Multi-Armed Bandit Problem with Temporally-Partitioned Rewards: When Partial Feedback Counts

Giulia Romano, Andrea Agostini, Francesco Trovè, Nicola Gatti and Marcello Restelli
Politecnico di Milano, Piazza Leonardo da Vinci 32, I-20133, Milan, Italy
{giulia.romano, francesco1.trovo, nicola.gatti, marcello.restelli}@polimi.it,
andrea1.agostini@mail.polimi.it

Abstract
There is a rising interest in industrial online applications where data becomes available sequentially. Inspired by the recommendation of playlists to users where their preferences can be collected during the listening of the entire playlist, we study a novel bandit setting, namely Multi-Armed Bandit with Temporally-Partitioned Rewards (TP-MAB), in which the stochastic reward associated with the pull of an arm is partitioned over a finite number of consecutive rounds following the pull. This setting, unexplored so far to the best of our knowledge, is a natural extension of delayed-feedback bandits to the case in which rewards may be delayed over a finite-time span after the pull instead of being fully disclosed in a single, potentially delayed round. We provide two algorithms to address TP-MAB problems, namely, TP-UCB-FR and TP-UCB-EW, which exploit the partial information disclosed by the reward collected over time. We show that our algorithms provide better asymptotic regret upper bounds than delayed-feedback bandit algorithms when a property characterizing a broad set of reward structures of practical interest, namely α-smoothness, holds. We also empirically evaluate their performance across a wide range of settings, both synthetically generated and from a real-world media recommendation problem.

1 Introduction
Sequential decision-making occurs in many real-world scenarios such as clinical trials, recommender systems, web advertising, and e-commerce. Inspired by these applications, many different flavours of the multi-armed bandit (MAB) setting have been investigated. A crucial role is played by the time the reward is observed. In many cases, the reward is subject to a delay, and such a delay, if not sufficiently short, can prevent the design of algorithms that are effective in practice. Online learning with delayed feedback has received considerable attention in recent years, and several results are available in the literature, e.g., see the seminal work by Joulani et al. [2013]. A major distinction in MABs with delayed feedback concerns the nature of the rewards, which may be stochastic [Mandel et al., 2015; Cella and Cesa-Bianchi, 2020] or adversarial [Bistritz et al., 2019; Thune et al., 2019; van der Hoeven and Cesa-Bianchi, 2021].

Our work focuses on a special class of bandit problems with stochastic and delayed rewards, in which we can get partial feedback over time. More precisely, we study a novel setting, namely MAB with Temporally-Partitioned Rewards (TP-MAB), in which the reward associated with an action, a.k.a. arm, chosen at a given round is collected during a finite number of rounds following the choice, according to an unknown probability distribution. In classical delayed-feedback bandits (see, e.g., Joulani et al. [2013]), the reward is concentrated in a single round that is (stochastically) delayed w.r.t. the round in which the learner pulled the corresponding arm. TP-MABs naturally extend this setting by allowing the reward to be partitioned into multiple elements that are collected with different delays. We call arm’s per-round reward the partial reward observed by the learner in a single round, which is assumed to be the realization of a random variable with an unknown probability distribution. We call arm’s cumulative reward the random variable given by the sum of all the per-round rewards obtained by pulling an arm. While the per-round reward can be observed round by round, the cumulative reward is revealed only at the end. Notice that, in a single round, the learner observes a per-round reward for each previously pulled arm whose cumulative reward is not terminated yet. Our goal is to find a policy to maximize the cumulative reward, exploiting the per-round rewards as intermediate signals on the arm performance.

Motivating applications. A motivating example for TP-MABs is recommending media content and, in particular, song playlists to a class of users (i.e., users sharing similar characteristics). In this setting, each arm corresponds to a playlist. The reward is measured in listening time (proportional to the user’s appreciation). The goal is to find the playlist that maximizes the reward. The recommendation system suggests a playlist to a new user at each round, whose appreciation is revealed through multiple steps. In particular, every partial observation corresponds to a song in the playlist, and the associated reward is positive if the user listens to that song and non-positive otherwise. The cumulative reward provided by recommending a playlist to a single user corresponds to the sum of the reward terms from all the playlist songs. Notice that the playlist cannot be trivially modeled as...
a collection of independent songs, as their order in the playlist affects the user’s behavior. In the classical delayed-feedback bandit setting, the feedback on the recommended playlist is obtained only once the user finishes listening to the entire playlist. However, the platform monitors whether every song is listened to or skipped by the user. Therefore, clues on the performances of the recommended arm can be exploited before the user finishes the playlist.

Another scenario captured by the TP-MAB framework is the evaluation of medical treatments taking place over a long period of time. In this setting, the per-round reward corresponds to the patient’s state of health at each daily/weekly medical check, and the goal is to find the treatment providing the greatest overall benefit to the patient. In the case of severe pathologies, such as cancer, this type of partial information would span several months if not years, providing valuable insights that would be otherwise ignored. Applying a standard delayed-MAB approach to this scenario, i.e., taking decisions only at the end of each treatment cycle, could negatively affect the time required to select an effective medical treatment. In this type of setting, we argue that the partial information provided by patients in periodic medical checks should be used to speed up the learning process.

**Original Contributions.** Initially, we focus on the lower bound of TP-MABs, showing that the TP-MAB setting has the same regret lower bound of the standard delayed-MAB setting when there is no further assumption about how the rewards are partitioned over time. Since in many practical applications of interest the cumulative reward of each arm does not concentrate excessively in a short sub-range of rounds, we introduce a property describing how the maximum per-round reward distributes. We call this property α-smoothness where $\alpha \geq 1$. In particular, the minimum value of $\alpha = 1$ corresponds to the case in which there is no structure and, therefore, the maximum per-round reward can be the entire cumulative reward. On the other hand, the maximum value of $\alpha$ is equal to the maximum delay and corresponds to the case in which the cumulative reward distributes evenly over time. Thus, the maximum per-round reward decreases as the value of $\alpha$ increases. We show that the lower bound of this setting is of a factor $1/\alpha$ smaller than that when $\alpha$-smoothness does not hold. Then, we design two novel algorithms, namely TP-UCB-FR and TP-UCB-EW, suited for the TP-MAB setting, which exploit partial feedback and the $\alpha$-smoothness property. We show that the regret of TP-UCB-FR is $O(\ln T/\alpha)$, where $T$ is the time horizon of the learning process, and the regret of TP-UCB-EW is $O(\ln T)$. A comprehensive analysis of the regret bounds of our and state-of-the-art algorithms in various settings can be found in Table 3 (in Appendix A for reasons of space). Finally, we experimentally show that our algorithms outperform the state of the art over synthetically generated and a real-world playlist recommendation scenario.

**Related Works.** To the best of our knowledge, ours is the first work addressing a bandit problem in which the reward from a pull is partitioned across multiple rounds. The most related works concern the Delayed-MAB setting, such as the seminal paper by Joulani et al. [2013], which summarizes the known results on the regret upper bounds of online learning algorithms. They also provide a modification of the well-known UCB1 algorithm from Auer et al. [2002] for the delayed-feedback setting, called Delayed-UCB1. More recently, a variety of delayed-feedback scenarios were studied investigating directions different from ours, such as linear and contextual [Arya and Yang [2020], Vernade et al. [2020a], Zhou et al. [2019]], non-stationary [Vernade et al. [2020b]] bandits under delayed feedback. Pike-Burke et al. [2018] and Cesa-Bianchi et al. [2018] also analyze the case of delayed, aggregated, and anonymous feedback. For clarity, we remark that, in our work, per-round rewards corresponding to different pulls can be received in the same round, and it is known from which arm they were generated. Many works apply bandits to practical scenarios, e.g., scheduling [Cayci et al. [2019]], advertising [Nuara et al. [2018], Castiglioni et al. [2022], Nuara et al. [2022]], pricing [Trovo et al. [2018]], and delayed feedback settings [Vernade et al. [2017]].

Works from the bandit literature, such as the ones by Dudik et al. [2011], Desautels et al. [2014], Neu et al. [2013], rely on known constant delays or maximum delay values. Similarly, in our work, we assume a maximum finite delay equal to $\tau_{\text{max}}$, which is compliant with the real-world scenarios we aim at modeling, e.g., in the above example of playlist recommendations, an infinite $\tau_{\text{max}}$ would correspond to a playlist of an infinite number of songs. According to the terminology used in the delayed-MAB literature, our setting is uncensored, meaning that the reward provided by a given action is eventually observed after a finite maximum delay. Conversely, many works in the field, such as, e.g., Manegueu et al. [2020] and Vernade et al. [2017], deals with random delays from an unbounded distribution with finite expectation.

# Problem Formulation

Consider a MAB problem with $K \in \mathbb{N}^*$ arms, over a time horizon of $T \in \mathbb{N}^*$ rounds. At every round $t \in [T]$, the learner pulls an arm $i \in \mathcal{A} = [K]$ and, from the pull of that arm, gets a per-round reward $x_{t,i} = x_{t,i,m_{t,i}}$ at every round $m \in \{t, \ldots, t + \tau_{\text{max}} - 1\}$, where $\tau_{\text{max}} \in \mathbb{N}^*$ is the time span over which the reward is partitioned. In particular, $\tau_{\text{max}} = 1$ is the maximum delay affecting the observation of a per-round reward, whose value is known to the learner. Therefore, at round $t + \tau_{\text{max}} - 1$, the cumulative reward from pulling arm $i$ at round $t$ is completely collected by the learner. Furthermore, we denote by $x_t = [x_{t,1}, \ldots, x_{t,\tau_{\text{max}}}]$ the vector of per-round rewards collected from pulling arm $i$ at round $t$. For every $j \in [\tau_{\text{max}}]$, the per-round reward $x_{t,j}$ is a realization of a random variable $X_{i,j}$ with support $[\mathcal{X}_j, \mathcal{X}_j]$. The cumulative reward collected from pulling arm $i$ at round $t$ is denoted by $r_t$, and it is the realization of the random variable $R_t := \sum_{j=1}^{\tau_{\text{max}}} X_{i,j}$, with support $[\mathcal{R}, \mathcal{R}]$, where $R_t := \sum_{j=1}^{\tau_{\text{max}}} X_{i,j}$, and $\mathcal{R} := \sum_{j=1}^{\tau_{\text{max}}} \mathcal{X}_j$. For every $i \in \mathcal{A}$ and $t \in [T]$, we assume that the

\[ We denote by $[n]$ the set $\{1, \ldots, n\}$.
variables $R_t^i$ are independent with mean $\mu_i := E[R_t^i]$.\footnote{W.l.o.g., we assume $X_t^i = 0, \forall i \in [K], \forall j \in [T_{\max}].$}

A policy $\Pi$ is an algorithm that at each round $t$ chooses an arm $i_t \in [K]$. The performance of a policy $\Pi$ is evaluated in terms of pseudo-regret, defined as the cumulative loss due to playing suboptimal arms during the time horizon $T$, formally:

$$R_T(\Pi) = T \mu^* - E \left[ \sum_{t=1}^{T} \mu_{i_t} \right],$$

where $\mu^* = \max_{i \in A} \{\mu_i\}$ is the expected reward of the optimal arm $i^*$, and the expectation is taken w.r.t. the stochasticity of the policy $\Pi$. Notice that we adopt the concept of pseudo-regret as for standard bandits, unlike what is done by Vernade et al. [2017], since our choice allows for a direct comparison with the vast prior work on delayed bandits.

In what follows, we cast the playlist recommendation problem, described in the introduction, in the TP-MAB setting.

**Example 1** (Playlist Recommendation). At each round $t$, a new user enters the platform, which provides a playlist suggestion. The different arms $i$ are the available playlists to suggest, each composed of $N$ songs. Songs are characterized by 4 listening levels (from “skipped” to “complete”), each associated with a different Bernoulli random variable representing the corresponding per-round reward. The vector of realized per-round rewards of song $k \in [N]$ is $[x_t^{i_1,k}, x_t^{i_2,k}, \ldots, x_t^{i_N,k}]$. Each variable assumes a value of 1 if the user reaches the corresponding level, and a value of 0 if the user stops listening to the song before that level. The cumulative reward $R_t^i$ for pulling arm $i$ at round $t$ is the sum of the rewards from the songs in the playlist, and the time span over which the platform observes the reward is $T_{\max} = 4N$.

We show that the TP-MAB problem has a lower-bound on the regret of the same order of the delayed-feedback bandit problem. The rationale is that no better lower bound is possible as delayed-feedback MABs with a finite delay are a subclass of TP-MABs whose reward vector $x_t^i$ has a single non-zero element for each $i \in A$ and $t \in [T]$. Most interestingly, the worst-case instance for the regret lower bound in the TP-MAB setting is the delayed-feedback bandit.\footnote{All the proofs are deferred to Appendix B for space reasons. See https://trovo.faculty.polimi.it/01papers/romano2022multi.pdf.}

**Theorem 1.** The regret of any uniformly efficient policy $\Pi$ applied to the TP-MAB problem is bounded from below by:

$$\lim_{T \to +\infty} \inf \frac{R_T(\Pi)}{\ln T} \geq \sum_{i: \mu_i < \mu^*} \Delta_i \frac{KL \left( \frac{\mu_i}{\bar{R}_{\max}}, \frac{\mu^*}{\bar{R}_{\max}} \right)}{\bar{R}_{\max}},$$

where $\Delta_i := \mu^* - \mu_i$ is the expected loss suffered by the learner if the arm $i$ is chosen instead of the optimal one $i^*$, $\bar{R}_{\max} := \max_{i \in [K]} R_i$, and $KL(p, q)$ is the Kullback-Leibler divergence between Bernoulli r.v. with means $p$ and $q$.\footnote{An uniformly efficient policy chooses the suboptimal arms on average $O(t^a)$ times ($0 < a < 1$) over $t$ rounds.}

The support of $Z_{t,k}^i$ is denoted by $[Z_{t,k}^i, \bar{Z}_{t,k}^i]$, where $Z_{t,k}^i := \sum_{j=(k-1)\phi+1}^{k\phi} X_{t,j}$, and $\bar{Z}_{t,k}^i := \sum_{j=(k-1)\phi+1}^{k\phi} X_{t,j}$. Intuitively, the $\alpha$-smoothness property states that the elements in $Z_{t,k}^i$ are independent and that, when $\alpha > 1$, the maximum reward $\bar{R}_i$ of a pull cannot be realized in a single time span corresponding to a $Z_{t,k}^i$ element. Formally:

**Def 1.** ($\alpha$-smoothness). In the TP-MAB setting, for $\alpha \in \{0, 1\}$, we say that the reward is $\alpha$-smooth if $\alpha$ and only if $\Delta_i \leq \bar{R}_{\max}$.

An example of $\alpha$-smooth environment with $\phi = 3$ is presented in Figure 1, where colors denote the elements $z_{t,k}^i$ that are the realizations of the variables $Z_{t,k}^i$.\footnote{We assume $\alpha$ is a factor of $\bar{R}_{\max}$ for the sake of presentation.

The following results also hold for generic $\alpha \in \{0, 1\}$.

Figure 1: Example of $\alpha$-smooth reward with $\phi = 3$.}

3 $\alpha$-Smoothness Property

From Theorem 1, we know that we cannot design algorithms with regret upper bounds better than those of the algorithms for the delayed-feedback bandit setting. Nonetheless, in practice, collecting per-round rewards can provide useful information on the cumulative reward of an arm. However, as already pointed out by Manegueu et al. [2020] for the standard delayed-feedback setting, zero rewards are ambiguous since they do not give any information on future rewards. In the general setting, small per-round rewards observed in the first rounds after the pull are not much informative to bound the values of future ones. To avoid this, we focus on those problems in which the maximum reward realized over a few rounds cannot exceed a fraction of the maximum reward $\bar{R}_i$.

Let us consider $\alpha \in \{T_{\max}\}$ s.t. $\alpha$ is a factor of $T_{\max}$, i.e., $T_{\max} := \phi$, and $\phi \in \mathbb{N}$. Let us define the vector $Z_{t,\alpha}^i := [Z_{t,1}^i, \ldots, Z_{t,\alpha}^i]$ whose element $Z_{t,k}^i$ is the random variable corresponding to the sum of a set of consecutive per-round rewards of cardinality $\phi$. Formally, for each $k \in [\alpha]$:

$$Z_{t,k}^i := \sum_{j=(k-1)\phi+1}^{k\phi} X_{t,j}.$$\footnote{We assume $\alpha$ is a factor of $\bar{R}_{\max}$ for the sake of presentation.

The following results also hold for generic $\alpha \in \{0, 1\}$.}

The support of $Z_{t,k}^i$ is denoted by $[Z_{t,k}^i, \bar{Z}_{t,k}^i]$, where $Z_{t,k}^i := \sum_{j=(k-1)\phi+1}^{k\phi} X_{t,j}$, and $\bar{Z}_{t,k}^i := \sum_{j=(k-1)\phi+1}^{k\phi} X_{t,j}$. Intuitively, the $\alpha$-smoothness property states that the elements in $Z_{t,k}^i$ are independent and that, when $\alpha > 1$, the maximum reward $\bar{R}_i$ of a pull cannot be realized in a single time span corresponding to a $Z_{t,k}^i$ element. Formally:

**Def 1.** ($\alpha$-smoothness). In the TP-MAB setting, for $\alpha \in \{0, 1\}$, we say that the reward is $\alpha$-smooth if and only if $\Delta_i \leq \bar{R}_{\max}$.\footnote{We assume $\alpha$ is a factor of $\bar{R}_{\max}$ for the sake of presentation.

The following results also hold for generic $\alpha \in \{0, 1\}$.

An example of $\alpha$-smooth environment with $\phi = 3$ is presented in Figure 1, where colors denote the elements $z_{t,k}^i$ that are the realizations of the variables $Z_{t,k}^i$.\footnote{We assume $\alpha$ is a factor of $\bar{R}_{\max}$ for the sake of presentation.

The following results also hold for generic $\alpha \in \{0, 1\}$.}

Consider the extreme values of parameter $\alpha$. When $\alpha = 1$, the reward has no constraint on how it distributes over time. This scenario includes the delayed-feedback bandit setting in which the cumulative reward provided by the arm pulled at $t$ is entirely collected at a single round (including the last possible round $t + T_{\max} - 1$). Note that, in this case, at each round before $t + T_{\max} - 1$, the sum of the future per-round rewards is in the range $[0, \bar{R}_i]$. Conversely, when $\alpha = T_{\max}$, the vector of aggregated rewards coincides with the vector of per-round
Algorithm 1 TP-UCB-FR

1: Input: $\alpha \in [\tau_{\max}]$, $\tau_{\max} \in \mathbb{N}^*$
2: for $t \in \{1, \ldots, K\}$ do  
3:  Pull arm $i_t = t$  
4:  for $t \in \{K + 1, \ldots, T\}$ do  
5:    for $i \in \{1, \ldots, K\}$ do  
6:      Compute $\hat{R}^i_t$ and $c^i_t$ as in Eqs (4)-(5)  
7:      $u^i_{t-1} \leftarrow \hat{R}^i_t + c^i_t$  
8:    Pull arm $i_t = \arg \max_{i \in \{1, \ldots, K\}} u^i_{t-1}$  
9:  Observe $x^i_{h,t-h+1}$ for $h \in \{t - \tau_{\max} + 1, \ldots, t\}$  

The rationale is to use the rewards coming from not fully-realized reward vectors by replacing the missing elements with fictitious realizations. At round $t$, fictitious reward vectors are associated to each arm pulled in the time span $H := \{t - \tau_{\max} + 1, \ldots, t\}$. We denote them by $x^i_{h} = [\hat{x}^i_{h,1}, \ldots, \hat{x}^i_{h,\tau_{\max}}]$ with $h \in H$, where $\hat{x}^i_{h,j} := x^i_{h,j}$, if $h + j \leq t$, and $\hat{x}^i_{h,j} = 0$, if $h + j > t$. The corresponding fictitious cumulative reward is $\hat{x}^i_{h} := \sum_{j=1}^{\tau_{\max}} \hat{x}^i_{h,j}$. The algorithm takes as input the smoothness $\alpha \in [\tau_{\max}]$, and the maximum delay $\tau_{\max}^\delta$. During the initialization phase, all arms are pulled once (Line 3). After that, at each round $t$, it computes the estimated expected reward for each arm $i$:

$$
\hat{R}^i_{t-1} := \frac{1}{n^i_{t-1}} \left( \sum_{h=1}^{\tau_{\max}} \tilde{r}^i_{h} I_{\{i_h = i\}} + \sum_{h \in H} \tilde{r}^i_{h} I_{\{i_h = i\}} \right),
$$

where $n^i_{t-1} := \sum_{h=1}^{t-\tau_{\max}+1} I_{\{i_h = i\}}$ is the number of times arm $i$ has been pulled by the policy up to round $t - 1$, and the confidence interval:

$$
c^i_{t-1} := \frac{2 \ln(t-1)}{\alpha n^i_{t-1}} + \frac{\phi(\alpha + 1)\hat{R}^i_t}{2n^i_{t-1}}. \tag{5}
$$

Finally, it pulls the arm with the largest upper confidence bound $u^i_{t-1}$ (Line 8), and observes its reward (Line 9).

We provide the following upper bound on the regret:

**Theorem 3.** In the TP-MAB setting with $\alpha$-smooth reward, the pseudo-regret of TP-UCB-FR after $T$ rounds is:

$$
R_T(\mathcal{U}_{FR}) \leq \sum_{i: \mu_i < \mu^*} \frac{4(\hat{R}^i_t)^2 \ln T}{\alpha \Delta_i} \left( 1 + \sqrt{\frac{1 + \frac{\alpha(\alpha + 1)\phi}{2\hat{R}^i_t} \ln T}{\Delta_i}} \right) + (\alpha + 1)\phi \sum_{i: \mu_i < \mu^*} \hat{R}^i_t + \left( 1 + \frac{\pi^2}{3} \right) \sum_{i: \mu_i < \mu^*} \Delta_i.
$$

We observe that the dominant term in $T$ has the order of $O\left( \sum_{i: \mu_i < \mu^*} \frac{n_{\text{max}} \ln T}{\Delta_i} \right)$, where $\Delta_{\text{max}} = \max \Delta_i$. When $\alpha = 1$, the upper bound scales as the one of classical MAB algorithms in stochastic settings. Notice that the pseudo-regret indirectly depends on $\tau_{\max}$ since $\hat{R}$ represents the cumulative reward obtained over $\tau_{\max}$ rounds. Let us compare this result with the one provided in Theorem 1 for general TP-MAB problems. Applying to Theorem 1 the inequality

$$
KL(p,q) \leq \frac{(p-q)^2}{q(1-q)}
$$

where for $p, q \in [0,1]$, derived using the fact that $\ln x \leq x - 1$, we get:

$$
\lim_{T \to +\infty} \frac{R_T(\mathcal{U})}{\ln T} \geq \sum_{i: \mu_i < \mu^*} \beta \Delta_i, \tag{6}
$$

where $\beta = \frac{\mu^*}{\Delta_{\text{max}}}.$

For $\alpha > 4(\hat{R}^i_t)^2/\beta$, the multiplicative factor in the dominant term of the upper bound provided in Theorem 3 is better than that in the lower bound in Theorem 1. This suggests that exploiting the $\alpha$-smoothness provides an improvement over the classical and delayed-feedback MABs.

### 4.2 The TP-UCB-EW Algorithm

The pseudo-code of TP-UCB-EW is provided in Algorithm 2. The key idea is to compute an upper confidence bound for the average of each set of $k$-th realized aggregated rewards $z^i_{k,h}$ from arm $i$ and use them to build an upper bound on the

---

8If these information are not available one should use $\alpha = 1$, meaning we are not assuming any structure on the reward, and use as $\tau_{\max}$ the largest delay observed so far.
Algorithm 2 TP-UCB-EW

1: Input: $\alpha \in [\tau_{\text{max}}]$, $\tau_{\text{max}} \in \mathbb{N}^*$
2: for $t \in \{1, \ldots, K\}$ do
3:   Pull arm $i_t = t$
4: for $t \in \{K + 1, \ldots, T\}$ do
5:   for $i \in \{1, \ldots, K\}$ do
6:     for $k \in \{1, \ldots, \alpha\}$ do
7:       Compute $\hat{Z}_{i-1,k}^t$ and $c_{i-1,k}$ as in Eq.s (7)-(8)
8:       $u_{i-1}^t \leftarrow \sum_{k=1}^{\alpha} (\hat{Z}_{i-1,k}^t + c_{i-1,k}^t)$
9:   Pull arm $i_t \in \arg \max_{i \in [K]} u_{i-1}^t$
10: Observe $x_{h,t-h+1}^t$ for $h \in \{t - \tau_{\text{max}} + 1, \ldots, t\}$

Overall average reward $R_i^t$. It takes as input the smoothness parameter $\alpha$, and the maximum delay parameter $\tau_{\text{max}}$. At first, it pulls each arm once (Line 3), while, in the following rounds, it computes the empirical mean:

$$\hat{Z}_{i-1,k}^t := \frac{\sum_{h=1}^{t-k} z_{i,k}^{h,1}(i_{h}=i)}{n_{t-1,k}},$$

(7)

where $n_{i-1,k}^t := \sum_{h=1}^{t-k} \mathbb{1}_{\{i_{h}=i\}}$ is the cardinality of the rewards observed up to round $t - 1$ for the $k$-th element of $Z_{i-1,\alpha}$, and the confidence bound:

$$c_{i-1,k}^t := \frac{R_i^t}{\alpha} \sqrt{\frac{2\ln(t-1)}{n_{t-1,k}}}.$$  (8)

We remark that $\hat{Z}_{i-1,k}^t + c_{i-1,k}^t$ is an upper confidence bound for the $k$-th element of $Z_{i-1,\alpha}$. Finally, the algorithm computes the upper bound $u_{i-1}^t$, summing the bounds above (Line 8), selects the arm $i$ choosing the largest $u_{i-1}^t$ (Line 9), and observes its reward (Line 10).

We provide the following upper bound on the regret:

**Theorem 4.** In the TP-MAB setting with $\alpha$-smooth reward, the pseudo-regret of TP-UCB-EW after $T$ rounds is:

$$R_T(U_{\text{EW}}) \leq \sum_{i, \mu_i < \mu^*} \frac{8(R_i^t)^2 \ln T}{\Delta_i} + \alpha \left( \phi + \frac{\pi^2}{3} \right) \sum_{i, \mu_i < \mu^*} \Delta_i.$$ 

Focusing on the dominant term in $T$ of the regret bound, we do not have an explicit improvement over the classical and delayed-feedback MAB algorithms. Therefore, in this case, the structure provided by the $\alpha$-smoothness seems not to affect the regret bound. Hence, from an asymptotic point of view, there is not a clear advantage from having $\alpha$-smooth rewards. However, the constant term is significantly smaller than that of TP-UCB-FR and allows TP-UCB-EW to be much more effective than TP-UCB-FR to tackle TP-MAB problems with a short time horizon.

## 5 Empirical Evaluation

We compare TP-UCB-FR and TP-UCB-EW algorithms with the UCB1 algorithm by Auer et al. [2002] and the Delayed-UCB1 algorithm by Joulani et al. [2013] in $\alpha$-smooth TP-MAB environments. Appendix A provides details on the adaptation of these two state-of-the-art algorithms to the TP-MAB problem. Notice that, for UCB1, we assume to immediately get the cumulative reward of a pull. Therefore, it represents a clairvoyant algorithm observing $R_i^t$ at round $t$. We compare the algorithms in three settings: two synthetically-generated environments and a real-world playlist recommendation scenario.7

**Setting #1.** At first, we evaluate the influence of the parameter $\alpha$. We model $K = 10$ arms, whose maximum reward is s.t. $R_i^t = 100 i$. The reward is collected over $\tau_{\text{max}} = 100$ rounds, the smoothness parameter is $\alpha = 20$, and the aggregated rewards are s.t. $Z_{i,k}^t \sim \mathcal{U}(0, 1]$, for each $k \in \alpha$.

We run the algorithms over a time horizon of $T = 10^3$ and average the results over 50 independent runs. In the results, TP-UCB-FR($\eta$) and TP-UCB-EW($\eta$) are s.t. the value of $\alpha$ taken as input is $\eta$, with $\eta \in \{5, 10, 20, 25, 50\}$.

**Results.** Figure 2 shows the pseudo-regret $R_T(U)$ over the time horizon and the vertical bars represent the 95% confidence intervals for the mean value. Let us focus on TP-UCB-FR(20) and TP-UCB-EW(20), for which $\eta$ is equal to the $\alpha$ of the environment. TP-UCB-EW(20) provides better results than Delayed-UCB1 over the entire time horizon, while TP-UCB-FR(20) is better than Delayed-UCB1 for $t > 10^4$ and better than TP-UCB-EW(20) for $t > 2 \cdot 10^4$. This suggests that TP-UCB-FR(20) is more suitable for longer time horizons, and this behavior is confirmed by the asymptotic order of Theorem 3. Notice that UCB1 obtains the reward as soon as an arm has been pulled, but it does not exploit the $\alpha$-smoothness property. *Vice versa*, our algorithms incorporate this information that, in some specific situations, allows us to beat even the non-delayed baseline.

During rounds $t \in [1, 7000]$, the Delayed-UCB1 algorithm outperforms TP-UCB-FR, since, during the initial rounds, incomplete samples may be far different from the corresponding unseen realizations, and, therefore, TP-UCB-FR initially pulls the suboptimal arms more often than Delayed-UCB1. Nonetheless, TP-UCB-FR outperforms Delayed-UCB1 over longer time horizons, as expected given the result in Theorem 3. TP-UCB-EW has a similar asymptotic behavior of those of UCB1 and Delayed-UCB1, i.e., the regret curves becomes parallel after $\approx 4000$ rounds. This is because the overall exploration term of the three algorithms is of the same order in $t$ and $\alpha$.

More details about the experiments are deferred to Appendix C.
Table 1: $R_T^{(\%)}(\mathcal{U})$ for Experimental Setting #2.

| $\tau_{\text{max}}$ | $\alpha$ | $R_T^{(\%)}(\mathcal{U}_{\text{FR}})$ | $R_T^{(\%)}(\mathcal{U}_{\text{EW}})$ |
|---------------------|----------|-------------------------------------|-------------------------------------|
| 100                 | 10       | 68.06% (0.26%)                      | 86.03% (0.59%)                     |
| 200                 | 20       | 95.42% (0.15%)                      | 80.38% (0.34%)                     |
| 100                 | 50       | 50.84% (0.11%)                      | 85.36% (0.33%)                     |
| 200                 | 100      | 81.55% (0.10%)                      | 78.70% (0.24%)                     |

Table 2: Pseudo-regret for the Spotify experimental setting.

| Algorithm               | $R_T(\mathcal{U})$ |
|-------------------------|---------------------|
| Delayed-UCB1            | 56473 (805)         |
| TP-UCB-FR               | 25367 (369)         |
| TP-UCB-EW               | 55000 (951)         |
| UCB1                    | 47368 (1289)        |

Figure 3: Pseudo-regret over time for the Spotify setting.

**Spotify Setting.** We apply the TP-MAB approach to solve the user recommendation problem presented in Example 1, using a dataset by Spotify [Brost et al., 2019]. We select the $K = 6$ most played playlist as the arms to be recommended, and each time a playlist $i$ is selected, the corresponding reward realization $x_i^t$ for the first $N = 20$ songs is sampled from the listening sessions of that playlist contained in the dataset. We recall that, in this setting, the maximum delay is $\tau_{\text{max}} = 4N = 80$, and the smoothness parameter is $\alpha = 20$. More details on the setting and the distributions of the reward for each playlist are provided in Appendix C. We average the results over 50 independent runs.

**Results.** Table 2 shows that the TP-UCB-FR algorithm provides the best performance among the analysed algorithms, outperforming UCB1 thanks to the exploitation of the $\alpha$-smoothness property. The regret over time in Figure 3 shows that the TP-UCB-FR provides worse performance than TP-UCB-EW only for a limited amount of rounds ($t < 4000$). This suggests that, in this specific scenario, the TP-UCB-FR algorithm represents a good candidate to provide playlist recommendations.

6 Conclusion and Future Works

This paper introduces the novel TP-MAB setting, which generalizes the delayed-feedback bandit setting with bounded delay. First, we show that the lower bound of the TP-MAB problem is the same as that of the standard delayed MAB problem. Then, we characterize a broad set of reward structures, by defining the $\alpha$-smoothness property, for which we provide a tighter lower bound. We design the TP-UCB-FR and the TP-UCB-EW algorithms, suited for the TP-MAB setting, which exploit the partial rewards collected over time and the $\alpha$-smoothness property. We show that the upper bounds on the regret for these algorithms are $O(\ln T/\alpha)$ and $O(\ln T)$, respectively. Finally, we empirically show that our algorithms outperform the state of the art over a wide range of settings generated from synthetic and real-world data.

An interesting future extension would be to consider generic functions regulating the relationship between the cumulative and delayed rewards.
References

[Arya and Yang, 2020] Sakshi Arya and Yuhong Yang. Randomized allocation with nonparametric estimation for contextual multi-armed bandits with delayed rewards. *Statistics & Probability Letters*, 164:108818, 2020.

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

[Bistritz et al., 2019] Ilai Bistritz, Zhengyuan Zhou, Xi Chen, Nicholas Bambos, and Jose Blanchet. Exp3 learning in adversarial bandits with delayed feedback. *NeurIPS*, 2019.

[Brost et al., 2019] Brian Brost, Rishabh Mehrotra, and Tristan Jehan. The music streaming sessions dataset. In *WWW*. ACM, 2019.

[Bubeck and Cesa-Bianchi, 2012] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multiarmed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[Castiglioni et al., 2022] Matteo Castiglioni, Alessandro Nuara, Giulia Romano, Giorgio Spadaro, Francesco Trovò, and Nicola Gatti. Safe online bid optimization with return-on-investment and budget constraints subject to uncertainty. *arXiv preprint arXiv:2201.07139*, 2022.

[Cayci et al., 2019] Semih Cayci, Atilla Eryilmaz, and Rayadurgam Srikant. Learning to control renewal processes with bandit feedback. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(2):1–32, 2019.

[Cella and Cesa-Bianchi, 2020] Leonardo Cella and Nicolò Cesa-Bianchi. Stochastic bandits with delay-dependent payoffs. In *AISTATS*, pages 1168–1177, 2020.

[Cesa-Bianchi et al., 2018] Nicolò Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Nonstochastic bandits with composite anonymous feedback. In *COLT*, pages 750–773, 2018.

[Desautels et al., 2014] Thomas Desautels, Andreas Krause, and Joel W. Burdick. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *Journal of Machine Learning Research*, 15(119):4053–4103, 2014.

[Dudik et al., 2011] Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. *arXiv preprint arXiv:1106.2369*, 2011.

[Joulani et al., 2013] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvari. Online learning under delayed feedback. In *ICML*, pages 1453–1461, 2013.

[Mandel et al., 2015] Travis Mandel, Yun-En Liu, Emma Brunskill, and Zoran Popović. The queue method: Handling delay, heuristics, prior data, and evaluation in bandits. In *AAAI*, volume 29, 2015.

[Manegueu et al., 2020] Anne Gael Manegueu, Claire Vernade, Alexandra Carpentier, and Michal Valko. Stochastic bandits with arm-dependent delays. In *ICML*, pages 3348–3356, 2020.

[Neu et al., 2013] Gergely Neu, András György, Csaba Szepesvari, and Andras Antos. Online markov decision processes under bandit feedback. *IEEE Transactions on Automatic Control*, 59(3):676–691, 2013.

[Nuara et al., 2018] Alessandro Nuara, Francesco Trovò, Nicola Gatti, and Marcello Restelli. A combinatorial-bandit algorithm for the online joint bid/budget optimization of pay-per-click advertising campaigns. In *AAAI*, volume 32, 2018.

[Nuara et al., 2022] Alessandro Nuara, Francesco Trovò, Nicola Gatti, and Marcello Restelli. Online joint bid/daily budget optimization of internet advertising campaigns. *Artificial Intelligence*, page 103663, 2022.

[Pike-Burke et al., 2018] Ciara Pike-Burke, Shipra Agrawal, Csaba Szepesvari, and Steffen Grunewalder. Bandits with delayed, aggregated anonymous feedback. In *ICML*, pages 4105–4113, 2018.

[Thune et al., 2019] Tobias Sommer Thune, Nicolò Cesa-Bianchi, and Yevgeny Seldin. Nonstochastic multiarmed bandits with unrestricted delays. In *NeurIPS*, 2019.

[Trovò et al., 2018] Francesco Trovò, Stefano Paladino, Marcello Restelli, and Nicola Gatti. Improving multiarmed bandit algorithms in online pricing settings. *International Journal of Approximate Reasoning*, 98:196–235, 2018.

[van der Hoeven and Cesa-Bianchi, 2021] Dirk van der Hoeven and Nicolò Cesa-Bianchi. Nonstochastic bandits and experts with arm-dependent delays. *arXiv preprint arXiv:2111.01589*, 2021.

[Vernade et al., 2017] Claire Vernade, Olivier Cappé, and Vianney Perchet. Stochastic bandit models for delayed conversions. In *UAI*, 2017.

[Vernade et al., 2020a] Claire Vernade, Alexandra Carpentier, Tor Lattimore, Giovanni Zappella, Beyza Ermis, and Michael Brueckner. Linear bandits with stochastic delayed feedback. In *ICML*, pages 9712–9721, 2020.

[Vernade et al., 2020b] Claire Vernade, Andras Gyorgy, and Timothy Mann. Non-stationary delayed bandits with intermediate observations. In *ICML*, pages 9722–9732, 2020.

[Zhou et al., 2019] Zhengyuan Zhou, Renyuan Xu, and Jose Blanchet. Learning in generalized linear contextual bandits with stochastic delays. In *NeurIPS*, volume 32, 2019.
Appendix of the Paper “Multi-Armed Bandit Problem with Temporally-Partitioned Rewards: When Partial Feedback Counts”

Algorithm 3 UCB1

1: for $t \in \{1, \ldots, K\}$ do
2:  Pull arm $i_t = t$ \hspace{1cm} \Comment{init phase}
3:  Observe the reward $r^i_t$ of the arm pulled at round $t$
4:  for $t \in \{K + 1, \ldots, T\}$ do
5:    for $i \in \{1, \ldots, K\}$ do
6:      \hspace{0.5cm} $\hat{R}^i_{t-1} \leftarrow \frac{1}{n^i_{t-1}} \sum_{h=1}^{t-1} r^i_h 1_{\{i_h = i\}}$
7:      \hspace{0.5cm} $c^i_{t-1} \leftarrow \frac{\hat{R}^i_t}{\sqrt{\frac{2 \ln n^i_{t-1}}{n^i_{t-1}}}}$
8:      \hspace{0.5cm} $u^i_{t-1} \leftarrow \hat{R}^i_{t-1} + c^i_{t-1}$
9:    Pull arm $i_t = \arg\max_{i \in [K]} u^i_{t-1}$
10: Observe the reward $r^i_t$ of the arm pulled at round $t$

A Baseline Algorithms Description

In this section, we report the details about the algorithms from the literature which we use as baselines in the experiments of Section 5. In particular, we compare the performances of the proposed TP-UCB-FR and the TP-UCB-EW with those of the baselines UCB1, assuming to obtain all the rewards corresponding to the pull of arm $i_t$ at time $t$, and Delayed-UCB1, which uses the realization of the pulls only when they are complete, i.e., with a constant delay of $\tau_{\max} - 1$.

A.1 Non-Delayed Feedback

We describe the version of the UCB1 algorithm, designed by Auer et al. [2002], in which the reward $r^i_t$ provided by pulling arm $i_t$ at round $t$ is observed by the learner at time $t$. We recall that this algorithm cannot be run in a TP-MAB setting, unless we are in the degenerate case $\tau_{\max} = 1$. It rather represents a clairvoyant algorithm having the information of the rewards $r^i_t$ without any delay. We denote its policy by $\Pi_{\text{UCB1}}$.

The pseudo-code of the UCB1 algorithm is reported in Algorithm 3. During the initialization phase, all the arms are pulled once (Line 2). Subsequently, at each round $t$, the learner computes the empirical mean of the cumulative rewards $\hat{R}^i_{t-1}$ collected up to round $t - 1$ (Line 6), where we denote by $n^i_{t-1} := \frac{1}{n^i} \sum_{h=1}^{t-1} 1_{\{i_h = i\}}$ the number of times the arm $i$ has been pulled up to round $t - 1$, and the confidence interval $c^i_{t-1}$ (Line 7). Finally, the learner pulls the arm with the largest upper confidence bound $u^i_{t-1}$ (Line 9), and observes the reward $r^i_t$ (Line 10).

We provide the following upper bound on the regret of the UCB1 algorithm (see the proof by [Auer et al., 2002]):

**Theorem 5.** The pseudo-regret of UCB1 after $T \in \mathbb{N}^+$ rounds on a MAB problem with $r^i_t$ rewards is:

$$R_T(\Pi_{\text{UCB1}}) \leq \sum_{i: \mu_i < \mu^*} 8(\overline{R})^2 \ln T + \left(1 + \frac{\pi^2}{3}\right) \sum_{i: \mu_i < \mu^*} \Delta_i.$$

A.2 Delayed Feedback

We show how to apply the Delayed-UCB1 algorithm, provided by Joulan et al. [2013] and originally designed for the Delayed-MAB setting, to the TP-MAB setting. In the TP-MAB problem, the realization of the cumulative reward $r^i_t$ is observed after $\tau_{\max} - 1$ rounds from the pull of the arm. As a consequence, one always waits for $\tau_{\max} - 1$ rounds before collecting the reward from a pull. This approach, corresponds to a delayed-feedback MAB setting in which the delay is known and deterministic. After such a delay, the learner updates the policy $\Pi_{\text{UCB1}}$ of Delayed-UCB1 with the value of the cumulative reward.

The pseudo-code of the Delayed-UCB1 algorithm applied to a generic TP-MAB setting is reported in Algorithm 4. During the initialization phase, all arms are pulled in a round robin fashion until at least one reward is collected (Line 2). Subsequently, at each round $t$, the learner computes the empirical mean $\hat{R}^i_{t-1}$ of the cumulative rewards collected up to round $t - 1$ (Line 5), where $s^i_{t-1} := \sum_{h=1}^{t-1} 1_{\{i_h = i\}}$ is the number of complete reward observed so far for arm $i$, and the confidence interval $c^i_{t-1}$ (Line 6). Finally, the learner pulls the arm with the largest upper confidence bound $u^i_{t-1}$ (Line 8), and observes the reward corresponding to the pull occurred at round $t - \tau_{\max} + 1$ (Line 9). When no sample is available for an arm $i$ its upper bound is set to $+\infty$. We provide the following upper bound on the regret of the Delayed-UCB1 algorithm (see Joulan et al. [2013]).
Algorithm 4 Delayed-UCB1

1: for $t \in \{1, \ldots, \tau_{\text{max}}\}$ do
2: Pull arm $i_t = ((t - 1) \mod K) + 1$
3: for $t \in \{\tau_{\text{max}} + 1, \ldots, T\}$ do
4: for $i \in \{1, \ldots, K\}$ do
5: $\hat{R}_i^{t-1} \leftarrow \frac{1}{s_i^{t-1}} \sum_{h=1}^{s_i^{t-1}} r_h^{t}\mathbb{1}_{\{i_t = i\}}$
6: $c_i^{t-1} \leftarrow \sqrt{\frac{2\ln(t-1)}{s_i^{t-1}}}$
7: $u_i^{t-1} \leftarrow \hat{R}_i^{t-1} + c_i^{t-1}$
8: Pull arm $i_t = \arg\max_{i \in [K]} u_i^{t-1}$
9: Observe reward $r_{i_t}^{t-\tau_{\text{max}}+1}$ of the arms pulled at round $t - \tau_{\text{max}} + 1$

Table 3: Summary of the known theoretical results (in black) and original contributions provided in this paper (in blue).

| Lower bound | MAB | Delayed-MAB | TP-MAB with $\alpha$-smoothness |
|-------------|-----|-------------|---------------------------------|
| $\sum_{i} C_i \ln T + \left(1 + \frac{\tau_{\text{max}}}{\tau_{\text{max}}}\right) \sum_{i} \Delta_i$ | $\sum_{i} \frac{1}{K} C_i \ln T - \frac{\Delta_i}{\tau_{\text{max}}} \ln T$ | $\sum_{i} \alpha KL\left(\frac{\mu_i - \mu^*}{\tau_{\text{max}}} / \frac{\tau_{\text{max}}}{\tau_{\text{max}}}\right)$ | $\sum_{i} \alpha KL\left(\frac{\mu_i - \mu^*}{\tau_{\text{max}}} / \frac{\tau_{\text{max}}}{\tau_{\text{max}}}\right)$ |
| $\sum_{i} C_i \ln T + \left(1 + \frac{\tau_{\text{max}}}{\tau_{\text{max}}}\right) \sum_{i} \Delta_i$ | $\sum_{i} C_i \ln T + \left(1 + \frac{\tau_{\text{max}}}{\tau_{\text{max}}}\right) \sum_{i} \Delta_i$ | $\sum_{i} C_i \ln T + \left(1 + \frac{\tau_{\text{max}}}{\tau_{\text{max}}}\right) \sum_{i} \Delta_i$ | $\sum_{i} C_i \ln T + \left(1 + \frac{\tau_{\text{max}}}{\tau_{\text{max}}}\right) \sum_{i} \Delta_i$ |
| $\sum_{i} C_i \ln T + \left(1 + \frac{\tau_{\text{max}}}{\tau_{\text{max}}}\right) \sum_{i} \Delta_i$ | $\sum_{i} C_i \ln T + \left(1 + \frac{\tau_{\text{max}}}{\tau_{\text{max}}}\right) \sum_{i} \Delta_i$ | $\sum_{i} C_i \ln T + \left(1 + \frac{\tau_{\text{max}}}{\tau_{\text{max}}}\right) \sum_{i} \Delta_i$ | $\sum_{i} C_i \ln T + \left(1 + \frac{\tau_{\text{max}}}{\tau_{\text{max}}}\right) \sum_{i} \Delta_i$ |

Theorem 6. The pseudo-regret of Delayed-UCB1 after $T \in \mathbb{N}^*$ rounds in the TP-MAB setting is:

$$R_T(\mathcal{U}_{\text{UCB1}}) \leq \sum_{i: \mu_i < \mu^*} \frac{8(\hat{R}_i^2)\ln T}{\Delta_i} + \left(1 + \frac{\pi^2}{3} + \tau_{\text{max}}\right) \sum_{i: \mu_i < \mu^*} \Delta_i. \quad (9)$$

Proof. The theorem follows from Theorem 7 by [Joulan et al., 2013], where the expected value of the maximum number of missing feedback of arm $i$ during the first $t$ time steps is $\mathbb{E}[G_{i,t}^t] < \tau_{\text{max}}$, where $G_{i,t}^t$ is the maximum number of missing feedbacks during the first $t$ rounds for arm $i$. \hfill \square

A.3 Theoretical Results Summary

Finally, we provide a table summarizing the results known in the literature and provided in this paper. Table 3 reports the lower and upper bounds on the regret for different algorithms and settings. Notice that the lower bound results hold for $T \to +\infty$. Moreover, in Table 3, we denote $\frac{8(\hat{R}_i^2)\ln T}{\Delta_i}$ by $C_i$ and $\sum_{i: \mu_i < \mu^*}$ by $\sum_{i}$. The assumption is that the instantaneous (for the MAB and Delayed-MAB settings) and cumulative (for the TP-MAB setting) rewards have support in $[0, \hat{R}_i]$. Moreover, in the Delayed-MAB setting, the maximum stochastic delay is $\tau_{\text{max}}$. The novel results have been highlighted in blue. UCB1 does not have guarantees in the Delayed-MAB and TP-MAB settings since it has been developed for a more restrictive scenario, i.e., $\tau_{\text{max}} = 1$.

The results related to the proposed algorithms, i.e., TP-UCB-FR and TP-UCB-EW, for the MAB setting have been derived fixing $\tau_{\text{max}} = 1$ and $\alpha = 1$ in the corresponding theorems. The results of the Delayed-MAB setting have been derived fixing $\alpha = 1$. We remark that TP-UCB-FR in the MAB setting has the same asymptotic order of upper bound of UCB1, while the upper bound of TP-UCB-EW reduces exactly to the one of UCB1 in this setting.

B Omitted Proofs

Theorem 1. The regret of any uniformly efficient policy $\mathcal{U}$ applied to the TP-MAB problem is bounded from below by:

$$\lim_{T \to +\infty} \inf \frac{R_T(\mathcal{U})}{\ln T} \geq \sum_{i: \mu_i < \mu^*} \frac{\Delta_i}{KL\left(\frac{\mu_i}{\tau_{\text{max}}} / \frac{\mu^*}{\tau_{\text{max}}}\right)}, \quad (1)$$
where $\Delta_i := \mu^* - \mu_i$ is the expected loss suffered by the learner if the arm $i$ is chosen instead of the optimal one $i^*$, $R_{\text{max}} := \max_{i \in [K]} R_i$, and $KL(p, q)$ is the Kullback-Leibler divergence between Bernoulli r.v. with means $p$ and $q$.

Proof. At first, notice that learning the optimal arm in a TP-MAB problem $\mathcal{P}$ for rewards $R_i^t$ taking values over a generic finite domain $[0, R_i^\infty)$, having range $R_i^\infty$, is equivalent to the problem of learning in a TP-MAB problem $\mathcal{P}'$ with reward $\frac{R_i^t}{R_{\text{max}}}$ having domain $[0, 1]$. Indeed, from a learning perspective, distinguish between two arms in the first setting requires the same sample complexity of distinguish between two arms in the second one. The expected reward of the $i$-th arm of the $\mathcal{P}'$ problem is $\mu_i' = \frac{\mu_i}{R_{\text{max}}}$ and the one corresponding to the optimal arm is $\mu^* = \frac{\mu^*}{R_{\text{max}}}$. Let us remark that determining the optimality of an arm in this problem is no harder than the one in which the reward is spread over the period $t$.

Let us consider for each problem $\mathcal{P}$ in the class of TP-MAB problems, its corresponding $\mathcal{P}'$ one. For each $\mathcal{P}'$, we build a corresponding Delayed-MAB equivalent problem, by delaying all the intermediate rewards corresponding to a pull at round $t$ to the round $t + \tau_{\text{max}} - 1$. Therefore, using the results on the lower bound of the Delayed-MAB problems provided by Vernade et al. [2017] (Lemma 15) we have that:

$$\lim_{T \to +\infty} \inf_{\mathcal{P}_T} \frac{\mathbb{E}[N_i(T)]}{\log(T)} \geq \frac{1}{KL\left(\frac{\mu_i}{R_{\text{max}}}, \frac{\mu^*}{R_{\text{max}}}\right)},$$

where $\mathbb{E}[N_i(T)]$ is the expected number of times an arm $i$ is selected over a time horizon of $T$ by the policy $\mathcal{U}$. Due to the equivalence depicted above, this result holds also for the original problems $\mathcal{P}$ in the class of TP-MAB problems. From the fact that $R_{\mathcal{T}}(\mathcal{U}) = \Delta_i \mathbb{E}[N_i(T)]$ and summing over the suboptimal arms, i.e., $i \neq i^*$, we get the theorem statement.

**Theorem 2.** The regret of any uniformly efficient policy $\mathcal{U}$ applied to the TP-MAB problem with the $\alpha$-smoothness property is bounded from below by:

$$\lim_{T \to +\infty} \inf_{\mathcal{P}_T} \frac{R_{\mathcal{T}}(\mathcal{U})}{\ln T} \geq \sum_{i : \mu_i < \mu^*} \frac{\Delta_i}{\alpha KL\left(\frac{\mu_i}{R_{\text{max}}}, \frac{\mu^*}{R_{\text{max}}}\right)}.$$

Proof. The proof follows the steps provided for Theorem 2.2 in the work by Bubeck and Cesa-Bianchi [2012] and generalize them to the setting in which multiple rewards, i.e., $\alpha$, are earned by a single arm pull. Let us define an auxiliary MAB setting in which:

- only two arms with expected value $\mu_1$ and $\mu_2$, with $\mu_2 < \mu_1 < 1$;
- all the arm have maximum reward equal to $R_i^\infty = R_{\text{max}}$;
- the reward $Z_{i,k}$ are i.i.d. over $k \in \{1, \ldots, \alpha\}$, meaning that the expected value of each of the element is $\frac{\mu_i}{\alpha}$;
- the reward are $Z_{i,k} \in \{0, R_{\text{max}}\}$, i.e., the reward are Bernoulli scaled by a factor $R_{\text{max}}$;
- pulling an arm at time $t$ provides $\alpha$ reward for the arm $\{Z_{i,1}, \ldots, Z_{i,\alpha}\}$, all observed by the learner at the time of the pull.

Let us remark that determining the optimality of an arm in this problem is no harder than the one in which the reward is spread over the period $\{1, \ldots, t + \tau_{\text{max}} - 1\}$. Therefore, the derivation of a lower bound for this problem would also provide a lower bound for the original TP-MAB setting with $\alpha$-smoothness. Moreover, let us recall that learning in a problem where the reward are scaled by a factor $R_{\text{max}}$, similarly to what has been done in Theorem 1, does not change the complexity of learning. From now on, we will consider as expected value of the two arms $\mu_{Z_1} := \frac{\mu_1}{R_{\text{max}}}$ and $\mu_{Z_2} := \frac{\mu_2}{R_{\text{max}}}$. Therefore, to compute the expected value of number of times an algorithm pulls the suboptimal arm $\mathbb{E}[N_2(T)]$ we can also use the scaled rewards. In what follows, we prove that the lower bound for the auxiliary problem for any uniformly efficient policy $\mathcal{U}$.

**Overall proof idea**

Let us consider a second instance of the above defined MAB such that arm 2 is optimal and $\mu_{Z_1} < \mu_{Z_2} < 1$. We refer to it as the modified bandit. Let $\varepsilon > 0$, since $x \mapsto KL(\mu_{Z_2}, x)$ is continuous one can find $\mu_{Z_2}' \in (\mu_{Z_1}, 1)$ such that:

$$KL(\mu_{Z_2}, \mu_{Z_2}') \leq (1 + \varepsilon) KL(\mu_{Z_2}, \mu_{Z_1}).$$

In what follows, we use the notation $\mathbb{E}'$, $\mathbb{P}'$ to denote the expected value and probability computed in the second bandit instance. The goal is to compare the behavior of the forecaster on the initial and modified bandits. The idea of the proof is to show that, with a big enough probability, the forecaster is not able to distinguish between the two problems. Then, using the fact that the forecaster is uniformly efficient by hypothesis, we show that the algorithm does not make too many mistake on the modified bandit and, in particular, provide a lower bound on the number of times the optimal arm is played. This reasoning implies a lower bound on the number of times the suboptimal arm 2 is played in the initial problem.

---

9An uniformly efficient policy chooses the suboptimal arms on average $\alpha(t^\alpha)$ times ($0 < \alpha < 1$) over $t$ rounds.
First step: $\mathbb{P}(C_t) = o(1)$
Let us define, for $s \in \{1, \ldots, t\}$, the empirical estimate of $KL(\mu_{Z_2}, \mu_{Z_2}')$ at round $t$ when the arm 2 is pulled $s$ times:

$$KL_{\alpha s} := \sum_{i=1}^{s} \sum_{k=1}^{\alpha} \ln \frac{\mu_{Z_2} Z_{n,k}^{2} + (1 - \mu_{Z_2}) (1 - Z_{n,k}^{2})}{\mu_{Z_2}' Z_{n,k}^{2} + (1 - \mu_{Z_2}') (1 - Z_{n,k}^{2})}. \tag{12}$$

We introduce the following event linking the behavior of the forecaster on the initial and modified bandits:

$$C_t := \left\{ \alpha N_2(t) < f_t \quad \text{and} \quad KL_{\alpha N_2(t)} \leq (1 - \varepsilon/2) \ln t \right\}, \tag{13}$$

where $f_t = \frac{1 - \varepsilon}{KL(\mu_{Z_2}, \mu_{Z_2}')} \ln t$. Following the proof of Theorem 2.2 from Bubeck and Cesa-Bianchi [2012], we have:

$$\mathbb{P}'(C_t) = \mathbb{E}[1_{C_t} \exp \left( -KL_{\alpha N_2(t)} \right)] \geq e^{- (1 - \varepsilon/2) \ln t \mathbb{P}(C_t)}, \tag{14}$$

where we used the change of measure identity for the first equality and use the fact that $KL_{\alpha N_2(t)} \leq (1 - \varepsilon/2) \ln t$ in $C_t$.\footnote{For any event $A$ in the $\sigma$-algebra generated by $\{Z_{n,k}^{2}\}_{n \in \{1, \ldots, t\}, k \in \{1, \ldots, \alpha\}}$, holds that $\mathbb{P}(A) = \mathbb{E}\left[1_{A} \exp \left(-KL_{\alpha N_2(t)}\right)\right]$.}

Combining Equation (14), the definition of $C_t$, and using the Markov’s inequality, we have:

$$\mathbb{P}(C_t) \leq t(1 - \varepsilon/2) \mathbb{P}'(C_t) \leq t(1 - \varepsilon/2) \mathbb{P}(\alpha N_2(t) < f_t) \leq t^{(1 - \varepsilon/2)} \mathbb{E}'[\hat{t} - N_2(t)] = o(1), \tag{15}$$

where with $o(1)$ we denote a quantity whose limit for $t \to +\infty$ is 0 and we used the fact that the policy $\U$ is uniformly efficient, i.e., $\mathbb{E}'[T_2(t)] = o(t^{\beta})$ with $\beta < 1$.

Second step: $\mathbb{P}(\alpha N_2(t) \leq f_t) = o(1)$
Using the Third step of the proof of Theorem 2.2 from Bubeck and Cesa-Bianchi [2012], we get:

$$o(1) = \mathbb{P}(C_t) \leq \mathbb{P} \left( \alpha T_2(t) < f_t \wedge E_1 \setminus E_t \wedge \max_{s \leq f_t/\alpha} KL_{\alpha s} \leq \frac{1 - \varepsilon/2}{1 - \varepsilon} KL(\mu_{Z_2}, \mu_{Z_2}') \right). \tag{16}$$

Using the strong law of large numbers the event $E_2$ is s.t. $\lim_{t \to +\infty} \mathbb{P}(E_2) = 1$, we infer that $\mathbb{P}(E_1) = \mathbb{P}(\alpha N_2(t) < f_t) = o(1)$, and that for $t \to +\infty$ we have $\mathbb{E}[N_2(t)] > f_t/\alpha$.

Final step
Using Equation (11) we have that, for $t \to +\infty$:

$$\mathbb{E}[N_2(t)] > f_t/\alpha = \frac{1 - \varepsilon}{\alpha KL(\mu_{Z_2}, \mu_{Z_2}') \ln t} \geq \frac{1 - \varepsilon}{\alpha(1 + \varepsilon) KL(\mu_{Z_2}, \mu_{Z_2}') \ln t}, \tag{17}$$

where the theorem statement follows from the arbitrariness of the value of $\varepsilon$, substituting $\mu_{Z_2}$ with $\mu_{\hat{\pi}, \max}$ and $\mu_{Z_2}$ with $\mu_{2, \max}$, and summing over all the suboptimal arms.

\[\square\]

Theorem 3. \textit{In the TP-MAB setting with $\alpha$-smooth reward, the pseudo-regret of TP-UCB-FR after $T$ rounds is:}

$$R_T(\U_{\text{FR}}) \leq \sum_{i: \mu_i < \mu_*} 4\left(\frac{\tilde{R}}{\alpha \Delta_i} \right)^2 \ln T \left(1 + \sqrt{1 + \frac{\alpha(\alpha + 1)\phi \Delta_i}{2\tilde{R} \ln T}} \right) + (\alpha + 1) \phi \sum_{i: \mu_i < \mu_*} \frac{\tilde{R}^i}{3} \sum_{\Delta_i},$$

\textit{Proof.} Let us define the true empirical mean of the cumulative reward of arm $i$ computed over $n_i^t$ samples as follows:

$$\hat{R}_i^t := \frac{1}{n_i} \sum_{h=1}^{t} \sum_{j=1}^{\tau_{i}^{h} \mathbb{1}_{(i_h = i)}} x_{i,j}.$$ 

We aim to bound the difference between $\hat{R}_i^t$ and the approximated empirical mean of the cumulative reward $\tilde{R}_i^t$ from arm $i$ computed over $n_i^t$ samples as in the TP-UCB-FR algorithm. Formally, we have:

$$\hat{R}_i^t - \tilde{R}_i^t = \frac{1}{n_i} \sum_{h=1}^{t} \sum_{j=1}^{\tau_{i}^{h} \mathbb{1}_{(i_h = i)}} x_{i,j} - \hat{R}_i^t = \frac{1}{n_i} \sum_{h=1}^{t} \sum_{j=1}^{\tau_{i}^{h} \mathbb{1}_{(i_h = i)}} x_{i,j} \quad \text{and} \quad \hat{R}_i^t - \tilde{R}_i^t = \frac{1}{n_i} \sum_{h=1}^{t} \sum_{j=1}^{\tau_{i}^{h} \mathbb{1}_{(i_h = i)}} x_{i,j} \quad \text{and} \quad \hat{R}_i^t - \tilde{R}_i^t = \frac{1}{n_i} \sum_{h=max(1, t+\tau_{i}^{h}+2)}^{t} \sum_{j=1}^{\tau_{i}^{h} \mathbb{1}_{(i_h = i)}} x_{i,j} \quad \text{and} \quad \hat{R}_i^t - \tilde{R}_i^t = \frac{1}{n_i} \sum_{h=max(1, t+\tau_{i}^{h}+2)}^{t} \sum_{j=1}^{\tau_{i}^{h} \mathbb{1}_{(i_h = i)}} x_{i,j}.$$ 

(18)
\[
\leq \frac{1}{n_t} \sum_{j=1}^{\alpha} \frac{R_i^j}{\alpha} = \frac{\phi}{n_t^i} \sum_{j=1}^{\alpha} j = \frac{\phi}{n_t^i} \frac{R_i^j (\alpha + 1)}{2} = \frac{\hat{R}(\alpha + 1) \phi}{2n_t^i},
\]

where, Equation (19) is due to the fact that \( R_i^j = 0 \) for each \( i \in [K] \), and the inequality in Equation (20) is due to the \( \alpha \)-smoothness of the environment.

Following the proof of Theorem 1 by Auer et al. [2002], we bound the expected number of time a suboptimal arm is pulled as follows:

\[
\mathbb{E}[N_i(t)] \leq \ell + \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{i,s=t}^{t+1} \mathbb{P}\{ (\hat{R}_{t,s}^i + c_{t,s}) \leq (\hat{R}_{t,s}^i + c_{t,s}) \},
\]

where \( \hat{R}_{t,s}^i \) and \( c_{t,s}^i \) are the empirical mean computed as in the TP-UCB-FR algorithm and the confidence bound, respectively.

Equation (22) implies that at least one of the following holds:

\[
\hat{R}_{t,s}^i \leq \mu^* - c_{t,s}^i, \quad \hat{R}_{t,s}^i \geq \mu_i + c_{t,s}^i, \quad \mu^* < \mu_i + 2c_{t,s}^i.
\]

Let us focus on Equation (23). We have that:

\[
\mathbb{P}\{ \hat{R}_{t,s}^i - \mu^* \leq -c_{t,s}^i \} = \mathbb{P}\{ \hat{R}_{t,s}^i - \mu^* \leq -c_{t,s}^i + \hat{R}_{t,s}^i - \hat{R}_{t,s}^i \}
\]

\[
\leq \mathbb{P}\{ \hat{R}_{t,s}^i - \mu^* \leq -c_{t,s}^i + \hat{R}_{t,s}^i - \hat{R}_{t,s}^i \}
\]

\[
\leq \exp \left\{ \frac{2 (\hat{R}^i \sqrt{\frac{2 \ln t}{\alpha s}})^2}{s^2} \sum_{t=1}^{\frac{2 \ln t}{\alpha s}} \left( \frac{\hat{R}_i}{\alpha} \right)^2 \right\} \leq e^{-4 \ln t} \leq t^{-4},
\]

where \( c_{t,s}^i := \hat{R}^i \sqrt{\frac{2 \ln t}{\alpha s}} + \frac{\pi (\alpha + 1)}{2s} \hat{R}^i \hat{R}_{t,s}^i \), \( \hat{R}_{t,s}^i \) is the empirical mean of the optimal arm \( i^* \) in the case \( s \) pulls occurred in the first \( t \) rounds, and we use the Hoeffding inequality in Equation (28).

Similarly, Equation (24) is bounded by:

\[
\mathbb{P}\{ \hat{R}_{t,s}^i - \mu_i \geq c_{t,s}^i \} \leq \mathbb{P}\{ \hat{R}_{t,s}^i - \mu_i \geq \frac{\hat{R}_i}{\alpha s} \}
\]

\[
\leq e^{-4 \ln t} = t^{-4},
\]

where we used the fact that \( \hat{R}_{t,s}^i \geq \hat{R}_i^i \) by construction of the latter, and we used the Hoeffding inequality to derive Equation (30).

Define:

\[
\ell := \left[ \frac{\hat{R}_i (\alpha + 1) \phi}{\Delta_i} + \frac{4(\hat{R}_i)^2 \ln t}{\alpha \Delta_i^2} \right] \left( 1 + \sqrt{1 + \frac{\alpha (\alpha + 1) \phi}{2 \hat{R}_i \ln t}} \right).
\]

We have that the following holds:

\[
\mu^* \geq \mu_i + 2c_{t,s}^i
\]

\[
\Delta_i \geq \frac{2}{\hat{R}_i} \sqrt{\frac{2 \ln t}{\alpha s}} + \frac{\hat{R}_i (\alpha + 1)}{2s}
\]

\[
s_i^2 \left( \frac{\Delta_i^2}{4} \right) - 2s_i \left( \frac{\Delta_i \hat{R}_i (\alpha + 1)}{4} + \frac{(\hat{R}_i)^2 \ln t}{\alpha} \right) + \phi \frac{(\hat{R}_i)^2 (\alpha + 1)^2}{4} \geq 0
\]
and, therefore, for \( s_i \geq \ell \) the inequality in Equation (25) is always false.

Finally, summing up the results derived above and using \( \ell \) as defined in Equation (31), we have:

\[
E[N_i(t)] \leq \left[ \frac{\mathcal{R}(\alpha + 1)}{\Delta_i} + \frac{4(\mathcal{R})^2 \ln t}{\alpha \Delta_i^2} \left( 1 + \frac{1}{1 + \frac{\Delta_i(\alpha + 1)\mathcal{R}}{2\mathcal{R}^2 \ln t}} \right) \right] + \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{s_i=1}^{s} \left[ \mathbb{P}\left( \hat{R}^\ast_{t,s} - \mu_i \leq -c_i^\ast_{t,s} \right) + \mathbb{P}\left( \hat{R}^\ast_{t,s} - \mu_i \geq c_i^\ast_{t,s} \right) \right].
\]

The theorem statement follows by the fact that \( \mathcal{R}_T(U_{\text{EW}}) = \sum_{i:\mu_i < \mu^\ast} \Delta_i E[N_i(T)] \).

**Theorem 4.** In the TP-MAB setting with \( \alpha \)-smooth reward, the pseudo-regret of TP-UCB-EW after \( T \) rounds is:

\[
\mathcal{R}_T(U_{\text{EW}}) \leq \sum_{i:\mu_i < \mu^\ast} \frac{8(\mathcal{R})^2 \ln T}{\Delta_i} + \alpha \left( \frac{\pi^2}{3} \right) \sum_{i:\mu_i < \mu^\ast} \Delta_i.
\]

**Proof.** Following the same proof strategy of Theorem 3, we want to bound the expected value of the number of pulls of suboptimal arms:

\[
E[N_i(t)] \leq l + \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{s_i=1}^{s} \mathbb{P}\left( \sum_{k=1}^{\alpha} (\tilde{Z}^i_{t,k,s} + c_i^\ast_{t,k,s}) \leq \sum_{k=1}^{\alpha} (\tilde{Z}^i_{t,k,s} + c_i^\ast_{t,k,s}) \right),
\]

where \( \tilde{Z}^i_{t,k,s} \) and \( c_i^\ast_{t,k,s} \) are the empirical mean computed as in the TP-UCB-EW algorithm and the confidence bound, respectively, of the optimal arm \( i^\ast \) in the case \( s \) pulls occurred in the first \( t \) rounds, and, \( \tilde{Z}^i_{t,k,s} \) and \( c_i^\ast_{t,k,s} \) are the empirical mean computed as in the TP-UCB-EW algorithm and the confidence bound, respectively, of the arm \( i \) in the case \( s_i \) pulls occurred in the first \( t \) rounds. Notice that in this case the number of samples collected from each one of the \( \alpha \) aggregated rewards are \( \leq s \) and \( \leq s_i \), respectively. Moreover, for values of \( t > t_{\max} \) the quantities regarding the suboptimal arm are estimated using at least one sample, e.g., \( 0 < n^i_{t,k,s} < s_i \). Conversely, for \( s \leq t_{\max} \) the optimal arm might have no sample available to estimate the expected value and the bound. However, since the values of the upper confidence bound is set \( + \infty \) if no sample is collected, the probability that it is smaller than the one of a suboptimal arm is 0, (i.e., \( \mathbb{P}\left( \sum_{k=1}^{\alpha} (\tilde{Z}^i_{t,k,s} + c_i^\ast_{t,k,s}) \leq \sum_{k=1}^{\alpha} (\tilde{Z}^i_{t,k,s} + c_i^\ast_{t,k,s}) \right) = 0 \).

As a consequence, the cases in which no sample is available for the optimal bound can be disregarded.

The condition above is satisfied if at least one of the following \( 2\alpha + 1 \) inequalities holds:

\[
\begin{align*}
\tilde{Z}^i_{t,k,s} - \mu^i_k & \leq -c_i^\ast_{t,k,s}, & \forall k \in \{1, \ldots, \alpha\} \\
\tilde{Z}^i_{t,k,s} - \mu_{i,k} & \geq c_i^\ast_{t,k,s}, & \forall k \in \{1, \ldots, \alpha\} \\
\sum_{k=1}^{\alpha} \mu^i_k - 2c_i^\ast_{t,k,s} & < 0,
\end{align*}
\]

where \( \mu_{i,k} := E[Z^i_{t,k,s}] \) and \( \mu^i_k := E[Z^i_{t,k,s}] \) are the expected value of the aggregated reward \( Z^i_{t,k,s} \) from arm \( i \), and \( Z^i_{t,k,s} \) from the optimal arm, respectively.

Let us focus on the \( k \)-th inequality in Equation (37). We have:

\[
\mathbb{P}(\tilde{Z}^i_{t,k,s} - \mu^i_k \leq -c_i^\ast_{t,k,s}) \leq \exp \left\{ -\frac{2(n^i_{t,k,s})^2(c_i^\ast_{t,k,s})^2}{\sum_{l=1}^{n^i_{t,k,s}} (\mathcal{R})^2} \right\}
\]
\[
0 \leq \frac{2n_{t,k,s}^i(c_{t,k,s}^i)^2\alpha^2}{\left(R_i\right)^2} \leq e^{-4\ln t} \leq t^{-4},
\]
where \(n_{t,k,s}^i\) is the number of samples available for the estimation of the expected value of \(Z_{t,k,s}^i\) if we pulled \(s\) times the arm \(i^*\) at round \(t\). Here, we assume that the estimates have at least one sample. If no samples are available, the original probability in Equation (36) is bounded by 0.

Similarly, for the inequalities in Equation (38), we have:

\[
\Pr(Z_{t,k,s}^i - \mu_{i,k} \geq c_{t,k,s}^i) \leq \exp\left\{-\frac{2(n_{t,k,s}^i)^2(c_{t,k,s}^i)^2}{\sum_{t=1}^{n_{t,k,s}^i} \left(\frac{R_i}{\alpha}\right)^2}\right\} \leq e^{-4\ln t} \leq t^{-4}.
\]

where \(n_{t,k,s}^i\) is the number of samples available for the estimation of the expected value of \(Z_{t,k,s}^i\) if we pulled \(s\) times the arm \(i\) at round \(t\).

Define \(l = \left[\alpha \phi - 1 + \frac{8(R_i)^2 \ln t}{\Delta_i^2}\right]\). Notice that \(l \geq \tau_{\max}\). We have that the inequality in Equation (39) is false. Indeed, we have that:

\[
\sum_{k=1}^{\alpha} \left(\mu_k^* - \mu_{i,k} - \frac{2(R_i)^2}{\Delta_i^2} \sum_{t=1}^{n_{t,k,s}^i} \frac{2\ln t}{s_t - \alpha \phi + 1}\right) \geq \Delta_i - 2\frac{R_i}{\alpha} \sqrt{\frac{2\ln t}{s_i - \alpha \phi + 1}} = \Delta_i - 2\frac{R_i}{\alpha} \sqrt{\frac{2\ln t}{s_i - \alpha \phi + 1}},
\]

where we used that \(\sum_{k=1}^{\alpha} \mu_k^* - \mu_{i,k} = \mu^* - \mu_i = \Delta_i\).

If \(s_i \geq \alpha \phi - 1 + \frac{8(R_i)^2 \ln t}{\Delta_i^2}\), we have that:

\[
s_i \geq \alpha \phi - 1 + \frac{8(R_i)^2 \ln t}{\Delta_i^2}
\]

\[
\frac{\Delta_i^2}{4(R_i)^2} \geq \frac{2\ln t}{s_i - \alpha \phi + 1}
\]

\[
\Delta_i - 2\frac{R_i}{\alpha} \sqrt{\frac{2\ln t}{s_i - \alpha \phi + 1}} \geq 0,
\]

which implies that the inequality in Equation (43) is false.

Finally, summing the above results we have that:

\[
\mathbb{E}[N(t)] \leq \left[\alpha \phi - 1 + \frac{8(R_i)^2 \ln(t)}{\Delta_i^2}\right] + \sum_{i=1}^{\alpha} \sum_{s=1}^{n_{i,k,s}^i} \sum_{k=1}^{\alpha} \Pr(Z_{t,k,s}^i - \mu_{i,k} \leq -c_{t,k,s}^i) + \Pr(Z_{t,k,s}^i - \mu_{i,k} \geq c_{t,k,s}^i)
\]

\[
\leq \alpha \phi + \frac{8(R_i)^2 \ln(t)}{\Delta_i^2} + \sum_{i=1}^{\alpha} \sum_{s=1}^{n_{i,k,s}^i} 2\alpha t^{-4}
\]

\[
\leq \frac{8(R_i)^2 \ln(t)}{\Delta_i^2} + \alpha \left[\phi + \frac{\pi^2}{3}\right].
\]

Recalling that \(\mathcal{R}_T(U_{\text{exp}}) = \sum_{i: \mu_i < \mu^*} \Delta_i \mathbb{E}[N_i(T)]\), concludes the proof.
Figure 4: Experiments for Setting #2 and uniform reward distribution: (a) $\tau_{\text{max}} = 100$, $\alpha = 10$, (b) $\tau_{\text{max}} = 200$, $\alpha = 20$.

C Experimental Settings Description and Additional Experiments

C.1 Technical Details

The code has been run on a server equipped with Intel(R) Xeon(R) CPU E5 - 4610 v2 @ 2.30GHz and 126 GiB of memory. The operating system was Ubuntu 16.04.3 LTS, and the experiments have been run on Python 3.5.2. The libraries used in the experiments, with the corresponding version were:

- numpy == 1.11.3
- tqdm == 4.14.0
- scipy == 0.18.1
- pandas == 0.20.3
- matplotlib == 3.3.4
- tikzplotlib == 0.9.8

For the experiments the total time spent was $\approx 468$ hours, where the generation of the parameters of the synthetic dataset took $\approx 27$ hours, the execution of the algorithms for Setting #1 $\approx 50$ hours, the execution of the algorithms for Setting #2 $\approx 320$ hours, the execution of the algorithms for Spotify Setting $\approx 5$ hours (considering the data preprocessing operations), the execution of the algorithms for Setting #4, presented in what follows, $\approx 66$ hours.

C.2 Experimental Settings

In what follows, we provide a detailed description of those setting which have been presented in Section 5 and further experiments confirming what has been showed in the main paper.

Setting #2 (main paper scenario) In this setting, each arm is described by a maximum reward $R_i$ and two vectors $a_i := [a_{i1}, \ldots, a_{i\alpha}]$ and $b_i := [b_{i1}, \ldots, b_{i\alpha}]$ of length $\alpha$. The aggregated reward $Z_{i,t,k}$ are distributed as $D_k = \frac{R_i}{\alpha} Beta(a_{ik}, b_{ik})$, $\forall k \in [\alpha]$. The results presented in the main paper are those corresponding to $a_i := 1_{\alpha}$ and $b_i := 1_{\alpha}$, where $1_{\alpha}$ is a vector of length $\alpha$ whose elements are all 1. This setting corresponds to a uniform distribution over $\frac{R_i}{\alpha}$ for each variable $Z_{i,t,k}$. The corresponding results are presented in Section 5. The regret over the entire time horizon is presented in Figure 4.

Spotify Setting The original Spotify dataset [Brost et al., 2019] consists of listening sessions with levels of appreciation for each song associated to a user on the Spotify service. Each listening session is truncated to 20 tracks (songs). Each row corresponds to the playback of one track pertaining to a specific listening session. The dataset describes how users sequentially interact with the streamed content they are presented with. More precisely, it contains information about when a user skips the playback of a track.

We preprocessed the available data as follows. At first, for computational reasons we analysed only a fraction of the Spotify dataset. Since we are interested in the listening sessions linked to a playlist, from that initial dataset we drop all the data associated with a context_type field context which is different from editorial playlist. Moreover, we discarded all the listening sessions with less than 20 songs and/or the user changed playlist during a single listening session (context_switch = true). This way, each listening session is composed of 20 song coming from a single playlist. We selected the 6 most
Table 4: Description of the arms in the Spotify Setting.

| i = 1 | i = 2 | i = 3 | i = 4 | i = 5 | i = 6 |
|-------|-------|-------|-------|-------|-------|
| \(\mu^i\) | 38.59 | 52.35 | 38.44 | 43.89 | 23.48 | 36.20 |
| \(\sigma^i\) | 21.83 | 20.11 | 23.09 | 23.14 | 23.48 | 23.8 |

listened playlists having no overlapping songs, and extracted from the dataset the listening sessions corresponding to them. The final dataset is available in the file `Spotify/spotifydf_012.csv` in the code provided in the supplementary material.

The process of recommending the playlists is modeled as follows.

**Example 2** (Playlist Recommendation Problem - Reprise). When a new user accesses the system, a playlist is proposed. This action corresponds to the selection of an arm \(i\) by the recommendation algorithm. The user will start the reproduction of the playlist, composed of exactly \(N = 20\) songs. For each song, at any time, the agent could decide to skip to the next song until the end of the playlist. We aim at finding the playlist that maximizes the overall listening time. Each song has a reward \(\tau\), representing increasing level of interest from the user. These levels correspond to the the realization of instantaneous reward \(X_{t,j}\) of Bernoulli r.v. that takes the value of 1 if the user has reached at least the corresponding level and 0 otherwise; The vector \(X^i_t\) has size equal to the number of songs of a playlist (i.e., aggregated rewards) times the number instant rewards returned by a song (i.e., \(\phi\)), and in this case \(\tau_{\text{max}} = 20 \times 4 = 80\). A summary of the expected rewards of the different playlists is provided in Table 4. Figure 5 shows an example of the reproduction of part of 5 songs of a playlist. Songs 1 and 3 were listened completely, while Song 2 was listened up to level the \(\text{skip}_2\). Song 4 and Song 5 were entirely skipped.

### C.3 Additional Experiments

#### Setting #2.1

In this experiment, the setting is the same as the one in Setting #2, except that we designed the rewards s.t. the first aggregated rewards after the pull are smaller than the last ones. Specifically, the distribution are defined by the following vectors:

- \(\tau_{\text{max}} = 100, \alpha = 10:\)
  \[
  a^i = [2, 4, 6, 8, 10, 10, 10, 10, 10, 10];
  \]
  \[
  b^i = [10, 10, 10, 10, 10, 10, 8, 6, 4, 2];
  \]

- \(\tau_{\text{max}} = 200, \alpha = 20:\)
  \[
  a^i = [2, \ldots, 18, 20, \ldots, 20];
  \]
  \[
  b^i = [20, \ldots, 20, 18, \ldots, 4, 2];
  \]

- \(\tau_{\text{max}} = 100, \alpha = 50:\)
  \[
  a^i = [2, \ldots, 48, 50, \ldots, 50];
  \]
  \[
  b^i = [50, \ldots, 50, 48, \ldots, 4, 2];
  \]

- \(\tau_{\text{max}} = 200, \alpha = 100:\)
  \[
  a^i = [2, \ldots, 98, 100, \ldots, 100];
  \]
  \[
  b^i = [100, \ldots, 100, 98, \ldots, 4, 2];
  \]

The corresponding results are provided in Figure 6. They are in line with the ones of Setting #2.

#### Setting #2.2

In this experiment, the setting is the same as the one in Setting #2, except that we designed the rewards s.t. the first aggregated rewards after the pull are larger than the last ones. Specifically, the distribution are defined by the following vectors:

- \(\tau_{\text{max}} = 100, \alpha = 10:\)
  \[
  a^i = [10, 10, 10, 10, 10, 10, 8, 6, 4, 2];
  \]
  \[
  b^i = [2, 4, 6, 8, 10, 10, 10, 10, 10, 10];
  \]

- \(\tau_{\text{max}} = 200, \alpha = 20:\)
  \[
  a^i = [20, \ldots, 20, 18, \ldots, 4, 2];
  \]
  \[
  b^i = [2, 4, \ldots, 18, 20, \ldots, 20];
  \]
$\tau_{\text{max}}$ = 100, $\alpha$ = 10:

\[ a^i = [50, \ldots, 50, 48, \ldots, 4, 2]; \]
\[ b^i = [2, 4, \ldots, 48, 50, \ldots, 50]; \]

$\tau_{\text{max}}$ = 200, $\alpha$ = 100:

\[ b^i = [100, \ldots, 100, 98, \ldots, 4, 2]; \]
\[ a^i = [2, 4, \ldots, 98, 100, \ldots, 100]. \]

The corresponding results are provided in Figure 7. They are in line with the ones of Setting #2.

**Setting #2.3** Finally, in this experiment, the setting is the same as the one in Setting #2, except that the reward distributions are randomly chosen.

Specifically, the distribution sampled used in the experiments are:

$\tau_{\text{max}}$ = 100, $\alpha$ = 10:

\[ a^i = [7, 7, 1, 5, 9, 8, 7, 5, 8, 6]; \]
\[ b^i = [10, 4, 9, 3, 5, 2, 3, 2, 10, 5, 9]; \]

$\tau_{\text{max}}$ = 200, $\alpha$ = 20:

\[ a^i = [10, 3, 5, 2, 6, 8, 9, 2, 6, 7, 6, 10, 4, 9, 8, 8, 9, 5, 1]; \]
\[ b^i = [9, 1, 2, 7, 1, 10, 8, 6, 4, 6, 2, 4, 10, 4, 4, 3, 9, 8, 2, 2]; \]
The overall results for the previous setting #2, #2.1, #2.2, and #2.3 are reported in Table 5, 6, 7, 8.

Summary for Setting #2

The corresponding results are provided in Figure 8. They are in line with the ones of Setting #2.

Figure 8: Experiments for Setting #2.3: (a) $\tau_{\text{max}} = 100$, $\alpha = 10$, (b) $\tau_{\text{max}} = 200$, $\alpha = 20$.

- $\tau_{\text{max}} = 100$, $\alpha = 50$:
  \[ a^i = [6, 9, 8, 2, 5, 9, 5, 2, 9, 6, 9, 4, 10, 9, 10, 5, 8, 2, 10, 7, 6, 10, 4, 5, 3, 4, 3, 1, 10, 5, 8, 2, 2, 3, 3, 1, 2, 9, 7, 9, 5, 9, 4, 4, 10, 7, 10, 5, 8, 8]; \]
  \[ b^i = [6, 2, 6, 10, 2, 8, 10, 6, 4, 4, 1, 5, 2, 4, 6, 3, 6, 7, 1, 2, 3, 4, 1, 10, 9, 10, 2, 1, 2, 4, 10, 10, 2, 7, 2, 6, 2, 1, 10, 1, 4, 3, 2, 8, 4, 1, 1, 9, 7, 10]; \]

- $\tau_{\text{max}} = 200$, $\alpha = 100$:
  \[ a^i = [2, 5, 2, 4, 2, 5, 6, 7, 3, 1, 9, 8, 1, 10, 2, 7, 4, 5, 6, 8, 10, 3, 4, 1, 3, 3, 6, 9, 5, 2, 1, 0, 8, 3, 1, 8, 7, 10, 9, 5, 6, 7, 5, 3, 9, 1, 8, 2, 6, 1, 9, 5, 3, 4, 8, 6, 10, 5, 6, 10, 10, 3, 5, 7, 2, 1, 1, 10, 4, 6, 3, 4, 4, 8, 7, 10, 7, 1, 7, 10, 7, 1, 3, 8, 2, 5, 3, 8, 9, 8, 9, 10, 1, 1, 8, 6, 5, 8, 1, 7, 4]; \]
  \[ b^i = [9, 2, 3, 1, 7, 7, 6, 1, 4, 1, 1, 9, 10, 2, 4, 2, 10, 4, 5, 5, 3, 2, 8, 7, 2, 1, 5, 8, 2, 5, 3, 9, 6, 2, 3, 5, 1, 1, 1, 4, 5, 9, 6, 6, 10, 1, 10, 8, 8, 7, 6, 9, 3, 4, 7, 10, 5, 1, 3, 3, 5, 6, 6, 6, 2, 6, 10, 1, 1, 5, 3, 3, 10, 5, 6, 7, 9, 3, 5, 2, 8, 4, 1, 5, 3, 9, 2, 5, 7, 6, 5, 7, 2, 2, 9, 8, 8, 6, 6, 2]. \]

The corresponding results are provided in Figure 8. They are in line with the ones of Setting #2.
Table 5: Summary of result for setting #2, $\tau_{\text{max}} = 100$, $\alpha = 10$.

| $\tau_{\text{max}}$ | $\alpha$ | Scenario       | Learner     | Regret     | Confidence Interval |
|----------------------|----------|----------------|-------------|-------------|---------------------|
| 100                  | 10       | TP-UCB-FR      | 379407.7536 | 641.3890868 |
| 100                  | 10       | TP-UCB-EW      | 476211.7734 | 1379.593546 |
| 100                  | 1        | Delayed-UCB1  | 550020.3093 | 3383.218936 |
| 100                  | 1        | UCB1           | 461295.3133 | 1198.377002 |
| 100                  | 2        | TP-UCB-FR      | 378590.4996 | 1444.810301 |
| 100                  | 2        | TP-UCB-EW      | 478543.3454 | 3282.169025 |
| 100                  | 2        | Delayed-UCB1  | 556264.2577 | 4563.491842 |
| 100                  | 2        | UCB1           | 464045.2915 | 3127.506071 |
| 100                  | 3        | TP-UCB-FR      | 377928.2537 | 550.2470147 |
| 100                  | 3        | TP-UCB-EW      | 477050.7314 | 1370.651113 |
| 100                  | 3        | Delayed-UCB1  | 552254.3013 | 2871.253395 |
| 100                  | 3        | UCB1           | 462051.9847 | 1022.873814 |
| 100                  | 4        | TP-UCB-FR      | 376004.9497 | 713.1333679 |
| 100                  | 4        | TP-UCB-EW      | 461523.0728 | 1159.826331 |
| 100                  | 4        | Delayed-UCB1  | 546401.0207 | 3116.186928 |
| 100                  | 4        | UCB1           | 445761.5334 | 1160.681727 |

Table 6: Summary of result for setting #2, $\tau_{\text{max}} = 200$, $\alpha = 20$.

| $\tau_{\text{max}}$ | $\alpha$ | Scenario       | Learner     | Regret     | Confidence Interval |
|----------------------|----------|----------------|-------------|-------------|---------------------|
| 200                  | 20       | TP-UCB-FR      | 1161392.507 | 653.9898656 |
| 200                  | 20       | TP-UCB-EW      | 969119.3579 | 2376.133933 |
| 200                  | 20       | Delayed-UCB1  | 1215396.1   | 11238.84718 |
| 200                  | 20       | UCB1           | 921857.7185 | 1262.074342 |
| 200                  | 2        | TP-UCB-FR      | 1159038.888 | 1855.39219 |
| 200                  | 2        | TP-UCB-EW      | 976387.8607 | 4103.790005 |
| 200                  | 2        | Delayed-UCB1  | 1214717.526 | 12958.26024 |
| 200                  | 2        | UCB1           | 922123.0453 | 3911.196296 |
| 200                  | 3        | TP-UCB-FR      | 1158406.886 | 719.1511692 |
| 200                  | 3        | TP-UCB-EW      | 971023.1429 | 2128.831649 |
| 200                  | 3        | Delayed-UCB1  | 1225998.654 | 12586.53841 |
| 200                  | 3        | UCB1           | 922097.5566 | 1084.342302 |
| 200                  | 4        | TP-UCB-FR      | 1150956.776 | 1373.38433 |
| 200                  | 4        | TP-UCB-EW      | 919231.1795 | 2971.381115 |
| 200                  | 4        | Delayed-UCB1  | 1224143.761 | 6816.67979 |
| 200                  | 4        | UCB1           | 863043.4276 | 2568.23259 |
Table 7: Summary of result for setting #2, $\tau_{\text{max}} = 100$, $\alpha = 50$.

| $\tau_{\text{max}}$ | $\alpha$ | Scenario | Learner      | Regret      | Confidence Interval |
|----------------------|----------|----------|--------------|--------------|--------------------|
| 100                  | 50       | 1        | TP-UCB-FR    | 280850.7628  | 200.0363298        |
| 100                  | 50       | 1        | TP-UCB-EW    | 470206.8356  | 610.8394845        |
| 100                  | 50       | 1        | Delayed-UCB1| 555004.3727  | 3611.482174        |
| 100                  | 50       | 1        | UCB1         | 461125.7678  | 433.190974         |
| 100                  | 50       | 2        | TP-UCB-FR    | 280469.8885  | 600.1158378        |
| 100                  | 50       | 2        | TP-UCB-EW    | 470948.6985  | 1810.491089        |
| 100                  | 50       | 2        | Delayed-UCB1| 551713.5918  | 3167.855141        |
| 100                  | 50       | 2        | UCB1         | 460454.4842  | 1535.465475        |
| 100                  | 50       | 3        | TP-UCB-FR    | 280432.6875  | 194.6246275        |
| 100                  | 50       | 3        | TP-UCB-EW    | 470851.5341  | 678.1378134        |
| 100                  | 50       | 3        | Delayed-UCB1| 552354.8852  | 2784.797814        |
| 100                  | 50       | 3        | UCB1         | 461262.8902  | 406.9041603        |
| 100                  | 50       | 4        | TP-UCB-FR    | 277350.6683  | 357.2049513        |
| 100                  | 50       | 4        | TP-UCB-EW    | 431428.2109  | 845.9105653        |
| 100                  | 50       | 4        | Delayed-UCB1| 533550.167   | 6134.964191        |
| 100                  | 50       | 4        | UCB1         | 419308.3464  | 840.25097          |

Table 8: Summary of result for setting #2, $\tau_{\text{max}} = 200$, $\alpha = 100$.

| $\tau_{\text{max}}$ | $\alpha$ | Scenario | Learner      | Regret      | Confidence Interval |
|----------------------|----------|----------|--------------|--------------|--------------------|
| 200                  | 100      | 1        | TP-UCB-FR    | 998723.9102  | 348.3923308        |
| 200                  | 100      | 1        | TP-UCB-EW    | 962166.9976  | 1574.53646        |
| 200                  | 100      | 1        | Delayed-UCB1| 1217054.205  | 13791.12121       |
| 200                  | 100      | 1        | UCB1         | 922801.461   | 681.1463488       |
| 200                  | 100      | 2        | TP-UCB-FR    | 997866.0232  | 1163.306506       |
| 200                  | 100      | 2        | TP-UCB-EW    | 962888.2947  | 845.9105653       |
| 200                  | 100      | 2        | Delayed-UCB1| 1223555.271  | 13076.51935       |
| 200                  | 100      | 2        | UCB1         | 924666.3352  | 1936.282782       |
| 200                  | 100      | 3        | TP-UCB-FR    | 995734.719   | 386.1528975       |
| 200                  | 100      | 3        | TP-UCB-EW    | 962419.0355  | 1671.591765       |
| 200                  | 100      | 3        | Delayed-UCB1| 122418.1588  | 14560.25523       |
| 200                  | 100      | 3        | UCB1         | 923018.9128  | 593.7216922       |
| 200                  | 100      | 4        | TP-UCB-FR    | 996058.5901  | 681.2301995       |
| 200                  | 100      | 4        | TP-UCB-EW    | 937032.8774  | 1813.90584        |
| 200                  | 100      | 4        | Delayed-UCB1| 1214671.825  | 12459.63383       |
| 200                  | 100      | 4        | UCB1         | 893569.8466  | 1098.403796       |
Table 9: Parameters used in Setting #4.

| Scenario   | $a^i$                      | $b^i$                      |
|------------|----------------------------|----------------------------|
| Scenario 1 | $[8, 2, 8, 7, 1, 5, 6, 3, 3, 10]$ | $[7, 2, 2, 4, 4, 1, 7, 1, 2]$ |
| Scenario 2 | $[7, 9, 9, 5, 8, 10, 4, 7, 2]$  | $[6, 4, 5, 10, 3, 7, 4, 6, 2, 2]$ |
| Scenario 3 | $[1, 9, 8, 4, 2, 8, 7, 5, 4, 1]$  | $[4, 10, 3, 2, 4, 8, 7, 6, 9, 3]$ |
| Scenario 4 | $[2, 10, 8, 3, 10, 7, 7, 9, 8, 6]$  | $[8, 8, 4, 9, 10, 4, 1, 6, 6, 6]$ |
| Scenario 5 | $[1, 9, 3, 5, 10, 3, 7, 10, 5, 8]$  | $[2, 2, 9, 1, 2, 4, 3, 1, 5, 1]$ |
| Scenario 6 | $[8, 6, 3, 8, 6, 9, 7, 9, 9]$  | $[1, 10, 2, 9, 10, 2, 7, 4, 5, 9]$ |
| Scenario 7 | $[10, 7, 8, 7, 10, 4, 1, 1, 3]$  | $[5, 9, 10, 5, 6, 2, 8, 5, 7]$ |
| Scenario 8 | $[7, 7, 1, 3, 4, 5, 6, 1, 1]$  | $[8, 7, 3, 8, 10, 2, 3, 6, 7, 1]$ |
| Scenario 9 | $[10, 8, 7, 8, 1, 2, 8, 3, 1, 1]$  | $[10, 10, 3, 6, 2, 9, 6, 4, 7, 8]$ |
| Scenario 10| $[2, 1, 10, 8, 10, 6, 2, 10, 5, 3]$  | $[7, 5, 2, 9, 4, 1, 7, 8, 6, 4]$ |

Table # Setting #4

In this setting, each arm is described by a maximum reward $R^i = \tau_{\text{max}} \cdot i$, and two vectors $a^i = [a^i_1, \ldots, a^i_\alpha]$ and $b^i = [b^i_1, \ldots, b^i_\alpha]$ of length $\alpha$. The aggregated rewards $Z^i_{t,k}$ are distributed as $D^i_k = \frac{R^i}{\alpha} \text{Beta}(a^i_k, b^i_k), \forall k \in [\alpha]$. In this experiment, we fix $\tau_{\text{max}} = 100, \alpha = 10, T = 10^5$, and we design ten scenarios differing in the vectors $a^i$ and $b^i$. The parameters characterizing such randomly generated scenarios are reported in Table 9. The results for each scenario are averaged over 50 independent runs. In Figure 9, we provide the average result over the 10 scenarios, with whiskers corresponding to 95% confidence intervals.

Figure 9 shows an aggregated result on the pseudo-regret $R^i(t)$ for the analysed algorithms. Even over randomly generated scenarios we see that the proposed method are able to provide a significant improvement over the Delayed-UCB1 algorithm. Moreover, consistently the TP-UCB-FR algorithm result to be the best one at the end of the analysed time horizon $T = 10^5$. Conversely, for shorter time horizon ($T \leq 0.35 \cdot 10^5$) the algorithm performing the best among the ones for the TP-MAB setting is the TP-UCB-EW, which strengthen the idea that this algorithm is better suited for shorter time horizons.

Setting #5 Finally, we provide an experiment over a longer time horizon of $T = 10^6$ in the same configuration depicted by Setting #1. The pseudo-regret over time for this experiment is provided in Figure 10. Let us focus on the regret of TP-UCB-FR(20), i.e., the TP-UCB-FR algorithm where parameter $\alpha$ corresponds to the one of the environment, and compare it with the regret of Delayed-UCB1. The regret of TP-UCB-FR(20) (red line) has a slower growth w.r.t. Delayed-UCB1 (purple line), and, consequently, the difference in terms of regret increases (logarithmically) over time. The parameter influencing the regret of TP-UCB-FR is $\alpha$, which characterizes the specific setting we are tackling. More specifically, if we fix the other parameters (e.g., $\tau_{\text{max}}$) and increase the value of alpha, we have a proportional improvement in the upper bound of the regret of TP-UCB-FR. Therefore, we expect to have an even larger improvement of our algorithm when the value of $\alpha$ is large.
Figure 10: Experiments for Setting # 5: $\tau_{\text{max}} = 100$, $\alpha = 20$
D Real-world Applications of the TP-MAB Framework

In this section, we report some additional real-world examples which can be modeled through the TP-MAB setting. The following scenarios are characterized by the $\alpha$-smoothness property with different values of the $\alpha$ parameters.

**Example 3 (E-commerce).** An agent periodically receives a batch of identical items to sell on an e-commerce platform. Every time a slot of $N$ items arrives, the agent decides a price $p_i$ to post on a website, which corresponds to the arm $i_t$ chosen for the round $t$. The selected time horizon to sell the items, which are perishable, is one month. Each day, the seller checks how many items have been ordered and collects the payments (i.e., rewards). In this example, the maximum delay is $\tau_{\text{max}} = 30$ days, and one round is equal to 1 day. The upper bound on the cumulative reward is $R_i = p_i N$. Notice that the partial reward of each round is also upper bounded by $p_i N$. This implies that the reward has no structure, and consequently the $\alpha$-smoothness in this setting holds with $\alpha = 1$.

**Example 4 (Lottery Ticket).** There are $K$ different lotteries to choose from. Lottery $i \in [K]$ has $N$ winning scratch cards, each with a prize of $M$. The probability to extract a winning ticket in lottery $i$ is $p_i$. The player has to choose a lottery at each time step. At each round, the player buys $n$ tickets and sequentially scratches them and observes the reward. If $N = 1$ the total amount the player can win is $M$ and the reward is 1-smooth. Indeed, suppose that the first $n - 1$ tickets are not winning. This does not preclude the possibility of still gaining the maximum cumulative reward with the last ticket. Conversely, if $N = n$ the total amount the player can win is $R_i = n M$, and the reward is $n$-smooth. More specifically, by scratching the first ticket, the player can get useful information on the cumulative reward if the reward is either zero or $M$. If the player observed a zero reward so far, the maximum achievable cumulative reward becomes $(N - 1)M$. Conversely if the player observed a positive reward, the overall reward is in the interval $[M, n M]$. 