

Complete Policy Regret Bounds for Tallying Bandits

Dhruv Malik

Carnegie Mellon University

DHRUVM@ANDREW.CMU.EDU

Yuanzhi Li

Carnegie Mellon University

YUANZHIL@ANDREW.CMU.EDU

Aarti Singh

Carnegie Mellon University

AARTISINGH@CMU.EDU

Editors: Po-Ling Loh and Maxim Raginsky

Abstract

Policy regret is a well established notion of measuring the performance of an online learning algorithm against an adaptive adversary. We study restrictions on the adversary that enable efficient minimization of the *complete policy regret*, which is the strongest possible version of policy regret. We identify a gap in the current theoretical understanding of what sorts of restrictions permit tractability in this challenging setting. To resolve this gap, we consider a generalization of the stochastic multi armed bandit, which we call the *tallying bandit*. This is an online learning setting with an m -memory bounded adversary, where the average loss for playing an action is an unknown function of the number (or tally) of times that the action was played in the last m timesteps. For tallying bandit problems with K actions and time horizon T , we provide an algorithm that w.h.p achieves a complete policy regret guarantee of $\tilde{O}(mK\sqrt{T})$, where the \tilde{O} notation hides only logarithmic factors. We additionally prove an $\tilde{\Omega}(\sqrt{mKT})$ lower bound on the expected complete policy regret of any tallying bandit algorithm, demonstrating the near optimality of our method.

Keywords: policy regret, bandits, online learning, reinforcement learning

1. Introduction

When decision making algorithms are deployed in the real world, the reward associated with choosing a decision is rarely static. Instead, an algorithm’s decision impacts the state of its environment, which in turn influences the quality of that same decision in the future. For instance, in recommender systems such as YouTube and Netflix, the choice to recommend a type of content is often instrumental in shaping the preferences of the user for that content genre. This creates a feedback loop between an algorithm and its environment, and results in a complex back and forth interaction.

Such dynamic and interactive settings are well modeled as online learning problems, where a player competes against an adaptive adversary. To measure the performance of the player, most of the literature on online learning has focused on a performance metric called the *traditional regret* (Auer et al., 2002b; Flaxman et al., 2005; Abernethy et al., 2008; Hazan and Kale, 2012). However, a significant line of work has established that when the adversary is adaptive, the traditional regret is a poor indicator of the performance of an algorithm (Merhav et al., 2002; Arora et al., 2012; Cesa-Bianchi et al., 2013; Heidari et al., 2016; Lindner et al., 2021). Instead, one typically opts for a stronger performance metric, known as *policy regret*. The policy regret accumulated over a time horizon T is defined with respect to a competitor class \mathcal{C}_T of deterministic policies (or length T action sequences). The policy regret with respect to \mathcal{C}_T , which we denote $\mathcal{R}_{\mathcal{C}_T}^{\text{pol}}$, compares the algorithm’s cumulative loss to that of the best policy in \mathcal{C}_T .

Much of the prior work on policy regret has focused on the restrictive assumption that \mathcal{C}_T contains only those action sequences that repeatedly play the same action (Arora et al., 2012; Cesa-Bianchi et al., 2013; Dekel et al., 2014; Arora et al., 2018). We allude to the policy regret with this choice of \mathcal{C}_T as the *constant action policy regret*. The strongest possible version of policy regret is when \mathcal{C}_T is the complete policy class (i.e., the set of all deterministic policies), and we abbreviate this as the *complete policy regret*. This challenging setting has recently received attention from the online learning community (Heidari et al., 2016; Seznec et al., 2020; Lindner et al., 2021). On the other hand, this performance metric is equivalent to the one that is standard in the closely related field of reinforcement learning, where a vast literature explores how to efficiently maximize cumulative reward (Kearns et al., 1999; Sutton et al., 2000; Kakade et al., 2003; Jin et al., 2018; Wang et al., 2020; Malik et al., 2021).

Unfortunately, prior work has shown that without restrictions on the adversary, obtaining non-trivial guarantees on even the constant action policy regret is impossible (Arora et al., 2012). Hence, to attain meaningful guarantees on the complete policy regret (CPR), it is necessary to restrict the adversary. Prior literature on policy regret studies different competitor classes \mathcal{C}_T , along with varying types of restrictions on the adversary, to demonstrate non-trivial guarantees on the corresponding policy regret $\mathcal{R}_{\mathcal{C}_T}^{\text{pol}}$. We comprehensively survey these restrictions, and identify a gap in the current theoretical understanding of when it is possible to attain meaningful guarantees on CPR. To resolve this gap, we make the following contributions:

- We introduce an online learning setting known as the *tallying bandit*. Here the average loss for playing an action is a function of the number (or tally) of times that action was played in the last m timesteps. The stochastic multi armed bandit (sMAB) is a special case of the tallying bandit, via a choice of $m = 1$. From a more practical angle, we view the tallying bandit as a step towards handling feedback loops that arise in applications such as recommender systems.
- For tallying bandit problems with K actions and time horizon T , we provide an algorithm, that given any $\delta \in (0, 1)$, achieves with probability at least $1 - \delta$ a complete policy regret guarantee of $\tilde{\mathcal{O}}\left(mK\sqrt{T}\log(T, m, K, 1/\delta)\right)$.
- We complement our algorithmic development with an $\tilde{\Omega}(\sqrt{mKT})$ minimax lower bound on the expected complete policy regret of any method designed for tallying bandits. This demonstrates the near optimality of our algorithm.

2. Problem Formulation

2.1. Online Learning & Complete Policy Regret

We begin by providing a generic formulation of online learning against adaptive adversaries, following rather closely the description of Arora et al. (2012). An online learning problem with time horizon T and action set \mathcal{X} is an iterative game between a player and an adaptive adversary. Throughout, we let K denote the cardinality of \mathcal{X} . Before the game begins, the adversary fixes a sequence of history dependent loss functions $\{f_t\}_{t=1}^T$, where f_t maps \mathcal{X}^t to the interval $[0, 1]$. At each timestep t of the game, the player chooses an action $a_t \in \mathcal{X}$. In the bandit feedback model, the player then observes the loss value $f_t(a_{1:t})$, where we have used $a_{1:t}$ as shorthand for $a_1, a_2 \dots a_t$. By contrast, in the full information model, the player observes $f_t(a_{1:t-1}, x)$ for all $x \in \mathcal{X}$.

The cumulative loss experienced by the player during this game is $\sum_{t=1}^T f_t(a_{1:t})$. Note that this is a random variable, since the player’s strategy can be random. In order to evaluate the performance of the player, we compare this cumulative loss to a baseline. In particular, we let $\mathcal{C}_T \subseteq \mathcal{X}^T$ be a competitor class of policies (or length T action sequences). Given some \mathcal{C}_T , one typically measures the player’s performance via either the *policy regret*, which we denote $\mathcal{R}_{\mathcal{C}_T}^{\text{pol}}$, or the *traditional regret*, which we denote $\mathcal{R}_{\mathcal{C}_T}^{\text{trad}}$. The policy regret (Arora et al., 2012; Cesa-Bianchi et al., 2013; Arora et al., 2018) is defined as

$$\mathcal{R}_{\mathcal{C}_T}^{\text{pol}} = \sum_{t=1}^T f_t(a_{1:t}) - \min_{(y_1, y_2 \dots y_T) \in \mathcal{C}_T} \sum_{t=1}^T f_t(y_1, y_2 \dots y_t). \quad (1)$$

Notably, this definition differs substantially from the traditional regret (Auer et al., 2002b; Flaxman et al., 2005; Abernethy et al., 2008; Hazan and Kale, 2012), given by

$$\mathcal{R}_{\mathcal{C}_T}^{\text{trad}} = \sum_{t=1}^T f_t(a_{1:t}) - \min_{(y_1, y_2 \dots y_T) \in \mathcal{C}_T} \sum_{t=1}^T f_t(a_{1:t-1}, y_t). \quad (2)$$

In the aforementioned adversarial online learning setup, the traditional regret lacks meaningful interpretation. Instead, one opts for the policy regret to measure the player’s performance. We refer the interested reader to Arora et al. (2012) for further details on the motivation for this choice.

Let $\mathcal{X}_{\text{const}}$ denote the set of constant action sequences, so that $\mathcal{X}_{\text{const}} = \{(x, x \dots x) \text{ s.t. } x \in \mathcal{X}\}$. We refer to $\mathcal{R}_{\mathcal{X}_{\text{const}}}^{\text{pol}}$ and $\mathcal{R}_{\mathcal{X}_{\text{const}}}^{\text{trad}}$ respectively as the *constant action policy regret* and *constant action traditional regret*. The constant action policy regret (and hence the constant action traditional regret) yields a weak measure of performance, since we are comparing the player’s performance to a very restricted baseline policy class. Expanding the competitor class \mathcal{C}_T yields stronger notions of performance. In our paper, we are interested in the challenging setting where \mathcal{C}_T is the complete policy class (the set of all length T action sequences), or equivalently where $\mathcal{C}_T = \mathcal{X}^T$. This choice of \mathcal{C}_T in Eq. (1) yields the strongest version of policy regret, and we refer to it as *complete policy regret (CPR)*, and denote it by $\mathcal{R}_T^{\text{cp}}$. An *optimal policy* is a policy in \mathcal{X}^T that minimizes the cumulative loss. We will often use the terminology “efficiently minimize the CPR”, which means to obtain a CPR bound that is sublinear in T and at most polynomial in all other problem dependent parameters. Our exclusive focus is on statistical (rather than computational) efficiency.

2.2. Restricting The Adversary

Prior work due to Arora et al. (2012) has shown that without any restrictions on the adversary, and even when $\mathcal{C}_T = \mathcal{X}_{\text{const}}$, for any player there exists an adversary such that the player’s constant action policy regret satisfies $\mathcal{R}_{\mathcal{X}_{\text{const}}}^{\text{pol}} = \tilde{\Omega}(T)$. To prove this lower bound, Arora et al. (2012) construct an adversary that is wholly unrestricted and hence extremely powerful. Thus, to obtain non-trivial upper bounds on even the constant action policy regret, it is necessary to weaken the adversary. One natural type of restriction that has been well studied in prior work, is to restrict the memory of the adversary (Arora et al., 2012; Cesa-Bianchi et al., 2013; Arora et al., 2018).

Definition 1 We say that an adversary is m -memory bounded if for all $t \geq m$, all $a_{1:t} \in \mathcal{X}^t$, all $a'_{1:t-m} \in \mathcal{X}^{t-m}$ and all f_t we have that

$$f_t(a_{1:t}) = f_t(a'_1, a'_2 \dots a'_{t-m}, a_{t-m+1} \dots a_t).$$

Hence, an m -memory bounded adversary is only permitted to define its loss function based on the player's most recent m actions. Prior work has shown that when m is sublinear in T and \mathcal{C}_T is sufficiently restricted (informally, when \mathcal{C}_T equals or is only slightly larger than $\mathcal{X}_{\text{const}}$), then the player can achieve policy regret $\mathcal{R}_{\mathcal{C}_T}^{\text{pol}}$ that is sublinear in T (Arora et al., 2012; Cesa-Bianchi et al., 2013; Dekel et al., 2014; Arora et al., 2018).

When $m = 1$, then the adversary is oblivious, and we overload notation and write $f_t(a_{1:t}) = f_t(a_t)$. Notably, for a fixed \mathcal{C}_T , the policy regret $\mathcal{R}_{\mathcal{C}_T}^{\text{pol}}$ equals the traditional regret $\mathcal{R}_{\mathcal{C}_T}^{\text{trad}}$ in this scenario. It is well known that against an oblivious adversary, a player can achieve sublinear constant action policy regret (Arora et al., 2012). Hence, it is natural to question whether a player can achieve sublinear CPR, when the adversary is oblivious. We show via the following counterexample that this is impossible. Note that the result of the counterexample holds even in the full information feedback model (as opposed to just bandit feedback). A similar result is given by Mohri and Yang (2018).

Counterexample 1 *Let $\mathcal{X} = \{x_1, x_2\}$. To define its sequence of loss functions, the adversary first samples a bit string b uniformly at random from $\{0, 1\}^T$. For each t , it then defines*

$$f_t(x_1) = b_t \text{ and } f_t(x_2) = 1 - b_t.$$

Attaining sublinear complete policy regret is then equivalent to making a sublinear number of mistakes when guessing the value of b_t . Since this is impossible, we have that $\mathbb{E}[\mathcal{R}_T^{\text{CP}}] = \tilde{\Omega}(T)$, where the expectation is over the sampling of b and the player's (possibly randomized) strategy.

Crucially, the above counterexample relies on the fact that even though the adversary is oblivious, its loss functions $f_t, f_{t'}$ for $t \neq t'$ are time varying and are constructed independent of each other. In this scenario, the player cannot predict anything about f_t via knowledge of $f_{t'}$ for $t' < t$. To evade such counterexamples, a different type of restriction on the adversary's power is to ensure that some knowledge of $f_{t'}$ leaks some information about f_t . This motivates the following definition.

Definition 2 *An m -memory bounded adversary is said to be g -restricted if the following is true. For each action $x \in \mathcal{X}$, there exists a base function $g_x : \cup_{m'=1}^m \mathcal{X}^{m'} \rightarrow [0, 1]$, such that*

$$f_t(a_{1:t}) \equiv f_t(a_{\max\{1, t-m+1\}:t}) = g_{a_t}(a_{\max\{1, t-m+1\}:t}).$$

Although we are not aware of prior work on online learning that uses g -restricted adversaries without additional restriction, in the sequel we will discuss prior work that consider g -restricted adversaries with significant additional restrictions (Heidari et al., 2016; Levine et al., 2017; Seznec et al., 2019, 2020; Lindner et al., 2021; Awasthi et al., 2022). For now, we note that a g -restricted adversary must define loss functions whose value for a fixed input cannot vary with time. With such a restriction, the player can learn information about each g_x (and hence about f_t) as the game progresses, and this enables the player to choose better actions over time. This restriction thus precludes the setting of Counterexample 1.

When the adversary is g -restricted, it is straightforward to achieve CPR bounds that are sublinear in T and depend exponentially on m . However, throughout our paper we interested in efficiently minimizing CPR, which means we desire bounds that scale polynomially with m . Unfortunately, there exist online learning games where the adversary is m -memory bounded and g -restricted, but it is impossible to efficiently minimize CPR. We demonstrate this in the following counterexample, which holds even in the full information feedback model (as opposed to just bandit feedback).

Counterexample 2 Let $\mathcal{X} = \{x_1, x_2\}$. Sample a tuple b of length $m - 1$ uniformly at random from \mathcal{X}^{m-1} . Define $g_{x_1} : \cup_{m'=1}^m \mathcal{X}^{m'} \rightarrow [0, 1]$ as

$$g_{x_1}(a_{1:m'}) = 1 \text{ if } m' < m \text{ and } g_{x_1}(a_{1:m}) = 1 - \prod_{i=1}^{m-1} \mathbb{I}(a_i = b_i).$$

Also define $g_{x_2} = 1$. Via the base functions g_{x_1}, g_{x_2} we define the adversary's loss functions as

$$f_t(a_{1:t}) \equiv f_t(a_{t-m+1:t}) = g_{a_t}(a_{t-m+1:t}).$$

The policy that cyclically plays actions $b_1, b_2 \dots b_{m-1}, x_1$ suffers a loss of zero at least once every m timesteps. Meanwhile, suffering zero loss for the player is at least as hard as identifying b , and a standard ‘‘needle in the haystack’’ argument (Du et al., 2020) shows that this requires $\tilde{\Omega}(2^m)$ timesteps. Hence we have that $\mathbb{E}[\mathcal{R}_T^{cp}] = \tilde{\Omega}(\min\{2^m, T\}/m)$, where the expectation is over the sampling of b and the player's (possibly randomized) strategy.

This counterexample demonstrates that even if the adversary is m -memory bounded and g -restricted, any player suffers CPR that scales exponentially with m . So, further restrictions on the adversary are necessary. A natural restriction is to enforce that each g_x has special structure. This is precisely the approach taken by works on *rotting* bandits (Heidari et al., 2016; Levine et al., 2017; Seznec et al., 2019, 2020), *improving* bandits (Heidari et al., 2016), *single peaked* bandits (Lindner et al., 2021) and *congested* bandits (Awasthi et al., 2022). Concretely, these works use base functions $\{g_x\}_{x \in \mathcal{X}}$ that have the following special ‘‘tallying’’ structure.

Definition 3 An m -memory bounded and g -restricted adversary is said to be h -tallying, if for each $x \in \mathcal{X}$ there exists $h_x : \{1, 2 \dots m\} \rightarrow [0, 1]$ such that

$$f_t(a_{1:t}) \equiv f_t(a_{\max\{1, t-m+1\}:t}) = g_{a_t}(a_{\max\{1, t-m+1\}:t}) = h_{a_t} \left(\sum_{t'=\max\{1, t-m+1\}}^t \mathbb{I}(a_{t'} = a_t) \right).$$

As discussed by the aforementioned works, this tallying structure is often a natural model in practice. For instance, Heidari et al. (2016) discuss a crowdsourcing setting where an agency utilizes workers to repeatedly perform the same task (such as classifying images) at each timestep. The agency picks a worker at each timestep, with the goal of picking a sequence of workers that makes the fewest number of mistakes when performing the task. Here, it is reasonable that an individual worker's performance changes as an (unknown) function of the *number* of times that the worker has already performed the task (for example, due to fatigue), thus motivating the tallying structure.

We emphasize that in addition to assuming the adversary is h -tallying, the aforementioned works of Heidari et al. (2016), Levine et al. (2017), Seznec et al. (2019), Seznec et al. (2020), Lindner et al. (2021) and Awasthi et al. (2022) make *supplemental* benign assumptions on the structure of the functions $\{h_x\}_{x \in \mathcal{X}}$, as we detail in Section 5. For instance, the rotting bandit setting of Heidari et al. (2016) assumes that h_x is an increasing function for each $x \in \mathcal{X}$. Under this benign assumption, they provide algorithms that efficiently minimize CPR. Notably, such strong assumptions on h_x enable this line of work to (often) tackle the case where $m = T$ (although algorithms designed for this case generally do not handle $m < T$), and additionally enables these works to (often) handle the more difficult scenario of when the losses are *stochastically* observed.

This exposes a gap in our understanding of when one can efficiently minimize CPR. In particular, it remains unclear whether we can attain this goal for h -tallying adversaries where we make *no* assumptions on the structure of each h_x . This motivates the following question.

Assume the adversary is m -memory bounded, g -restricted and h -tallying. Without any assumptions on the functions $\{h_x\}_{x \in \mathcal{X}}$, and in the bandit feedback model with (possibly) stochastically observed losses, when is it possible to efficiently minimize the complete policy regret?

The remainder of this paper is devoted to resolving this question. To this end, in the sequel we define the *tallying bandit*, and provide upper and lower bounds on the achievable CPR in this setting.

3. Tallying Bandits

Let us formally introduce the tallying bandit setting.

Definition 4 *An online learning game is an (m, g, h) -tallying bandit if the adversary is m -memory bounded, g -restricted and h -tallying, and if after playing action a_t the player observes a random variable $\tilde{h}_{a_t}(y_t) \in [0, 1]$ satisfying*

$$\mathbb{E} \left[\tilde{h}_{a_t}(y_t) \right] = h_{a_t}(y_t) = g_{a_t}(a_{\max\{1, t-m+1\}:t}) = f_t(a_{1:t}),$$

where $y_t = \sum_{t'=\max\{1, t-m+1\}}^t \mathbb{I}(a_{t'} = a_t)$.

We assume the cardinality K of the action set \mathcal{X} is finite, and also that m is known (although we discuss how to relax this in Section 6). With this definition of the setting in hand, we can restate our goal of efficiently minimizing the CPR. Concretely, we desire an algorithm, which when given an (m, g, h) -tallying bandit problem, has a CPR bound that is polynomial in K, m and is sublinear in T . The tallying bandit strictly generalizes the well studied stochastic multi armed bandit (sMAB) (Lai and Robbins, 1985; Auer et al., 2002a), simply via a choice of $m = 1$. Hence, we generalize the study of sMAB to $m > 1$. We remark that tallying bandit is a special case of the *rested* bandit, a general framework for nonstationary MAB where the reward of an arm evolves when it is pulled, and we defer detailed discussion of this to Section 5.

Recall that in sMAB, the optimal policy plays the same optimal arm at each timestep. Hence, in sMAB obtaining zero constant action traditional regret is equivalent to obtaining zero constant action policy regret and also zero CPR. Given that the tallying bandit is highly structured and generalizes sMAB via $m = 1$, it is natural to question whether minimizing constant action policy (or traditional) regret implies minimizing CPR. The following counterexample answers this question in the negative when $m > 1$. More generally, this counterexample shows that even when the adversary is restricted (as in tallying bandits), minimizing the constant action policy (or traditional) regret can lead to solutions whose cumulative loss is $\tilde{\Omega}(T)$ larger than the minimum achievable total loss.

Counterexample 3 *Let $\mathcal{X} = \{x_1, x_2\}$ and $m = 2$. Define $h_{x_1}(1) = h_{x_2}(1) = 0$ and $h_{x_1}(2) = h_{x_2}(2) = 1$. A policy that fixes either of the two actions, and then plays this action at every timestep, incurs cumulative loss $T - 1$ but has zero constant action policy regret and zero constant action traditional regret. Meanwhile, the complete policy regret of this policy is $T - 1$, since the optimal policy that alternates playing actions x_1 and x_2 incurs zero cumulative loss.*

Thus far, our motivation for the tallying bandit setting has been primarily theoretical, to resolve the gap in our understanding of when we can efficiently minimize CPR. Nevertheless, in similar vein to [Heidari et al. \(2016\)](#), [Lindner et al. \(2021\)](#) and [Awasthi et al. \(2022\)](#), we believe that the tallying bandit is a simple approximation for various practical settings. For instance, in recommender systems the reward associated with an action is rarely static, because the stimulus of recommended content influences user preferences ([Cosley et al., 2003](#); [Sinha et al., 2016](#)). Moreover, literature on psychology and cognition suggests that humans often forget prior stimuli and do not always encode them in permanent memory ([Klatzky, 1980](#); [Chessa and Murre, 2007](#)). Thus, one way to model a user’s preferences is via an (unknown) function of the *number* of times a content genre has been recommended in a *recent* time interval, motivating both the h -tallying structure as well as bounded m . Nevertheless, the tallying bandit is just one plausible model, and we suggest possible extensions in [Section 6](#).

Let us now discuss potential avenues for efficiently minimizing CPR in tallying bandit problems. One approach is to observe that any tallying bandit problem can be cast as a reinforcement learning (RL) problem, where each state corresponds to a sequence of actions taken in the last m timesteps. However, methods for solving such RL problems typically scale with the cardinality of the state space ([Azar et al., 2013, 2017](#); [Jin et al., 2018](#)), and such approaches would suffer $\tilde{\Omega}(K^m)$ CPR.

Hence it is necessary to leverage the additional properties of tallying bandit problems. To gain intuition, let us consider the simplified setting of deterministic bandit feedback, where for each $t, a_{1:t} \in \mathcal{X}^t$ the player observes $f_t(a_{1:t})$ with no noise. Since the loss functions f_t are fully defined by the functions $\{h_x\}_{x \in \mathcal{X}}$, it is natural to consider the following algorithm, which we denote ALG_{det} . First, the algorithm queries $h_x(y)$ at each $(x, y) \in \mathcal{X} \times \{1, 2 \dots m\}$. This yields full information about the loss functions f_t . Then, the algorithm plans offline an optimal sequence of actions for the remaining timesteps. ALG_{det} is formally specified as [Algorithm 2](#) in [Appendix C](#), and the following result shows that its CPR is minimax optimal (upto constant factors).

Proposition 1 *Consider any (m, g, h) -tallying bandit problem with deterministic bandit feedback. Then the complete policy regret of [Algorithm 2](#) (ALG_{det}) is almost surely upper bounded as $\mathcal{R}_T^{\text{cp}} \leq m(K + 1)$. Moreover, there exists an (m, g, h) -tallying bandit problem, such that the (expected) complete policy regret of any (possibly randomized) algorithm on this problem with deterministic feedback is lower bounded as $\mathbb{E}[\mathcal{R}_T^{\text{cp}}] \geq mK/128$.*

The proof of [Proposition 1](#) is included in [Appendix C](#). Due to the optimality of ALG_{det} given deterministic feedback, it is reasonable to extend it to handle stochastic feedback. One natural way to do so is via an “explore then exploit” modification, which has been studied even for sMAB ([Slivkins, 2019](#)). Consider the following “explore then exploit” algorithm, which we denote $\text{ALG}_{\text{stoch}}$. It queries repeatedly to receive stochastic realizations of $h_x(y)$ at each $(x, y) \in \mathcal{X} \times \{1, 2 \dots m\}$, and constructs tight confidence intervals for these $h_x(y)$. It then uses these estimated values of $h_x(y)$ to plan offline an optimal sequence of actions for the remaining timesteps. $\text{ALG}_{\text{stoch}}$ is provably efficient, in the sense that its CPR is sublinear in T and polynomial in m, K . However, a standard argument ([Slivkins, 2019](#)) shows that the dependency on T for $\text{ALG}_{\text{stoch}}$ scales as $\tilde{\Theta}(T^{2/3})$, and it is unclear whether this dependency is optimal for tallying bandits. In the forthcoming section, we show that this dependency on T can be significantly improved.

4. Main Results

We now turn to our main results. In Section 4.1, we formulate Algorithm 1, a method designed for solving tallying bandit problems, and prove an $\tilde{O}(mK\sqrt{T})$ upper bound on its CPR, where \tilde{O} hides only logarithmic factors. In Section 4.2, we prove an $\tilde{\Omega}(\sqrt{mKT})$ lower bound on the CPR of any method designed to solve tallying bandit problems. This shows that Algorithm 1 is nearly optimal.

4.1. Upper Bound

To formulate our algorithm for tallying bandits, it is natural to exploit the h -tallying structure of the problem, by building estimates of $h_x(y)$ for each $(x, y) \in \mathcal{X} \times \{1, 2 \dots m\}$. As discussed in Section 3, “explore then exploit” algorithms such as $\text{ALG}_{\text{stoch}}$ incur a poor dependency on T . Instead, it is critical to balance exploration and exploitation, by estimating $h_x(y)$ only by playing those actions that will not increase the regret too fast. To this end, we introduce a key definition. Recall that a deterministic policy π is length T sequence of actions. We say that a deterministic policy π is \sqrt{T} -cyclic if $\pi_{k\sqrt{T}+t} = \pi_t$ for each $1 \leq t \leq \sqrt{T}$ and each $0 \leq k \leq \sqrt{T} - 1$. The basis for our algorithm relies on the key claim that for any tallying bandit problem, there exists a \sqrt{T} -cyclic policy that is nearly optimal. This claim is formalized as Lemma 3 in Appendix A.

Assuming this claim to be true, to solve tallying bandits it is tempting to leverage algorithms designed for multi armed bandits with expert advice (Auer et al., 2002b), where we treat each \sqrt{T} -cyclic policy as an expert. However, such an approach would only guarantee $\tilde{O}(T^{3/4})$ CPR, since there are $K^{\sqrt{T}}$ experts. Instead, our algorithm draws inspiration from the successive elimination (SE) algorithm, which has been applied to sMAB (Even-Dar et al., 2002; Slivkins, 2019). Before we apply SE in our setting, let us recall SE in the context of sMAB. The method proceeds in epochs. Within each epoch s , it maintains a set A_s of feasible arms, where an arm is feasible only if its estimated optimality gap lies within a confidence interval of size C_{s-1} . The algorithm pulls each arm in A_s repeatedly to obtain a sharper estimate of its optimality gap. Then the method uses these sharper estimates to prune A_s and create a smaller set A_{s+1} , where A_{s+1} contains only those arms in A_s whose estimated optimality gap is smaller than some $C_s < C_{s-1}$.

To apply SE in our tallying bandit setting, we define the initial set A_1 as the set of all \sqrt{T} -cyclic policies, and treat each such policy to be analogous to an “arm” in sMAB. Our aforementioned key claim ensures that there is some policy in A_1 that is guaranteed to be nearly optimal. However, there are two salient technical issues that prevent a naive application of SE to solve tallying bandits. First, note that the cardinality of A_1 is $K^{\sqrt{T}}$. This is too large to apply a traditional SE approach, since naively estimating the optimality gap of each policy in A_s by repeatedly playing the policy and applying a concentration inequality would incur large regret. To resolve this, we exploit the h -tallying structure of the problem to modify SE, and estimate the optimality gap of each policy in A_s by iteratively estimating $h_x(y)$ for each $(x, y) \in \mathcal{X} \times \{1, 2 \dots m\}$. Second, note that unlike in the sMAB, in the tallying bandit the prior history of actions affects the loss of the current action, which biases the estimation of the optimality gap of the policies in A_s . To handle this, we modify SE to incorporate an additional overheard step before the estimation, and our proof shows that this overhead step removes the bias from the estimation without incurring much additional regret.

With this outline in mind, let us present our method, which is formalized in Algorithm 1. To define Algorithm 1, we say that to *execute* a \sqrt{T} -cyclic policy π for $k \leq \sqrt{T}$ periods means to choose the action sequence $\pi_1, \pi_2 \dots \pi_{k\sqrt{T}}$. We also define for each \sqrt{T} -cyclic policy π and $(x, y) \in \mathcal{X} \times \{1, 2 \dots m\}$, the quantity $N_{xy}(\pi)$ via the following procedure. Execute π for $n + 1 \leq$

Algorithm 1: Successive Elimination for Tallying Bandits (SE-TB)

Inputs: memory capacity m , time horizon T , failure probability tolerance $\delta \in (0, 1)$, number of actions K

- 1 Define $S = \log_2 \left(\frac{\sqrt{T}}{4Km} + 1 \right)$.
- 2 Define $n_s = 2^s$, $T_s = 2n_s Km \sqrt{T}$ and $C_s = \sqrt{\frac{32Km}{n_s \sqrt{T}} \log \left(\frac{2KmS}{\delta} \right)}$.
- 3 Construct A_1 to be the set of all \sqrt{T} -cyclic policies.
- 4 **for** $s \in \{1, 2 \dots S\}$ **do**
- 5 **for** $x \in \mathcal{X}$ **do**
- 6 **for** $y \in \{1, 2 \dots m\}$ **do**
- 7 Select $\pi_{sxy} \in \arg \max_{\pi' \in A_s} \{N_{xy}(\pi')\}$, where N_{xy} is defined in Eq. (3).
- 8 **if** $N_{xy}(\pi_{sxy}) = 0$ **then**
- 9 Execute π_{sxy} for $2n_s$ periods and store nothing.
- 10 **else**
- 11 Execute π_{sxy} for n_s periods and store nothing.
- 12 Execute π_{sxy} for n_s periods and store $\{\tilde{h}_x(y)_{s,k}\}_{k=1}^{n_s N_{xy}(\pi_{sxy}) \sqrt{T}}$.
- 13 **end**
- 14 **end**
- 15 **end**
- 16 **for** $\pi \in A_s$ **do**
- 17 Define $\hat{\mu}_s(\pi) = \sum_{(x,y) \in \mathcal{X} \times \{1,2 \dots m\}} N_{xy}(\pi) \frac{1}{n_s N_{xy}(\pi_{sxy}) \sqrt{T}} \sum_{k=1}^{n_s N_{xy}(\pi_{sxy}) \sqrt{T}} \tilde{h}_x(y)_{s,k}$.
- 18 **end**
- 19 Select $\hat{\pi}_s \in \arg \min_{\pi \in A_s} \hat{\mu}_s(\pi)$.
- 20 Construct $A_{s+1} = \{\pi \in A_s \text{ s.t. } \hat{\mu}_s(\pi) \leq \hat{\mu}_s(\hat{\pi}_s) + 2C_s\}$.
- 21 **end**

\sqrt{T} periods so that we have played the action sequence $\pi_1 \dots \pi_{n\sqrt{T}}, \pi_{n\sqrt{T}+1} \dots \pi_{(n+1)\sqrt{T}}$. Then use this action sequence to define

$$N_{xy}(\pi) = \frac{1}{\sqrt{T}} \sum_{t=n\sqrt{T}+1}^{(n+1)\sqrt{T}} \mathbb{I}(\pi_t = x) \cdot \mathbb{I} \left(y = \sum_{t'=\max\{1,t-m+1\}}^t \mathbb{I}(\pi_{t'} = x) \right). \quad (3)$$

Intuitively, $N_{xy}(\pi)$ is the fraction of times that the player (stochastically) observes the loss value $h_x(y)$ when they repeatedly play the \sqrt{T} -cyclic policy π . In Lemma 2 in Appendix A, we show that if $m \leq \sqrt{T}$, then as long as $n \geq 1$, the number $N_{xy}(\pi)$ is well defined and independent of n , and also independent of any action sequence that was played before we executed π for $n+1$ periods. It suffices to consider tallying bandit problems where $m \leq \sqrt{T}$ (as we show in our proofs). Hence, lines 11 and 12 in Algorithm 1 are well defined, because when we execute π for $2n_s \geq 2$ periods, then in the latter $n_s \geq 1$ periods we observe (stochastic instantiations of) the loss value $h_x(y)$ for a total of $n_s N_{xy}(\pi) \sqrt{T}$ times, and we have denoted these observations as $\{\tilde{h}_x(y)_{s,k}\}_{k=1}^{n_s N_{xy}(\pi_{sxy}) \sqrt{T}}$. Note that various steps in Algorithm 1, such as line 7, require knowledge of $N_{xy}(\pi)$, but this can be

computed offline when m is known. Indeed, the only steps of Algorithm 1 that are online (or incur regret) are lines 9, 11 and 12. Let us now analyze the performance of Algorithm 1.

Theorem 1 *For any (m, g, h) -tallying bandit problem and any input $\delta \in (0, 1)$, with probability at least $1 - \delta$ the complete policy regret of Algorithm 1 (SE-TB) is upper bounded as*

$$\mathcal{R}_T^{cp} \leq 1200Km\sqrt{T} \left(\sqrt{\log(2Km \log(T)/\delta)} + \log_2 \left(\sqrt{T}/(2Km) \right) \right).$$

The proof of Theorem 1 is deferred to Appendix A. This result guarantees that given any tallying bandit problem, Algorithm 1 efficiently minimizes CPR, with a favorable dependency on m, K, T . Nevertheless, we acknowledge that our result has the following two limitations.

Knowledge of m . Algorithm 1 requires m as an input, which is unrealistic in practice. It is possible to modify Algorithm 1 to be adaptive to an unknown m , albeit at the expense of polynomially worse (and not sharp) dependency on m, K . Our focus is on obtaining a sharp characterization of the achievable CPR when m is known, and so we relegate discussion of this modification to Section 6. Sharply characterizing the minimax CPR when m is unknown remains an important open question.

Computational Efficiency. Algorithm 1 is computationally inefficient. We emphasize that our exclusive focus is on statistical (rather than computational) efficiency, since our work is only a first step. We believe this is a worthwhile endeavor, since attaining sublinear CPR is a non-trivial task riddled with subtleties, even in settings that make much stronger assumptions than we do. For instance, the improving (Heidari et al., 2016) and single peaked (Lindner et al., 2021) bandit settings enforce $m = T$, require monotonicity and convexity conditions on $\{h_x\}_{x \in \mathcal{X}}$, and also require the losses are observed *deterministically*. Even with these strong requirements, the best known CPR guarantees are *asymptotic* bounds that may decay *arbitrarily* slowly, and their algorithms cannot handle $m < T$. Hence we believe that our effort to provide nearly optimal non-asymptotic bounds on the CPR, in our realistic and practically motivated setting where $m \leq T$ and losses are observed stochastically, is worthwhile. Nevertheless, devising computationally efficient algorithms for the tallying bandit remains an important future direction, and we believe this is a non-trivial task. Indeed, even for the congested bandit (Awasthi et al., 2022), which is the tallying bandit with the additional strong assumption that $\{h_x\}_{x \in \mathcal{X}}$ are increasing, existing algorithms are computationally inefficient.

4.2. Lower Bound

It is reasonable to question whether the dependency of Algorithm 1 on m, K, T is optimal. Since the tallying bandit is equivalent to sMAB when $m = 1$, a classical result (Slivkins, 2019) shows that any tallying bandit algorithm suffers $\tilde{\Omega}(\sqrt{KT})$ expected CPR. However, the correct dependency on m is unclear when $m > 1$, due to the highly structured nature of the tallying bandit. For instance, the proof of Theorem 1 shows that any tallying bandit problem can be equivalently cast as a Markov decision process (MDP), where it takes at most m timesteps to transition from any state to any other state in this MDP. One may wonder whether we can utilize such structure to design a smarter algorithm that exchanges the *multiplicative* dependence on m in the result of Theorem 1 for an *additive*

dependence on m . Concretely, one may desire a bound that scales as $\tilde{\mathcal{O}}\left(\text{poly}(m, K) + K\sqrt{T}\right)$. The following result shows that this is impossible.

Theorem 2 *There exists an (m, g, h) -tallying bandit problem and a numerical constant $c > 0$, such that the (expected) complete policy regret of any (possibly randomized) algorithm on this problem is lower bounded as*

$$\mathbb{E} [\mathcal{R}_T^{\text{cp}}] \geq c \cdot \max \left\{ mK, \sqrt{mKT} \right\}.$$

The proof of Theorem 2 is deferred to Appendix B. This result demonstrates that Algorithm 1 is nearly minimax optimal, and its suboptimality is bounded by $\tilde{\mathcal{O}}\left(\sqrt{mK} \log(T, m, K)\right)$. A comment on the proof technique of Theorem 2 is in order. Our proof reduces the tallying bandit setting to that of best arm identification in sMAB problems (Audibert and Bubeck, 2010; Slivkins, 2019). We construct a tallying bandit problem where minimizing CPR is at least as hard as identifying the best arm in an sMAB problem with $\tilde{\Theta}(mK)$ arms. Indeed, when $m = 1$ then tallying bandit is equivalent to sMAB, and Theorem 2 recovers the classical lower bound on the expected regret suffered by any algorithm designed for sMAB with K arms. Notably, a key component of our proof is to non-trivially upper bound the cumulative loss of the optimal policy in our construction, to ensure that we get a multiplicative (in lieu of additive) dependence on m in our lower bound.

5. Related Work

Policy Regret. The incompatibility of the traditional regret with an adaptive adversary was first identified by Merhav et al. (2002), who studied the full information feedback model. The notion of policy regret was formalized by the foundational work of Arora et al. (2012). They provide an algorithm which efficiently minimizes constant action policy regret $\mathcal{R}_{\mathcal{X}_{\text{const}}}^{\text{pol}}$ against generic m -memory bounded adversaries. It is unclear how to apply this algorithm to our setting, since our focus is on minimizing the CPR $\mathcal{R}_T^{\text{cp}}$. Arora et al. (2012) do also consider minimizing the policy regret $\mathcal{R}_{\mathcal{C}_T}^{\text{pol}}$ when $\mathcal{C}_T \supseteq \mathcal{X}_{\text{const}}$. For instance, when \mathcal{C}_T is the set of all piecewise constant sequences with at most s switches, they provide an algorithm whose policy regret satisfies $\mathcal{R}_{\mathcal{C}_T}^{\text{pol}} \leq \tilde{\mathcal{O}}\left(m(Ks)^{1/3}T^{2/3}\right)$. Our Counterexample 3 shows that this algorithm cannot minimize CPR in the tallying bandit setting, since the optimal policy in Counterexample 3 has $\tilde{\Theta}(T)$ switches. Cesa-Bianchi et al. (2013), Dekel et al. (2014) and Arora et al. (2018) study the constant action policy regret, and do not discuss CPR. The results of Mohri and Yang (2018) can be extended to yield policy regret guarantees relative to rather large comparator classes, but do not provide CPR guarantees.

Reinforcement Learning (RL). Minimizing CPR is equivalent to the performance metric used in RL, which is to maximize the total collected reward (Sutton and Barto, 2018). As we show (see Lemma 1), any tallying bandit problem can be cast as an RL problem. However, RL methods typically scale with the cardinality of the state space (Azar et al., 2013, 2017; Jin et al., 2018). Hence, applying off the shelf RL algorithms to solve tallying bandit problems would incur $\tilde{\Omega}(K^m)$ CPR.

Restless & Rested Bandits. In restless bandits, the reward of an arm evolves according to a stochastic process, independently of the actions chosen by the player (Whittle, 1981; Garivier and Moulines, 2011; Besbes et al., 2014). This is incompatible with tallying bandits. By contrast, tallying bandits is a special case of rested bandits, which is a general non-stationary MAB framework

where an arm’s reward changes when it is pulled. Tekin and Liu (2012) and Cortes et al. (2020) both study rested bandits where an arm’s reward evolves according to a stochastic process, but both consider notions of regret that are significantly weaker than the CPR. A different variant of rested bandits is studied by Bouneffouf and Féraud (2016), who assume that the dynamics of how the reward changes is known upto a constant factor, and hence their results are incomparable to ours.

Recharging, Recovering, Blocking, Delay-Dependent & Last Switch Dependent Bandits. This line of work studies settings where an arm’s reward changes according to the number of timesteps that have passed since the arm was last pulled (Kleinberg and Immorlica, 2018; Basu et al., 2019; Pike-Burke and Grunewalder, 2019; Cella and Cesa-Bianchi, 2020; Laforgue et al., 2021). Such settings typically cannot be cleanly classified as either rested or restless bandits, but are related to both. The models in these works for how the reward evolves are different than our tallying structure, where the reward of an arm instead depends on the number of times that arm was played.

Rotting Bandits. The rotting bandits setting (Heidari et al., 2016; Levine et al., 2017; Seznec et al., 2019, 2020) is a special case of our tallying bandit formulation, and merits close comparison to our work. This setting enforces $m = T$, and an arm’s reward is assumed to be a decreasing function of the number of times that arm has been pulled. In our language, this means the $\{h_x\}_{x \in \mathcal{X}}$ functions are increasing. This strong assumption enables these works to efficiently minimize CPR even though $m = T$, although we note that algorithms for this setting cannot handle general $m < T$. By contrast, in our setting we make no assumptions on the structure of the functions $\{h_x\}_{x \in \mathcal{X}}$, and assuming $m < T$ is necessary. Indeed, in our setting, when m is $\tilde{\Omega}(T)$ then the result of Theorem 2 shows that any player suffers $\tilde{\Omega}(T)$ worst case CPR.

Improving & Single Peaked Bandits. The improving bandit (Heidari et al., 2016) and single peaked bandit (Lindner et al., 2021) are both special cases of our tallying bandit setting, and deserve special attention. In improving bandits, the reward of an arm is an increasing, concave function of the number of times it has been pulled. The single peaked bandit generalizes this, so that the reward function of an arm is initially increasing and concave, but may become decreasing at some point. In our language, this means that the $\{h_x\}_{x \in \mathcal{X}}$ functions are decreasing and convex, or decreasing and convex and then possibly increasing after some point. Both settings enforce $m = T$, and algorithms for these settings do not handle general $m < T$. By contrast, since we make no assumptions on the structure of the functions $\{h_x\}_{x \in \mathcal{X}}$, our Theorem 2 shows that when m is $\tilde{\Omega}(T)$ then the worst case CPR scales as $\tilde{\Omega}(T)$. We remark that the CPR bounds in these works are asymptotic, whereas we provide non-asymptotic guarantees. We also remark that these works require the losses to be observed deterministically, and they only provide a heuristic to handle stochastic observations of the loss.

Congested Bandits. In concurrent work, Awasthi et al. (2022) introduced the congested bandit, which is a special case of the tallying bandit. Their formulation considers arbitrary $m \leq T$, and the reward of an arm is a decreasing function of the number of times it has been pulled. Hence the congested bandit is the tallying bandit with the additional assumption that the $\{h_x\}_{x \in \mathcal{X}}$ functions are increasing. This additional assumption enables them to provide an $\tilde{O}\left(\sqrt{mKT}\right)$ CPR bound, although we note that their algorithm is computationally inefficient.

6. Discussion

In this paper, we studied conditions under which it is possible to efficiently minimize CPR in online learning. To this end, it is necessary to restrict the adversary, and we considered several natural restrictions on the adversary that have appeared in prior work. We then exposed a gap in our understanding of when it is possible to efficiently minimize CPR. To resolve this gap, we introduced the tallying bandit setting, and formulated an algorithm whose CPR (after discarding logarithmic factors) is w.h.p at most $\tilde{O}(mK\sqrt{T})$. We also provided a lower bound of $\tilde{\Omega}(\sqrt{mKT})$ on the expected CPR of any tallying bandit algorithm, demonstrating the near optimality of our method.

Our Algorithm 1 required as input the true value of m . In practice, this knowledge is unrealistic, and one instead might only have an upper bound \bar{m} on the true value of m . Let us describe a modified version of Algorithm 1 that can be used in this setting. Recall from the proof of Theorem 1 that Algorithm 1 proceeds in epochs, and in each epoch s it stores a set A_s of policies whose (average) loss is $\tilde{O}(2^{-s}T^{-1/4})$ greater than that of the optimal policy. Hence, for the setting with unknown m , we can run \bar{m} instantiations of Algorithm 1. After each epoch s , we identify the instantiation that has the policy with the minimum estimated (average) loss, and denote this instantiation as m_s . We then discard those instantiations whose policies have (average) loss that is $\tilde{\Omega}(2^{-s}T^{-1/4})$ greater than the (average) loss of the best policy stored by m_s . With such an approach, we are guaranteed to never discard the instantiation corresponding to the true m . And via the techniques used in the proof of Theorem 1, we can show that if we do not discard an instantiation corresponding to $m' \neq m$, then playing policies stored by this instantiation does not incur large regret. This approach yields a $\tilde{O}(\sqrt{T})$ upper bound on the CPR, at the expense of polynomial factors of \bar{m}, K .

A number of open directions remain. A natural open question is resolving the gap between our upper and lower bounds on the achievable CPR in the tallying bandit. Separately, although Algorithm 1 is nearly statistically optimal, it is computationally inefficient. However, since the tallying bandit is highly structured, it is possible that the computational efficiency of even our own Algorithm 1 can be improved. For instance, can we design a data structure, which stores policies in a manner that allows efficient elimination of suboptimal policies after each epoch? Devising computationally efficient algorithms for tallying bandits is an important direction for future work. Finally, we view the tallying bandit as only a first step towards modeling interactive settings like recommender systems. For example, consider the following generalization of tallying bandits, where the loss of an action is a function of a *weighted* sum of the number of times the action has been played in the past, where more recent plays are given more weight. This naturally corresponds to a model of human memory, where more importance is placed on more recent events. Can we design efficient algorithms for such settings?

Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE1745016. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

- Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Conference on Learning Theory*, 2008.
- Raman Arora, Ofer Dekel, and Ambuj Tewari. Online bandit learning against an adaptive adversary: From regret to policy regret. In *International Conference on Machine Learning*, 2012.
- Raman Arora, Michael Dinitz, Teodor V. Marinov, and Mehryar Mohri. Policy regret in repeated games. In *Advances in Neural Information Processing Systems*, 2018.
- Jean-Yves Audibert and Sébastien Bubeck. Best arm identification in multi-armed bandits. In *Conference on Learning Theory*, 2010.
- Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(23):235256, 2002a.
- Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002b.
- Pranjal Awasthi, Kush Bhatia, Sreenivas Gollapudi, and Kostas Kollias. Congested bandits: Optimal routing via short-term resets, 2022.
- Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert J. Kappen. Minimax pac bounds on the sample complexity of reinforcement learning with a generative model. *Machine Learning*, 91(3):325349, 2013.
- Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, 2017.
- Soumya Basu, Rajat Sen, Sujay Sanghavi, and Sanjay Shakkottai. Blocking bandits. In *Advances in Neural Information Processing Systems*, 2019.
- Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in Neural Information Processing Systems*, 2014.
- Djallel Bouneffouf and Raphael Féraud. Multi-armed bandit problem with known trend. *Neurocomputing*, 205(C):16–21, 2016.
- Leonardo Cella and Nicolò Cesa-Bianchi. Stochastic bandits with delay-dependent payoffs. In *International Conference on Artificial Intelligence and Statistics*, pages 1168–1177, 2020.
- Nicolò Cesa-Bianchi, Ofer Dekel, and Ohad Shamir. Online learning with switching costs and other adaptive adversaries. In *Advances in Neural Information Processing Systems*, 2013.
- Antonio G. Chessa and Jaap M. J. Murre. A neurocognitive model of advertisement content and brand name recall. *Marketing Science*, 26(1):130–141, 2007.
- Corinna Cortes, Giulia DeSalvo, Vitaly Kuznetsov, Mehryar Mohri, and Scott Yang. Discrepancy-based algorithms for non-stationary rested bandits. *arXiv preprint arXiv:1710.10657*, 2020.

- Dan Cosley, Shyong K. Lam, Istvan Albert, Joseph A. Konstan, and John Riedl. Is seeing believing? how recommender system interfaces affect users' opinions. In *Conference on Human Factors in Computing Systems*, 2003.
- Ofer Dekel, Jian Ding, Tomer Koren, and Yuval Peres. Bandits with switching costs: $T^{2/3}$ regret. In *Symposium on Theory of Computing*, 2014.
- Simon S. Du, Sham M. Kakade, Ruosong Wang, and Lin F. Yang. Is a good representation sufficient for sample efficient reinforcement learning? In *International Conference on Learning Representations*, 2020.
- Eyal Even-Dar, Shie Mannor, and Yishay Mansour. PAC bounds for multi-armed bandit and markov decision processes. In *Conference on Computational Learning Theory*, 2002.
- Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Symposium on Discrete Algorithms*, 2005.
- Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*, 2011.
- Elad Hazan and Satyen Kale. Online submodular minimization. *Journal of Machine Learning Research*, 13(93):2903–2922, 2012.
- Hoda Heidari, Michael Kearns, and Aaron Roth. Tight policy regret bounds for improving and decaying bandits. In *International Joint Conference on Artificial Intelligence*, 2016.
- Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.
- Chi Jin, Zeyuan Allen-Zhu, Sébastien Bubeck, and Michael I Jordan. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, 2018.
- Sham Kakade, Michael Kearns, and John Langford. Exploration in metric state spaces. In *International Conference on Machine Learning*, 2003.
- Michael Kearns, Yishay Mansour, and Andrew Y. Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. In *International Joint Conference on Artificial Intelligence*, 1999.
- Roberta L. Klatzky. *Human memory: structures and processes*. Freeman, 1980.
- R. Kleinberg and N. Immorlica. Recharging bandits. In *IEEE Symposium on Foundations of Computer Science*, 2018.
- Pierre Laforgue, Giulia Clerici, Nicolò Cesa-Bianchi, and Ran Gilad-Bachrach. Break your bandit routine with LSD rewards: a last switch dependent analysis of satiation and seasonality. *arXiv preprint arXiv:2110.11819*, 2021.
- T.L Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):422, 1985.

- Nir Levine, Koby Crammer, and Shie Mannor. Rotting bandits. In *Advances in Neural Information Processing Systems*, 2017.
- David Lindner, Hoda Heidari, and Andreas Krause. Addressing the long-term impact of ML decisions via policy regret. In *International Joint Conference on Artificial Intelligence*, 2021.
- Dhruv Malik, Aldo Pacchiano, Vishwak Srinivasan, and Yuanzhi Li. Sample efficient reinforcement learning in continuous state spaces: A perspective beyond linearity. In *International Conference on Machine Learning*, 2021.
- N. Merhav, E. Ordentlich, G. Seroussi, and M.J. Weinberger. On sequential strategies for loss functions with memory. *IEEE Transactions on Information Theory*, 48(7):1947–1958, 2002.
- Mehryar Mohri and Scott Yang. Competing with automata-based expert sequences. In *International Conference on Artificial Intelligence and Statistics*, 2018.
- Ciara Pike-Burke and Steffen Grunewalder. Recovering bandits. In *Advances in Neural Information Processing Systems*, 2019.
- Julien Seznec, Andrea Locatelli, Alexandra Carpentier, Alessandro Lazaric, and Michal Valko. Rotting bandits are no harder than stochastic ones. In *International Conference on Artificial Intelligence and Statistics*, 2019.
- Julien Seznec, Pierre Menard, Alessandro Lazaric, and Michal Valko. A single algorithm for both restless and rested rotting bandits. In *International Conference on Artificial Intelligence and Statistics*, 2020.
- Ayan Sinha, David F Gleich, and Karthik Ramani. Deconvolving feedback loops in recommender systems. In *Advances in Neural Information Processing Systems*, 2016.
- Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning*, 12(1-2):1–286, 2019.
- Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2018.
- Richard S Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 2000.
- Cem Tekin and Mingyan Liu. Online learning of rested and restless bandits. *IEEE Transactions on Information Theory*, 58(8):5588–5611, 2012.
- Ruosong Wang, Simon S Du, Lin Yang, and Sham Kakade. Is long horizon RL more difficult than short horizon RL? In *Advances in Neural Information Processing Systems*, 2020.
- Peter Whittle. Arm-Acquiring Bandits. *The Annals of Probability*, 9(2):284–292, 1981.

Appendix A. Analysis of Algorithm 1

In this section, we analyze the complete policy regret of Algorithm 1, and prove Theorem 1. Before we formally prove Theorem 1, we first state below two key lemmas, that will be useful throughout. The first lemma shows an equivalence between tallying bandit problems and Markov decision processes (MDPs) (Sutton and Barto, 2018). The second lemma verifies that the $N_{xy}(\pi)$ quantity defined in Algorithm 1 is well defined. We additionally introduce new quantities μ and π^* which will be useful for our proofs. With this outline in mind, let us begin the analysis.

Lemma 1 *Any (m, g, h) -tallying bandit problem can be equivalently expressed as a finite horizon Markov decision process (MDP).*

The proof of this Lemma 1 is given in Appendix A.9. For the sake of brevity, we have not stated the explicit details of the reduction (for instance, the definition of state space or transition function of the corresponding MDP) in the statement of this lemma. Nevertheless, these details are readily found in the proof.

Recall that in Section 4.1, to facilitate the definition of Algorithm 1 we defined the quantity $N_{xy}(\pi)$ via the following procedure. Execute π for $n + 1$ periods so that we have played the action sequence $\pi_1, \pi_2 \dots \pi_{n\sqrt{T}}, \pi_{n\sqrt{T}+1} \dots \pi_{(n+1)\sqrt{T}}$. Then use this action sequence to define

$$N_{xy}(\pi) = \frac{1}{\sqrt{T}} \sum_{t=n\sqrt{T}+1}^{(n+1)\sqrt{T}} \mathbb{I}(\pi_t = x) \cdot \mathbb{I}\left(y = \sum_{t'=\max\{1, t-m+1\}}^t \mathbb{I}(\pi_{t'} = x)\right).$$

The next lemma shows that $N_{xy}(\pi)$ is always well defined if $1 \leq n \leq \sqrt{T} - 1$ and $m \leq \sqrt{T}$.

Lemma 2 *Consider any (m, g, h) -tallying bandit problem where $m \leq \sqrt{T}$, and fix any \sqrt{T} -cyclic policy π , any $x \in \mathcal{X}$ and any $y \in \{1, 2 \dots m\}$. When defined via the aforementioned procedure, the quantity $N_{xy}(\pi)$ is well defined for any $1 \leq n \leq \sqrt{T} - 1$. Furthermore, $N_{xy}(\pi)$ is independent of any action sequence that was played before π was executed for $n + 1$ periods.*

The proof of this Lemma 2 is deferred to Appendix A.8. Now for each \sqrt{T} -cyclic policy π , we define the quantity $\mu(\pi)$ as follows

$$\mu(\pi) = \sum_{(x,y) \in \mathcal{X} \times \{1,2 \dots m\}} N_{xy}(\pi) h_x(y).$$

Via this definition and the result of Lemma 2, it is immediate that when $m \leq \sqrt{T}$, if we play an arbitrary action sequence and then execute π for $n + 1$ periods, then the (expected) cumulative loss experienced in the final period (i.e., the $(n + 1)$ th period) is $\mu(\pi)\sqrt{T}$. We use this notion of μ to define the policy π^* as

$$\pi^* \in \arg \min_{\pi \in A_1} \mu(\pi).$$

Recall for this definition that A_1 as defined in Algorithm 1 is the set of all \sqrt{T} -cyclic policies. With these definitions in hand, we are now in a position to formally prove Theorem 1.

A.1. Proof of Theorem 1

First, we note that if $m > \sqrt{T}$, then the statement of the theorem is trivially true since the complete policy regret is always upper bounded by T . Hence, for the remainder of the proof it suffices to assume that $m \leq \sqrt{T}$. For any policy π , which is a length T deterministic sequence of actions, let $\ell_t(\pi)$ denote the expected loss suffered at timestep t while playing π . Our next lemma shows that for any tallying bandit problem, the loss suffered by the optimal policy (i.e., the policy in \mathcal{X}^T that experiences the minimum cumulative expected loss) can be well approximated by $T\mu(\pi^*)$.

Lemma 3 *Given any (m, g, h) -tallying bandit problem, let π^{**} denote an optimal policy in \mathcal{X}^T . Then*

$$T\mu(\pi^*) - \sum_{t=1}^T \ell_t(\pi^{**}) \leq (m+1)\sqrt{T}.$$

The proof of Lemma 3 is deferred to Appendix A.7. Now let ℓ^s denote the loss experienced in epoch $s \in \{1, 2, \dots, S\}$ of Algorithm 1. The following lemma bounds the cumulative loss of Algorithm 1 relative to $T\mu(\pi^*)$.

Lemma 4 *Assume that $m \leq \sqrt{T}$. With probability at least $1 - \delta$, the total loss of Algorithm 1 relative to $T\mu(\pi^*)$ can be upper bounded as*

$$\sum_{s=1}^S \ell^s - T\mu(\pi^*) \leq Km\sqrt{T} \left(5 \log_2 \left(\frac{\sqrt{T}}{4Km} + 1 \right) + 400 \sqrt{\log \left(\frac{2Km \log(T)}{\delta} \right)} \right).$$

The proof of this Lemma 4 is provided in Appendix A.2. With the results of Lemma 3 and Lemma 4 in hand, we now utilize them to prove Theorem 1 as follows. Note that the complete policy regret $\mathcal{R}_T^{\text{cp}}$ of Algorithm 1 satisfies

$$\begin{aligned} \mathcal{R}_T^{\text{cp}} &= \sum_{s=1}^S \ell^s - \sum_{t=1}^T \ell_t(\pi^{**}) \\ &= \sum_{s=1}^S \ell^s - T\mu(\pi^*) + T\mu(\pi^*) - \sum_{t=1}^T \ell_t(\pi^{**}) \\ &\leq Km\sqrt{T} \left(5 \log_2 \left(\frac{\sqrt{T}}{4Km} + 1 \right) + 400 \sqrt{\log \left(\frac{2Km \log(T)}{\delta} \right)} \right) + (m+1)\sqrt{T} \\ &\leq 1200Km\sqrt{T} \left(\sqrt{\log(2Km \log(T)/\delta)} + \log_2 \left(\sqrt{T}/(2Km) \right) \right). \end{aligned}$$

This completes the proof of Theorem 1. ■

A.2. Proof of Lemma 4

To facilitate the proof, we require the following critical lemma, which bounds the loss incurred by Algorithm 1 in each epoch $s \in \{1, 2, \dots, S\}$. For the statement of the following lemma, note that completing any epoch $s \in \{1, 2, \dots, S\}$ takes a total of $T_s = 2n_s Km\sqrt{T}$ timesteps.

Lemma 5 Assume that $m \leq \sqrt{T}$. With probability at least $1 - \delta$, we have simultaneously for each epoch $s \in \{2, 3, \dots, S\}$ that the total loss relative to $T_s \mu(\pi^*)$ is bounded as

$$\ell^s - T_s \mu(\pi^*) \leq Km \left(\sqrt{T} + 8n_s \sqrt{T} C_{s-1} \right).$$

The proof of this Lemma 5 is provided in Appendix A.3. Observe that by the result of Lemma 5, we are guaranteed with probability at least $1 - \delta$ that

$$\begin{aligned} \sum_{s=1}^S \ell^s - T \mu(\pi^*) &= \sum_{s=1}^S (\ell^s - T_s \mu(\pi^*)) \\ &\leq 4Km \sqrt{T} + \sum_{s=2}^S (\ell^s - T_s \mu(\pi^*)) \\ &\leq 4Km \sqrt{T} + \sum_{s=2}^S Km \left(\sqrt{T} + 8n_s \sqrt{T} C_{s-1} \right) \\ &\leq 5SKm \sqrt{T} + 8Km \sum_{s=2}^S n_s \sqrt{T} C_{s-1}. \end{aligned} \tag{4}$$

Now substituting in the definitions

$$n_s = 2^s \text{ and } C_{s-1} = \sqrt{\frac{32Km}{n_{s-1} \sqrt{T}} \log \left(\frac{2KmS}{\delta} \right)} = \sqrt{\frac{64Km}{n_s \sqrt{T}} \log \left(\frac{2KmS}{\delta} \right)},$$

which were provided in Algorithm 1, into the final term on the RHS of Eq. (4) yields that

$$\begin{aligned} 8Km \sum_{s=2}^S n_s \sqrt{T} C_{s-1} &= 8Km \sum_{s=2}^S n_s \sqrt{T} \sqrt{\frac{64Km}{n_s \sqrt{T}} \log \left(\frac{2KmS}{\delta} \right)} \\ &= 64K^{1.5} m^{1.5} \sqrt{\log \left(\frac{2KmS}{\delta} \right)} \sum_{s=2}^S n_s \sqrt{T} \sqrt{\frac{1}{n_s \sqrt{T}}} \\ &= 64K^{1.5} m^{1.5} \sqrt{\log \left(\frac{2KmS}{\delta} \right)} T^{1/4} \sum_{s=2}^S \sqrt{n_s} \\ &= 64K^{1.5} m^{1.5} \sqrt{\log \left(\frac{2KmS}{\delta} \right)} T^{1/4} \sum_{s=2}^S 2^{s/2} \\ &\leq 400K^{1.5} m^{1.5} \sqrt{\log \left(\frac{2KmS}{\delta} \right)} T^{1/4} 2^{S/2}. \end{aligned}$$

Finally, we recall the definition of $S = \log_2 \left(\frac{\sqrt{T}}{4Km} + 1 \right)$ to observe that

$$\begin{aligned}
8Km \sum_{s=2}^S n_s \sqrt{T} C_{s-1} &\leq 400K^{1.5} m^{1.5} \sqrt{\log \left(\frac{2KmS}{\delta} \right)} T^{1/4} 2^{S/2} \\
&= 400K^{1.5} m^{1.5} \sqrt{\log \left(\frac{2KmS}{\delta} \right)} T^{1/4} \sqrt{\left(\frac{\sqrt{T}}{4Km} + 1 \right)} \\
&\leq 400K^{1.5} m^{1.5} \sqrt{\log \left(\frac{2KmS}{\delta} \right)} T^{1/4} \frac{T^{1/4}}{\sqrt{Km}} \\
&= 400Km \sqrt{\log \left(\frac{2KmS}{\delta} \right)} \sqrt{T} \\
&\leq 400Km \sqrt{\log \left(\frac{2Km \log(T)}{\delta} \right)} \sqrt{T}
\end{aligned} \tag{5}$$

Combining Eq. (4) with Eq. (5) yields the result. ■

A.3. Proof of Lemma 5

To facilitate the proof, we leverage the following critical lemma, which bounds the gap of the average value μ of policies in A_s versus $\mu(\pi^*)$.

Lemma 6 *Assume that $m \leq \sqrt{T}$. The event*

$$\bigcap_{s=1}^S \bigcap_{\pi \in A_s} \{ \mu(\pi) - \mu(\pi^*) \leq 4C_{s-1} \},$$

occurs with probability at least $1 - \delta$.

The proof of this Lemma 6 is provided in Appendix A.4. Let us now return to the main proof. For ease in notation, let ℓ^{sxy} denote the total loss experienced in epoch s of Algorithm 1 while executing the policy π_{sxy} for $2n_s$ epochs. Hence we have $\ell^s = \sum_{(x,y) \in \mathcal{X} \times \{1,2,\dots,m\}} \ell^{sxy}$.

Note that within a single epoch $s > 1$, for each $(x, y) \in \mathcal{X} \times \{1, 2, \dots, m\}$ we execute the policy π_{sxy} for $2n_s$ periods, where each period takes \sqrt{T} timesteps. Recall the fact that when $m \leq \sqrt{T}$, if we play an arbitrary action sequence and then execute π_{sxy} for $n + 1$ periods for $n \geq 1$, then the (expected) cumulative loss experienced in the final period (i.e., the $(n + 1)$ th period) is $\mu(\pi_{sxy})\sqrt{T}$. In particular, this fact implies that if we execute π_{sxy} for $2n_s$ periods, then the total loss experienced ℓ^{sxy} during these $2n_s$ periods is upper bounded by

$$\ell^{sxy} \leq \mu(\pi_{sxy})\sqrt{T}(2n_s - 1) + \sqrt{T} \leq 2n_s\mu(\pi_{sxy})\sqrt{T} + \sqrt{T}.$$

Hence we can use Lemma 6 to upper bound

$$\begin{aligned}
\ell^{sxy} - 2n_s\mu(\pi^*)\sqrt{T} &\leq 2n_s\mu(\pi_{sxy})\sqrt{T} + \sqrt{T} - 2n_s\mu(\pi^*)\sqrt{T} \\
&= \sqrt{T} + 2n_s\sqrt{T}(\mu(\pi_{sxy}) - \mu(\pi^*)) \\
&\leq \sqrt{T} + 8n_s\sqrt{T}C_{s-1}.
\end{aligned}$$

This bound holds uniformly for each $(x, y) \in \mathcal{X} \times \{1, 2 \dots m\}$, and hence we have that

$$\ell^s - T_s \mu(\pi^*) = \sum_{(x,y) \in \mathcal{X} \times \{1,2 \dots m\}} \left(\ell^{sxy} - 2n_s \mu(\pi^*) \sqrt{T} \right) \leq Km \left(\sqrt{T} + 8n_s \sqrt{T} C_{s-1} \right).$$

This completes the proof. \blacksquare

A.4. Proof of Lemma 6

To facilitate the proof, we require the following two critical helper results. The first result bounds the error incurred when estimating $\mu(\pi)$ via the stochastic realizations

$\left\{ \left\{ \tilde{h}_x(y)_{s,k} \right\}_{k=1}^{n_s N_{xy}(\pi_{sxy}) \sqrt{T}} \right\}_{(x,y) \in \mathcal{X} \times \{1,2 \dots m\}}$.

The second result shows that while running Algorithm 1, which is based on successive elimination of inferior policies over epochs $s \in \{1, 2 \dots S\}$, at any epoch s we never eliminate π^* from our set A_s of feasible policies.

Lemma 7 *Assume that $m \leq \sqrt{T}$. Fix any $s \in \{1, 2 \dots S\}$, and let B_s denote the event that for all $\pi \in A_s$ we simultaneously have that*

$$|\hat{\mu}_s(\pi) - \mu(\pi)| \leq C_s.$$

Then B_s occurs with probability at least $1 - \delta/S$.

Lemma 8 *Assume that $m \leq \sqrt{T}$. The event $\cap_{s=1}^S B_s$, where the event B_s is defined in Lemma 7, implies the event that*

$$\pi^* \in \cap_{s=1}^S A_s \text{ and } \cap_{s=1}^S \{0 \leq \hat{\mu}_s(\pi^*) - \hat{\mu}_s(\hat{\pi}_s) \leq 2C_s\}.$$

The proofs of Lemma 7 and Lemma 8 are provided in Appendix A.5 and Appendix A.6 respectively.

Let us now return to the proof. By the result of Lemma 7 and a union bound, the event $\cap_{s=1}^S B_s$ occurs with probability at least $1 - \delta$. Furthermore, the result of Lemma 8 shows that the event $\cap_{s=1}^S B_s$ implies the event

$$\pi^* \in \cap_{s=1}^S A_s. \tag{6}$$

So on the event $\cap_{s=1}^S B_s$, note that for any s and any $\pi \in A_s$ we have

$$\begin{aligned} \mu(\pi) - \mu(\pi^*) &\stackrel{(i)}{\leq} \hat{\mu}_{s-1}(\pi) - \mu(\pi^*) + C_{s-1} \\ &\stackrel{(ii)}{\leq} \hat{\mu}_{s-1}(\hat{\pi}_{s-1}) - \mu(\pi^*) + 3C_{s-1} \\ &\stackrel{(iii)}{\leq} \hat{\mu}_{s-1}(\pi^*) - \mu(\pi^*) + 3C_{s-1} \\ &\stackrel{(iv)}{\leq} \mu(\pi^*) - \mu(\pi^*) + 4C_{s-1} \\ &= 4C_{s-1}, \end{aligned}$$

where step (i) follows from Lemma 7, step (ii) follows from the definition of A_s and the fact that $\pi \in A_s$, step (iii) follows from the definition of $\hat{\pi}_{s-1}$ and Eq. (6), and step (iv) follows again from Lemma 7 and Eq. (6). This completes the proof. \blacksquare

A.5. Proof of Lemma 7

For the proof of this lemma, it is useful to define the quantity C_{sxy} as

$$C_{sxy} = \sqrt{\frac{32}{n_s N_{xy}(\pi_{sxy}) \sqrt{T}} \log\left(\frac{2KmS}{\delta}\right)},$$

for each $(s, x, y) \in \{1, 2 \dots S\} \times \mathcal{X} \times \{1, 2 \dots m\}$. Fix any $(x, y) \in \mathcal{X} \times \{1, 2 \dots m\}$. Note that by Hoeffding's bound (Hoeffding, 1963), we are guaranteed that the event

$$\left| h_x(y) - \frac{1}{n_s N_{xy}(\pi_{sxy}) \sqrt{T}} \sum_{k=1}^{n_s N_{xy}(\pi_{sxy}) \sqrt{T}} \tilde{h}_x(y)_{s,k} \right| \leq C_{sxy}, \quad (7)$$

occurs with probability at least $1 - \delta/(KmS)$. A union bound then ensures that the above event occurs simultaneously for all $(x, y) \in \mathcal{X} \times \{1, 2 \dots m\}$ with probability at least $1 - \delta/S$. We now claim that this (simultaneous) event is a subset of B , which is sufficient to complete the proof.

To establish the claim, note that on this event, we are guaranteed for any $\pi \in A_s$ that

$$\begin{aligned} & |\mu(\pi) - \hat{\mu}_s(\pi)| \\ &= \left| \sum_{(x,y) \in \mathcal{X} \times \{1,2\dots m\}} N_{xy}(\pi) h_x(y) - \sum_{(x,y) \in \mathcal{X} \times \{1,2\dots m\}} N_{xy}(\pi) \frac{1}{n_s N_{xy}(\pi_{sxy}) \sqrt{T}} \sum_{k=1}^{n_s N_{xy}(\pi_{sxy}) \sqrt{T}} \tilde{h}_x(y)_{s,k} \right| \\ &= \left| \sum_{(x,y) \in \mathcal{X} \times \{1,2\dots m\}} N_{xy}(\pi) \left(h_x(y) - \frac{1}{n_s N_{xy}(\pi_{sxy}) \sqrt{T}} \sum_{k=1}^{n_s N_{xy}(\pi_{sxy}) \sqrt{T}} \tilde{h}_x(y)_{s,k} \right) \right| \\ &\leq \sum_{(x,y) \in \mathcal{X} \times \{1,2\dots m\}} N_{xy}(\pi) C_{sxy}, \end{aligned}$$

where the final step follows from the triangle inequality and Eq. (7). Continuing the above, we have that

$$\begin{aligned} |\mu(\pi) - \hat{\mu}_s(\pi)| &\leq \sum_{(x,y) \in \mathcal{X} \times \{1,2\dots m\}} N_{xy}(\pi) C_{sxy} \\ &= \sqrt{\frac{32}{n_s \sqrt{T}} \log\left(\frac{2KmS}{\delta}\right)} \sum_{(x,y) \in \mathcal{X} \times \{1,2\dots m\}} \frac{N_{xy}(\pi)}{\sqrt{N_{xy}(\pi_{sxy})}} \\ &\stackrel{(i)}{\leq} \sqrt{\frac{32}{n_s \sqrt{T}} \log\left(\frac{2KmS}{\delta}\right)} \sum_{(x,y) \in \mathcal{X} \times \{1,2\dots m\}} \sqrt{N_{xy}(\pi)} \\ &\stackrel{(ii)}{\leq} \sqrt{Km} \sqrt{\frac{32}{n_s \sqrt{T}} \log\left(\frac{2KmS}{\delta}\right)} \\ &= C_s, \end{aligned}$$

where step (i) follows from the fact that $N_{xy}(\pi_{sxy}) \geq N_{xy}(\pi)$ by its definition in Algorithm 1, and step (ii) follows from the fact that $N_{xy}(\pi) \in [0, 1]$, that $\sum_{(x,y) \in \mathcal{X} \times \{1,2\dots m\}} N_{xy}(\pi) = 1$ as well as the Cauchy-Schwarz inequality. This establishes the claim and hence completes the proof. \blacksquare

A.6. Proof of Lemma 8

Assume that the event $\cap_{s'=1}^S B_{s'}$ is true. On this event, we prove the lemma by induction on s . First we demonstrate the base case of $s = 1$, which is that $\pi^* \in A_1$ and $0 \leq \widehat{\mu}_1(\pi^*) - \widehat{\mu}_1(\widehat{\pi}_1) \leq 2C_1$. Then for the inductive step we show that if the event $\pi^* \in A_{s-1}$ and $0 \leq \widehat{\mu}_{s-1}(\pi^*) - \widehat{\mu}_{s-1}(\widehat{\pi}_{s-1}) \leq 2C_{s-1}$ occurs, then we also have that the event

$$\pi^* \in A_s \text{ and } 0 \leq \widehat{\mu}_s(\pi^*) - \widehat{\mu}_s(\widehat{\pi}_s) \leq 2C_s,$$

is also true.

For the base case, note that by definition we are guaranteed $\pi^* \in A_1$. And by the definition of $\widehat{\pi}_1$, we know that $0 \leq \widehat{\mu}_1(\pi^*) - \widehat{\mu}_1(\widehat{\pi}_1)$. Furthermore, recalling the definition of the event B_1 in Lemma 7, on the event B_1 we have that

$$\widehat{\mu}_1(\pi^*) - \mu(\pi^*) \leq C_1 \text{ and } \mu(\widehat{\pi}_1) - \widehat{\mu}_1(\widehat{\pi}_1) \leq C_1.$$

Putting these equations together and using the fact that $\mu(\pi^*) \leq \mu(\widehat{\pi}_1)$ ensures that

$$\widehat{\mu}_1(\pi^*) - \widehat{\mu}_1(\widehat{\pi}_1) \leq 2C_1.$$

This verifies the base case.

For the inductive step, assume that $\pi^* \in A_{s-1}$ and $0 \leq \widehat{\mu}_{s-1}(\pi^*) - \widehat{\mu}_{s-1}(\widehat{\pi}_{s-1}) \leq 2C_{s-1}$ occurs. Then the definition of A_s and the inductive hypothesis directly imply that $\pi^* \in A_s$. Hence, it is true by definition of $\widehat{\pi}_s$ that $0 \leq \widehat{\mu}_s(\pi^*) - \widehat{\mu}_s(\widehat{\pi}_s)$. Then recalling the definition of the event B_s in Lemma 7, on the event B_s we have that

$$\widehat{\mu}_s(\pi^*) - \mu(\pi^*) \leq C_s \text{ and } \mu(\widehat{\pi}_s) - \widehat{\mu}_s(\widehat{\pi}_s) \leq C_s.$$

Putting these equations together and using the fact that $\mu(\pi^*) \leq \mu(\widehat{\pi}_s)$ ensures that

$$\widehat{\mu}_s(\pi^*) - \widehat{\mu}_s(\widehat{\pi}_s) \leq 2C_s.$$

This verifies the inductive step. As argued earlier, this is sufficient to complete the proof. \blacksquare

A.7. Proof of Lemma 3

Note that since $\mu(\pi^*) \in [0, 1]$, the statement is trivial for $m \geq \sqrt{T}$. Hence, assume for the remainder of the proof that $m < \sqrt{T}$. There exists some $k \in \{0, 1, \dots, \sqrt{T} - 1\}$ such that

$$\sum_{t=k\sqrt{T}+1}^{(k+1)\sqrt{T}} \ell_t(\pi^{**}) \leq \frac{1}{\sqrt{T}} \sum_{t=1}^T \ell_t(\pi^{**}).$$

Define the \sqrt{T} -cyclic policy π by letting $\pi_t = \pi_{k\sqrt{T}+t}^{**}$ for all $1 \leq t \leq \sqrt{T}$. Note that via the MDP characterization provided in Lemma 1, we can equivalently think of the tallying bandit problem as some MDP that we denote M . The proof of Lemma 1 shows that regardless of the state we start at, either playing $\pi_1, \pi_2, \dots, \pi_m$ or playing $\pi_{k\sqrt{T}+1}^{**}, \pi_{k\sqrt{T}+2}^{**}, \dots, \pi_{k\sqrt{T}+m}^{**}$ leads to the same state in M .

From that state, either playing $\pi_{m+1}, \pi_{m+2} \dots \pi_{\sqrt{T}}$ or playing $\pi_{k\sqrt{T}+m+1}^{**}, \pi_{k\sqrt{T}+m+2}^{**} \dots \pi_{(k+1)\sqrt{T}}^{**}$ leads to the same sequence of states, and hence the same sequence of (expected) losses. Hence we have shown that after a single first execution of π , we can bound the loss as

$$\sum_{t=1}^{\sqrt{T}} \ell_k(\pi) \leq m + \sum_{t=k\sqrt{T}+1}^{(k+1)\sqrt{T}} \ell_k(\pi^{**}) \leq m + \frac{1}{\sqrt{T}} \sum_{t=1}^T \ell_t(\pi^{**}).$$

Repeating this argument for \sqrt{T} executions of π , we have shown that

$$\sum_{t=1}^T \ell_k(\pi) \leq m\sqrt{T} + \sqrt{T} \sum_{t=k\sqrt{T}+1}^{(k+1)\sqrt{T}} \ell_k(\pi^{**}) \leq m\sqrt{T} + \sum_{t=1}^T \ell_t(\pi^{**}). \quad (8)$$

Now, we observe that $T\mu(\pi) \leq \sum_{t=1}^T \ell_k(\pi) + \sqrt{T}$, where we used the fact that when $m \leq \sqrt{T}$, if we play an arbitrary action sequence and then execute π for $n+1$ periods for $n \geq 1$, then the (expected) cumulative loss experienced in the final period (i.e., the $(n+1)$ th period) is $\mu(\pi)\sqrt{T}$. Finally, we note that $\mu(\pi^*) \leq \mu(\pi)$ by definition, and so we have that

$$T\mu(\pi^*) \leq T\mu(\pi) \leq \sum_{t=1}^T \ell_k(\pi) + \sqrt{T},$$

which combined with Eq. (8) implies that

$$T\mu(\pi^*) - \sum_{t=1}^T \ell_t(\pi^{**}) \leq (m+1)\sqrt{T}.$$

This completes the proof. ■

A.8. Proof of Lemma 2

To prove this result, we leverage the MDP characterization of tallying bandits provided by Lemma 1. Let M denote the MDP corresponding to the given (m, g, h) -tallying bandit problem. Note that by the proof of Lemma 1, and by the assumption that $m \leq \sqrt{T}$, after executing π for $n \geq 1$ periods we have arrived at the state $(\pi_{n\sqrt{T}-m+1} \dots \pi_{n\sqrt{T}})$. And since π is \sqrt{T} -cyclic we are guaranteed that

$$(\pi_{n\sqrt{T}-m+1} \dots \pi_{n\sqrt{T}}) = (\pi_{\sqrt{T}-m+1} \dots \pi_{\sqrt{T}}).$$

So if we execute π for one more period (i.e., the $(n+1)$ th period) from this starting state, then regardless of n we will observe an identical sequence of states, since the transition function of the MDP M is deterministic. Hence, the quantity

$$\sum_{t=n\sqrt{T}+1}^{(n+1)\sqrt{T}} \mathbb{I}(\pi_t = x) \cdot \mathbb{I}\left(y = \sum_{t'=\max\{1, t-m+1\}}^t \mathbb{I}(\pi_{t'} = x)\right),$$

used to define $N_{xy}(\pi)$ is independent of n , ensuring that $N_{xy}(\pi)$ is well defined. It remains to establish the claim that $N_{xy}(\pi)$ is independent of the action sequence that was played before π was

executed for $n + 1$ periods. Denote this prior action sequence as $a_{1:k}$ for any finite value of k . Note that regardless of what $a_{1:k}$ is, after we execute π for n periods we still arrive at the state

$$(\pi_{n\sqrt{T}-m+1} \cdots \pi_{n\sqrt{T}}) = (\pi_{\sqrt{T}-m+1} \cdots \pi_{\sqrt{T}}),$$

in the MDP M . Then if we execute π for one more period (i.e., the $(n + 1)$ th period) from this starting state, then regardless of $a_{1:k}$ we will observe an identical sequence of states, since the transition function of the MDP M is deterministic. This ensures that $N_{xy}(\pi)$ is well defined. ■

A.9. Proof of Lemma 1

Given an (m, g, h) -tallying bandit problem, for ease in notation let the (finite) action set \mathcal{X} be denoted as $\{1, 2 \dots K\}$. To define the MDP M , let its state space be $(\{0\} \cup \mathcal{X})^m$ and let its action space be \mathcal{X} . Let the initial state be the length m vector $(0, 0 \dots 0)$. For each state $s = (s_1, s_2 \dots s_m)$ and action i , define the deterministic transition function $\mathcal{T}_M : (\{0\} \cup \mathcal{X})^m \times \mathcal{X} \rightarrow (\{0\} \cup \mathcal{X})^m$ of the MDP as

$$\mathcal{T}_M(s, i) = (s_2, s_2 \dots s_{m-1}, s_m, i).$$

Let the (finite) horizon of the MDP be the time horizon T of the tallying bandit problem. Finally, define for each state s the (expected) reward function $R_M : (\{0\} \cup \mathcal{X})^m \rightarrow [0, 1]$ of the MDP as

$$R_M(s) = 1 - h_{s_m} \left(\sum_{t=1}^m \mathbb{I}(s_t = s_m) \right).$$

It is immediate the taking actions in the tallying bandit problem corresponds to taking actions in the state space of this MDP. ■

Appendix B. Proof of Theorem 2

First note that as an immediate consequence of Proposition 1, we must have $\mathbb{E} [\mathcal{R}_T^{\text{cp}}] \geq mK/128$. So for the remainder of the proof, we focus on showing that $\mathbb{E} [\mathcal{R}_T^{\text{cp}}] \geq c\sqrt{mKT}$ for some numerical constant $c > 0$. Also note that due to the result of Proposition 1, we can assume for this proof that $m \leq T/100$, because the complete policy regret scales linearly with T in the regime that $m > T/100$.

At a high level, our proof will proceed via a reduction to best arm identification in stochastic multi armed bandit problems (Slivkins, 2019). Roughly speaking, we will show the existence of an (m, g, h) -tallying bandit problem, such that minimizing the complete policy regret in this problem is at least as hard as identifying the best arm in a stochastic multi armed bandit problem with $\Theta(mK)$ arms.

To this end, we first recall Lemma 1, which was originally stated at the beginning of Appendix A and proved in Appendix A.9. We have restated it here for convenience since it shall be useful for our proof of Proposition 1.

Lemma 1 *Any (m, g, h) -tallying bandit problem can be equivalently expressed as a finite horizon Markov decision process (MDP).*

Now, construct an (m, g, h) -tallying bandit problem using the following procedure, where we assume that m is at least some sufficiently large universal constant. Sample (x^*, y^*) uniformly at random from $\mathcal{X} \times \{23m/24, 23m/24 + 1 \dots m\}$. Define $h_x(y) = 1/2$ for each $(x, y) \in \mathcal{X} \times \{1, 2 \dots m\}$ such that $(x, y) \neq (x^*, y^*)$. Also define $h_{x^*}(y^*) = 1/2 - \epsilon$ for some $\epsilon \in (0, 1)$ to be specified later.

We now define the stochastic bandit feedback model for this tallying bandit problem as follows. When the player plays action x , and this action x has been played a total of y times in the past m timesteps (including the current timestep), then the player receives as feedback a Bernoulli random variable with mean $h_x(y)$. It is immediate that this feedback model meets the criteria outlined in Definition 4.

Let us now upper bound the cumulative loss incurred by the optimal policy in this tallying bandit problem. To do so, consider the policy π , which is a length T sequence of actions, that we define as follows. Choose some $x \in \mathcal{X}$ such that $x \neq x^*$. We define π to choose action x^* for y^* timesteps and then choose action x for $m - y^*$ timesteps, and then repeat this length m sequence over and over. Recalling the definition of a \sqrt{T} -cyclic policy that was stated in Section 4.1, we can analogously say that π is an m -cyclic policy, such that within each period of length m it plays x^* for y^* times and then plays x for $m - y^*$ times.

By Lemma 1, there exists an MDP M that is equivalent to the constructed tallying bandit problem. Recalling the characterization of this MDP M provided in the proof of Lemma 1, let us understand the sequence of states in M that we arrive at when we follow π . In the first m timesteps, π plays x^* for y^* times and then plays x for $m - y^*$ times, and so we arrive at the state

$$(x^*, x^* \dots x^*, x, x \dots x) \equiv (x^*)^{y^*} \times x^{m-y^*}. \quad (9)$$

Then for the next y^* timesteps, π plays x^* repeatedly. This leads to the progression of states given by

$$\begin{aligned} & (x^*)^{y^*-1} \times x^{m-y^*} \times (x^*) \\ & (x^*)^{y^*-2} \times x^{m-y^*} \times (x^*)^2 \\ & (x^*)^{y^*-3} \times x^{m-y^*} \times (x^*)^3 \\ & \vdots \\ & (x^*) \times x^{m-y^*} \times (x^*)^{y^*-1} \\ & x^{m-y^*} \times (x^*)^{y^*}. \end{aligned} \quad (10)$$

Then for the next $m - y^*$ timesteps, it plays x , so that the state after these timesteps is

$$(x^*, x^* \dots x^*, x, x \dots x) \equiv (x^*)^{y^*} \times x^{m-y^*},$$

and we have arrived back at the state listed in Eq. (9).

Let us now use this insight to bound the expected loss incurred by following π . Critically, for every period of m timesteps after the very first period, Eq. (10) shows that we observe (stochastic

instantiations of) the loss value $h_{x^*}(y^*)$ for exactly $y^* \in \{23m/24, 23m/24 + 1 \dots m\}$ timesteps. We hence upper bound the expected cumulative loss incurred by π as

$$\begin{aligned}
 (1/2)m + ((1/2 - \epsilon)y^* + (1/2)(m - y^*))\frac{T - m}{m} &= m/2 + (-\epsilon y^* + (1/2)m)\frac{T - m}{m} \\
 &= -\epsilon y^*\frac{T - m}{m} + \frac{T - m}{2} + m/2 \\
 &\leq -\epsilon\frac{23m}{24}\frac{T - m}{m} + T/2 \\
 &= -\epsilon\frac{23(T - m)}{24} + T/2 \\
 &= T/2 - \epsilon 23T/24 + 23m\epsilon/24.
 \end{aligned} \tag{11}$$

Let us now consider the performance of any algorithm which attempts to solve this tallying bandit problem. Instead of the algorithm operating in the usual oracle model, where it might need m actions to arrive at a state that it deems beneficial, let us strengthen the algorithm by equipping it with a generative model. Concretely, we strengthen the algorithm so that at any timestep, it can query any state in M and receive a stochastic instantiation of the $h_x(y)$ loss value corresponding to that state. Given this generative model, it is immediate that an algorithm that is attempting to minimize its cumulative loss (or equivalently, minimize its complete policy regret), is attempting to maximize the number of times it queries states whose loss value is $h_{x^*}(y^*)$.

Hence, we can interpret this algorithm as running best arm identification on a classical stochastic multi armed bandit problem with $mK/24$ arms. Note that the number of arms in this stochastic multi armed bandit problem is $mK/24$, since (x^*, y^*) was sampled uniformly at random from a set of cardinality $mK/24$. The remainder of the argument closely follows that of Slivkins (Slivkins, 2019). So pick $\epsilon = \sqrt{cmK/T}$, for some numerical constant $c > 0$ whose precise value can be found in the proofs of Corollary 2.9 and Theorem 2.10 of Slivkins (Slivkins, 2019). These two results show that for each timestep less than or equal to T , with probability at least $1/12$ the algorithm does not select a state whose loss value is $h_{x^*}(y^*)$. In particular, this means that the expected cumulative loss of this algorithm is at least

$$(1/12)T/2 + (11/12)T(1/2 - \epsilon) = T/24 + 11T/24 - 11\epsilon T/12 = T/2 - 11\epsilon T/12. \tag{12}$$

Putting together Eq. (11) and Eq. (12), we hence have that the expected complete policy regret of this algorithm is lower bounded as

$$\begin{aligned}
 \mathbb{E} [\mathcal{R}_T^{\text{cp}}] &\geq T/2 - 11\epsilon T/12 - (T/2 - \epsilon 23T/24 + 23m\epsilon/24) \\
 &= -11\epsilon T/12 + \epsilon 23T/24 - 23m\epsilon/24 \\
 &= \epsilon T/24 - 23m\epsilon/24 \\
 &\geq c\sqrt{mKT},
 \end{aligned}$$

where in the final step we used our assumption that $m \leq T/100$, substituted in the definition of $\epsilon = \sqrt{cmK/T}$ and redefined the value of the numerical constant c . This completes the proof. ■

Appendix C. Proof of Proposition 1

First, we recall Lemma 1, which was originally stated at the beginning of Appendix A and proved in Appendix A.9. We have restated it here for convenience since it shall be useful for our proof of Proposition 1.

Lemma 1 *Any (m, g, h) -tallying bandit problem can be equivalently expressed as a finite horizon Markov decision process (MDP).*

We now formally define the algorithm ALG_{det} below in Algorithm 2. Subsequently, we prove the upper bound in Proposition 1, and then we prove the lower bound in Proposition 1.

Algorithm 2: ALG_{det}

Inputs: memory capacity m , time horizon T

- 1 **for** $x \in \mathcal{X}$ **do**
- 2 **for** $y \in \{1, 2 \dots m\}$ **do**
- 3 Choose action x .
- 4 Observe and store $h_x(y)$.
- 5 **end**
- 6 **end**
- 7 Plan (offline) an optimal policy $\bar{\pi} = \{\bar{\pi}_k\}_{k=mK+1}^T$ to play for remaining $T - mK$ timesteps.
- 8 Choose actions according to $\bar{\pi}$ for remaining $T - mK$ timesteps.

C.1. Proof of Upper Bound in Proposition 1

First note that the offline planning step in Algorithm 2 is statistically (although perhaps not computationally) feasible, since the player has full information about the loss functions once it stores $h_x(y)$ for each $(x, y) \in \mathcal{X} \times \{1, 2 \dots m\}$.

Let π^{**} denote the optimal policy for the problem, so that $\pi^{**} \in \mathcal{X}^T$ is a length T sequence of actions such that following this sequence achieves the minimum loss. Let $\ell_t(\pi^{**})$ denote the loss incurred at the t th timestep while playing the action sequence $\{\pi_t^{**}\}_{t=1}^T$. Let $\ell(\bar{\pi})$ denote the total loss incurred by the final step of Algorithm 2 which starts playing $\bar{\pi}$ at timestep $mK + 1$.

To complete the proof, we will leverage the MDP characterization of tallying bandit problems that was established in the proof of Lemma 1. Let M denote the equivalent MDP for this tallying bandit problem. Observe that $\bar{\pi}$ is the optimal policy for the remaining $T - mK$ timesteps assuming that $\bar{\pi}$ is initialized at the initial state $s = (K, K \dots K)$ in this MDP M . Also note that playing the action sequence $\{\pi_t^{**}\}_{t=mK+1}^T$ from the state s leads to a cumulative loss that is upper bounded by

$$m + \sum_{t=m(K+1)+1}^T \ell_t(\pi^{**}),$$

which follows because regardless of the initial state, we arrive at state $s' = (\pi_{mK+1}^{**}, \pi_{mK+2}^{**} \dots \pi_{m(K+1)}^{**})$ in M after playing the length m action sequence $\{\pi_t^{**}\}_{t=mK+1}^{m(K+1)}$. From state s' , the losses we ex-

perience when playing $\{\pi_t^{**}\}_{t=m(K+1)+1}^T$ at each timestep $t \geq m(K+1)+1$ are exactly $\ell_t(\pi^{**})$. The optimality of $\bar{\pi}$ hence implies that

$$\ell(\bar{\pi}) \leq m + \sum_{t=m(K+1)+1}^T \ell_t(\pi^{**}). \quad (13)$$

Hence, by naively upper bounding the loss incurred in the first mK timesteps, the complete policy regret of Algorithm 2 is upper bounded via Eq. (13) as

$$\begin{aligned} \mathcal{R}_T^{\text{cp}} &\leq mK + \ell(\bar{\pi}) - \sum_{t=mK+1}^T \ell_t(\pi^{**}) \\ &\leq mK + m + \sum_{t=m(K+1)+1}^T \ell_t(\pi^{**}) - \sum_{t=mK+1}^T \ell_t(\pi^{**}) \\ &= mK + m - \sum_{t=mK+1}^{m(K+1)} \ell_t(\pi^{**}) \\ &\leq m(K+1). \end{aligned}$$

This verifies the upper bound in Proposition 1. ■

C.2. Proof of Lower Bound in Proposition 1

Construct an (m, g, h) -tallying bandit problem via the following procedure. Sample an action x^* uniformly at random from \mathcal{X} , and keep its identity hidden from the user. Define the functions $\{h_x\}_{x \in \mathcal{X}}$ as

$$h_{x^*}(y) = \begin{cases} 1 & \text{if } y < m \\ 0 & \text{if } y = m \end{cases} \quad \text{and } h_x = 1 \text{ if } x \neq x^*.$$

It is immediate that the optimal policy always plays the action x^* , and its cumulative loss is precisely $m - 1$. Meanwhile, to obtain zero loss at any timestep, the player must identify x^* . Note that to identify whether a certain action x equals x^* , the player must play x for m consecutive times, in order to receive the feedback $h_x(m)$. Since there are K actions, and identifying whether an action is the correct one requires m queries, a standard counting argument (Du et al., 2020) reveals that (in expectation over a possibly randomized strategy) the player makes at least $mK/64$ queries before observing $h_{x^*}(m)$. Hence, the (expected) complete policy regret of the (possibly randomized) player is lower bounded as

$$\mathbb{E} [\mathcal{R}_T^{\text{cp}}] \geq mK/64 - (m - 1) \geq mK/128,$$

where we assume that K is larger than some numerical constant. This verifies the lower bound in Proposition 1. ■