate a separate class of stochastic multi-armed bandits problem (Abbasi-Yadkori et al. [2011]) by modifying its general mechanism. In this work, we introduce a novel setting named Long-Memory AutoRegressive Bandits (LM-ARBs), where the environment-generated reward evolves according to the fractionally integrated autoregressive process of the autoregressive order *p* and the fractional differencing parameter *d*, an extension of the autoregressive bandits characterized only by the autoregressive process. Then, we provide an optimistic regret-minimization algorithm `Long-Memory AutoRegressive Upper Confidence Bound (ARLM-UCB)` that suffers a sublinear regret of order $\mathcal{O}\left(\frac{(p+1)^2\sqrt{n}T^{2d+0.5}\log^2(T)\log\log(T)}{(1-\Gamma)^2}\right)$, where n is the number of actions, T is the optimization horizon, and $\Gamma < 1$ is a stability index of the fractionally differenced process. Finally, we conduct numerical experiments in synthetic environments to validate our algorithm effectiveness w.r.t. bandit baselines.

1 Introduction

Many real-world sequential decision-making problems require the learner to select an action that determines a long-term reward evolution, creating a temporal dependence for future rewards over a long time horizon. When analyzing this reward, the agent must account for a much slower decay of temporal dependence between the current reward and the sequence of past observations. Autoregressive Fractionally Integrated Moving-Average (ARFIMA) (Hosking [1981], McLeod and Hipel [1986], Granger and Joyeux [1980]) is widely used to model the long-term persistence in temporal dependence in the real-world phenomena, such as stock market volatility, temperature variations, earthquake magnitude sequences, and traffic data (Bakar and Hafner [2019], Yuan et al. [2014], Kondo Lembang et al. [2021], Doodipala [2020]). In the context of Reinforcement Learning (Sutton and Barto [2020]), this method flexibly allows one to model the long-term behavior of reward sequences. For example, the learner's ability to capture long-range dependence in stock price return dynamics enables more accurate forecasting of future trends, volatility clustering, and regime shifts that are not evident in short memory models (Liu [2000]).

In this paper, we model the reward of a decision-making process as an ARFIMA process, whose parameters depend on the action selected by the agent at every round. This scenario can be viewed as a separate class of stochastic bandit algorithms (Abbasi-Yadkori et al. [2011]), where the temporal structure of the reward is governed by the long memory ARFIMA process whose action-dependent parameters are unknown to the agent. In this setting, the agent faces a multi-staged challenge of estimating the ARFIMA parameters responsible for generating the reward in the given environment. Given such a complex learning process, this scenario displays remarkable differences to more traditional non-stationary learning problems. In our problem, the environment does not change by any exogenous sources of non-stationarity, which is often represented by a smooth changing in the

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

mean of rewards for each arm over time, studied by Trella et al. [2024] in the bandit context. That said, the reward dynamics in our proposed setting solely depends on actions selected by an agent.

1.1 Original Contribution

In this work, we propose a novel setting named Long-Memory AutoRegressive Bandits (LM-ARBs), in which the reward follows an ARFIMA$(p, d, 0)$ process, where p stands for AR order and the fractional differencing parameter d. We then devise a new optimistic algorithm `AutoRegressive Long-Memory Upper Confidence Bound (ARLM-UCB)` to learn a long-term optimal policy in online settings and show that it suffers a sublinear regret of order $\mathcal{O}\left(\frac{(p+1)^2\sqrt{n}T^{2d+0.5}\log^2(T)\log\log(T)}{(1-\Gamma)^2}\right)$, where n is the number of actions, T is the optimization horizon and $\Gamma < 1$ is a stability index of the fractionally differenced process. In the end, we empirically evaluate `ARLM-UCB` and compare its performance with other bandit baselines in our setting, illustrating that our proposed algorithm outperforms a number of popular multi-armed bandit (MAB) benchmarks that do and do not account for the temporal dynamics in the reward evolution process.

1.2 Related Work

This work proposes a modification of a traditional MAB learning problem by incorporating long-range temporal dependency into the reward evolution process. Many related work on creating temporal dynamics in bandit settings focused on addressing challenges like delayed feedback (Tang et al. [2024]), influence of past actions on future rewards (Tang et al. [2021]), or the non-stationarity of AR dynamics (Chen et al. [2023]). Bacchiocchi et al. [2024] presented AutoRegressive Bandits (ARBs) setting. This setting explicitly models autoregressive (AR) sequences, where the reward depends on several past observations of the length of a finite AR order p, which eventually makes this method neglect the long-term dependence of rewards over extended horizons. Thus, we acknowledge that the prevailing concensus highlights the importance of constructing temporal dependence in bandit frameworks.

Limited studies have addressed MAB settings with long-range temporal dependence between rewards. A recent study by Qin et al. [2023] introduces a framework for contextual bandits, where rewards depend on long-range temporal dependence between past actions and contexts. The major limitation of this method hinders in the assumption of sparsity in the reward structure, where only a finite number of contexts, much smaller from their total number, influence the current reward, which may not be realistic for many real-world settings. For instance, in scenarios where rewards depend on dense or complex long-range temporal patterns (e.g., cumulative effects across many past contexts), this assumption may fail to capture the full dependency structure, impairing the ability to model richer temporal dynamics.

The ARFIMA process has been well studied in the classical time series literature. Asymptotic theory for ARFIMA estimation was developed by Dahlhaus [1989] for maximum likelihood methods and Fox and Taqqu [1986] for general long memory processes. In the frequency domain, Robinson [1995] made seminal contributions by proposing Gaussian semiparametric estimators of the fractional differencing parameter and establishing their asymptotic properties, while Beran [1995] advanced time domain approaches including maximum likelihood estimation with rigorous asymptotic theory. However, only a few studies have addressed MAB settings with long-range temporal dependence between rewards. The key limitation of many existing machine learning methods is their inability to analyze temporal dependence on a long-range horizon of past observations. This notion was particularly validated by different studies (Al-Selwi et al. [2023], Zucchet et al. [2023]), demonstrating that traditional sequential modeling algorithms do not learn long-range dependence.

Gupta et al. [2021] proposed fractional dynamical systems with long-range filtering operations on state vectors, which closely relate to ARFIMA processes. In the context of deep learning, Zhao et al. [2020] have proposed a modification to a traditional recurrent neural network (RNN) architecture that enables capturing long memory from a time series perspective via new memory filter component directly incorporating ARFIMA process. Motivated by successes in extending machine learning frameworks with long memory processes, we directly implement ARFIMA modeling in our bandit setting.

2 Problem Formulation

In this section, we briefly discuss the formal representation of the ARFIMA(p, d, q) process in terms of the parameters used throughout this document and introduce the LM-ARB setting, the evolution of rewards, the formal problem of the learner, key assumptions, and definitions of policies and regret (Section 2.1). Subsequently, we establish several essential assumptions for convergent evolution of the utilized ARFIMA process (Section 2.2), present the closed-form solution for the optimal policy in our setting (Section 2.3), and describe the stochastic properties of the AR reward process (Section 2.4).

Notation. We will employ the following notation across the paper. Let $a, b \in \mathbb{N}$, with $a \leq b$, we introduce the symbols: $[\![a, b]\!] :\in \{a, \dots, b\}$ and $[\![b]\!] :\in \{1, \dots, b\}$. Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ be real-valued vectors, we denote with $\langle \mathbf{x},\mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y} = \Sigma_{i=1}^n x_i y_i$ the inner product. For a positive semi-definite matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, we denote $||\mathbf{x}||_{\mathbf{A}}^2 = \mathbf{x}^\top \mathbf{A}\mathbf{x}$ the weighted 2-norm. We define a zero-mean random variable ε is σ^2-subgaussian if $\mathbb{E}[e^{\lambda\varepsilon}] \leq e^{\frac{\lambda^2\sigma^2}{2}}$.

2.1 Long-Memory Autoregressive Bandits

The AutoRegressive Fractionally Integrated Moving-Average(ARFIMA) characterizes the fractional long-term evolution of the autoregressive component. The ARFIMA(p, d, q) process $\{X_t, t \in \mathbb{Z}\}$ is represented through the following form:

$$\left(1 - B\right)^d \left(1 - \Sigma_{i=1}^p \phi_i B^i\right) X_t = \left(1 + \Sigma_{i=1}^q \theta_i B^i\right) \varepsilon_t, \tag{1}$$

where ϕ_i, θ_j, ($i \in [\![p]\!]$ and $j \in [\![q]\!]$) are the coefficients of the model, B is the backshift operator defined as $B^j X_t = X_{t-j}$ for $j \in \mathbb{N}$, ε_t is the zero-mean $i.i.d.$ σ^2-subgaussian error term of the system, and $(1 - B)^d = \sum_{j=0}^\infty \psi_j(d)$, where $\psi_j(d) = \prod_{i=1}^j \frac{i-1-d}{i}$ for $j \geq 1$, with $\psi_0(d) \equiv 1$. We employ this process representation across this paper to create a long memory-reward evolution dynamics within our environment. For simplicity, we consider $q = 0$ in our proposed LM-ARB setting, with the potential to be further generalized.

We denote x_t to be an AR process of order p, which is represented in the form of Bacchiocchi et al. [2024]:

$$x_t = \phi_0(u_t) + \sum_{i=1}^p \phi_i(u_t) x_{t-i} + \varepsilon_t = \langle \boldsymbol{\phi}(u_t), \mathbf{z}_{t-1} \rangle + \varepsilon_t, \tag{2}$$

where $\boldsymbol{\phi}(u) := (\phi_0(u), \dots, \phi_p(u))^\top \in \mathbb{R}^{p+1}$ is the *parameter vector* containing the the *unknown parameters* $(\phi_i(u_t))_{i\in[\![p]\!]} \in \mathbb{R}^p$ depending on the choice of an action u_t, $\mathbf{z}_{t-1} = (1, x_{t-1}, \dots, x_{t-p})^\top \in \mathcal{Z} :\in \{1\} \times \mathcal{X}^p$ ($\mathcal{X} \subseteq \mathbb{R}$ is the reward space) is the *vector of past estimated rewards* expressing the past history of estimations, and ε_t is a random $i.i.d.$ zero mean σ^2 sub-Gaussian noise, conditioned to the past.

We introduce a novel Long-Memory Autoregressive Bandits (LM-ARB) setting, where at each round $t \in \mathbb{N}$, the environment generates a noisy long-term (LT) reward y_t that evolves according to the ARFIMA$(p, d, 0)$ process of the following form:

$$y_t = (1 - B)^d x_t = (1 - B)^d (\phi_0(u_t) + \sum_{i=1}^p \phi_i(u_t) x_{t-i} + \varepsilon_t). \tag{3}$$

where $y_t \in \mathcal{Y}$, with the reward space $\mathcal{Y} \subseteq \mathbb{R}$, and d is an unknown fractional differencing rate, whose true value is stored within the environment for reward generation. After observing the reward y_t, the learner estimates a fractionally differencing rate (learning rate) $\hat{d}_t$ to convert y_t to an approximate short memory AR(p) reward in the following way:

$$\tilde{x}_t = (1 - B)^{-\hat{d}_t} y_t = \Sigma_{j=0}^\infty \psi_j(-\hat{d}_t) y_{t-j}, \tag{4}$$

where $\hat{d}_t$ is an estimated rate of learning, whose value is estimated by the agent through a defined loss-minimization mechanism, which we will later introduce in Section 3.

In this way, the agent approximates the *short memory* autoregressive reward $\tilde{x}_t$ from an infinite sequence of past observations of long memory rewards $(y_t, y_{t-1}, \dots)$, reducing the setting to AR(p)

to analyze the behavior of the system. In practice, the infinite-sum approximation must be reduced to a finite window size for the computational efficiency. Furthermore, in this setting, we presuppose that all the rewards played prior to the first round $t = 1$ are zero. Therefore, in our study, we truncate the infinite summation with the number of rounds played $t \in [\![T]\!]$, which gives us the following reward approximation we use across our learning horizon:

$$\hat{x}_t = \Sigma_{j=0}^{t} \psi_j(-\hat{d}_t) y_{t-j}. \tag{5}$$

2.2 Assumptions

We introduce the following assumptions, which we will utilize across the paper, and comment on their roles.

Assumption 2.1. (Non-negativity). $c \leq \phi_i(u)$ for every $u \in \mathcal{U}, i \in [\![p]\!]$ and $c \in (0, 1)$.

Assumption 2.2. (Stability). $\max_{u \in \mathcal{U}} \Sigma_{i=1}^{p} \phi_i(u) \leq \Gamma$ for $\Gamma < 1$.

Assumption 2.3. (Boundedness). $\max_{u \in \mathcal{U}} \phi_0(u) \leq m$ for $m \in (0, \infty)$

Assumption 2.4. (Long-memoryness). $0 < d < 0.5$

The Assumption 2.1 enforces the non-negativity of AR coefficients. Many real-world processes (i.e., pricing, stock markets, temperature anomalies etc.) are characterized by this assumption, where processes violating such will generate unrealistic and counterintuitive behaviors. The Assumption 2.2 ensures that the sum of $(\phi_i(u))_{i \in [\![p]\!]}$ is bounded to a value $\Gamma \in [0, 1)$ and Assumption 2.3 enforces the boundedness on $\phi_0(u)$ and the sequence of environment rewards. These latter assumptions guarantee the stability of the considered ARFIMA process, preventing it from diverging for any action sequence played. Finally, Assumption 2.4 is a necessary requirement that enables the ARFIMA process to model long memory temporal sequences (Box et al. [2015]).

2.3 Policy and Regret:

We model the learner's behavior by a deterministic policy $\underline{\pi} = (\underline{\pi}_t)_{t \in \mathbb{N}}$, defined for every round $t \in \mathbb{N}$ as $\underline{\pi} : \mathcal{H}_{t-1} \to \mathcal{U}$ that maps the history of observations $H_{t-1} := (\hat{x}_0, u_1, \hat{x}_1, \ldots, u_{t-1}, \hat{x}_{t-1}) \in \mathcal{H}_{t-1}$ to an action $u_t = \underline{\pi}(H_{t-1}) \in \mathcal{U}$, where $\mathcal{H}_{t-1} := \mathcal{X} \times (\mathcal{U} \times \mathcal{X})^{-1}$ is the set of length histories $t - 1$. The performance of a policy is evaluated in terms of the expected cumulative estimated reward over the horizon $T \in \mathbb{N}$:

$$J_T(\underline{\pi}) = \mathbb{E}[\Sigma_{t=1}^{T} \hat{x}_t], \tag{6}$$

The regret suffered by playing a policy $\underline{\pi}$, competing against the optimal policy $\underline{\pi}^*$ on a learning horizon $T \in \mathbb{N}$ is given by:

$$\hat{R}(\underline{\pi}, T) = J^* - \mathbb{E}[\Sigma_{t=1}^{T} \hat{x}_t] = \mathbb{E}[\Sigma_{t=1}^{T} \hat{r}_t], \tag{7}$$

where $\hat{r}_t := x_t^* - \hat{x}_t$ is the instantaneous policy regret and $(x_t^*)_{t \in [\![T]\!]}$ is the sequence of short memory AR rewards observed playing the optimal policy $\underline{\pi}^*$.

In the LM-ARB setting, because the converted analyzed reward $\hat{x}_t$ is of AR(p) process exhibiting short memory behavior, we employ the definition of the optimal policy of Bacchiocchi et al. [2024] expressed as follows:

Theorem 2.5. *(Optimal Policy) Under Assumptions 2.1 and 2.2, an optimal policy $\underline{\pi}^*$ maximizing the expected reward $J_T(\underline{\pi})$, for every round $t \in \mathbb{N}$ and history $H_{t-1} \in \mathcal{H}_{t-1}$ is given by:*

$$\underline{\pi}_t^*(H_{t-1}) \in \arg\max_{u \in \mathcal{U}} \langle \boldsymbol{\phi}(u), \hat{\mathbf{z}}_{t-1} \rangle. \tag{8}$$

Some important comments on the implementation of this theorem in our setting come in order. First, the optimal action depends on the past p reconstructed AR rewards. Thus, we preserve the Markovian property of the reward policy $\underline{\pi}^*$ by representing LM-ARB as a Markov Decision Process (MDP) with the state representation $\hat{\mathbf{z}}_{t-1} = (1, \hat{x}_{t-1}, \ldots, \hat{x}_{t-p})$ (Puterman [1994]), defined in the same way as in the ARB setting with true AR reward process $\{x_t\}$. On the other hand, defining the optimal policy in terms of a sequence of environment-generated rewards $(y_t)_{t \in \mathbb{N}}$ will create additional challenges in policy evaluation due to the complex structure of the reward-generating ARFIMA$(p, d, 0)$ process. Second, in every round $t \in \mathbb{N}$, the optimal action maximizes the instantaneous reward expected $\mathbb{E}[\hat{x}_t | H_{t-1}] = \langle \boldsymbol{\phi}(u), \hat{\mathbf{z}}_{t-1} \rangle$. This is because the Assumption 2.1 establishes the non-negativity of parameters, ensuring the meaningful evolution of the ARFIMA process, compatible with real-world settings. In this way, the optimal action maximizes both the expected immediate reward (i.e., myopic policy) and the expected cumulative reward.

2.4 On the Stochastic Properties of the AR Reward Process x_t

Estimating the fractional differencing rate $\hat{d}_t$ by the agent requires strong theoretical guarantees for the asymptotic convergence of the true parameter. The stochastic properties of a process for which the agent approximates the reward output must satisfy those necessary for asymptotic convergence. We encapsulate important properties of x_t in the following theorem:

Theorem 2.6. *(Geometric Ergodicity) Under Assumptions 2.2 and 2.3, the AR reward process x_t is strictly stationary and geometrically ergodic.*

This theorem presents a required condition about x_t when showing the asymptotic convergence of the differencing rate estimates:

Theorem 2.7. *(Asymptotic Convergence) Provided that Theorem 2.6 and Assumptions 2.1-2.4 hold, the asymptotic convergence of the estimated learning rate $\hat{d}_t$ to the true rate d is guaranteed by McAleer and Ling [2008] at the following rate:*

$$\hat{d}_t = d + \mathcal{O}\left(\sqrt{\frac{\log\log(t)}{t}}\right) \quad a.s., \tag{9}$$

where $a.s.$ represents the convergences in an almost sure sense.

A particular comment deserves the relevance of a non-zero lower bound for AR coefficients made in Assumption 2.1 to 2.7. The reward evolution process in the addressed setting is conditioned by persistent time-varying AR (TVAR) coefficients (Jiang [2023]), whose configuration depends on the arm played in the current round $t \in \mathbb{N}$. The nonzero lower bound condition guaranteed by Assumption 2.1 for every $(\phi_i(u))_{i\in[\![p]\!]}$ preserves the invertibility and regularity conditions required for consistent long-run inference. This property allows the estimated fractional differencing parameter $\hat{d}_t$ to converge asymptotically to its true value d of the environment over T. The complete proofs for 2.6 and 2.7 are outlined in Appendix B.

3 Algorithm

In this section, we present the algorithm `AutoRegressive Long-Memory Upper Confidence Bound (ARLM-UCB)`, an optimistic regret-minimizing algorithm for the LM-ARB setting whose pseudocode is presented in Algorithm 1. `ARLM-UCB` leverages the myopic optimal policy defined in Theorem 2.5 implements an incremental regularized least squares procedure across the given grid of fractional differencing values G_d to estimate the unknown parameters d and then $\boldsymbol{\phi}(u)$ for every action $u \in \mathcal{U}$ independently. The algorithm requires knowledge of the order p of the ARFIMA process.

The algorithm is based on the following procedure. `ARLM-UCB` starts by initializing for every action $u \in \mathcal{U}$ the Gram matrix $\hat{\mathbf{V}}_0(u) = \lambda \mathbf{I}_{p+1}$, where $\lambda > 0$ is the Ridge regularization parameter, the vectors $\hat{\mathbf{b}}_0(u) = \hat{\boldsymbol{\phi}}_0(u) = \mathbf{0}_{p+1}$ and the vector of estimated short memory observations $\hat{\mathbf{z}}_0 = (1, 0, \dots, 0)^\top$ for storing estimated short memory rewards $\hat{x}_t$ converted from the long memory rewards y_t generated by the environment. The algorithm also initializes the coefficients of decay as $\hat{\psi}_j = \prod_{i=1}^{j} \frac{i-1-d}{i}$ and $\psi_0 = 1$ utilized for the reward conversion process.

Then, for every round $t \in \mathbb{N}$, the algorithm computes the *Upper Confidence Bound* index (line 5) defined for every $u \in \mathcal{U}$ as follows:

$$\begin{aligned} u_t \in \arg\max_{u\in\mathcal{U}} \mathrm{UCB}_t(u) := \langle \hat{\boldsymbol{\phi}}_{t-1}(u), \hat{\mathbf{z}}_{t-1}\rangle + \mathcal{B}_{\hat{\mathbf{z}}_{t-1}-\mathbf{z}_{t-1}} \|\hat{\boldsymbol{\phi}}_{t-1}(u)\|_{\hat{\mathbf{V}}_{t-1}^{-1}(u)} \\ + \beta_{t-1}(u)\left(\|\hat{\mathbf{z}}_{t-1}\|_{\hat{\mathbf{V}}_{t-1}^{-1}(u)} + \mathcal{B}_{\hat{\mathbf{z}}_{t-1}-\mathbf{z}_{t-1}}\right). \end{aligned} \tag{10}$$

where $\mathcal{B}_{\hat{\mathbf{z}}_{t-1}-\mathbf{z}_{t-1}} = \sqrt{p}t^{\hat{d}_{t-1}-0.5}\log(t)\sqrt{\log(\log(t)+1)}$ arises from the asymptotic boundedness of $\hat{\mathbf{z}}_{t-1} - \mathbf{z}_{t-1}$. Similarly to `Lin-UCB` (Abbasi-Yadkori et al. [2011]), the index $\mathrm{UCB}_t(u)$ is designed to be optimistic, i.e., $\langle \hat{\boldsymbol{\phi}}_{t-1}(u), \hat{\mathbf{z}}_{t-1}\rangle \leq \mathrm{UCB}_t(u)$ in high probability for every $u \in \mathcal{U}$. Then, the agent plays an optimistic action $u_t \in \arg\max_{u\in\mathcal{U}} \mathrm{UCB}_t(u)$ and observes the ARFIMA$(p, d, 0)$ long memory (LM) reward y_t (line 6) of the form as in Equation 3.

Algorithm ARLM-UCB

1: **Input:** regularization parameter $\lambda > 0$, grid of fractional differencing

2: values G_d, exploration coefficient $(\beta_{t-1})_{t\in\mathbb{N}}$.

3: **Initialize:** $t \leftarrow 1, \hat{\mathbf{V}}_0(u) = \mathbf{V}_0(u,d) = \lambda \mathbf{I}_{p+1}, \hat{\mathbf{b}}_0(u) = \mathbf{b}_0(u,d) = \mathbf{0}_{p+1}, \hat{\boldsymbol{\phi}}_0(u) = \boldsymbol{\phi}_0(u,d) = \mathbf{0}_{p+1}, \hat{\mathbf{z}}_0 = (1,0,\dots,0)^\top$, fractional differencing parameter $\hat{d}_0$.

4: **for** $t \in [\![T]\!]$ **do**

5: Compute $u_t \in \arg\max_{u\in\mathcal{U}} \mathrm{UCB}_t(u) := \langle \hat{\boldsymbol{\phi}}_{t-1}(u), \hat{\mathbf{z}}_{t-1}\rangle + \mathcal{B}_{\hat{\mathbf{z}}_{t-1}-\mathbf{z}_{t-1}} \|\hat{\boldsymbol{\phi}}_{t-1}(u)\|_{\hat{\mathbf{V}}_{t-1}^{-1}(u)} + \beta_{t-1}(u)\left(\|\hat{\mathbf{z}}_{t-1}\|_{\hat{\mathbf{V}}_{t-1}^{-1}(u)} + \mathcal{B}_{\hat{\mathbf{z}}_{t-1}-\mathbf{z}_{t-1}}\right)$,
where $\mathcal{B}_{\hat{\mathbf{z}}_{t-1}-\mathbf{z}_{t-1}} = \sqrt{p}t^{\hat{d}_{t-1}-0.5}\log(t)\sqrt{\log(\log(t)+1)}$

6: Play action u_t and observe the LM reward y_t

7: **for** $\hat{d} \in G_d$ **do**

8: Compute $x_t(\hat{d}) = \Sigma_{j=0}^{t}\psi_j(-\hat{d})y_{t-j}$ and $\mathbf{z}_{t-1}(\hat{d}) = (1, x_{t-1}(\hat{d}), \dots, x_{t-p}(\hat{d}))^\top$

9: **for** $u \in \mathcal{U}$ **do**

10: $\hat{\mathbf{V}}_t(u,\hat{d}) = \hat{\mathbf{V}}_{t-1}(u,\hat{d}) + \mathbf{z}_{t-1}(\hat{d})\mathbf{z}_{t-1}^\top(\hat{d})\mathbb{1}_{\{u=u_t\}}$

11: $\hat{\mathbf{b}}_t(u,\hat{d}) = \hat{\mathbf{b}}_{t-1}(u,\hat{d}) + \mathbf{z}_{t-1}(\hat{d})^\top x_t(\hat{d})\mathbb{1}_{\{u=u_t\}}$

12: $\boldsymbol{\phi}_t(u,\hat{d}) = \hat{\mathbf{V}}_t^{-1}(u,\hat{d})\hat{\mathbf{b}}_t(u,\hat{d})$

13: Compute loss $L_t(\hat{d}) = \sum_{u\in\mathcal{U}}\sum_{i=1}^{t}\{x_{t-i+1}(\hat{d}) - \langle \boldsymbol{\phi}_{t-i+1}(u,\hat{d}), \mathbf{z}_{t-i}(\hat{d})\rangle\}^2\mathbb{1}_{\{u=u_t\}}$

14: Choose $\hat{d}_t = \arg\min_{\hat{d}\in G_d} L_t(\hat{d})$ and $\hat{x}_t = x_t(\hat{d}_t)$

15: **for** $u \in \mathcal{U}$ **do**

16: $\hat{\mathbf{V}}_t(u) = \hat{\mathbf{V}}_{t-1}(u) + \hat{\mathbf{z}}_{t-1}\hat{\mathbf{z}}_{t-1}^\top\mathbb{1}_{\{u=u_t\}}$

17: $\hat{\mathbf{b}}_t(u) = \hat{\mathbf{b}}_{t-1}(u) + +\hat{\mathbf{z}}_{t-1}\hat{x}_t\mathbb{1}_{\{u=u_t\}}$

18: $\hat{\boldsymbol{\phi}}_t(u) = \hat{\mathbf{V}}_t^{-1}\hat{\mathbf{b}}_t(u)$

19: Update $\hat{\mathbf{z}}_t = (1, \hat{x}_t, \dots, \hat{x}_{t-p+1})^\top$

20: $t \leftarrow t+1$

Once the LM reward y_t is observed, the agent calculates a short memory converted reward $\hat{x}_t$ and later the parameter vector $\hat{\boldsymbol{\phi}}_t(u)$ on a grid of fractional differencing values G_d using the following minimalized squared distance loss procedure. For each $d \in G_d$, the agent estimates the short memory reward using a truncated infinite series ARFIMA sum $\hat{x}_t = (1-B)^{-\hat{d}}y_t = \Sigma_{j=0}^{t}\hat{\psi}_j(-\hat{d})y_{t-j}$ (line 8). The agent then continues with updating the Gram matrix estimate $\mathbf{V}_t(u,\hat{d})$, the vector $\mathbf{b}_t(u,\hat{d})$, and the estimate $\boldsymbol{\phi}_t(u,\hat{d})$ (lines 10-12). Finally, using all previous short memory samples up to round $t \in \mathbb{N}$ $(\hat{x}_i)_{i\in\mathbb{N}}$, and previous short memory observation vectors $(\hat{\mathbf{z}}_i)_{i\in\mathbb{N}}$ and estimates $(\hat{\boldsymbol{\phi}}_i(u))_{i\in\mathbb{N}}$, the agent estimates the loss function (line 13) for the selected $\hat{d}$ and all $u \in \mathcal{U}$ defined as follows:

$$L_t(\hat{d}) = \sum_{u\in\mathcal{U}}\sum_{i=1}^{t}\{x_{t-i+1}(\hat{d}) - \langle \boldsymbol{\phi}_{t-i+1}(u,\hat{d}), \mathbf{z}_{t-i}(\hat{d})\rangle\}^2\mathbb{1}_{\{u=u_t\}} \tag{11}$$

The estimate of the fractional differencing parameter $\hat{d}_t$ and the estimated short memory reward sample $\hat{x}_t$ that minimize the loss function $L_t(\hat{d})$ are selected at the end of a search over a fractional differencing values grid G_d in line 14. Using $\hat{x}_t$, the agents proceeds to compute $\hat{\mathbf{V}}_t(u)$, the vector $\hat{\mathbf{b}}_t(u)$, and the estimate $\hat{\boldsymbol{\phi}}_t(u)$ for the current round (lines 16-18). The observation vector is then updated with a new reward sample as $\hat{\mathbf{z}}_t = (1, \hat{x}_t, \dots, \hat{x}_{t-p+1})^\top$ (line 19), so that it can be used in the next round along with the estimate $\hat{\boldsymbol{\phi}}_t(u)$.

4 Regret Analysis

In this section, we conduct analysis of the regret of `ARLM-UCB`. We first present the formal self-normalized concentration inequality and compare it with existing results in the literature (Section 4.1). Then, we provide the the bound on the expected cumulative (policy) regret (Section 4.2). The complete proofs of the theorems stated in the section are presented in Appendices B.3 and B.4.

4.1 Concentration Inequality for the Parameter Vectors

We first present the concentration result for the estimates $\hat{\boldsymbol{\phi}}_t(u)$ of the true parameters $\boldsymbol{\phi}(u)$, for every action $u \in \mathcal{U}$. At each round $t \in \mathbb{N}$, for the chosen action $u_t \in \mathcal{U}$ and for each fractional differencing coefficient is $\hat{d} \in G_d$, where the selected fractional differencing coefficient is $\hat{d} \in G_d$, we solve the Ridge regression problem as:

$$\hat{\boldsymbol{\phi}}_t(u) := \arg\min_{\tilde{\boldsymbol{\phi}} \in \mathbb{R}^{p+1}} \sum_{l \in \mathcal{O}_t(u_t)} (x_l - \langle \tilde{\boldsymbol{\phi}}, \hat{\mathbf{z}}_{l-1} \rangle)^2 + \lambda \|\tilde{\boldsymbol{\phi}}\|_2^2 = \hat{\mathbf{V}}_t(u_t)^{-1} \hat{\mathbf{b}}_t(u_t), \tag{12}$$

where $\mathcal{O}_t(u)$ is the set of rounds, where the action u was played, i.e., $\mathcal{O}_t(u) :\in \{\tau \in \mathbb{N} : u_\tau = u\}$. The following theorem formulates the concentration of $\hat{\boldsymbol{\phi}}_t(u)$ around $\boldsymbol{\phi}(u)$ over the rounds:

Theorem 4.1. *(Self-normalized concentration) Let $u \in \mathcal{U}$ be an action and $(\hat{\boldsymbol{\phi}}_t(u))_{t \in \mathcal{O}_\infty(u)}$ be the sequence of solutions of the Ridge regression problems of Algorithm 1. Then, under Assumption 2.1 and 2.2, for every $\lambda \geq 0$ and $\delta \in (0,1)$, with probability at least $1-\delta$, and for all rounds $t \in \mathbb{N}$, we have the following.*

$$\|\hat{\boldsymbol{\phi}}_t(u) - \boldsymbol{\phi}(u)\|_{\hat{\mathbf{V}}_t(u)} \leq \sqrt{\lambda}\|\boldsymbol{\phi}(u)\|_2 + c_1(t)\|\boldsymbol{\phi}(u)\|_2 + c_2(t) + \sigma\sqrt{2\log\left(\frac{1}{\delta}\right) + \log\left(\frac{\det(\hat{\mathbf{V}}_t(u))}{\lambda^{p+1}}\right)},$$

where $c_1(t) := \mathcal{O}\left(\sqrt{p}t^d \log(t)\sqrt{\log\log(t)}\right)$ and $c_2(t) := \mathcal{O}\left(\sqrt{p+1}t^d \log(t)\sqrt{\log\log(t)}\right)$

Theorem 4.1 resembles the self-normalized concentration inequality of Bacchiocchi et al. [2024], whose idea originates from Theorem 1 of Abbasi-Yadkori et al. [2011]. Likewise, in case of `AR-UCB`, the exploration coefficients $\beta_t(u)$ are different for every action $u \in \mathcal{U}$. However, the important novelty that distinguishes the algorithm `ARLM-UCB` from the former is that the agent estimates $\hat{\boldsymbol{\phi}}_t(u)$ using estimated AR rewards $(\hat{x}_t)_{t\in\mathbb{N}}$ calculated from past t environment-generated ARFIMA rewards $(y_t)_{t\in\mathbb{N}}$. This notion results in the derivation of the term $c_1(t)\|\boldsymbol{\phi}(u)\|_2 + c_2(t)$ that emphasizes the convergence of the estimated reward $\hat{x}_t$ with the coefficient $\hat{d}_t$ at every round $t \in \mathbb{N}$ to the true AR reward $x_t = (1-B)^{-d} y_t$ obtained through direct undifferencing the environment reward y_t.

Using the results in Theorem 4.1, we select the coefficient β_t based on the knowledge of the upper bounds specified in Assumption 2.2 and Assumption 2.3 for every $t \in \mathbb{N}$:

$$\beta_t(u) := \sqrt{\lambda(m^2+1)} + c_1(t)\sqrt{m^2+1} + c_2(t) + \sigma\sqrt{2\log\left(\frac{1}{\delta}\right) + \log\left(\frac{\det(\hat{\mathbf{V}}_t(u))}{\lambda^{p+1}}\right)}, \tag{13}$$

where $c_1(t) = \sqrt{p}t^d \log(t)\sqrt{\log(\log(t)+1)}$ and $c_2(t) = \sqrt{p+1}t^d \log(t)\sqrt{\log(\log(t)+1)}$. This formula is constructed with three terms. The first term is a *bias* term, the second one is a *estimation error* term, and the third one is a *concentration* term. The *bias* term is derived by utilizing Assumptions 2.2 and 2.3, which guarantee that $\|\boldsymbol{\phi}(u)\|_2 \leq \sqrt{m^2+\Gamma^2} \leq \sqrt{m^2+1}$. The *estimation error* term arises from the fact that our learner is required learn the true fractional differencing rate d throughout the learning interval T. Two components of this term, $c_1(t)$ and $c_2(t)$, arise from their respective bounds derived in Equation 4.1. In this way, the exploration coefficient $\beta_t(u)$ ensures that, with probability $1-\delta$, the following inequality holds universally for every action $u \in \mathcal{U}$:

$$\|\hat{\boldsymbol{\phi}}_t(u) - \boldsymbol{\phi}(u)\|_{\hat{\mathbf{V}}_t(u)} \leq \beta_t(u) \tag{14}$$

The *bias* and *concentration* terms mimic those for $\beta_t(u)$ of Bacchiocchi et al. [2024], highlighting the independence of the simultaneous knowledge of Γ and c (Assumptions 2.1 and 2.2) introduced in our setting. This feature for `ARLM-UCB` is plausible for learning, as the true values of these parameters are unknown in practice.

4.2 Regret Bound

In this section, we derive a bound on the expected policy regret bound for `ARLM-UCB`:

Theorem 4.2. *Let $\delta = (2T)^{-1}$. Under Assumptions 2.1-2.4,* ARLM-UCB *suffers a cumulative expected (policy) regret bounded by (highlighting the dependence on $\Gamma, p, m, \sigma, n,$ and T):*

$$\mathbb{E}[R(\texttt{ARLM-UCB}, T)] \leq \left(\frac{(p+1)^2 (m+\sigma)\sqrt{n} T^{2d+0.5} \log^2(T) \log\log(T)}{(1-\Gamma)^2} \right).$$

The regret bound in Theorem 4.2 expands the one for AR-UCB. It's worth noting that, unlike in the case of AR-UCB, with $p = 0$ and $\Gamma = 0$, our problem becomes ARFIMA$(0, d, 0)$, where we obtain the regret rate $\mathcal{O}\left((m+\sigma)\sqrt{n}T^{2d+0.5}\log^2(T)\log\log(T)\right)$. This rate does not reduce ARLM-UCB to standard MAB problem, unlike in the case of Bacchiocchi et al. [2024], where $p = 0$ gives the regret $\mathcal{O}\left((m+\sigma)\sqrt{nT}\right)$ tight to standard MABs. This notion suggests that introducing the fractional differencing in the reward-evolution process generally makes the learning problem more complex and difficult to handle by regular multi-armed bandits. All the theoretical derivations allowing us to achieve this upper bound are proved in Appendix B.

5 Experiments

This section presents numerical experiments on the ARLM-UCB, highlighting how this algorithm can outperform various competing bandit baselines in synthetically-generated domains. Appendix A features the bandit comparison on real-world data. The full code of our algorithm implementation is available at https://github.com/uladcharn/LM-ARM. All the algorithms were implemented in Python 3.12, and run over an Apple M1 with 8 GB RAM, in no more than a couple of hours.

We compare ARLM-UCB with the following baselines: (a) UCB1 (Auer et al. [1995]), an algorithm developed for stochastic MABs, (b) EXP3 (Auer et al. [2002]), an algorithm designed for adversarial MABs, (c) its finite-memory adaptive adversaries B-EXP3 (Dekel et al. [2012]), (d) AR2 (Chen et al. [2023]) designed to manage non-stationary AR(1) processes, and (e) AR-UCB (Bacchiocchi et al. [2024]) designed to operate in stationary AR(p) environments.

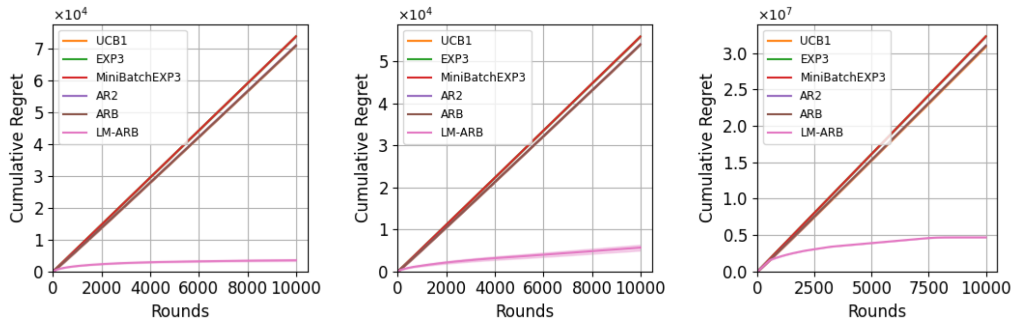
We evaluate the selected bandits in three synthetic scenarios with different properties that govern the reward evolution processes. For all synthetic experiments, we set the number of rounds to be $T = 10000$, the true fractional differencing rate $d = 0.35$, and the grid of fractional differencing rates $G_d = \{0.01, 0.02 \ldots, 0.48, 0.49\}$, s.t. $d \in G_d$. The three settings have their ARFIMA$(p, d, 0)$ process order $p \in \{0, 1, 2\}$, number of actions $n \in \{5, 7\}$, number of exploration rounds $K \in \{5, 75, 550\}$, and scale $m \in \{1.6, 7.4, 920\}$. The values of AR coefficients $\boldsymbol{\phi}(u)$ are drawn randomly from a uniform distribution for each action $u \in \mathcal{U}$ and for each setting. The standard deviations of the noise in three environments are $\sigma \in \{1, 1.6, 10\}$. The selected hyperparameters of AR-UCB and ARLM-UCB are $\lambda = 1$ and $\bar{m} \in \{1.6, 7.5, 1000\}$. Table 1 summarizes the parameter settings for each experimental scenario.

Setting	p	n	m	$\bar{m}$	σ	K
A	0	5	7.4	7.5	1.25	5
B	1	5	1.6	1.6	0.9	75
C	2	7	920	1000	10	550

Table 1: Setting description

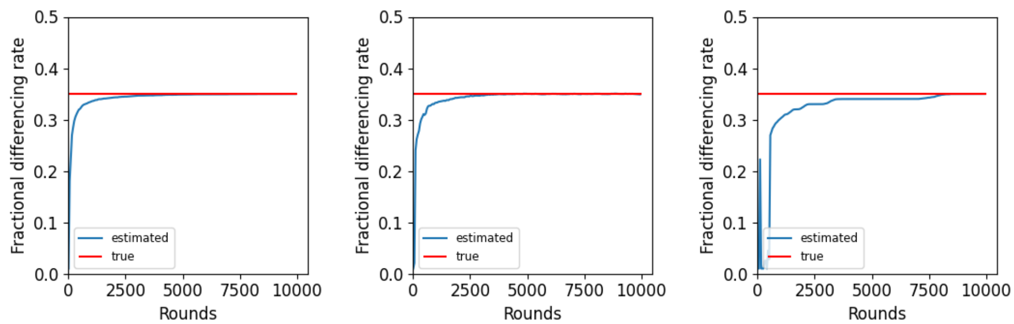
5.1 Results

Figure 1 shows the average cumulative regrets for three settings. We observe that ARLM-UCB consistently outperforms all competing bandit algorithms, always demonstrating sublinear behavior. On the other hand, all other bandits exhibit linear regret, since they are unable to process the long memory rewards and converge to sublinear regret over the learning horizon. That is because only ARLM-UCB has a specific mechanism for modeling long memory dynamics, allowing for the precise estimation and analysis of the underlying reward-governing long memory persistence.



(a) Setting A. (b) Setting B. (c) Setting C.

Figure 1: Cumulative Regret of `ARLM-UCB` and multiple baselines (100 runs, mean $\pm$std).



(a) Setting A. (b) Setting B. (c) Setting C.

Figure 2: The convergence dynamics of selected fractional differencing parameters $\hat{d}$.

5.2 On the Convergence of the Fractional Differencing Rate Estimate

Figures 2 display the average estimates of the sequence of fractional differencing rates $(\hat{d}_t)_{t\in\mathbb{N}}$ by `ARLM-UCB` on the optimization horizon T (blue) and a straight vertical line (red) representing the true fractional differencing rate $d = 0.35$ as a benchmark. We immediately observe that the learner-selected rates always start at near-zero values in earlier round. This is due to the fact that our agent did not sufficiently explore the environment, which loosens his sense of present long-range dependence. However, the agent quickly regains the perception of long memory over more rounds and eventually converges in his estimates of $(\hat{d}_t)_{t\in\mathbb{N}}$ to the true rate d.

6 Conclusion

In this work, we addressed the online sequential decision-making problem, where the ARFIMA long memory temporal structure between rewards is present. We first formulate a LM-ARB setting by introducing a range of necessary assumptions and a selection of the optimal policy applicable in the context of our problem. We then propose a new online algorithm `ARLM-UCB` that learns the online parameters for each action and searches the grid for the true fractional differencing rate corresponding to the long memory reward y_t. `ARLM-UCB` employs the idea that this bandit setting can be reduced to ARBs and solved using a linear contextual bandit with thoughtful selection of a fractional differencing rate. We also present a novel regret bound for this algorithm, accounting for the need to estimate the fractional differencing parameter in the addressed setting. Finally, we provided a variety of numerical experiments to assess the performance of `ARLM-UCB` based on cumulative regret and validate our solution. Future research directions should focus on extending the presented approach by designing a similar setting incorporating moving average part in the reward-generating mechanism. It is also promising in the long-term to derive a policy evaluation directly on a sequence of ARFIMA rewards.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24 (NeurIPS 2011)*, pages 2312–2320, 2011.

Safwan Mahmood Al-Selwi, Mohd Fadzil Hassan, Said Jadid Abdulkadir, and Amgad Muneer. Lstm inefficiency in long-term dependencies regression problems. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 30(3):16–31, 2023.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 322–331. IEEE, 1995.

Peter Auer, Nicolò Cesa-Bianchi, and Gábor Lugosi. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 31(1):145–157, 2002.

Francesco Bacchiocchi, Gianmarco Genalti, Davide Maran, Marco Mussi, Marcello Restelli, Nicola Gatti, and Alberto Maria Metelli. Autoregressive bandits. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*. PMLR, 2024.

S. H. A. Bakar and C. M. Hafner. Forecasting realised volatility using arfima and har models. *Quantitative Finance*, 19(12):1925–1934, 2019.

Jan Beran. Maximum likelihood estimation of the differencing parameter for invertible short and long memory ARIMA models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(4):659–672, 1995.

George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 5th edition, 2015. ISBN 978-1-118-67502-1.

Uladzimir Charniauski and Yao Zheng. Autoregressive bandits in near-unstable or unstable environment. *American Journal of Undergraduate Research*, 21(2):15–26, 2024. doi: 10.33697/ajur.2024.116.

Qinyi Chen, Negin Golrezaei, and Djallel Bouneffouf. Non-stationary bandits with auto-regressive temporal dependency. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2023.

Rainer Dahlhaus. Efficient parameter estimation for self-similar processes. *The Annals of Statistics*, 17(4): 1749–1766, 1989.

Ofer Dekel, Ambuj Tewari, and Raman Arora. Online bandit learning against an adaptive adversary: From regret to policy regret. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 27–34. PMLR, 2012.

Mallikarjuna Doodipala. Time series analysis for long memory process of air traffic using arfima. *International Journal of Scientific & Technology Research*, 9(3):6268–6272, 2020.

Robert Fox and Murad S. Taqqu. Large-sample properties of parameter estimates for strongly dependent stationary gaussian time series. *The Annals of Statistics*, 14(2):517–532, 1986.

C. W. J. Granger and R. Joyeux. An introduction to long-memory time series models and fractional differencing. *Journal of Time Series Analysis*, 1(1):15–29, 1980.

Gaurav Gupta, Chenzhong Yin, Jyotirmoy V. Deshmukh, and Paul Bogdan. Non-markovian reinforcement learning using fractional dynamics. *arXiv preprint arXiv:2107.13790*, 2021.

J. R. M. Hosking. Fractional differencing. *Biometrika*, 68(1):165–176, 1981.

Xing-Qi Jiang. Time varying coefficient AR and VAR models. In *The Practice of Time Series Analysis*. Springer, 2023.

Ferry Kondo Lembang, Lexy Janzen Sinay, and Asrul Irfanullah. Arfima modelling for tectonic earthquakes in the maluku region. *Indonesian Journal of Statistics and Its Applications*, 5(1):39–49, 2021.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020. ISBN 9781108486828.

Ming Liu. Modeling long memory in stock market volatility. *Journal of Econometrics*, 99(1):139–171, 2000.

Michael McAleer and Shiqing Ling. A general asymptotic theory for time-series models. Technical report, Hong Kong University of Science and Technology, 2008.

A. I. McLeod and K. Hipel. Fractional time series modelling. *Technometrics*, 28(2):101–111, 1986.

D.Feigin Paul and L.Tweedie Richard. Random coefficient autoregressive processes: a markov chain analysis of stationarity and finiteness of moments. *Journal of Time Series Analysis*, 6(1):1–14, 1985. doi: 10.1111/j.1467-9892.1985.tb00394.x.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. John Wiley Sons, 1994. ISBN 978-0-471-61977-2.

Yuzhen Qin, Yingcong Li, Fabio Pasqualetti, Maryam Fazel, and Samet Oymak. Stochastic contextual bandits with long horizon rewards. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*, pages 8611–8619. Association for the Advancement of Artificial Intelligence, 2023.

L.Tweedie Richard. Criteria for rates of convergence of markov chains, with application to queueing and storage theory. In C.Kingman J., F. and E.H.Reuter G., editors, *Probability, Statistics and Analysis*, London Mathematical Society Lecture Note Series, page 260–276. Cambridge University Press, 1983. doi: 10.1017/CBO9780511662430.016.

P. M. Robinson. Gaussian semiparametric estimation of long range dependence. *The Annals of Statistics*, 23(5): 1630–1661, 1995.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2020.

Wei Tang, Chien-Ju Ho, and Yang Liu. Bandit learning with delayed impact of actions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Yifu Tang, Yingfei Wang, and Zeyu Zheng. Stochastic multi-armed bandits with strongly reward-dependent delays. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 3043–3051. PMLR, 02–04 May 2024.

Anna L. Trella, Walter Dempsey, Finale Doshi-Velez, and Susan A. Murphy. Non-stationary latent auto-regressive bandits. *arXiv preprint arXiv:2402.03110*, 2024.

JQ (Justin) Veenstra and A. Ian McLeod. *arfima: Fractional ARIMA (and Other Long Memory) Time Series Modeling*, 2022. R package version 1.8-1.

Naiming Yuan, Zuntao Fu, and Shida Liu. Extracting climate memory using fractional integrated statistical model: A new perspective on climate prediction. *Scientific Reports*, 4:6577, 2014.

Jingyu Zhao, Feiqing Huang, Jia Lv, Yanjie Duan, Zhen Qin, Guodong Li, and Guangjian Tian. Do RNN and LSTM have long memory? In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11365–11375. PMLR, 13–18 Jul 2020.

Nicolas Zucchet, Robert Meier, Simon Schug, Asier Mujika, and João Sacramento. Online learning of long-range dependencies. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, pages 10477–10493. Curran Associates, Inc., 2023.

A Real-world Data Experiments

In this section, we compare `ARLM-UCB` with the same baselines as in Section 5 on the stock prices of three technological companies: Apple, Meta, and Netflix. For each company listed, the data is obtained from Kaggle and contains daily closing prices and daily trading volumes from 2014 to 2023, 2004 to 2024, and 2010 to 2024, respectively. We convert the closing prices to absolute log-returns and parallelly shift this series by 1 to ensure stationarity and prepare our series of log-return for further processing explained. We then discretize the prices into $n = 4$ price bands (i.e., our actions) based on whether the magnitude of a return is positive/negative before being converted to an absolute value and whether the volatility exceeds the value in the third quartile of the volatility distribution of each stock. Table 2 summarizes the action selection for our real-world experiment.

Log-Return	Trading Volume	Arm Label
Positive	Exceeds 3rd quartile	1
Negative	Exceeds 3rd quartile	2
Positive	Below 3rd quartile	3
Negative	Below 3rd quartile	4

Table 2: A summary of an action selection method

A.1 Setting Configuration

We construct the simulation environment for each stock using the following methodology. First, we find the true fractionally differencing value d corresponding to each data set using the R package `arfima` (Veenstra and McLeod [2022]). We fit the ARFIMA$(p, d, 0)$ model on the absolute log-return series to estimate the global fractional differencing rate d. Then we un-difference the original series of log-returns with d and convert. The earlier parallel shift of our log-return series ensures the positivity of the reconstructed price series, which is a necessary condition for our bandit analysis. Finally, we estimate the hidden autoregressive parameters for each action using standard regression methods, taking into account the adjacent values of the past p at each time point. Each selected value of p represents the number of significant AR lags in reward-governing ARFIMA model fit to each data. To determine the noise standard deviation σ for each dataset, we compute the square root of the weighted mean of all variances produced from estimating each arm.

Table 3 summarizes the parameter settings for each dataset considered. We globally set the number of rounds $T = 10000$, the grid of fractional differencing rates $G_d = \{0.01, 0.02, \dots, 0.48, 0.49\}$, s.t. $d \in G_d$. Each setting has estimated AR parameters $m \in \{0.012, 0.135, 0.007\}$ and $\Gamma \in \{0.75, 0.959, 0.995\}$ with the number of exploration rounds $K \in \{150, 250\}$. The estimated Gaussian noise standard deviations are $\sigma \in \{0.013, 0.014, 0.027\}$.

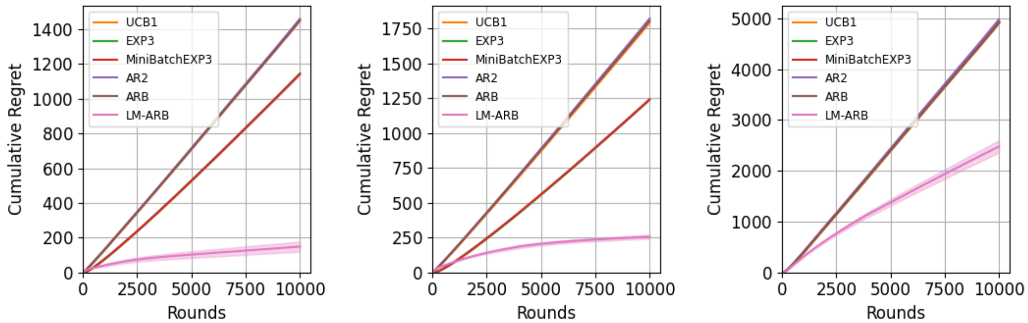
We select $\lambda \in \{0.001, 0.01, 1\}$, where $\lambda \in \{0.01, 0.001\}$ is set for settings with Γ close to 1 numerically. Charniauski and Zheng [2024] showed that setting close-to-zero values for λ could help AR-UCB achieve smaller regrets in such near-unstable (e.g., with $\Gamma \approx 1$) settings. This notion should hold similarly in LM-ARB case, since we aim to estimate the underlying AR process of undifferenced rewards $\hat{x}_t$. We also choose $\bar{m} \in \{0.01, 0.0.015, 0.15\}$ that are set to around 20% greater value than m in each respective environment.

Stock Data	p	m	$\bar{m}$	λ	Γ	d	σ	K
Apple	1	0.012	0.015	0.001	0.959	0.25	0.013	150
Netflix	2	0.135	0.15	1	0.75	0.21	0.027	250
Google	3	0.007	0.01	0.01	0.995	0.31	0.014	150

Table 3: Setting description

A.2 Results

Figure 3 illustrates the average cumulative regrets for three datasets. The `ARLM-UCB` demonstrates the sublinear convergence in all three cases, achieving the smallest cumulative regret. On the other



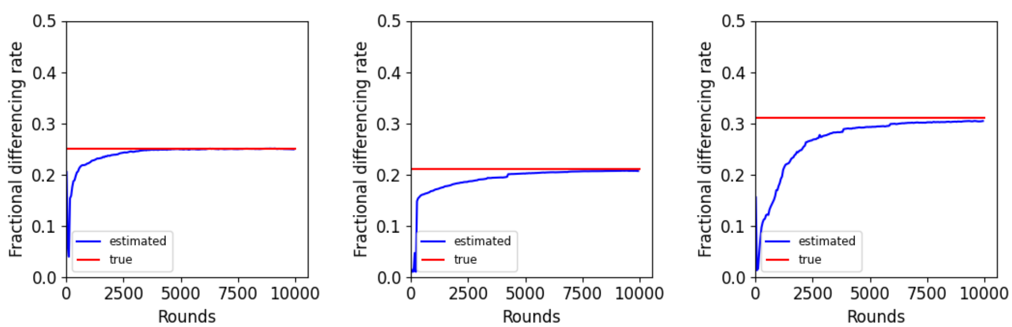
(a) Apple (b) Netflix (c) Google

Figure 3: Cumulative regret of `ARLM-UCB` and baselines on real data (100 runs, mean $\pm$std).

hand, neither of the other bandits achieve sublinear regret in all considered cases. We also observe that both `EXP3` and `B-EXP3` suffer the exponential regret in all three scenarios, which is explicitly seen in 3a and 3b. This might be due to the structure of arms and parameter values presented in each of three environments. These observations make `ARLM-UCB` the algorithm with the best performance over the competitors.

A.3 On the Convergence of the Fractional Differencing Rate Estimate

Figure 4 demonstrates the average estimates of the sequence of fractional differencing rates $(\hat{d}_t)_{t\in\mathbb{N}}$ on the optimization horizon T (blue) and a straight vertical line (red) representing the true fractional differencing rate for each dataset. These results replicate the ones achieved on synthetic data cases displayed in Section 5.2. We see that the estimated rates rapidly converge to the true rate, starting at low values. These plots also demonstrate that the estimate $\hat{d}_t$ is converging regardless of the true rate d established in the environment.



(a) Apple (b) Netflix (c) Google

Figure 4: The convergence dynamics of selected fractional differencing parameters $\hat{d}$.

B Omitted Proofs

B.1 Proof of Theorem 2.6

Proof. We prove the important property of geometric ergodicity of the reward process x_t. For the process x_t, we consider the process expressed in Equation 2.

We define the companion vector state $X_t := (x_{t-1}, \dots, x_{t-p})^\top \in \mathbb{R}^p$ for all $t \in \mathbb{N}$. We rewrite the reward evolution from Equation 2 as follows:

$$X_t = A(u_t)X_{t-1} + b(u_t) + \xi_t,$$

where we define:

$$A(u_t) := \begin{pmatrix} \phi_1(u_t) & \phi_2(u_t) & \cdots & \phi_p(u_t) \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \dots & 1 \end{pmatrix} \in \mathbb{R}^{p\times p}, b(u_t) := \begin{pmatrix} \phi_0(u_t) \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \in \mathbb{R}^p, \xi_t := \begin{pmatrix} \varepsilon_t \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \in \mathbb{R}^p,$$

and the transition probability is:

$$P(x, \mathcal{A}) = \int_{\mathcal{A}} \varphi(z - m_t(x))dz,$$

for $x \in \mathbb{R}^p$, the mean process $m_t(x) = A(u_t)x + b(u_t)$, and $\mathcal{A} \in \mathbb{B}^p$, the class of Borel sets of $\mathbb{R}^p$.

Because ξ_t is defined in terms of Gaussian noise, $P(x, \mathcal{A}) > 0$ and $\{X_t\}$ is ν_p-irreducible for the Lebesgue measure ν_p on $(\mathbb{R}^p, \mathcal{B}^p)$.

We prove by showing that Tweedie's drift criterion (Richard [1983]) holds, i.e. there is a small set $G \subset \mathbb{R}^p$ with $\nu_p(G) > 0$ and a non-negative continuous function $V(x)$, s.t.

$$\mathbb{E}\left[V(X_t)|X_{t-1} = x\right] \leq (1-\delta)V(x), x \notin G \tag{15}$$

and

$$\mathbb{E}\left[V(X_t)|X_{t-1} = x\right] \leq M, x \in G \tag{16}$$

for $0 < \delta < 1$ and $0 < M < \infty$.

By Assumptions 2.2 and 2.3, we observe that $\rho(A(u_t)) = \Gamma < 1$ and $\sup_t \mathbb{E}[\|b(u_t)\|] \leq m$.

We choose Lyapunov criterion as $V(x) = 1 + \|x\|^2$. We are now able to observe the following:

$$\begin{aligned} \mathbb{E}\left[V(X_t)|X_{t-1} = x\right] &= \mathbb{E}\left[1 + \|X_t\|^2 | X_{t-1} = x\right] \\ &\leq 1 + \Gamma^2\|x\|^2 + m^2 + \sigma^2 \leq 1 + \Gamma^2 V(x) + m^2 + \sigma^2. \end{aligned}$$

Denote $\delta = 1 - \Gamma^2 - \frac{1-\Gamma^2+m^2+\sigma^2}{V(x)}$ and $G := \{x : \|x\| \leq L\}$, s.t. $V(x) \geq 1 + \frac{m^2+\sigma^2}{1-\Gamma^2}$ for every $\|x\| > L$. We obtain that conditions stated in Equations 15 and 16 hold.

Moreover, we observe that $\mathbb{E}[f(X_t)|X_{t-1} = x]$ is continuoues w.r.t. x for every bounded function $f(\cdot)$. Thus, $\{X_t\}$ is a Feller chain.

By Paul and Richard [1985], G is a small set. By referring to Theorem 4(ii) in Richard [1983] and Theorem 1 in Paul and Richard [1985], X_t is geometrically ergodic with a unique strictly stationary solution.

□

B.2 Proof of Theorem 2.7

Proof. Let $u \in \mathcal{U}$. For each arm $u \in \mathcal{U}$, the reward y_t at every round $t \in \mathbb{N}$ evolves according to ARFIMA$(p, d, 0)$ process as stated in Equation 3. Observe that the process for each arm evolves independently, regardless of when or whether the arm is pulled.

Assumptions 2.1-2.4, Theorem 2.6 and the noise $\varepsilon_t \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$ for $t \in \mathbb{N}$ guarantee that x_t is strictly stationary and ergodic, with $\mathbb{E}[x_t^2] < \infty$.

For ARFIMA$(p, d, 0)$, we also have the following:

$$x_t = \sum_{i=0}^{\infty} \tilde{\psi}_{0i}(d)\varepsilon_{t-i}(d) \text{ and } \varepsilon_t(d) = (1-B)^d x_t = \sum_{i=0}^{\infty} \tilde{\psi}_i(d)x_{t-i} - \sum_{i=0}^{\infty}\left[\sum_{n=0}^{\infty} \tilde{\psi}_n(d)\phi_{i-n}(u_t)\right] x_{t-k},$$

from which we observe that

$$\sum_{i=0}^{\infty} \tilde{\psi}_i(d)x_{t-i} - \sum_{i=0}^{\infty}\left[\sum_{n=0}^{\infty} \tilde{\psi}_n(d)\phi_{i-n}(u_t)\right] x_{t-k} \leq \sum_{i=0}^{\infty} \tilde{\psi}_i(d)x_{t-i} - \sum_{i=0}^{\infty}\left[\sum_{n=0}^{\infty} c \cdot \tilde{\psi}_n(d)\right] \leq \sum_{i=0}^{\infty} \tilde{\psi}_i(d)x_{t-i}.$$

For the last term of the above inequality, using the conditions of stationarity and ergodicity of the reward x_t, McAleer and Ling [2008] verified that the estimated rates $(\hat{d}_t)_{t\in\mathbb{N}}$ converge to the true rate d as stated in Equation 9 and that concludes the proof.

$\square$

B.3 Proof of Theorem 4.1

Before we develop the self-normalized concentration, we introduce the context vector bound (Lemma B.1) by Bacchiocchi et al. [2024].

Lemma B.1. *Let $(\mathbf{z}_t^*)_{t\in[\![T]\!]}$ be the sequence of observation vectors observed by executing the learner's policy. If $\mathbf{z}_0 = (1,0,\dots,0)^\top$, then, for every $\delta \in (0,1)$, with probability at least $1-\delta$, simultaneously for every $t \in [\![T]\!]$, it holds that:*

$$\|\mathbf{z}_{t-1}\|_2 \leq \sqrt{1 + p\left(\frac{m+\eta}{1-\Gamma}\right)^2},$$

where $\eta = \sqrt{2\sigma^2 \log(T/\delta)}$.

Proof of Theorem 4.1. We first consider an action at a time; then we obtain the final result with a union bound over $\mathcal{U} := [\![n]\!]$.

Let $u \in \mathcal{U}$. Observe that the estimates of an action u change only when u is pulled. Let $l \in \mathbb{N}$ be an index and let $t_l(u) \in \mathbb{N}$ be the round in which the action u is pulled for the l-th time, i.e., $\{t_l(u) : l \in \mathbb{N}\} = \mathcal{O}_\infty(u)$. Thus, we have the following: [mention somewhere that a.s. notation is omitted]

$$\boldsymbol{\phi}_{t_l}(u) = \hat{\mathbf{V}}^{-1}_{t_l(u)}(u)\hat{\mathbf{b}}_{t_l(u)}(u) = \left(\lambda \mathbf{I}_{p+1} + \sum_{j=1}^{l} \hat{\mathbf{z}}_{t_j(u)-1}\hat{\mathbf{z}}^\top_{t_j(u)-1}\right)^{-1} \sum_{j=1}^{l} \hat{\mathbf{z}}_{t_j(u)-1}\hat{x}_{t_j}$$

$$= \left(\lambda \mathbf{I}_{p+1} + \underbrace{\sum_{j=1}^{l} \hat{\mathbf{z}}_{t_j(u)-1}\hat{\mathbf{z}}^\top_{t_j(u)-1} - \sum_{j=1}^{l} \mathbf{z}_{t_j(u)-1}\mathbf{z}^\top_{t_j(u)-1}}_{\Delta_1} + \sum_{j=1}^{l} \mathbf{z}_{t_j(u)-1}\mathbf{z}^\top_{t_j(u)-1}\right)^{-1} \cdot$$

$$\left(\underbrace{\sum_{j=1}^{l} \hat{\mathbf{z}}_{t_j(u)-1}\hat{x}_{t_j} - \sum_{j=1}^{l} \mathbf{z}_{t_j(u)-1}x_{t_j}}_{\Delta_2} + \sum_{j=1}^{l} \mathbf{z}_{t_j(u)-1}x_{t_j}\right)$$

$$= \left(\lambda \mathbf{I}_{p+1} + \Delta_1 + \sum_{j=1}^{l} \mathbf{z}_{t_j(u)-1}\mathbf{z}^\top_{t_j(u)-1}\right)^{-1} \left(\Delta_2 + \sum_{j=1}^{l} \mathbf{z}_{t_j(u)-1}x_{t_j}\right)$$

$$= \left(\mathbf{V}_{t_l(u)}(u) + \Delta_1\right)^{-1}\left(\Delta_2 + \mathbf{b}_{t_l(u)}(u)\right) = \left(\mathbf{V}_{t_l(u)}(u) + \Delta_1\right)^{-1}\Delta_2 + \left(\mathbf{V}_{t_l(u)}(u) + \Delta_1\right)^{-1}\mathbf{b}_{t_l(u)}(u)$$

$$\stackrel{(a)}{=} \underbrace{\mathbf{V}^{-1}_{t_l(u)}(u)\mathbf{b}_{t_l(u)}(u) - \mathbf{V}^{-1}_{t_l(u)}(u)\Delta_1\hat{\mathbf{V}}^{-1}_{t_l(u)}(u)\mathbf{b}_{t_l(u)}(u)}_{P_1} + \underbrace{\mathbf{V}^{-1}_{t_l(u)}(u)\Delta_2 - \mathbf{V}^{-1}_{t_l(u)}(u)\Delta_1\hat{\mathbf{V}}^{-1}_{t_l(u)}(u)\Delta_2}_{P_2}.$$

where passage (a) arises from the observation that

$$\left(\mathbf{V}_{t_l(u)}(u) + \Delta_1\right)^{-1} = \mathbf{V}_{t_l(u)}(u) - \mathbf{V}_{t_l(u)}(u)\Delta_1\left(\mathbf{V}_{t_l(u)}(u) + \Delta_1\right)^{-1} = \mathbf{V}_{t_l(u)}(u) - \mathbf{V}_{t_l(u)}(u)\Delta_1\hat{\mathbf{V}}_{t_l(u)}(u)$$

Before we decompose terms P_1 and P_2, we demonstrate the following decomposition of $\mathbf{V}^{-1}_{t_l(u)}(u)\Delta_1\hat{\mathbf{V}}^{-1}_{t_l(u)}(u)$, observing that $\Delta_1 = \hat{\mathbf{V}}_{t_l(u)}(u) - \mathbf{V}_{t_l(u)}(u)$:

$$\mathbf{V}^{-1}_{t_l(u)}(u)\Delta_1\hat{\mathbf{V}}^{-1}_{t_l(u)}(u) = \mathbf{V}^{-1}_{t_l(u)}(u)(\hat{\mathbf{V}}_{t_l(u)}(u) - \mathbf{V}_{t_l(u)}(u))\hat{\mathbf{V}}^{-1}_{t_l(u)}(u)$$

$$= (\mathbf{V}_{t_l(u)}^{-1}(u)\hat{\mathbf{V}}_{t_l(u)}(u) - \mathbf{I}_{p+1})\hat{\mathbf{V}}_{t_l(u)}^{-1}(u) = \mathbf{V}_{t_l(u)}^{-1}(u) - \hat{\mathbf{V}}_{t_l(u)}^{-1}(u).$$

We now begin with the term P_1. The following holds:

$$P_1 = \mathbf{V}_{t_l(u)}^{-1}(u)\mathbf{b}_{t_l(u)}(u) - \mathbf{V}_{t_l(u)}^{-1}(u)\Delta_1\hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\mathbf{b}_{t_l(u)}(u) =$$

$$= \mathbf{V}_{t_l(u)}^{-1}(u)\sum_{j=1}^{l}\mathbf{z}_{t_j(u)-1}x_{t_j} - \mathbf{V}_{t_l(u)}^{-1}(u)\Delta_1\hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\sum_{j=1}^{l}\mathbf{z}_{t_j(u)-1}x_{t_j}$$

$$= \mathbf{V}_{t_l(u)}^{-1}(u)\sum_{j=1}^{l}\mathbf{z}_{t_j(u)-1}(\langle\phi(u), \mathbf{z}_{t_j(u)-1}\rangle + \varepsilon_{t_j})$$

$$+\mathbf{V}_{t_l(u)}^{-1}(u)\Delta_1\hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\sum_{j=1}^{l}\mathbf{z}_{t_j(u)-1}(\langle\phi(u), \mathbf{z}_{t_j(u)-1}\rangle + \varepsilon_{t_j})$$

$$\overset{(b)}{=} \phi(u) - \lambda\mathbf{V}_{t_l(u)}^{-1}(u)\phi(u) + \mathbf{V}_{t_l(u)}^{-1}(u)\underbrace{\sum_{j=1}^{l}\mathbf{z}_{t_j(u)-1}\varepsilon_{t_j}}_{\mathbf{s}_{t_j}} + \Delta_1\hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\phi(u)$$

$$-\lambda\mathbf{V}_{t_l(u)}^{-1}(u)\Delta_1\hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\phi(u) + \mathbf{V}_{t_l(u)}^{-1}(u)\Delta_1\hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\underbrace{\sum_{j=1}^{l}\mathbf{z}_{t_j(u)-1}\varepsilon_{t_j}}_{\mathbf{s}_{t_j}}$$

$$= \phi(u) - \lambda\mathbf{V}_{t_l(u)}^{-1}(u)\phi(u) + \mathbf{V}_{t_l(u)}^{-1}(u)\mathbf{s}_{t_j} - \Delta_1\mathbf{V}_{t_l(u)}^{-1}(u)\phi(u)$$

$$+\mathbf{V}_{t_l(u)}^{-1}(u)\Delta_1\Delta_1\hat{\mathbf{V}}_{t_l(u)}^{-1}\phi(u) + \lambda\mathbf{V}_{t_l(u)}^{-1}(u)\Delta_1\hat{\mathbf{V}}_{t_l(u)}^{-1}\phi(u) - \mathbf{V}_{t_l(u)}^{-1}(u)\Delta_1\hat{\mathbf{V}}_{t_l(u)}^{-1}\mathbf{s}_{t_j}$$

$$= \phi(u) - \lambda\mathbf{V}_{t_l(u)}^{-1}(u)\phi(u) + \mathbf{V}_{t_l(u)}^{-1}(u)\mathbf{s}_{t_j} - \Delta_1\mathbf{V}_{t_l(u)}^{-1}(u)\phi(u)$$

$$+\Delta_1\mathbf{V}_{t_l(u)}^{-1}(u)\phi(u) - \Delta_1\hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\phi(u) + \lambda\mathbf{V}_{t_l(u)}^{-1}(u)\phi(u) - \lambda\hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\phi(u)$$

$$-\mathbf{V}_{t_l(u)}^{-1}(u)\mathbf{s}_{t_j} + \hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\mathbf{s}_{t_j} = \phi(u) - \lambda\hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\phi(u) - \Delta_1\hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\phi(u) + \hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\mathbf{s}_{t_j},$$

where the passage (b) arises from the observation that $\sum_{j=1}^{l}\mathbf{z}_{t_j-1}(\langle\boldsymbol{\phi}(u), \mathbf{z}_{t_j-1}\rangle) = \sum_{j=1}^{l}\mathbf{z}_{t_j-1}\mathbf{z}_{t_j-1}^{\top}\boldsymbol{\phi}(u)$.

We then proceed with P_2 in a similar fashion as follows:

$$P_2 = \mathbf{V}_{t_l(u)}^{-1}(u)\Delta_2 - \mathbf{V}_{t_l(u)}^{-1}(u)\Delta_1\hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\Delta_2 = \mathbf{V}_{t_l(u)}^{-1}(u)\Delta_2 - \mathbf{V}_{t_l(u)}^{-1}(u)\Delta_2 + \hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\Delta_2 = \hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\Delta_2.$$

Thus, we achieve

$$\boldsymbol{\phi}_{t_l}(u) = \phi(u) - \lambda\hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\phi(u) - \Delta_1\hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\phi(u) + \hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\mathbf{s}_{t_j} + \hat{\mathbf{V}}_{t_l(u)}^{-1}(u)\Delta_2.$$

Moving $\phi(u)$ on the left side of the equation and taking the Gramian norm of both sides, we have the following inequality

$$\|\boldsymbol{\phi}_{t_l}(u) - \phi(u)\|_{\hat{\mathbf{V}}_{t_l(u)}^{-1}(u)} \le \sqrt{\lambda}\|\phi(u)\|_2 + \|\Delta_1\|_2\|\phi(u)\|_2 + \|\Delta_2\|_2 + \|\mathbf{s}_{t_j}\|_{\hat{\mathbf{V}}_{t_l(u)}^{-1}(u)}.$$

We begin deriving the bounds for $\|\Delta_1\|_2$ and $\|\Delta_2\|_2$. First, $\|\Delta_1\|_2$ can be decomposed as

$$\|\Delta_1\|_2^2 = \|\sum_{j=1}^{l}\hat{\mathbf{z}}_{t_j(u)-1}\hat{\mathbf{z}}_{t_j(u)-1}^{\top} - \sum_{j=1}^{l}\mathbf{z}_{t_j(u)-1}\mathbf{z}_{t_j(u)-1}^{\top}\|_2^2$$

$$= \|\sum_{j=1}^{l}\hat{\mathbf{z}}_{t_j(u)-1}\hat{\mathbf{z}}_{t_j(u)-1}^{\top} - \sum_{j=1}^{l}\mathbf{z}_{t_j(u)-1}\hat{\mathbf{z}}_{t_j(u)-1}^{\top} + \sum_{j=1}^{l}\mathbf{z}_{t_j(u)-1}\hat{\mathbf{z}}_{t_j(u)-1}^{\top} - \sum_{j=1}^{l}\mathbf{z}_{t_j(u)-1}\mathbf{z}_{t_j(u)-1}^{\top}\|_2^2$$

$$\leq \| \sum_{j=1}^{l} \underbrace{[\hat{\mathbf{z}}_{t_j(u)-1} - \mathbf{z}_{t_j(u)-1}]}_{\mathbf{e}_{t_j(u)-1}} \hat{\mathbf{z}}_{t_j(u)-1}^{\top} \|_2^2 + \| \sum_{j=1}^{l} \mathbf{z}_{t_j(u)-1} \underbrace{[\hat{\mathbf{z}}_{t_j(u)-1}^{\top} - \mathbf{z}_{t_j(u)-1}^{\top}]}_{\mathbf{e}_{t_j(u)-1}^{\top}} \|_2^2$$

$$\leq \sum_{j=1}^{l} \|\mathbf{e}_{t_j(u)-1} \hat{\mathbf{z}}_{t_j(u)-1}^{\top}\|_2^2 + \sum_{j=1}^{l} \|\mathbf{z}_{t_j(u)-1} \mathbf{e}_{t_j(u)-1}^{\top}\|_2^2,$$

and $\|\Delta_2\|_2$ as

$$\|\Delta_2\|_2^2 = \| \sum_{j=1}^{l} \hat{\mathbf{z}}_{t_j(u)-1} \hat{x}_{t_j} - \sum_{j=1}^{l} \mathbf{z}_{t_j(u)-1} x_{t_j} \|_2^2$$

$$= \| \sum_{j=1}^{l} \hat{\mathbf{z}}_{t_j(u)-1} \hat{x}_{t_j} - \sum_{j=1}^{l} \mathbf{z}_{t_j(u)-1} \hat{x}_{t_j} + \sum_{j=1}^{l} \mathbf{z}_{t_j(u)-1} \hat{x}_{t_j} - \sum_{j=1}^{l} \mathbf{z}_{t_j(u)-1} x_{t_j} \|_2^2$$

$$\leq \| \sum_{j=1}^{l} \hat{\mathbf{z}}_{t_j(u)-1} \hat{x}_{t_j} - \mathbf{z}_{t_j(u)-1} \hat{x}_{t_j} \|_2^2 + \| \sum_{j=1}^{l} \mathbf{z}_{t_j(u)-1} \hat{x}_{t_j} - \mathbf{z}_{t_j(u)-1} x_{t_j} \|_2^2$$

$$\leq \sum_{j=1}^{l} \|\mathbf{e}_{t_j(u)-1} \hat{x}_{t_j}\|_2^2 + \sum_{j=1}^{l} \|\mathbf{z}_{t_j(u)-1} (\hat{x}_{t_j} - x_{t_j})\|_2^2.$$

To be able to bound the 2-norms of Δ_1 and Δ_2, we first introduce the following decomposition of the reward difference:

$$|x_t - \hat{x}_t| = |\sum_{j=0}^{\infty} \psi_j(-d) y_{t-j} - \sum_{j=0}^{t} \psi_j(-\hat{d}_t) y_{t-j}| \leq \underbrace{|\sum_{j=1}^{t} \psi_j(-d) y_{t-j} - \sum_{j=1}^{t} \psi_j(-\hat{d}_t) y_{t-j}|}_{B_1} + \underbrace{|\sum_{j=t+1}^{\infty} \psi_j(-d) y_{t-j}|}_{B_2}.$$

We let $x_t = \sum_{j=0}^{\infty} \psi_j(-d) y_{t-j} = y_t + \sum_{j=1}^{\infty} \prod_{i=1}^{j} \frac{i-1+d}{i} y_{t-j}$, $\tilde{x}_t = \sum_{j=0}^{t} \psi_j(-d) y_{t-j} = y_t + \sum_{j=1}^{t} \prod_{i=1}^{j} \frac{i-1+d}{i} y_{t-j}$, and $\hat{x}_t = \sum_{j=0}^{t} \psi_j(-\hat{d}_t) y_{t-j} = y_t + \sum_{j=1}^{t} \prod_{i=1}^{j} \frac{i-1+\hat{d}_t}{i} y_{t-j}$.

Few important observations about y_t. First, by Theorem 2.6, the process x_t is strictly stationary and ergodic, so the variance of x_t is bounded by a finite constant, i.e., $\text{Var}(x_t) \leq \sigma_{x_t}^2$, for every round $t \in \mathbb{N}$. We also have that, by Lemma B.3, $x_t \leq \frac{m+\eta}{1-\Gamma}$ for every $t \in \mathbb{N}$, where $\eta = \sqrt{2\sigma^2 \log(T/\delta)}$, and so is $y_t = (1-B)^d x_t \leq \frac{m+\eta}{1-\Gamma}$ and with a finite variance, respectively.

With these observations, we begin with the term B_1. We denote $f(d) = \tilde{x}_t$ and $f(\hat{d}_t) = \hat{x}_t$, for which we observe the following through Taylor decomposition:

$$B_1 = |\hat{x}_t - \tilde{x}_t| = |f(\hat{d}_t) - f(d)| \leq |\hat{d}_t - d| \cdot \left| \frac{\partial f(d)}{\partial d} \right| = |\hat{d}_t - d| \sum_{j=1}^{t} \psi_j'(d) y_{t-i}.$$

We intermediately provide the decomposition of the term $\psi_j'(d)$ through the log-derivative, which holds for every $j \in [\![t]\!]$:

$$\psi_j'(-d) = \psi_j(-d) \sum_{i=1}^{j} \frac{1}{j-1+d} = \psi_j(-d) \cdot \mathcal{O}(\log(j)) = \mathcal{O}(j^{-1+d} \log(j)),$$

where we exploit the notion that $\psi_j(-d) = \mathcal{O}(j^{-1+d})$ from Theorem 3.1 of McAleer and Ling [2008].

Therefore, using Theorem 2.6 and Cauchy-Schwarz, we bound $\mathbb{E}[B_1^2]$ as

$$\mathbb{E}[B_1^2] \leq (\hat{d}_t - d)^2 \left(\sum_{j=1}^{t} \mathcal{O}(j^{-1+d} \log j) y_{t-i} \right)^2 = \mathcal{O}\left(\frac{\log\log(t)}{t} \right) \cdot \mathcal{O}(t^{2d} \log t) = \mathcal{O}(t^{2d-1} \log^2(t) \log\log(t)),$$

so the Root Mean Square of B_1 is bounded by $\mathcal{O}(t^{d-0.5}\log(t)\sqrt{\log\log(t)})$.

We proceed by bounding B_2 in a similar fashion. Given that $\psi_j(-d) = \mathcal{O}(j^{-1+d})$, we have by Cauchy-Schwarz that

$$\mathbb{E}[B_2^2] \leq \left(\sum_{j=t+1}^{\infty} \psi_j(-d) y_{t-j}\right)^2 \leq \mathcal{O}(t^{2d-1}),$$

from which we see that the Root Mean Square of B_2 is bounded by $\mathcal{O}(t^{d-0.5})$.

Thus, the bound for the reward difference is:

$$|x_t - \hat{x}_t| = \mathcal{O}\left(t^{d-0.5}\log(t)\sqrt{\log\log(t)}\right) + \mathcal{O}(t^{d-0.5}) = \mathcal{O}\left(t^{d-0.5}\log(t)\sqrt{\log\log(t)}\right),$$

where the first term dominates the latter by the log-factor.

We now bounding $\|\Delta_1\|_2$. Applying Theorem 2.7, we observe that the following holds:

$$\begin{aligned}\sum_{j=1}^{l} \|\mathbf{e}_{t_j(u)-1}\hat{\mathbf{z}}_{t_j(u)-1}^\top\|_2^2 &= \sum_{j=1}^{l} \|\mathbf{e}_{t_j(u)-1}\|_2^2 \|\hat{\mathbf{z}}_{t_j(u)-1}\|_2^2 \\ &\leq \sqrt{1+p\left(\frac{m+\eta}{1-\Gamma}\right)^2} \cdot \sum_{j=1}^{l}\sum_{i=1}^{p} (\hat{x}_{t_j(u)-i-1} - x_{t_j(u)-i-1})^2 \overset{(a)}{=} \mathcal{O}\left(pt^{2d}\log^2(t)\log\log(t)\right),\end{aligned}$$

where passage (a) follows from $\|\hat{\mathbf{z}}_t\|_2^2 \leq \sqrt{1+p\left(\frac{m+\eta}{1-\Gamma}\right)^2} = \mathcal{O}(1)$ for every $t \in [\![T]\!]$. By a similar notion, the same bound holds for the second term $\sum_{j=1}^{l}\|\mathbf{z}_{t_j(u)-1}\mathbf{e}_{t_j(u)-1}^\top\|_2^2$ as well.

Thus, we have the following bound for the norm of Δ_1:

$$\|\Delta_1\|_2 = \mathcal{O}\left(\sqrt{p}t^d\log(t)\sqrt{\log\log(t)}\right) := c_1(t)$$

We then bound $\|\Delta_2\|_2$ in a similar fashion. We start with the term $\sum_{j=1}^{l}\|\mathbf{e}_{t_j(u)-1}\hat{x}_{t_j}\|_2^2$:

$$\sum_{j=1}^{l}\|\mathbf{e}_{t_j(u)-1}\hat{x}_{t_j}\|_2^2 \leq \left(\frac{m+\eta}{1-\Gamma}\right)^2 \sum_{j=1}^{l}\sum_{i=1}^{p}(\hat{x}_{t_j(u)-i-1} - x_{t_j(u)-i-1})^2 \overset{(a)}{=} \mathcal{O}\left(pt^{2d}\log^2(t)\log\log(t)\right),$$

where in passage (a) we similarly observe that $x_t \leq \frac{m+\eta}{1-\Gamma} = \mathcal{O}(1)$ for every $t \in [\![T]\!]$ by Lemma B.1.

We finish by bounding the second term $\sum_{j=1}^{l}\|\mathbf{z}_{t_j(u)-1}(\hat{x}_{t_j} - x_{t_j})\|_2^2$ using all the previous observations as follows:

$$\sum_{j=1}^{l}\|\mathbf{z}_{t_j(u)-1}(\hat{x}_{t_j} - x_{t_j})\|_2^2 \leq \sqrt{1+p\left(\frac{m+\eta}{1-\Gamma}\right)^2}\sum_{j=1}^{l}(\hat{x}_{t_j} - x_{t_j})^2 = \mathcal{O}\left(t^{2d}\log^2(t)\log\log(t)\right)$$

Thus, we have the following bound for the norm of Δ_2:

$$\|\Delta_2\|_2 = \mathcal{O}\left(\sqrt{p+1}t^d\log(t)\sqrt{\log\log(t)}\right) := c_2(t).$$

Therefore, the following inequality holds:

$$\|\boldsymbol{\phi}_{t_l}(u) - \boldsymbol{\phi}(u)\|_{\hat{\mathbf{V}}^{-1}_{t_l(u)}(u)} \leq \sqrt{\lambda}\|\boldsymbol{\phi}(u)\|_2 + c_1(t)\|\boldsymbol{\phi}(u)\|_2 + c_2(t) + \|\mathbf{s}_{t_l(u)}\|_{\hat{\mathbf{V}}^{-1}_{t_l(u)}}.$$

Finally, let $\mathcal{F}_{t_l(u)} = \sigma(\mathbf{z}_0, u_1, \mathbf{z}_1, u_2, \ldots, \mathbf{z}_{t_l(u)-1}, u_{t_l(u)})$ be the filtration generated by all events realized at round $t_l(u)$. Let us now consider the stochastic processes $(\varepsilon_{t_l(u)})_{l\in\mathbb{N}}$ and $(\mathbf{z}_{t_l(u)-1})_{l\in\mathbb{N}}$. We observe that $\varepsilon_{t_l(u)}$ is $\mathcal{F}_{t_l(u)}$-measurable and conditionally σ^2-subgaussian and that $\mathbf{z}_{t_l(u)-1}$ is

$\mathcal{F}_{t_l(u)-1}$-measurable. Thus, by applying Theorem 1 of Abbasi-Yadkori et al. [2011], we have that simultaneously for all $l \in \mathbb{N}$ with probability $1-\delta$:

$$\|\mathbf{s}_{t_l(u)}\|_{\hat{\mathbf{V}}^{-1}_{t_l(u)}} \le \sigma \sqrt{2\log\frac{1}{\delta} + \log\frac{\det \hat{\mathbf{V}}_{t_l(u)}(u)}{\lambda^{p+1}}}$$

for all actions $u \in \mathcal{U}$ and the rounds $t \in \mathbb{N}$.

□

B.4 Proof of the Upper Regret Bound

Before we derive the regret bound, we reconstruct the results of Bacchiocchi et al. [2024] for the External-to-Policy Regret Bound (Lemma B.3) in our setting. We first introduce their results for Policy-Regret-Decomposition (Lemma B.2) to be used for our purposes.

Lemma B.2. *(Policy-Regret-Decomposition). Let $(x_t^*)_{t\in\mathbb{N}}$ be the sequence of rewards by executing the optimal policy π^* and let $(x_t)_{t\in\mathbb{N}}$ be the sequence of rewards by executing the learner's policy π. Then, for every $t \in \mathbb{N}$, it holds that:*

$$\hat{r}_t = r_t + \epsilon_t = \sum_{i=1}^{p} \phi_i(u_t^*) r_{t-i} + \rho_t + \epsilon_t,$$

where $r_t := x_t^* - x_t$ is the instantaneous policy regret, $\rho_t := \langle \boldsymbol{\phi}(u_t^*) - \boldsymbol{\phi}(u_t), \mathbf{z}_{t-1}\rangle$ is the instantaneous external regret, $\epsilon_t = x_t - \hat{x}_t$ is the error term representing the difference between the converted and the true AR rewards, the $u_t^* = \pi_t(H_{t-1}^*)$, and $r_{t-i} = 0$ if $i \ge t$.

Lemma B.3. *(External-to-Policy Regret Bound) Let π be the learner's policy and $T \in \mathbb{N}$ be the horizon. Under Assumptions 2.1 and 2.2, it holds that:*

$$\begin{aligned}\hat{R}(\pi, T) &= \mathbb{E}\left[\sum_{t=1}^{T}\left[\sum_{i=1}^{p} \phi_i(u_t^*) r_{t-i} + \rho_t + \eta_t\right]\right] \\ &\le \left(\frac{\Gamma p}{1-\Gamma} + 1\right)\mathcal{Q}(\underline{\pi}, T) + \mathcal{O}\left(T^{d+0.5}\log(T)\sqrt{\log\log(T)}\right),\end{aligned}$$

where $\mathcal{Q}(\pi, T) := \mathbb{E}[\sum_{t=1}^{T} \rho_t]$ is the cumulative expected external regret.

Proof of Lemma B.3. We use the results of the regret decomposition in Lemma B.2. We then express our cumulative regret as the following Triangular Inequality:

$$\hat{R}(\pi, T) \le \mathbb{E}\left|\left[\sum_{t=1}^{T}\left[\sum_{i=1}^{p} \phi_i(u_t^*) r_{t-i} + \rho_t\right]\right]\right| + \left|\sum_{i=1}^{T} \epsilon_t\right| = \left|\sum_{t=1}^{T} r_t\right| + \left|\sum_{i=1}^{T} \epsilon_t\right|.$$

Bacchiocchi et al. [2024] proved that $\sum_{t=1}^{T} r_t \le \left(\frac{\Gamma p}{1-\Gamma} + 1\right)\mathcal{Q}(\underline{\pi}, T)$. We now create the bound for the sum of error terms in the following way:

$$\begin{aligned}\left|\sum_{t=1}^{T} \epsilon_t\right| &= \left|\sum_{t=1}^{T} (x_t - \hat{x}_t)\right| \le \sum_{t=1}^{T} |x_t - \hat{x}_t| \\ &\overset{(a)}{=} \sum_{t=1}^{T} \mathcal{O}\left(t^{d-0.5}\log(t)\sqrt{\log\log(t)}\right) = \mathcal{O}\left(T^{d+0.5}\log(T)\sqrt{\log\log(T)}\right),\end{aligned}$$

where the passage (a) arises from the result for the bound of $\hat{x}_t - x_t$ obtained in Appendix B.3.

□

To derive the upper bound of regret, we also make use of the Elliptic Potential Lemma (Lattimore and Szepesvári [2020], Lemma 19.4) in our derivations of the bound of regret for our setting.

Lemma B.4. *(Elliptic Potential Lemma). Let $\mathbf{V_0} \in \mathbb{R}^{b\times b}$ be a positive definite matrix and let $\mathbf{u}_1, \dots, \mathbf{u}_n \in \mathbb{R}^b$ be a sequence of vectors such that $\|\mathbf{u}_t\|_2 \leq L < +\infty$ for all $t \in [\![n]\!]$. Let $\mathbf{V_t} = \mathbf{V_0} + \sum_{s=1}^T \mathbf{u}_s\mathbf{u}_s^\top$, then:*

$$\sum_{t=1}^n \min\{1, \|\mathbf{u}_t\|_{\mathbf{V_{t-1}^{-1}}}\} \leq 2d\left(\frac{tr(\mathbf{V_0}) + nL^2}{b\det(\mathbf{V_0})^{1/b}}\right).$$

Proof of Theorem 4.2. Let $\delta \in (0,1)$, and define, as in the main paper, for every round $t \in [\![T]\!]$ and action $u \in \mathcal{U}$:

$$\beta_t(u) := \sqrt{\lambda(m^2+1)} + c_1(t)\sqrt{m^2+1} + c_2(t) + \sigma\sqrt{2\log\left(\frac{1}{\delta}\right) + \log\left(\frac{\det(\hat{\mathbf{V}}_t(u))}{\lambda^{p+1}}\right)},$$

where $c_1(t) = \sqrt{p}t^d\log(t)\sqrt{\log(\log(t)+1)}$ and $c_2(t) = \sqrt{p+1}t^d\log(t)\sqrt{\log(\log(t)+1)}$.

Let us define the confidence set $\mathcal{C}_t(u) :\in \{\boldsymbol{\phi} \in \mathbb{R}^{p+1} : \|\boldsymbol{\phi} - \hat{\boldsymbol{\phi}}_{t-1}(u)\|_{\mathbf{V}_{t-1}(u)} \leq \beta_{t-1}(u)\}$ and the optimistic estimate of the parameter vector $\boldsymbol{\phi}(u)$:

$$\tilde{\boldsymbol{\phi}}_{t-1}(u) = \arg\max_{\boldsymbol{\phi}\in\mathcal{C}_t(u)} \langle \boldsymbol{\phi}, \hat{\mathbf{z}}_{t-1}\rangle.$$

By Theorem 4.1, we have that, for every action $u \in \mathcal{U}$ and round $t \in [\![T]\!]$, the true parameter vector satisfies $\phi(u) \in \mathcal{C}_t(u)$ with a probability of at least $1-\delta$. Therefore, with the same probability, we have:

$$\begin{aligned}
\langle \phi(u_t^*) - \phi(u_t), \hat{\mathbf{z}}_{t-1}\rangle &= \langle \phi(u_t^*) - \phi(u_t), \mathbf{z}_{t-1}\rangle + \langle \phi(u_t^*) - \phi(u_t), \hat{\mathbf{z}}_{t-1} - \mathbf{z}_{t-1}\rangle \\
&\leq 2\beta_{t-1}(u_t)\left(\|\mathbf{z}_{t-1}\|_{\mathbf{V}_{t-1}(u)^{-1}} + \mathcal{O}\left(\sqrt{p}t^{d-0.5}\log(t)\sqrt{\log\log(t)}\right)\right),
\end{aligned}$$

where the first term is derived by Bacchiocchi et al. [2024] using the Cauchy-Schwartz inequality, for which it holds that $\langle \mathbf{v}, \mathbf{w}\rangle = \|\mathbf{v}\|_{\mathbf{V}_{t-1}(u)^{-1}} \cdot \|\mathbf{w}\|_{\mathbf{V}_{t-1}(u)^{-1}}$ for every couple of vectors $\mathbf{v}, \mathbf{w}$, and the second term follows the result of Lemma B.3.

We also introduce the following notion about the external regret derived by Bacchiocchi et al. [2024]:

$$\rho_t = \langle \phi(u_t^*) - \phi(u_t), \mathbf{z}_{t-1}\rangle \leq \|\mathbf{z}_{t-1}\|_2 + m.$$

By Theorem 2.7 we have:

$$\|\mathbf{z}_{t-1}\|_2 \leq \sqrt{1 + p\left(\frac{m+\eta}{1-\Gamma}\right)^2} := L,$$

where $\eta = \sqrt{2\sigma^2\log(T/\delta)}$, and, consequently, we have:

$$\rho_t \leq L + m := C_1.$$

We then proceed as follows:

$$\begin{aligned}
\rho_t &\leq 2\min\left\{C_1, \beta_{t-1}(u_t)\left(\|\mathbf{z}_{t-1}\|_{\mathbf{V}_{t-1}(u)^{-1}} + \mathcal{O}\left(\sqrt{p}t^{d-0.5}\log(t)\sqrt{\log\log(t)}\right)\right)\right\} \\
&\leq 2\max\{C_1, \beta_{t-1}(u_t)\}\min\left\{1, \|\mathbf{z}_{t-1}\|_{\mathbf{V}_{t-1}(u)^{-1}} + \mathcal{O}\left(\sqrt{p}t^{d-0.5}\log(t)\sqrt{\log\log(t)}\right)\right\} \\
&= 2\max\{C_1, \beta_{t-1}(u_t)\}\min\left\{1, \|\mathbf{z}_{t-1}\|_{\mathbf{V}_{t-1}(u)^{-1}}\right\} \\
&\quad +2\max\{C_1, \beta_{t-1}(u_t)\}\min\left\{1, \mathcal{O}\left(\sqrt{p}t^{d-0.5}\log(t)\sqrt{\log\log(t)}\right)\right\}.
\end{aligned}$$

Summing all over $t \in [\![T]\!]$, we get the following bound on the cumulative external regret:

$$\mathcal{Q}(\texttt{ARLM-UCB}) = \sum_{t=1}^T \rho_t \leq \sqrt{T\sum_{t=1}^T \rho_t^2}$$

$$\leq 2\max\{C_1, \beta_{T-1}\}\sqrt{T\sum_{t=1}^{T}\left[\min\left\{1, \|\mathbf{z}_{t-1}\|_{\mathbf{V}_{t-1}(u)^{-1}}\right\}\left(1+\mathcal{O}\left(\sqrt{p}t^{d-0.5}\log(t)\sqrt{\log\log(t)}\right)\right)\right]^2},$$

with $\beta_{T-1} := \max_{u\in\mathcal{U}}\beta_{T-1}(u)$ passage about β_{T-1} holds since the sequence $\beta_t(u_t)$ is non-decreasing and thus each term can be bounded with their value at $t = T$. Futhermore, the last inequality follows from an application of Cauchy-Schwartz inequality and the following observation:

$$\left[\min\left\{1, \|\mathbf{z}_{t-1}\|_{\mathbf{V}_{t-1}(u)^{-1}}\right\}\left(1+\mathcal{O}\left(\sqrt{p}t^{d-0.5}\log(t)\sqrt{\log\log(t)}\right)\right)\right]^2$$

$$= \min\left\{1, \|\mathbf{z}_{t-1}\|^2_{\mathbf{V}_{t-1}(u)^{-1}}\right\} + \min\left\{1, \|\mathbf{z}_{t-1}\|^2_{\mathbf{V}_{t-1}(u)^{-1}}\right\}\mathcal{O}\left(pt^{2d-1}\log^2(t)\log\log(t)\right)$$

$$+ \min\left\{1, \|\mathbf{z}_{t-1}\|^2_{\mathbf{V}_{t-1}(u)^{-1}}\right\}\mathcal{O}\left(\sqrt{p}t^{d-0.5}\log(t)\sqrt{\log\log(t)}\right).$$

Applying The Elliptic Potential Lemma, Bacchiocchi et al. [2024] proved that

$$\sum_{t=1}^{T}\min\left\{1, \|\mathbf{z}_{t-1}\|^2_{\mathbf{V}_{t-1}(u)^{-1}}\right\} \leq 2n(p+1)\log\left(1+\frac{TL^2}{n\lambda(p+1)}\right).$$

Similarly, we may show that

$$\sum_{t=1}^{T}\min\left\{1, \|\mathbf{z}_{t-1}\|^2_{\mathbf{V}_{t-1}(u)^{-1}}\right\}\mathcal{O}\left(\sqrt{p}t^{d-0.5}\log(t)\sqrt{\log\log(t)}\right)$$

$$\leq n(p+1)\log\left(1+\frac{TL^2}{n\lambda(p+1)}\right)\mathcal{O}\left(\sqrt{p}T^{d+0.5}\log(T)\sqrt{\log\log(T)}\right),$$

and

$$\sum_{t=1}^{T}\min\left\{1, \|\mathbf{z}_{t-1}\|^2_{\mathbf{V}_{t-1}(u)^{-1}}\right\}\mathcal{O}\left(pt^{2d-1}\log^2(t)\log\log(t)\right)$$

$$\leq n(p+1)\log\left(1+\frac{TL^2}{n\lambda(p+1)}\right)\mathcal{O}\left(pT^{2d}\log^2(T)\log\log(T)\right).$$

Plugging in these results in the bound for cumulative external regret, we obtain the following:

$$\mathcal{Q}(\texttt{ARLM-UCB}) = \sum_{i=1}^{T}\rho_t \leq \max\{C_1, \beta_{T-1}\}\sqrt{n(p+1)\log\left(1+\frac{TL^2}{n\lambda(p+1)}\right)}$$

$$\cdot\sqrt{T+\mathcal{O}\left(\sqrt{p}T^{d+1.5}\log(T)\sqrt{\log\log(T)}\right)+\mathcal{O}\left(pT^{2d+1}\log^2(T)\log\log(T)\right)}$$

$$= \max\{C_1, \beta_{T-1}\}\sqrt{n(p+1)\log\left(1+\frac{TL^2}{n\lambda(p+1)}\right)\left(T+\mathcal{O}\left(pT^{2d+1}\log^2(T)\log\log(T)\right)\right)}.$$

Finally, we bound the tern β_{T-1}:

$$\beta_{T-1} := \sqrt{\lambda(m^2+1)}+c_1(T-1)\sqrt{m^2+1}+c_2(T-1)+\sigma\max_{u\in\mathcal{U}}\sqrt{2\log\left(\frac{1}{\delta}\right)+\log\left(\frac{\det(\hat{\mathbf{V}}_{T-1}(u))}{\lambda^{p+1}}\right)},$$

$$\leq \sqrt{\lambda(m^2+1)}+c_1(T-1)\sqrt{m^2+1}+c_2(T-1)+\sigma\sqrt{2\log\left(\frac{1}{\delta}\right)+(p+1)\log\left(\frac{\lambda(p+1)+TL^2}{\lambda(p+1)}\right)}.$$

We then set $\delta = (2T)^{-1}$. By highlighting the dependence on m, p, σ, Γ, and T, we have:

$$\beta_{T-1} = \mathcal{O}\left(m\left(1+T^d\log(T)\sqrt{(p+1)\log\log(T)}\right)+\sigma\sqrt{p+1}\right),$$

and

$$C_1 = \mathcal{O}\left(1 + \sqrt{p}\frac{m+\sigma}{1-\Gamma}\right).$$

These results hold with probability $1 - 2\delta$:

$$\begin{aligned}
\mathcal{Q}(\texttt{ARLM-UCB}) = \sum_{i=1}^{T} \rho_t \leq \mathcal{O}\left(\frac{(m+\sigma)\, T^d \log(T) \sqrt{n(p+1)\log(\log(T))}}{1-\Gamma}\right) \\
\cdot \sqrt{T + \mathcal{O}\left(pT^{2d+1} \log^2(T) \log\log(T)\right)} \\
= \left(\frac{(p+1)(m+\sigma)\sqrt{n} T^{2d+0.5} \log^2(T) \log\log(T)}{1-\Gamma}\right).
\end{aligned}$$

Finally, applying Lemma B.3, this results in:

$$\hat{R}(\texttt{ARLM-UCB}, T) \leq \left(\frac{(p+1)^2(m+\sigma)\sqrt{n} T^{2d+0.5} \log^2(T) \log\log(T)}{(1-\Gamma)^2}\right).$$

□

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim that our algorithm can operate within the LM-ARB setting and learn the optimal policy outlined in Theorem 2.5. The results of numerical experiments presented in Section 5 support this claim, demonstrating the sublinear convergence of the cumulative regret achieved by our algorithm and showcasing the best performance across multiple popular bandit baselines, including the `AR-UCB` from Bacchiocchi et al. [2024] designed to manage AR(p) processes.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In the discussion of Theorem 2.5 in Section 2, we explain our motivation to define the optimal policy in terms of converted short-term AR reward rather than directly analyzing the environment-generated long memory reward due to the complexity of the reward-generating ARFIMA process. Although our algorithm only maximizes the optimal policy w.r.t. short-term converted AR rewards, as described in the same section, we believe that it is inspiring to develop a policy evaluation mechanism that directly leverage the environment-generated ARFIMA rewards.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All of our assumptions are outlined as Assumptions 2.1, Assumptions 2.2, Assumptions 2.3 and Assumptions 2.4 that are introduced in Section 2.2. Any further conditions are outlined in the statement of our theorems and lemmas, the proofs for all of which can be found in Appendix B.

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The full code of our algorithm implementation is available at https://github.com/uladcharn/LM-ARM. The full details of our numerical studies are included in Section 5. Additional experiments validating our algorithm's performance on real-world data are presented in Appendix A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The full code of our algorithm implementation is available at https://github.com/uladcharn/LM-ARM.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The full details of our numerical studies are included in Section 5, and the details about real-world data experiments are described in Appendix A.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The plots for all of our numerical experiments introduced in Section 5 and Appendix A display the standard deviation fills around the cumulative regret graph for each algorithm.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All the algorithms were implemented in Python 3.12, and run over an Apple M1 with 8 GB RAM, in no more than a couple of hours.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read and acknowledged the code of ethics. Our paper is a theoretical contribution to the field of reinforcement learning theory, and as such conforms to the guidelines outlined.

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our paper is a theoretical contribution to the field of reinforcement learning theory, and as such any societal impact will at the very least be second-order impacts not directly tied to our work. We discuss that Algorithm 1 can analyze the long memory rewards conditioned by the ARFIMA process. This novel algorithmic capability allows it be employed for modeling long memory temporal sequences in real-world tasks across various fields, such as finances, seismology, or meteorology.

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper poses no such risks.

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: In our study, we use the existing Python code for a range of selected bandit algorithms originally written by Bacchiocchi et al. [2024], which is protected by the MIT license.

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our paper does not release new assets.

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our study does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our study does not involve crowdsourcing nor research with human subjects.

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.