Weighted Tallying Bandits:
Overcoming Intractability via Repeated Exposure Optimality

Dhruv Malik [1] **Conor Igoe** [1] **Yuanzhi Li** [1] **Aarti Singh** [1]

Abstract

In human-interactive applications of online learning, a human's preferences or abilities are often a function of the algorithm's recent actions. Motivated by this, a significant line of work has formalized settings where an action's loss is a function of the number of times it was played in the prior m timesteps, where m corresponds to a bound on human memory capacity. To more faithfully capture decay of human memory with time, we introduce the Weighted Tallying Bandit (WTB), which generalizes this setting by requiring that an action's loss is a function of a *weighted* summation of the number of times it was played in the last m timesteps. WTB is intractable without further assumption. So we study it under Repeated Exposure Optimality (REO), a condition requiring the existence of an action that when repetitively played will eventually yield smaller loss than any other action sequence. We study the minimization of complete policy regret (CPR), which is the strongest notion of regret, in WTB under REO. Since m is often unknown, we only assume access to an upper bound M on m. We show that for problems with K actions and horizon T, a simple modification of the successive elimination algorithm has $\widetilde{\mathcal{O}}\left(\sqrt{KT} + (m+M)K\right)$ CPR. Upto an additive (in lieu of mutliplicative) factor in $(m+M)K$, this recovers the classical guarantee for the far simpler stochastic multi-armed bandit with traditional regret. We additionally show that in our setting, any algorithm will suffer additive CPR of $\widetilde{\Omega}\left(mK + M\right)$, demonstrating our result is near optimal. Our method is computationally efficient, and we experimentally demonstrate its practicality and superiority over various baselines.

[1]Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA. Correspondence to: Dhruv Malik <dhruvm@andrew.cmu.edu>.

1. Introduction

When an online learning algorithm is deployed in an interactive application, its decisions impact the state of its environment. In turn, this impacts the quality of subsequent decisions made by the algorithm. This is especially true in human-centered applications such as recommender systems or crowdsourcing. For instance, consider a crowdsourcing setting where at each timestep we want to select a worker to perform a task, without prior knowledge of any worker's ability. The task may be complex or require some fine-tuning, and each worker might need a calibration period where they repeatedly perform the task, before they start exhibiting their true ability. The existence of such a calibration period has been extensively demonstrated in visuomotor tasks (Adams, 1961), such as throwing darts (Wunderlich et al., 2020) or shooting a basketball (Phatak et al., 2020). Hence, an algorithm that asks workers to alternately perform the task, without intelligently allowing each worker time to calibrate themselves to the task, may bias its estimation of each worker's true ability. This interaction between an algorithm and its environment distinguishes it from classical non-interactive frameworks such as the multi-armed bandit.

To capture one aspect of this interactivity, a significant research thrust in online learning has studied settings where an action's loss is described by the number of times that action was recently played in the prior m timesteps (Heidari et al., 2016; Levine et al., 2017; Seznec et al., 2019; 2020; Lindner et al., 2021; Awasthi et al., 2022; Malik et al., 2022). The quantity m typically corresponds to a bound on human memory capacity or capability. For example, in the aforementioned scenario m would be the number of timesteps required by a worker to fine-tune and calibrate themselves to the task, before revealing their true ability.

Of course, such settings are an approximation to reality. For instance, psychological research shows that humans typically have better memory for more recently occurring events (Klatzky, 1980; Ricker et al., 2016). So if we play an action once in the previous m timesteps, its impact on the present may greatly differ depending on whether it was played on the previous timestep or m timesteps ago. In the context of the aforementioned crowdsourcing setting, a worker may need a shorter calibration period if they per-

formed the task on the previous timestep, as opposed to many timesteps ago. However, prior formalizations are oblivious to this difference. Motivated by these considerations, we make the following contributions:

- We introduce the Weighted Tallying Bandit (WTB), which generalizes prior formalizations by requiring that an action's loss is described by a *weighted summation* of the number of times that action was played in the prior m timesteps. Since this setting is dynamic and interactive, we eschew the traditional regret, and instead study the minimization of the strongest notion of regret known as the complete policy regret (CPR).

- We show that minimizing CPR in WTB is generally intractable. So we study it under the additional condition of Repeated Exposure Optimality (REO), which enforces the existence of an action that when repetitively played m times will yield smaller loss than other action sequences. In the context of the aforementioned example, REO is interpreted as the existence of a worker that once calibrated to the task, will perform better than other (calibrated or uncalibrated) workers. We motivate this condition via literature on human physiology.

- For WTB problems with K actions and horizon T that satisfy REO, and in the regime where only an upper bound M on the true value of m is known, we show that a slight modification of the classical successive elimination algorithm achieves a CPR guarantee (upto a logarithmic factor) of $\widetilde{\mathcal{O}}\left(\sqrt{KT} + (m + M)K\right)$. Besides an additive factor in $(m + M)K$, this matches the lower bound on the weaker traditional regret of the stochastic multi-armed bandit (which is the special $m = 1$ case of WTB with REO).

- One may desire an algorithm that is fully adaptive to m and requires no such upper bound M. We show this is impossible. Concretely, we show that any algorithm with sublinear CPR must require such an upper bound M, and then show that a linear dependency on this input M is necessary. This implies a lower bound of $\widetilde{\Omega}\left(\sqrt{KT} + mK + M\right)$ on the achievable CPR in our setting, highlighting our algorithm's near optimality.

- Via numerical simulations, we demonstrate our method's efficiency and superiority over various baselines.

2. Problem Formulation

2.1. Weighted Tallying Bandit

We begin by formally introducing the Weighted Tallying Bandit as an online learning game with bandit feedback over time horizon T, where the player has an action set $\mathcal{X}$ with finite cardinality K. A long line of prior work has studied the scenario where an action's loss at any timestep is a function of the number of timesteps it was played in the prior m timesteps (Heidari et al., 2016; Levine et al., 2017; Seznec et al., 2019; 2020; Lindner et al., 2021; Awasthi et al., 2022; Malik et al., 2022). We refer to these settings as "tallying" settings. Our goal is to generalize this, to the case where an action's loss is a function of a weighted tally of the number of times it was played in the past m timesteps.

To this end, we first introduce some notation. Assume the player has played the game for t timesteps, and for each timestep $1 \leq t' \leq t$ the player plays action $a_{t'}$. For a fixed action $x \in \mathcal{X}$, we define the vector $y^{t,x,m} \in \{0,1\}^m$ in a componentwise fashion as follows. For each component $1 \leq i \leq m$, let $y_i^{t,x,m} = \mathbb{I}(a_{t-i+1} = x)$ if $t - i + 1 \geq 1$ else let $y_i^{t,x,m} = 0$ if $t - i + 1 < 1$. Hence, the vector $y^{t,x,m}$ stores the timesteps where action x was played in the previous m timesteps upto (and including) the current timestep t. With this notation in hand, we are now in a position to formally define the Weighted Tallying Bandit.

Definition 2.1 (Weighted Tallying Bandit (WTB))**.** An online learning game is said to be an (m, w, h)-weighted tallying bandit with memory capacity m, if there exists an integer $m \geq 1$, a set of vectors $\{w_x\}_{x \in \mathcal{X}} \subset (0, 1]^m$, and a set of functions $\{h_x\}_{x \in \mathcal{X}}$ each mapping from $\mathbb{R}$ to $[0, 1]$, such that the following is true. For each $x \in \mathcal{X}$, the expected loss incurred at timestep t by playing action $a_t = x$ is given by $h_x\left(w_x^\top y^{t,x,m}\right)$, and the player observes as feedback a random observation $\widetilde{h}_x(w_x^\top y^{t,x,m}) \in [0, 1]$, that is independent of all other random observations, and satisfies $\mathbb{E}\left[\widetilde{h}_x(w_x^\top y^{t,x,m})\right] = h_x\left(w_x^\top y^{t,x,m}\right)$.

The quantities $m, \{w_x\}_{x \in \mathcal{X}}, \{h_x\}_{x \in \mathcal{X}}$ are all generally unknown, and the player only learns about them via bandit feedback over time. When $m = 1$, then WTB recovers the stochastic multi-armed bandit (sMAB) (Lai & Robbins, 1985; Auer et al., 2002a). However, WTB with $m \geq 2$ is often a better model for human-centered domains that require calibration. To see this, let us concretize the crowdsourcing setting introduced in Section 1. Assume the task to be performed is throwing a dart at a dartboard, and each worker is a different darts player. Without prior knowledge of any player's true ability to hit the dartboard, our goal is to discover which of the K players is best, by picking (at each timestep) a player to throw a dart and seeing whether they hit or miss. At first glance, this appears to be an sMAB problem, where each player has some true ability, which we stochastically observe each time we query the player.

Unfortunately, this sMAB formulation is agnostic to the calibration period that darts players require before they can exhibit their true performance. The existence of such a

calibration period has been demonstrated in the literature on visuomotor calibration. For instance, Wunderlich et al. (2020) show that when professional darts players toss darts in a row, the first toss is significantly less accurate than the remainder of the darts, although the performance stabilizes after the first dart toss. They attribute this phenomenon to the *warm-up decrement* (Adams, 1961; Anshel & Wrisberg, 1993; Anshel, 1995), which describes the decline in performance due to a break in a specific motor skill, as well as its recovery once the skill is resumed. Simply put, a player performs better once they are "in motion" and have fine-tuned their movement parameters after their first toss.

This phenomenon affects the design of algorithms for our darts setting, since we do not observe the true performance of a dart player until after their first toss. Furthermore, this *cannot* be resolved by simply having each player toss once, so that they are calibrated, and then running a standard sMAB algorithm while assuming the players stay calibrated forever. Indeed, Wunderlich et al. (2020) demonstrate that even small interruptions in the dart tosses (such as the time required for the player to retrieve their darts from the board) can cause the player to "reset", and subconsciously lose their fine-tuned movement parameters. Hence, the sMAB is hence a poor model for this setting. By contrast, the WTB with $m \geq 2$ is a more faithful model, since m describes the number of times a player must toss a dart in a row before we (stochastically) observe their true performance. The "reset" phenomenon that exists in this motivating example (as well as our forthcoming examples) requires that if we model this problem with WTB, then m should be non-trivially smaller than the horizon T. We assume this throughout our paper.

WTB more naturally models this phenomenon than the aforementioned tallying settings (Heidari et al., 2016; Levine et al., 2017; Seznec et al., 2019; 2020; Lindner et al., 2021; Awasthi et al., 2022; Malik et al., 2022), which are all special cases of the WTB where w_x is the all ones vector $\vec{1}$ for each $x \in \mathcal{X}$. As a stylized example, assume the task is shooting basketball free-throws, and that we need to find the best of two players x_1, x_2. Consider two different sequences of selecting players — x_1, x_2, x_1 versus x_2, x_1, x_1. Phatak et al. (2020) show that players require a calibration period of length at least 3 while shooting free-throws, and that their shooting performance improves with each successive free-throw. This implies that picking x_1, x_2, x_1 (i.e., x_1 shoots, then x_2, then x_1 again) will cause x_1 to have a worse expected performance on her final shot, relative to her performance if we select x_2, x_1, x_1 (i.e., x_2 shoots, then x_1 shoots twice). If we model this with WTB where $m = 3$ and $w_x = \vec{1}$, then we cannot distinguish these two scenarios, since in both cases x_1 shot twice in the past m timesteps. By contrast, WTB with $w \neq \vec{1}$ allows us to model different losses for these two scenarios. For instance, if we set $w_{x_1} = [1, 1/2, 1/4]$ and $h_{x_1}(z) = 1 - z/3$, then this model

says that selecting x_2, x_1, x_1 will cause x_1 to have better performance on her final shot than if we selected x_1, x_2, x_1.

More broadly, WTB significantly generalizes prior tallying settings, by allowing us to better approximate the decay in memory strength that occurs with passage of time, that has been documented extensively by studies on human memory (Klatzky, 1980; Ricker et al., 2016). This more naturally models the human-centered applications that motivate tallying settings. For instance, Malik et al. (2022) motivate their study via recommender systems, arguing that recommended content impacts human preferences, and assume the quantity $m \ll T$ bounds the length of time that a human remembers past recommendations. But their formulation is agnostic to how recently a piece of content was recommended *within* this window of length m. So if some content was recommended k times in the past m timesteps, then their framework requires that this incurs the same loss regardless of the ordering of those k recommendations. This is rather limiting, since human preferences today may depend only mildly on recommendations that occurred $\widetilde{\Omega}(m)$ timesteps ago. Our WTB formulation is more fine grained, and allows for the possibility of different losses incurred by each of the different orderings of those k recommendations.

2.2. Complete Policy Regret

A key property of WTB is that the loss incurred by an action depends on the past actions of the algorithm. In such dynamic scenarios, it has been established that the traditional regret is inappropriate to measure an algorithm's performance (Arora et al., 2012). Instead, one opts for the stronger notion of policy regret (Cesa-Bianchi et al., 2013; Arora et al., 2018). In line with prior work on tallying settings (Heidari et al., 2016; Levine et al., 2017; Seznec et al., 2019; 2020; Lindner et al., 2021; Awasthi et al., 2022; Malik et al., 2022), we study the minimization of the complete policy regret (CPR), which is the strongest possible notion of regret. Given an (m, w, h)-weighted tallying bandit and an algorithm that plays action sequence $(a_1, a_2 \ldots a_T) \in \mathcal{X}^T$, the CPR $\mathcal{R}^{\mathrm{cp}}$ of the algorithm is defined as

$$\mathcal{R}^{\mathrm{cp}} = \sum_{t=1}^{T} h_{a_t}\left(w_{a_t}^\top y^{t, a_t, m}\right) -$$

$$\min_{(x_1, x_2 \ldots x_T) \in \mathcal{X}^T} \sum_{t=1}^{T} h_{x_t}\left(w_{x_t}^\top y^{t, x_t, m}\right). \quad (1)$$

Following prior convention, we refer to any length T sequence of actions (i.e., an element of $\mathcal{X}^T$) as a policy. The CPR is hence the algorithm's cumulative loss, relative to the minimum loss achieved by the best policy in $\mathcal{X}^T$. Minimizing CPR is thus equivalent to minimizing the cumulative loss of the algorithm, and we note that this performance metric is identical to the one used in reinforcement learn-

ing (Jin et al., 2018; Wang et al., 2020; Igoe et al., 2022). If the CPR of an algorithm is sublinear in T and polynomial in m, K then we say it has statistically efficient CPR.

Prior work has shown that in the case of WTB with $w_x = \vec{1}$ for each $x \in \mathcal{X}$, without any further assumption, there exists an algorithm with statistically efficient CPR (Malik et al., 2022). Unfortunately, the following result shows that such an algorithm does not exist in WTB with $w_x \neq \vec{1}$.

Proposition 2.2. *For any $m \geq 1$, there exists an (m, w, h)-weighted tallying bandit with $K = 2$ such that the following is true. Any (possibly randomized) algorithm has expected CPR satisfying $\mathbb{E}\left[\mathcal{R}^{cp}\right] = \widetilde{\Omega}\left(\min\{2^m, T\}/m\right)$.*

The proof of Proposition 2.2 is deferred to Appendix C. At a high level, the proof shows that if $w_x \neq \vec{1}$ then h_x can take on $\widetilde{\Omega}\left(2^m\right)$ different values, and so discovering the optimal sequence of actions requires $\widetilde{\Omega}\left(2^m\right)$ queries. This result demonstrates that if we desire an algorithm with statistically efficient CPR, then we must impose structure on the WTB setting that restricts the set of optimal action sequences. We motivate and formalize such structure in the sequel.

2.3. Repeated Exposure Optimality

To motivate additional structure in the types of problems that are modeled by WTB, we recall the darts setting illustrated in Section 2.1. Notably, if we ask a player to toss darts in a row, then on the *first* toss, their uncalibrated performance is poor and not necessarily indicative of their subsequent performance. But on successive tosses after the first toss, Wunderlich et al. (2020) show that their calibrated performance stabilizes and is *better* than the uncalibrated performance on the first toss. A similar observation holds for shooting free-throws (Phatak et al., 2020). So if we let $x^\star \in \mathcal{X}$ denote the player with the best calibrated performance, then this implies that the calibrated performance $x^\star$ is better than not only the calibrated performances of player $x \neq x^\star$, but also the uncalibrated performances of all players. We formalize this insight in the following condition.

Definition 2.3 (Repeated Exposure Optimality (α-REO)). An (m, w, h)-weighted tallying bandit satisfies the Repeated Exposure Optimality condition with parameter α, if there exists action $x^\star \in \mathcal{X}$, such that for each $x \in \mathcal{X}$ and each $y \in \{1\} \times \{0, 1\}^{m-1}$ we have $h_{x^\star}\left(\|w_{x^\star}\|_1\right) \leq h_x\left(w_x^\top y\right) + \alpha$.

The α-REO condition thus requires that there is some action $x^\star \in \mathcal{X}$, which when played repetitively for at least m times in a row, will have smaller loss (upto the suboptimality α) than other action sequences. Two remarks are in order, to understand this condition in the context of prior work. First, observe that even when we additionally impose the α-REO condition on WTB, the sMAB remains a special case of this setting via a choice of $\alpha = 0, m = 1$. Second, significant prior work on tallying settings has focused on when the

loss functions $\{h_x\}_{x \in \mathcal{X}}$ are monotonic. For instance, the improving bandit (Heidari et al., 2016) is a special case of WTB under significant additional restrictions, including (but not limited to) the facts that $\{w_x\}_{x \in \mathcal{X}} = \{\vec{1}\}$ and $\{h_x\}_{x \in \mathcal{X}}$ are decreasing. We note that this property of decreasing $\{h_x\}_{x \in \mathcal{X}}$ functions is a special case of the 0-REO condition.

We have motivated REO via the warm-up decrement phenomenon documented in the psycho-physiological literature. And we believe REO may be relevant in other interactive settings such as recommender systems, as we discuss in Section 6. Yet, we acknowledge our setting fails to model many subtleties that arise in human-centered applications. A complete study of these subtleties is beyond the scope of our paper, and we relegate discussion of avenues for future work to Section 6. With the REO condition thus formalized, we devote the remainder of our paper to the following question:

Fix any (m, w, h)-weighted tallying bandit satisfying α-REO. Is there a computationally efficient algorithm, with statistically efficient CPR, that solves this problem?

3. Main Results

We present two categories of results. In Section 3.1 we present a statistically and computationally efficient algorithm that can solve WTB problems satisfying REO. This method requires only an upper bound M on the true memory capacity m, whose exact value is often unknown. In Section 3.2, we show the impossibility of an algorithm that is fully adaptive to an unknown m (i.e., does not require knowledge of an upper bound $M < T$ on m). We also show that if such an upper bound $M < T$ on m is known, then the dependency of our method on M is optimal.

3.1. An Algorithm for WTB with REO

Before we present our algorithm, we consider some natural approaches. Since WTB is a subclass of reinforcement learning (RL) problems, one may attempt to use RL algorithms to solve it. But even when $\{w_x\}_{x \in \mathcal{X}} = \{\vec{1}\}$, such algorithms suffer $\widetilde{\Omega}\left(K^m\right)$ CPR (Awasthi et al., 2022; Malik et al., 2022). One may also attempt to extend the classical UCB algorithm from sMAB to WTB as follows. Solve the problem in epochs of length m, where at the beginning of each epoch, we select the action that minimizes the usual UCB estimate, and then play it m times in a row instead of just once. Then we record the loss observed in the most recent play, since this is an unbiased estimate of the action's eventual loss, and use it to update the action's UCB estimate. While this seems like a reasonable heuristic, each epoch has an m-length overhead which substantially increases regret.

A different idea is to adapt algorithms from prior talling settings for our problem. But prior tallying settings that are

---

Algorithm 1 Successive Elimination for WTB with REO

Require: upper bound M on memory capacity m, time horizon T, failure probability tolerance $\delta \in (0, 1)$

1: Define $S = \log_2 \left(\frac{T}{4KM} + 1 \right)$ and $A_1 = \mathcal{X}$.

2: Define $n_s = KM2^s/|A_s|$ and $C_s = \sqrt{\frac{32}{n_s} \log \left(\frac{2KS}{\delta} \right)}$.

3: **for** $s \in \{1, 2 \dots S\}$ **do**

4:     **for** $x \in A_s$ **do**

5:         Execute action x for $n_s \geq m$ times.

6:         Execute action x for n_s times and store $\{\widetilde{h}_x(\|w_x\|_1)_{s,k}\}_{k=1}^{n_s}$.

7:         Define $\widehat{\mu}_s(x) = \frac{1}{n_s} \sum_{k=1}^{n_s} \widetilde{h}_x(\|w_x\|_1)_{s,k}$.

8:     **end for**

9:     Select $\widehat{x}_s \in \arg\min_{x \in A_s} \widehat{\mu}_s(x)$.

10:    Construct $A_{s+1} = \{x \in A_s \text{ s.t. } \widehat{\mu}_s(x) \leq \widehat{\mu}_s(\widehat{x}_s) + 2C_s\}$.

11: **end for**

---

comparable to ours have CPR bounds that scale *multiplicatively* with m (see Section 5 for details). We show that due to the additional presence of REO, we can solve not just these tallying settings but also WTB with a CPR guarantee that is only *additive* in m. The algorithm that achieves this bound is a slightly modified version of successive elimination (SE), and is presented in Algorithm 1. This is inspired by Malik et al. (2022), who adapt SE for their tallying bandit setting, although their modification is more involved. By contrast, our modification is simple, since REO permits us to only estimate the eventual loss of each action. We now present our main result, which bounds this algorithm's CPR.

Theorem 3.1. *Fix any (m, w, h)-weighted tallying bandit problem satisfying Repeated Exposure Optimality with parameter α. When Algorithm 1 is run with inputs $M \geq m$ and $\delta \in (0, 1)$, then with probability at least $1 - \delta$ it has complete policy regret upper bounded as*

$$\mathcal{R}^{cp} \leq 4KM + Km\log(T) + 800\sqrt{KT \log\left(2K\log(T)/\delta\right)} + \alpha T. \quad (2)$$

The proof of Theorem 3.1 is deferred to Appendix A. Let us highlight some key aspects of this result.

Comparison to sMAB & Tallying Settings. Recall that in the classical sMAB, which is a special case of WTB with 0-REO via $m = 1$, any algorithm suffers $\widetilde{\Omega}\left(\sqrt{KT}\right)$ traditional regret. Theorem 3.1 thus shows that the much larger class of WTB with $\widetilde{\mathcal{O}}\left(\sqrt{K/T}\right)$-REO problems can be solved with essentially this guarantee on CPR, upto a logarithmic factor and an additive dependence on mK. Our guarantee scales more favorably than those obtained for prior comparable tallying settings (see Section 5 for details).

Efficiency & Practicality. Algorithm 1 is computationally

efficient and scalable. Its total runtime over T iterations is $\widetilde{\mathcal{O}}\left(T + K\log(T)\right)$ and the space complexity required at any timestep is $\widetilde{\mathcal{O}}(K)$. This is in contrast to results on prior comparable tallying settings (see Section 5 for details). Moreover, implementing Algorithm 1 does not require exact knowledge of unknown quantities such as $\{h_x\}_{x \in \mathcal{X}}$, $\{w_x\}_{x \in \mathcal{X}}$, α or m; an upper bound M on m suffices. While Algorithm 1 appears to require the time horizon T as an input, we note that the method is already performing a doubling trick. This means that for any T representable on a computer (say $T \leq 2^{64} = 2^{2^6}$), and since $C_s = \widetilde{\mathcal{O}}(\log\log T)$, a short numerical computation reveals that redefining C_s as $\sqrt{\frac{64}{n_s} \log(2K/\delta)}$ and picking $\delta < 0.009$ ensures that we can get the same CPR bound (upto constants) as Eq. (2), even without providing T as an input.

Statistical Optimality In Various Regimes. In the regime where m is known (so $M = m$) and REO is satisfied with $\alpha = 0$, the guarantee of Theorem 3.1 is optimal within a single logarithmic factor. To see this, note that in the RHS of Eq. (2), the Km term cannot be improved due to Proposition 1 of Malik et al. (2022), and the $\sqrt{KT}$ term is of course tight due to the classical sMAB lower bound. Moreover, when m is known and REO is satisfied with $\alpha = \widetilde{\Theta}\left(\sqrt{mK/T}\right)$, then the proof of Theorem 2 of Malik et al. (2022) shows that there is a regime of non-trivial $0 < \alpha \ll 1$ where the dependence on αT in Theorem 3.1 cannot be improved, and so Theorem 3.1 is optimal (within a logarithmic factor). We note that it is unclear whether the αT term in Eq. (2) is optimal for *all* $\alpha > 0$, and investigating this is an interesting future direction. We defer our investigation into the optimal dependency on M (when m is unknown) to Section 3.2.

Best Arm Identification. Algorithm 1 can also be used to identify actions whose eventual loss is near that of $x^\star$. In particular, after T rounds (or S epochs), with probability at least $1 - \delta$ any action $x \in A_{S+1}$ satisfies $h_x(\|w_x\|_1) \leq h_{x^\star}(\|w_{x^\star}\|_1) + 4C_s = h_{x^\star}(\|w_{x^\star}\|_1) + \widetilde{\mathcal{O}}\left(\sqrt{K/T}\right)$.

The proof of Theorem 3.1 requires care to ensure optimal dependencies, but the technique is standard, and our contribution is not a novel analysis route. Rather, our contribution is to demonstrate that a classical algorithm for the canonical sMAB can be easily adapted to solve a much more general, and ostensibly more complex, class of problems that are practically motivated. Prior tallying settings that are comparable to WTB have inherent computational and statistical difficulties (see Section 5 for details). Our formalization of REO and Theorem 3.1 identifies well motivated structure that permits statistically and computationally efficient solutions to problems arising in interactive domains.

3.2. Adaptivity To Memory Capacity

While Algorithm 1 does not require knowledge of the true memory capacity m, it does require an upper bound M on m. Theorem 3.1 suggests that the CPR of Algorithm 1 scales linearly in this input M, which is disadvantageous when it is difficult to non-trivially upper bound m. In general, we desire an algorithm which scales more favorably (or not at all) with the input M. For instance, this could be achieved via an algorithm that maintains a confidence interval of the true value m, and adaptively queries to refine its estimate of m, in order to improve or remove its dependency on M. We now show that such an algorithm cannot exist, even in the simpler "tallying setting" that is a special case of WTB, and in the case when REO is satisfied with parameter $\alpha = 0$.

To this end, we introduce some notation. For any positive integers T, M, K with $M \leq T$, let $\mathsf{UTB}_{T,M,K}$ denote the set of unweighted tallying bandit problems (i.e., WTB problems where w_x is the all ones vector for each action x), that each have horizon length T, number of actions K, and memory capacity $m \in \{1, 2 \ldots M\}$, and that satisfy 0-REO. For any possibly randomized algorithm $\mathcal{A}$ and any unweighted tallying bandit problem tb, let m_{tb} denote the memory capacity of tb, and let $\mathcal{R}^{\mathrm{cp}}(\mathcal{A}, \mathsf{tb})$ denote the expected CPR of algorithm $\mathcal{A}$ when it is used to solve tb. And for a choice of $\epsilon = (\epsilon_1, \epsilon_2, \epsilon_3)$ satisfying $\epsilon_1, \epsilon_2 \in (0, 1)$ and $\epsilon_3 \in [0, \epsilon_2)$, and a choice of function $f : \mathbb{R}^2 \to \mathbb{R}$, let $\overline{\mathcal{A}}_{\epsilon, f}$ be the set of algorithms $\mathcal{A}$ which, when given as input any positive integers T, M, K with $M \leq T$ (and no other information), satisfy for each problem instance $\mathsf{tb} \in \mathsf{UTB}_{T,M,K}$ that $\mathbb{E}[\mathcal{R}^{\mathrm{cp}}(\mathcal{A}, \mathsf{tb})] \leq \min\left\{T/4, f(m_{\mathsf{tb}}, K)\left(T^{1-\epsilon_1} + T^{\epsilon_3}M^{1-\epsilon_2}\right)\right\}$.

An algorithm $\mathcal{A}$ in the set $\overline{\mathcal{A}}_{\epsilon, f}$ thus has a benign dependence on M in the following sense. When given positive integers T, M, K with $M \leq T$, and a problem instance $\mathsf{tb} \in \mathsf{UTB}_{T,M,K}$, the algorithm $\mathcal{A}$ does not a priori know the memory capacity m_{tb} of tb, and only knows the upper bound M. Nonetheless, the CPR of $\mathcal{A}$ when solving tb scales *sublinearly* in M. Unfortunately, the following result demonstrates that such an algorithm does not exist.

Theorem 3.2. *For each ϵ satisfying $\epsilon_1, \epsilon_2 \in (0, 1)$ and $\epsilon_3 \in [0, \epsilon_2)$, and each function f, the corresponding set $\overline{\mathcal{A}}_{\epsilon, f}$ is the empty set.*

The proof is deferred to Appendix B. The result reveals a "price for adaptivity" (see, for e.g., Locatelli & Carpentier (2018) for similar results in a different context), showing that if we only have a bound M on the unknown true memory capacity, then any algorithm's CPR cannot be sublinear in both M and T. We concretize this via two corollaries. The following corollary is stated for when we have no nontrivial bound on the memory capacity (i.e., that $M = T$).

Corollary 3.3. *Fix any function $f : \mathbb{R}^2 \to \mathbb{R}$. There is no*

(possibly randomized) algorithm which has expected CPR bounded by $\widetilde{o}(T)f(m_{\mathsf{tb}}, K)$ for each $\mathsf{tb} \in \mathsf{UTB}_{T,T,K}$.

The result shows it is impossible to have an algorithm whose CPR is sublinear in T for all unweighted tallying bandit instances tb with horizon T that satisfy 0-REO, even at the expense of arbitrarily poor dependence on m_{tb}, K. So, to obtain a sublinear CPR guarantee of the sort afforded by Theorem 3.1, it is necessary to have knowledge of a bound $M < T$ on the true memory capacity. The next corollary is stated for when we have such a non-trivial bound $M < T$.

Corollary 3.4. *Fix any function $f : \mathbb{R}^2 \to \mathbb{R}$. There is no (possibly randomized) algorithm, which given an input M, has expected CPR bounded by $f(m_{\mathsf{tb}}, K)\left(\widetilde{o}(T) + \widetilde{o}(M)\right)$ for each $\mathsf{tb} \in \mathsf{UTB}_{T,M,K}$.*

The result thus shows that we cannot hope to have an algorithm with sublinear dependency on both M and T for all unweighted tallying bandit instances tb with horizon T that satisfy 0-REO, even at the expense of arbitrarily bad dependence on K, m_{tb}. Note that ignoring logarithmic factors, Theorem 3.1 shows that Algorithm 1 has $\widetilde{\mathcal{O}}\left(\sqrt{KT} + K(M + m_{\mathsf{tb}})\right)$ CPR for each $\mathsf{tb} \in \mathsf{UTB}_{T,M,K}$. Combined with our earlier discussion of Theorem 3.1, Corollary 3.4 thus shows that any algorithm must suffer $\widetilde{\Omega}\left(\sqrt{KT} + m_{\mathsf{tb}}K + M\right)$ CPR, highlighting the near optimality of Algorithm 1 for WTB problems satisfying 0-REO, even in the regime when we only have an upper bound M on the unknown true memory capacity.

4. Numerical Results

In this section, we evaluate the performance of Algorithm 1 (denoted SE), in different domains which are modeled as WTB problems satisfying REO. In each domain, we compare this performance to the following baselines — (A) The EXP3 algorithm (Auer et al., 2002b), which has sublinear traditional regret in our setting (B) The batched version of EXP3 described by Arora et al. (2012), denoted as EXP3B, which has sublinear CPR in our setting (C) The modified UCB algorithm described in Section 3.

4.1. Synthetic Losses on Unweighted Tallying Bandit

We let $\{w_x\}_{x \in \mathcal{X}} = \{\vec{1}\}$, and fix some $x^\star \in \mathcal{X}$. We define $h_x = 0.5$ for each $x \in \mathcal{X}$, with the modification that $h_{x^\star}(\|w_{x^\star}\|_1) = 0.35$. So the losses are identical, except until we play $x^\star$ at least m times, implying that this instance satisfies 0-REO. To define the feedback model, we say the random variable $\widetilde{h}_x(w_x^\top y^{t,x,m})$ has distribution Bernoulli($h_x(w_x^\top y^{t,x,m})$). When $m = 1$, this is a hard instance for sMAB (Slivkins, 2019), and UCB is optimal. For $m > 1$, we note that the UCB variant will perform best when $h_x \approx h_{x^\star}(\|w_{x^\star}\|_1)$, since the losses incurred dur-

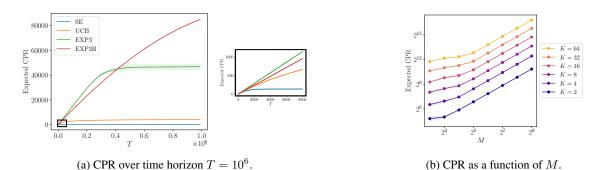(a) CPR over time horizon $T = 10^6$.

(b) CPR as a function of M.

Figure 1: We plot the expected CPR of each algorithm. In both plots, each datapoint is obtained by averaging over 20 problem instances, and the shaded region depicts ± 1 standard error around the mean. In (a) we fix $K = 5$, $m = 3$ and $M = 3$. In (b) we fix $m = 4$ and $T = 10^6$.

ing the m-length overhead are nearly equal to the eventual losses of repetitively playing $x^\star$. Hence, we consider our experimental design to be as favorable to the UCB variant as possible. In Figure 1a, we plot the expected CPR of each method over time. As expected, SE outperforms each baseline. In Appendix D.1, we present similar results for other choices of m, K, M, and also for a problem where α-REO is satisfied with $\alpha > 0$. Separately, we study SE's CPR as a function of its input M, for the same fixed m, T, K. In Figure 1b, we observe that SE's CPR is at most a linear function of M, as predicted by Theorem 3.1.

4.2. Synthetic Losses on Weighted Tallying Bandit

We now consider a WTB problem satisfying REO where $\{w_x\}_{x \in \mathcal{X}} \neq \{\vec{1}\}$. We relegate the discussion of the precise loss functions used to Appendix D.2. The optimal policy, and hence the CPR, are difficult to compute for this problem. So in lieu of CPR, we plot the expected cumulative loss of each algorithm in excess of SE's loss (hence the CPR at any time is obtained by applying a constant shift to each algorithm's excess loss). The results are shown in Figure 2a, and demonstrate our method's superiority over the baselines.

4.3. Simulated Dart Throwing Tournament

Motivated by prior work showing the existence of a calibration period in motor tasks (Adams, 1961; Phatak et al., 2020; Wunderlich et al., 2020), we simulate a simplified dart throwing tournament with $K = 20$ players. As discussed in Section 2.3, Wunderlich et al. (2020) show that while a player's first toss is uncalibrated and not necessarily indicative of their subsequent performance, in immediately subsequent tosses the performance calibrates, stabilizes and is better than that of the first toss. We model each (random) instance of the tournament as a WTB with $m = 2$ and arbitrary w, where each player $x \in \mathcal{X}$ has expected loss function sampled from $h_x(w_{x,1}) \sim \text{Unif}[0.68, 0.72]$ and $h_x(\|w_x\|_1) \sim \text{Unif}[0.58, 0.62]$. We obtained the bounds

for these distributions from Wunderlich et al. (2020), who showed that most players' average performance was concentrated in these intervals. To define the feedback model, we require the random variable $\widetilde{h}_x(w_x^\top y^{t,x,m})$ has distribution Bernoulii($h_x(w_x^\top y^{t,x,m})$). While our experimental design eschews some real world subtleties that may occur while throwing darts (for instance, missing a throw might affect the player's confidence on the next throw), we believe it reasonably models the calibration period required to throw darts optimally. In Figure 2b, we plot the CPR of each method over time. SE outperforms each baseline.

5. Related Work

Tallying Settings with $m = T$. A significant thrust of prior work studies tallying settings that are special cases of WTB with $\{w_x\}_{x \in \mathcal{X}} = \{\vec{1}\}$, and require that $m = T$ (Heidari et al., 2016; Levine et al., 2017; Seznec et al., 2019; 2020; Lindner et al., 2021; Metelli et al., 2022). Of course, to ensure tractability they enforce various additional types of assumptions, typically in the form of monotonicity on the $\{h_x\}_{x \in \mathcal{X}}$ functions. Results here do not apply to the case when $m < T$, because $m < T$ causes complications in the design of algorithms since an action's loss "resets" if it is not played. Since our paper is primarily motivated by applications where $m < T$, we do not view these works as directly comparable to ours. Nevertheless, we note that upto an additive factor in mK and a logarithmic factor, the CPR guarantees in all these works generally scale less favorably than the rates provided by our Theorem 3.1.

Tallying Settings with $m < T$. A different body of prior work studies tallying settings that are special cases of WTB with $\{w_x\}_{x \in \mathcal{X}} = \{\vec{1}\}$, and like us, they are motivated by applications where $m < T$ (Awasthi et al., 2022; Malik et al., 2022). These settings are more comparable to ours, since they do not enforce that $m = T$. The tallying bandit (Malik et al., 2022) makes no assumptions be-

(a) Excess loss in WTB with $\{w_x\}_{x \in \mathcal{X}} \neq \{\vec{1}\}$.
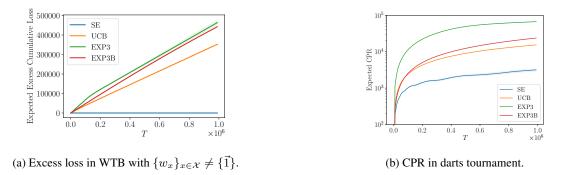

(b) CPR in darts tournament.

Figure 2: In (a), we plot as a function of time the expected cumulative loss of each algorithm in excess of that of SE, in the WTB instance where $\{w_x\}_{x \in \mathcal{X}} \neq \{\vec{1}\}$ described in Section 4.2, with $K = 5$, $m = 4$ and $M = 4$. In (b), we plot as a function of time the expected CPR of each algorithm in the simulated darts tournament described in Section 4.3, and truncate the y-axis below 10^2 for illustrative purpose. In both (a) & (b), data is obtained by averaging over 20 problem instances, and the shaded region depicts ± 1 standard error around the mean.

yond $\{w_x\}_{x \in \mathcal{X}} = \{\vec{1}\}$, and here any algorithm must suffer $\widetilde{\Omega}\left(\sqrt{mKT}\right)$ CPR. They adapt successive elimination, and our algorithm is heavily inspired by theirs. The congested bandit (Awasthi et al., 2022) specializes the tallying bandit by requiring that the $\{h_x\}_{x \in \mathcal{X}}$ functions are increasing, and even here the best known upper bound is $\widetilde{\mathcal{O}}\left(\sqrt{mKT}\right)$. The best CPR bounds of these settings thus seem to scale multiplicatively with m. Moreover, the computational complexities of the best known algorithms in these settings are exponential in T, m respectively. By contrast, our REO condition allows for a computationally efficient method with a near optimal CPR guarantee that is only additive in m.

Related Non-Tallying Settings. A massive body of work studies settings where the loss of each action evolves over time in some structured fashion (for instance, according to a stochastic process or according to the number of timesteps since the action was last played (Whittle, 1981; Garivier & Moulines, 2011; Tekin & Liu, 2012; Besbes et al., 2014; Bouneffouf & Féraud, 2016; Kleinberg & Immorlica, 2018; Basu et al., 2019; Pike-Burke & Grunewalder, 2019; Cella & Cesa-Bianchi, 2020; Cortes et al., 2020; Laforgue et al., 2021). The models for the evolution of loss in all these works are different than our (weighted) tallying setting.

Policy Regret. Many works study policy regret against generic m-memory bounded adversaries, and their algorithms apply to our setting (Arora et al., 2012; Cesa-Bianchi et al., 2013; Dekel et al., 2014; Arora et al., 2018; Mohri & Yang, 2018). However, these results are agnostic to our problem's structure, and a direct application would result in a suboptimal CPR bound that is worse than our Theorem 3.1.

6. Discussion

In this paper, we formulated the Weighted Tallying Bandit, which generalizes prior tallying settings so that the loss at a timestep is a function of a weighted summation of the number of times it was recently played. To ensure tractability, we introduced the Repeated Exposure Optimality condition, which we motivated via human-centered applications where one's best performance requires a calibration period before it stabilizes. We showed that a simple modification of the classical successive elimination algorithm solves this setting, and our numerical results demonstrate its superiority over alternative baselines. Finally, we showed that while our algorithm required as input a non-trivial upper bound $M < T$ on m, any algorithm with sublinear CPR requires such an input, and our method's dependency on this input M is optimal. This implies our algorithm's CPR is near optimal for WTB problems satisfying 0-REO.

Our work is primarily theoretical, so we do not foresee short-term negative societal impact. In the long-term, our work could increase the deployment of interactive algorithms amongst humans. To ensure this deployment is benign, it is necessary (but not sufficient) to understand the power and limitations of such algorithms, which is the aim of our work.

We acknowledge our work has certain limitations. From a theoretical perspective, while there is a regime of non-trivial α (see Section 3.1) where Theorem 3.1's dependence on α cannot be improved, it is unclear whether this dependence is optimal for *all* values of α. Investigating this is an interesting future direction. More practically, a limitation is that while our setting models calibration periods, it fails to model other subtleties arising in the human-centered domains that motivate our work. For e.g., in strenuous tasks, a calibrated individual may fatigue after repeatedly performing a task for long $m < T$. Here, the best model for losses associated

with repeatedly playing an action is an initial period where the individual calibrates and their performance improves to "sweet-spot", but then performance deteriorates as fatigue accumulates (this is analogous to the $m = T$ setting considered by Lindner et al. (2021)). Our setting handles the initial calibration period, but cannot handle the latter deterioration phase. Handling this is a key direction for future work.

Our concrete motivating examples for REO are primarily derived from the psycho-physiological literature. Nevertheless, we expect it may additionally apply to recommender systems. For e.g., it is plausible that a user needs to see a type of content multiple times before she decides her preferences for it, but if she is not shown the content for a while, then she forgets its details and requires re-calibration to re-affirm her preference for that content relative to more recent recommendations. In this case, the goal is to explore her *eventual* preferences, and then repetitively select the item she *eventually* prefers most. This setting is reasonably modeled via WTB with REO. Incorporating this idea into a recommender system is an interesting practical direction.

Acknowledgements

References

Adams, J. A. The second facet of forgetting: A review of warmup decrement. *Psychological Bulletin*, 58:257–273, 1961.

Anshel, M. H. Examining warm-up decrement as a function of interpolated open and closed motor tasks: Implications for practice strategies. *Journal of Sports Sciences*, 13: 247–256, 1995.

Anshel, M. H. and Wrisberg, C. A. Reducing warm-up decrement in the performance of the tennis serve. *Journal of Sport & Exercise Psychology*, 15(3):290–303, 1993.

Arora, R., Dekel, O., and Tewari, A. Online bandit learning against an adaptive adversary: From regret to policy regret. In *International Conference on Machine Learning*, 2012.

Arora, R., Dinitz, M., Marinov, T. V., and Mohri, M. Policy regret in repeated games. In *Advances in Neural Information Processing Systems*, 2018.

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

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002b.

Awasthi, P., Bhatia, K., Gollapudi, S., and Kollias, K. Congested bandits: Optimal routing via short-term resets. In *International Conference on Machine Learning*, 2022.

Basu, S., Sen, R., Sanghavi, S., and Shakkottai, S. Blocking bandits. In *Advances in Neural Information Processing Systems*, 2019.

Besbes, O., Gur, Y., and Zeevi, A. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in Neural Information Processing Systems*, 2014.

Bouneffouf, D. and Féraud, R. Multi-armed bandit problem with known trend. *Neurocomputing*, 205(C):16–21, 2016.

Cella, L. and Cesa-Bianchi, N. Stochastic bandits with delay-dependent payoffs. In *International Conference on Artificial Intelligence and Statistics*, pp. 1168–1177, 2020.

Cesa-Bianchi, N., Dekel, O., and Shamir, O. Online learning with switching costs and other adaptive adversaries. In *Advances in Neural Information Processing Systems*, 2013.

Cortes, C., DeSalvo, G., Kuznetsov, V., Mohri, M., and Yang, S. Discrepancy-based algorithms for non-stationary rested bandits. *arXiv preprint arXiv:1710.10657*, 2020.

Dekel, O., Ding, J., Koren, T., and Peres, Y. Bandits with switching costs: $T^{2/3}$ regret. In *Symposium on Theory of Computing*, 2014.

Du, S. S., Kakade, S. M., Wang, R., and Yang, L. F. Is a good representation sufficient for sample efficient reinforcement learning? In *International Conference on Learning Representations*, 2020.

Garivier, A. and Moulines, E. On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*, 2011.

Heidari, H., Kearns, M., and Roth, A. Tight policy regret bounds for improving and decaying bandits. In *International Joint Conference on Artificial Intelligence*, 2016.

Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Igoe, C., Ghods, R., and Schneider, J. Multi-agent active search: A reinforcement learning approach. *IEEE Robotics and Automation Letters*, 7(2):754–761, 2022.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, 2018.

Klatzky, R. L. *Human memory: structures and processes*. Freeman, 1980.

Kleinberg, R. and Immorlica, N. Recharging bandits. In *IEEE Symposium on Foundations of Computer Science*, 2018.

Laforgue, P., Clerici, G., Cesa-Bianchi, N., and Gilad-Bachrach, R. Break your bandit routine with LSD rewards: a last switch dependent analysis of satiation and seasonality. *arXiv preprint arXiv:2110.11819*, 2021.

Lai, T. and Robbins, H. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1): 4–22, 1985.

Levine, N., Crammer, K., and Mannor, S. Rotting bandits. In *Advances in Neural Information Processing Systems*, 2017.

Lindner, D., Heidari, H., and Krause, A. Addressing the long-term impact of ML decisions via policy regret. In *International Joint Conference on Artificial Intelligence*, 2021.

Locatelli, A. and Carpentier, A. Adaptivity to smoothness in x-armed bandits. In *Conference On Learning Theory*, 2018.

Malik, D., Li, Y., and Singh, A. Complete policy regret bounds for tallying bandits. In *Conference on Learning Theory*, 2022.

Metelli, A. M., Trovò, F., Pirola, M., and Restelli, M. Stochastic rising bandits. In *International Conference on Machine Learning*, 2022.

Mohri, M. and Yang, S. Competing with automata-based expert sequences. In *International Conference on Artificial Intelligence and Statistics*, 2018.

Phatak, A., Mujumdar, U., Rein, R., Wunderlich, F., Garnica, M., and Memmert, D. Better with each throw — a study on calibration and warm-up decrement of real-time consecutive basketball free throws in elite nba athletes. *German Journal of Exercise and Sport Research*, 50:273–279, 2020.

Pike-Burke, C. and Grunewalder, S. Recovering bandits. In *Advances in Neural Information Processing Systems*, 2019.

Ricker, T. J., Vergauwe, E., and Cowan, N. Decay theory of immediate memory: From brown (1958) to today (2014). *Quarterly Journal of Experimental Psychology*, 69(10): 1969–1995, 2016.

Seznec, J., Locatelli, A., Carpentier, A., Lazaric, A., and Valko, M. Rotting bandits are no harder than stochastic ones. In *International Conference on Artificial Intelligence and Statistics*, 2019.

Seznec, J., Menard, P., Lazaric, A., and Valko, M. A single algorithm for both restless and rested rotting bandits. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Slivkins, A. Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning*, 12(1-2):1–286, 2019.

Tekin, C. and Liu, M. Online learning of rested and restless bandits. *IEEE Transactions on Information Theory*, 58 (8):5588–5611, 2012.

Wang, R., Du, S. S., Yang, L., and Kakade, S. Is long horizon RL more difficult than short horizon RL? In *Advances in Neural Information Processing Systems*, 2020.

Whittle, P. Arm-Acquiring Bandits. *The Annals of Probability*, 9(2):284–292, 1981.

Wunderlich, F., Heuer, H., Furley, P., and Memmert, D. A serial-position curve in high-performance darts: The effect of visuomotor calibration on throwing accuracy. *Psychological Research*, 84:2057–2064, 2020.

A. Analysis of Algorithm 1

In this section, we analyze the complete policy regret of Algorithm 1, and prove Theorem 3.1. As discussed in Section 3, our analysis is overall rather standard, although we require a careful choice of parameters to ensure optimal dependencies in the final result. Thus, many of the computations closely follow those of Malik et al. (Malik et al., 2022), but we nevertheless provide the entire argument for the sake of completeness. Before we formally prove Theorem 3.1, we first introduce the function $\mu : \mathcal{X} \to [0, 1]$ which will be useful for our proofs. For any action $x \in \mathcal{X}$ let us define $\mu(x)$ as

$$\mu(x) = h_x(\|w_x\|_1).$$

And for each epoch $s \in \{1, 2 \dots S\}$ in the outer loop of Algorithm 1, define $T_s = 2|A_s|n_s$, where A_s, n_s are defined in Algorithm 1. With this definition in hand, we are now in a position to formally prove Theorem 3.1.

A.1. Proof of Theorem 3.1

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. Define the policy $\pi^\star$ as

$$\pi^\star \in \arg\min_{\pi \in \mathcal{X}^T} \sum_{t=1}^{T} \ell_t(\pi),$$

so that $\pi^\star$ is an optimal policy (i.e., a policy that suffers the minimum cumulative expected loss). Note that the definition of an (m, w, h)-weighted tallying bandit ensures that for any timestep t, there exists an action $x \in \mathcal{X}$ and $y \in \{1\} \times \{0, 1\}^{m-1}$ such that $\ell_t(\pi^\star) = h_x(w_x^\top y)$. Thus, the α-REO condition ensures that

$$\mu(x^\star) = h_{x^\star}(\|w_{x^\star}\|_1) \le h_x(w_x^\top y) + \alpha = \ell_t(\pi^\star) + \alpha.$$

In particular, this implies that

$$\sum_{t=1}^{T} \ell_t(\pi^\star) \ge T\mu(x^\star) - \alpha T. \tag{3}$$

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 $T\mu(x^\star)$.

Lemma A.1. *With probability at least $1 - \delta$, the total loss of Algorithm 1 relative to $T\mu(x^\star)$ can be upper bounded as*

$$\sum_{s=1}^{S} \ell^s - T\mu(x^\star) \le 4KM + Km\log(T) + 800\sqrt{KT\log\left(\frac{2K\log(T)}{\delta}\right)}.$$

The proof of this Lemma A.1 is provided in Appendix A.2. With the result of Lemma A.1 in hand, we now utilize it to prove Theorem 3.1 as follows. Note via Eq. (3) and Lemma A.1 that the complete policy regret $\mathcal{R}^{\text{cp}}$ of Algorithm 1 satisfies

$$\mathcal{R}^{\text{cp}} = \sum_{s=1}^{S} \ell^s - \sum_{t=1}^{T} \ell_t(\pi^\star)$$

$$\le \sum_{s=1}^{S} \ell^s - T\mu(x^\star) + \alpha T$$

$$\le 4KM + Km\log(T) + 800\sqrt{KT\log\left(\frac{2K\log(T)}{\delta}\right)} + \alpha T.$$

This completes the proof of Theorem 3.1. ∎

A.2. Proof of Lemma A.1

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 $T_s = 2|A_s|n_s$ timesteps.

Lemma A.2. *With probability at least* $1 - \delta$*, we have simultaneously for each epoch* $s \in \{2, 3 \ldots S\}$ *that the total loss relative to* $T_s\mu(x^\star)$ *is bounded as*

$$\ell^s - T_s\mu(x^\star) \leq |A_s| \left(m + 4(2n_s - m)C_{s-1}\right).$$

The proof of this Lemma A.2 is provided in Appendix A.3. Observe that by the result of Lemma A.2, we are guaranteed with probability at least $1 - \delta$ that

$$
\begin{aligned}
\sum_{s=1}^{S} \ell^s - T\mu(x^\star) &= \sum_{s=1}^{S} \left(\ell^s - T_s\mu(x^\star)\right) \\
&\leq 4KM + \sum_{s=2}^{S} \left(\ell^s - T_s\mu(x^\star)\right) \\
&\leq 4KM + \sum_{s=2}^{S} |A_s| \left(m + 4(2n_s - m)C_{s-1}\right) \\
&\leq 4KM + SKm + 8\sum_{s=2}^{S} |A_s| n_s C_{s-1}.
\end{aligned}
\tag{4}
$$

Recall the definitions $n_s = KM2^s/|A_s|$ and $T_s = 2|A_s|n_s$ provided in Algorithm 1. Also note that

$$C_{s-1} = \sqrt{\frac{32}{n_{s-1}} \log\left(\frac{2KS}{\delta}\right)} = \sqrt{\frac{32}{n_s|A_s|/(2|A_{s-1}|)} \log\left(\frac{2KS}{\delta}\right)} = \sqrt{\frac{64|A_{s-1}|}{n_s|A_s|} \log\left(\frac{2KS}{\delta}\right)}.$$

Substituting the above relations into the final term on the RHS of Eq. (4), we get that

$$
\begin{aligned}
8\sum_{s=2}^{S} |A_s| n_s C_{s-1} &= 8\sum_{s=2}^{S} |A_s| n_s \sqrt{\frac{64|A_{s-1}|}{n_s|A_s|} \log\left(\frac{2KS}{\delta}\right)} \\
&= 8\sqrt{\log\left(\frac{2KS}{\delta}\right)} \sum_{s=2}^{S} |A_s| n_s \sqrt{\frac{64|A_{s-1}|}{n_s|A_s|}} \\
&= 8\sqrt{\log\left(\frac{2KS}{\delta}\right)} \sum_{s=2}^{S} |A_s| \sqrt{n_s} \sqrt{\frac{64|A_{s-1}|}{|A_s|}} \\
&= 8\sqrt{\log\left(\frac{2KS}{\delta}\right)} \sum_{s=2}^{S} |A_s| \sqrt{\frac{KM2^s}{|A_s|}} \sqrt{\frac{64|A_{s-1}|}{|A_s|}} \\
&= 8\sqrt{\log\left(\frac{2KS}{\delta}\right)} \sum_{s=2}^{S} \sqrt{KM2^s} \sqrt{64|A_{s-1}|} \\
&\leq 8\sqrt{\log\left(\frac{2KS}{\delta}\right)} \sum_{s=2}^{S} K\sqrt{M2^s} \sqrt{64} \\
&\leq 800\sqrt{\log\left(\frac{2KS}{\delta}\right)} K\sqrt{M} 2^{S/2}.
\end{aligned}
$$

Now recall from Algorithm 1 the definition of $S = \log_2\left(\frac{T}{4KM} + 1\right)$. Substituting this into the equation above, we get that

$$
\begin{aligned}
8\sum_{s=2}^{S} |A_s| n_s C_{s-1} &\leq 800\sqrt{\log\left(\frac{2KS}{\delta}\right)} K\sqrt{M} 2^{S/2} \\
&= 800\sqrt{\log\left(\frac{2KS}{\delta}\right)} K\sqrt{M}\sqrt{\frac{T}{4KM} + 1} \\
&\leq 800\sqrt{\log\left(\frac{2KS}{\delta}\right)} K\sqrt{M}\sqrt{\frac{T}{KM}} \\
&= 800\sqrt{\log\left(\frac{2KS}{\delta}\right)} \sqrt{K}\sqrt{T}.
\end{aligned}
\tag{5}
$$

Combining Eq. (4) with Eq. (5) and using the upper bound $S \leq \log(T)$ yields the result. ∎

A.3. Proof of Lemma A.2

To facilitate the proof, we leverage the following critical lemma, which bounds the gap of the value $\mu(x)$ of each action $x \in A_s$ versus $\mu(x^\star)$.

Lemma A.3. *The event*

$$
\cap_{s=2}^{S} \cap_{x \in A_s} \left\{\mu(x) - \mu(x^\star) \leq 4C_{s-1}\right\},
$$

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

The proof of this Lemma A.3 is provided in Appendix A.4. Let us now return to the main proof. For any epoch $s > 1$ and any action $x \in A_s$, let ℓ^s denote the total loss experienced in epoch s of Algorithm 1 while executing the action x for $2n_s$ times. Hence we have $\ell^s = \sum_{x \in A_s} \ell^{sx}$.

Note that within a single epoch $s > 1$, for each $x \in A_s$ we execute x for $2n_s$ times. For the latter $2n_s - m$ times that x is executed, action x has been played m times in the previous m timesteps. Hence, for the latter $2n_s - m$ times that x is executed, the expected loss of playing the action x is $h_x(m) = \mu(x)$. Thus, we have that

$$
\ell^{sx} \leq m + (2n_s - m)\mu(x).
$$

Hence, for each $s > 1$ and $x \in A_s$, we can use Lemma A.3 to upper bound

$$
\begin{aligned}
\ell^{sx} - 2n_s\mu(x^\star) &\leq m + (2n_s - m)\mu(x) - 2n_s\mu(x^\star) + m\mu(x^\star) \\
&\leq m + (2n_s - m)\left(\mu(x) - \mu(x^\star)\right) \\
&\leq m + 4(2n_s - m)C_{s-1}.
\end{aligned}
$$

This bound holds uniformly for each $x \in A_s$. Recalling that $T_s = 2|A_s|n_s$, we hence have that

$$
\begin{aligned}
\ell^s - T_s\mu(x^\star) &= \sum_{x \in A_s} \ell^{sx} - 2|A_s|n_s\mu(x^\star) \\
&= \sum_{x \in A_s} \left(\ell^{sx} - 2n_s\mu(x^\star)\right) \\
&\leq \sum_{x \in A_s} \left(m + 4(2n_s - m)C_{s-1}\right) \\
&= |A_s|\left(m + 4(2n_s - m)C_{s-1}\right).
\end{aligned}
$$

This completes the proof. ∎

A.4. Proof of Lemma A.3

To facilitate the proof, we require the following two critical helper results. The first result bounds the error incurred when estimating $\mu(x)$ via the stochastic realizations $\{\widetilde{h}_x(m)_{s,k}\}$. The second result shows that while running Algorithm 1, which is based on successive elimination of inferior actions over epochs $s \in \{1, 2 \dots S\}$, at any epoch s we never eliminate $x^\star$ from our set A_s of feasible actions.

Lemma A.4. *Fix any $s \in \{1, 2 \dots S\}$, and let B_s denote the event that for all actions $x \in A_s$ we simultaneously have that*

$$|\widehat{\mu}_s(x) - \mu(x)| \leq C_s.$$

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

Lemma A.5. *The event $\cap_{s=1}^S B_s$, where the event B_s is defined in Lemma A.4, implies the event that*

$$x^\star \in \cap_{s=1}^S A_s \text{ and } \cap_{s=1}^S \left\{0 \leq \widehat{\mu}_s(x^\star) - \widehat{\mu}_s(\widehat{x}_s) \leq 2C_s\right\}.$$

The proofs of Lemma A.4 and Lemma A.5 are provided in Appendix A.5 and Appendix A.6 respectively.

Let us now return to the proof. By the result of Lemma A.4 and a union bound, the event $\cap_{s=1}^S B_s$ occurs with probability at least $1 - \delta$. Furthermore, the result of Lemma A.5 shows that the event $\cap_{s=1}^S B_s$ implies the event

$$x^\star \in \cap_{s=1}^S A_s. \tag{6}$$

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

$$
\begin{aligned}
\mu(x) - \mu(x^\star) &\overset{(i)}{\leq} \widehat{\mu}_{s-1}(x) - \mu(x^\star) + C_{s-1} \\
&\overset{(ii)}{\leq} \widehat{\mu}_{s-1}(\widehat{x}_{s-1}) - \mu(x^\star) + 3C_{s-1} \\
&\overset{(iii)}{\leq} \widehat{\mu}_{s-1}(x^\star) - \mu(x^\star) + 3C_{s-1} \\
&\overset{(iv)}{\leq} \mu(x^\star) - \mu(x^\star) + 4C_{s-1} \\
&= 4C_{s-1},
\end{aligned}
$$

where step (i) follows from Lemma A.4, step (ii) follows from the definition of A_s and the fact that $x \in A_s$, step (iii) follows from the definition of $\widehat{x}_{s-1}$ and Eq. (6), and step (iv) follows again from Lemma A.4 and Eq. (6). This completes the proof. ∎

A.5. Proof of Lemma A.4

Fix any $x \in \mathcal{X}$. Recalling the definition of C_s provided in Algorithm 1, Hoeffding's bound (Hoeffding, 1963) ensures that the event

$$|\widehat{\mu}_s(x) - \mu(x)| = \left|\mu(x) - \frac{1}{n_s} \sum_{k=1}^{n_s} \widetilde{h}_x(m)_{s,k}\right| \leq \sqrt{\frac{32}{n_s} \log\left(\frac{2KS}{\delta}\right)} = C_s, \tag{7}$$

occurs with probability at least $1 - \delta/(KS)$. Since $|A_s| \leq K$, a union bound then ensures that the above event occurs simultaneously for all $x \in A_s$ with probability at least $1 - \delta/S$. ∎

A.6. Proof of Lemma A.5

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 $x^\star \in A_1$ and $0 \leq \widehat{\mu}_1(x^\star) - \widehat{\mu}_1(\widehat{x}_1) \leq 2C_1$. Then for the inductive step we show that if the event $x^\star \in A_{s-1}$ and $0 \leq \widehat{\mu}_{s-1}(x^\star) - \widehat{\mu}_{s-1}(\widehat{x}_{s-1}) \leq 2C_{s-1}$ occurs, then we also have that the event

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

is also true.

For the base case, note that by definition we are guaranteed $x^\star \in A_1$. And by the definition of $\widehat{x}_1$, we know that $0 \leq \widehat{\mu}_1(x^\star) - \widehat{\mu}_1(\widehat{x}_1)$. Furthermore, recalling the definition of the event B_1 in Lemma A.4, on the event B_1 we have that

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

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

$$\widehat{\mu}_1(x^\star) - \widehat{\mu}_1(\widehat{x}_1) \leq 2C_1.$$

This verifies the base case.

For the inductive step, assume that $x^\star \in A_{s-1}$ and $0 \leq \widehat{\mu}_{s-1}(x^\star) - \widehat{\mu}_{s-1}(\widehat{x}_{s-1}) \leq 2C_{s-1}$ occurs. Then the definition of A_s and the inductive hypothesis directly imply that $x^\star \in A_s$. Hence, it is true by definition of $\widehat{x}_s$ that $0 \leq \widehat{\mu}_s(x^\star) - \widehat{\mu}_s(\widehat{x}_s)$. Then recalling the definition of the event B_s in Lemma A.4, on the event B_s we have that

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

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

$$\widehat{\mu}_s(x^\star) - \widehat{\mu}_s(\widehat{x}_s) \leq 2C_s.$$

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

B. Proof of Theorem 3.2

Assume for the sake of contradiction that the statement is false. Then there exists some ϵ satisfying the given conditions and some function f, such that $\overline{\mathcal{A}}_{\epsilon,f}$ is not empty. This implies the existence of an algorithm $\mathcal{A}$, such that when it is given as input any positive integers T, K, M with $M \leq T$, the algorithm $\mathcal{A}$ satisfies that

$$\mathbb{E}[\mathcal{R}^{\mathrm{cp}}(\mathcal{A}, \mathsf{tb})] \leq \min\left\{ T/4, f(m_{\mathsf{tb}}, K)\left(T^{1-\epsilon_1} + T^{\epsilon_3}M^{1-\epsilon_2}\right)\right\} \text{ for all } \mathsf{tb} \in \mathsf{UTB}_{T,M,K}. \tag{8}$$

If $\mathcal{A}$ was a randomized algorithm, then this implies the existence of a deterministic algorithm with the same property. So we can assume without loss of generality that $\mathcal{A}$ is deterministic.

Fix some integer $K \geq 2$. Pick some sufficiently large T, M such that the following conditions hold simultaneously

$$M < T/4 \text{ and } f(1, K)\left(T^{1-\epsilon_1} + T^{\epsilon_3}M^{1-\epsilon_2}\right) < M/2. \tag{9}$$

To see these conditions are simultaneously feasible, recall that $\epsilon_1 \in (0, 1)$ and $0 \leq \epsilon_3 < \epsilon_2 < 1$. Let $\gamma = \min\{\epsilon_1, \epsilon_2 - \epsilon_3\} > 0$. So if we choose $M = T^{1-\gamma/2}$, then since this M satisfies $M = T^{1-\gamma/2} < T$, we have that

$$\begin{aligned}
f(1, K)\left(T^{1-\epsilon_1} + T^{\epsilon_3}M^{1-\epsilon_2}\right) &< f(1, K)\left(T^{1-\epsilon_1} + T^{\epsilon_3}T^{1-\epsilon_2}\right) \\
&= f(1, K)\left(T^{1-\epsilon_1} + T^{1-(\epsilon_2-\epsilon_3)}\right) \\
&\leq 2f(1, K)T^{1-\gamma}.
\end{aligned}$$

So for sufficiently large T, we have for this choice of $M = T^{1-\gamma/2}$ that $M < T/4$ and also that $f(1, K)\left(T^{1-\epsilon_1} + T^{\epsilon_3}M^{1-\epsilon_2}\right) < M/2$. This shows that Eq. (9) is feasible.

We will now define two unweighted tallying bandit problems, each of which have K actions. Recall that in an unweighted tallying bandit problem with memory capacity m, the loss associated with playing an action at a given timestep is fully defined by the number of times that action was played in the last m timesteps. Concretely, assume that in some unweighted tallying bandit problem tb, we play action x on the current timestep, and the total number of times it has been played in the

last m timesteps (including the current timestep) is $1 \leq y \leq m$. Then there exists a function $h_{\mathsf{tb},x} : \{1, 2 \ldots m\} \to [0, 1]$, such that denote the loss associated with playing this action is given by $h_{\mathsf{tb},x}(y)$. We will use this notation to instantiate the forthcoming unweighted tallying bandit problems.

With this notation in hand, let us instantiate the unweighted tallying bandit problem tb_A with memory length $m_{\mathsf{tb}_A} = 1$ as follows. For action x_1, we have that $h_{\mathsf{tb}_A,x_1} = 1/2$. For action x_2, we have that $h_{\mathsf{tb}_A,x_2} = 1$. And for every other action x, let $h_{\mathsf{tb}_A,x} = 1$. We say that whenever the player plays action x, the player almost surely observes $h_{\mathsf{tb}_A}(1)$. Notice that since $m_{\mathsf{tb}_A} = 1$, and there is no stochasticity in the observation of losses when we play any action, tb_A is indeed a deterministic multi-armed bandit problem.

Now, we instantiate the tallying bandit problem tb_B with memory length $m_{\mathsf{tb}_B} = M$ as follows. For action x_1 we define $h_{\mathsf{tb}_B,x_1} = 1/2$. For action x_2 we define

$$
h_{\mathsf{tb}_B,x_2}(y) = \begin{cases} 1 \text{ if } 1 \leq y < M \\ 0 \text{ if } y = M \end{cases} .
$$

And for every other action x, we define $h_{\mathsf{tb}_B,x} = 1$. Once again, we enforce that there is no stochasticity in the player's observation of losses when the player plays an action. So the feedback model in tb_B is deterministic.

Let the horizon length for problems tb_A and tb_B be the T chosen as per Eq. (9). Note also that both problem instances satisfy REO with parameter $\alpha = 0$, and so $\mathsf{tb}_A, \mathsf{tb}_B \in \mathsf{UTB}_{T,M,K}$. So via our assumption of the determinism of $\mathcal{A}$, via Eq. (8) and via the fact that $m_{\mathsf{tb}_A} = 1$, we have that

$$
\mathcal{R}^{\mathrm{cp}}(\mathcal{A}, \mathsf{tb}_A) \leq f(m_{\mathsf{tb}_A}, K) \left(T^{1-\epsilon_1} + T^{\epsilon_3} M^{1-\epsilon_2}\right) = f(1, K) \left(T^{1-\epsilon_1} + T^{\epsilon_3} M^{1-\epsilon_2}\right) < M/2, \tag{10}
$$

where the final inequality follows due to Eq. (9). And again by our assumption of the determinism of $\mathcal{A}$ and via Eq. (8), we have that

$$
\mathcal{R}^{\mathrm{cp}}(\mathcal{A}, \mathsf{tb}_B) \leq T/4. \tag{11}
$$

When $\mathcal{A}$ is run on problem tb_A, there are 2 cases. Either $\mathcal{A}$ plays x_2 for M times in a row (at some point in its execution for T timesteps while solving tb_A) or it does not.

Consider the first case, where $\mathcal{A}$ plays x_2 for M times in a row on problem tb_A. Then we have that $\mathcal{R}^{\mathrm{cp}}(\mathcal{A}, \mathsf{tb}_A) \geq M/2$. This is because the optimal strategy for tb_A always plays x_1 on each timestep, and so any timestep where $\mathcal{A}$ plays action $x \neq x_1$ will add $h_{\mathsf{tb}_A,x} - h_{\mathsf{tb}_A,x_1} = 1 - 1/2 = 1/2$ to the CPR of $\mathcal{A}$. This is a contradiction to Eq. (10).

Consider the second case, where $\mathcal{A}$ never plays x_2 for M times in a row on problem tb_A. Note that the deterministic algorithm $\mathcal{A}$ can be viewed as a length T sequence of functions, where the tth function maps the past $t-1$ action choices and loss observations to the action played at timestep t. Also note that the observed loss of a playing an action in tb_B is different from playing the same action in tb_A, if and only if that action was x_2 and it was played M times in the prior M timesteps (including the current timestep).

Thus, since $\mathcal{A}$ never plays x_2 for M times in a row on problem tb_A, it sees the *identical* sequence of loss outputs when it is deployed on tb_B, and hence makes the identical sequence of actions as it would have if deployed in tb_A, which in turn implies that it never plays x_2 for M times in a row on problem tb_B. Since $M < T/4$ via Eq. (9), and playing any action $x \neq x_2$ will always yield loss at least $1/2$, the strategy that always plays x_2 is optimal and has cumulative loss of $M - 1$. So, since Eq. (9) implies that

$$
T/2 - (M-1) > T/2 - M > T/2 - T/4 = T/4,
$$

we have that

$$
\mathcal{R}^{\mathrm{cp}}(\mathcal{A}, \mathsf{tb}_B) \geq T/2 - (M-1) > T/4.
$$

This is a contradiction to Eq. (11).

In either case, we have arrived at a contradiction. Hence, we have shown that for each ϵ satisfying the given conditions and each function f, the corresponding set $\overline{\mathcal{A}}_{\epsilon,f}$ is the empty set. This completes the proof. ∎

C. Proof of Proposition 2.2

In this section, we provide a formal proof of Proposition 2.2. Let $\mathcal{X} = \{x_1, x_2\}$. Let w be defined componentwise as $w_i = \frac{1}{2^i}$ for each $1 \leq i \leq m$. Set $w_x = w$ for each $x \in \mathcal{X}$. Now sample a bit string $y^\star$ uniformly at random from $\{0,1\}^{m-1}$, whose identity is kept hidden from the player. Define $h_{x_1} = 1$ and define

$$h_{x_2}(w_{x_2}^\top y^{t,x_2,m}) = h_{x_2}(w^\top y^{t,x_2,m}) = \begin{cases} 1 \text{ if } w^\top y^{t,x_2,m} \neq w^\top(1, y^\star) \\ 0 \text{ if } w^\top y^{t,x_2,m} = w^\top(1, y^\star) \end{cases}.$$

We assume that there is no stochasticity in the loss feedback experienced by the player. This defines an (m, w, h)-weighted tallying bandit game. For ease in notation in the sequel, we also define $v \in \mathcal{X}^{m-1}$ as

$$v_i = \begin{cases} x_2 \text{ if } y_i^\star = 1 \\ x_1 \text{ if } y_i^\star = 0 \end{cases}.$$

We claim that with our choice of w, if $y \neq y' \in \{1\} \times \{0,1\}^{m-1}$ then $w^\top y \neq w^\top y'$. We defer the formal proof of this claim for now, and use this claim to complete the proof of Proposition 2.2. Critically, the claim implies that we incur non-unit loss at timestep t if and only if we play action x_2 at timestep t and have $y^{t,x_2,m} = (1, y^\star)$. Equivalently, we incur non-unit loss at timestep t if and only if our action sequence for the timesteps $t, t-1 \ldots t-m$ is (x_2, v). Thus, the policy that cyclically plays $v_{m-1}, v_{m-2} \ldots v_1, x_2$ incurs a loss of zero at least once every m timesteps. Meanwhile, identifying the optimal policy is at least as hard as playing the action sequence (x_2, v), which in turn is at least as hard as identifying $y^\star$.

A standard "needle in the haystack" argument (Du et al., 2020) then shows that identifying $y^\star$ requires $\widetilde{\Omega}(2^m)$ timesteps. In turn, since the cyclic policy $v_{m-1}, v_{m-2} \ldots v_1, x_2$ incurs a loss of zero at least once every m timesteps, this implies that the expected CPR $\mathbb{E}[\mathcal{R}^{\mathrm{cp}}]$ of any (possibly randomized) algorithm is lower bounded by $\widetilde{\Omega}(\min\{2^m, T\}/m)$, where the expectation is over the (possible) randomization of the algorithm as well as the sampling of $y^\star$.

Let us now return to prove our claim that if $y \neq y' \in \{1\} \times \{0,1\}^{m-1}$ then $w^\top y \neq w^\top y'$. Assume for the sake of contradiction that $y \neq y'$ but $w^\top y = w^\top y'$. Let $J \subseteq \{2, 3 \ldots m\}$ be the set of coordinates that y, y' differ, and let $j^\star = \min J$. Note that J is non-empty by assumption, and so $j^\star$ is well defined. Assume without loss of generality that $y_{j^\star} - y'_{j^\star} = 1$ (the case when $y_{j^\star} - y'_{j^\star} = -1$ is completely symmetric). Then observe that

$$
\begin{aligned}
0 &= w^\top(y - y') \\
&= \sum_{j=1}^m w_j(y_j - y'_j) \\
&= \sum_{j \in J} w_j(y_j - y'_j) \\
&= w_{j^\star} + \sum_{j \neq j^\star \in J} w_j(y_j - y'_j).
\end{aligned}
\tag{12}
$$

We can now lower bound Eq. (12) as

$$
\begin{aligned}
0 &= w_{j^\star} + \sum_{j \neq j^\star \in J} w_j(y_j - y'_j) \\
&\geq w_{j^\star} - \sum_{j \neq j^\star \in J} w_j|y_j - y'_j| \\
&= w_{j^\star} - \sum_{j \neq j^\star \in J} w_j \\
&\geq w_{j^\star} - \sum_{j = j^\star + 1}^m w_j.
\end{aligned}
\tag{13}
$$

Now substituting in our choice of w into Eq. (13), we find that

$$0 \geq w_{j^\star} - \sum_{j=j^\star+1}^{m} w_j = \frac{1}{2^{j^\star}} - \sum_{j=j^\star+1}^{m} \frac{1}{2^j} = \frac{1}{2^{j^\star}} \left(1 - \sum_{j=1}^{m-j^\star} \frac{1}{2^j} \right) > 0,$$

which of course is a contradiction. This proves our claim that if $y \neq y' \in \{1\} \times \{0,1\}^{m-1}$ then $w^\top y \neq w^\top y'$. ∎

D. Extended Numerical Results & Details

D.1. Unweighted Tallying Bandit

In this section, we first present additional experimental results for the unweighted tallying bandit problem, where the loss functions are identical to those described in Section 4.1. Hence, this problem satisfies 0-REO. Here, we vary the values of m, K, M, and plot the CPR of each method over time. The results are shown in Figure 3. In each case, we observe that SE outperforms the baselines. These results also show that the performance of Algorithm 1 is robust to using a conservative upper bound M on m.

We now present results for a different unweighted tallying bandit problem, where α-REO is satisfied with $\alpha > 0$. For each action $x \in \mathcal{X}$, we define $w_x = \vec{1}/(4m)$. We fix an action $x^\star \in \mathcal{X}$ and a different action $x^{\star\star} \in \mathcal{X}$, and then define the loss functions $\{h_x\}_{x \in \mathcal{X}}$ as

$$h_x(y^{t,x,m}) = \begin{cases} 1 - w_x^\top y^{t,x,m} - 0.15 & \text{if } x = x^\star, y^{t,x,m} = \vec{1} \\ 1 - w_x^\top y^{t,x,m} - \frac{m-1}{2m} - 0.2 & \text{if } x = x^{\star\star}, y^{t,x,m} = (1,0,0\ldots0) \\ 1 - w_x^\top y^{t,x,m} & \text{otherwise} \end{cases}.$$

A numerical computation reveals that this unweighted tallying bandit problem instance satisfies α-REO with $\alpha = \max\{0, -0.2 + \frac{2m-3}{4m}\}$. We empirically study the performance of Algorithm 1 relative to the baselines on this problem instance, with a choice of $m = 4, K = 5, M = 4$. Since the optimal policy in this problem is not obvious, the CPR is difficult to compute. So in lieu of the CPR, we plot the expected cumulative loss of each algorithm in excess of SE's loss (hence the CPR at any time is obtained by applying a constant shift to each algorithm's excess loss). The results are shown in Figure 4, where we observe that our method outperforms all others.

D.2. Weighted Tallying Bandit

Here, we describe the loss functions that were used to define the WTB problem instance described in Section 4.2. For each action $x \in \mathcal{X}$, we define w_x in the following fashion. First we define the vector $v \in \mathbb{R}^m$ coordinate wise by setting its ith coordinate as $v_i = 1/2^i$. Then we set $w_x = v/(2\|v\|_1)$ for each $x \in \mathcal{X}$. We fix an action $x^\star \in \mathcal{X}$, and then define the loss functions $\{h_x\}_{x \in \mathcal{X}}$ as

$$h_x(y^{t,x,m}) = \begin{cases} 1 - w_x^\top y^{t,x,m} - 0.15 & \text{if } x = x^\star, y^{t,x,m} = \vec{1} \\ 1 - w_x^\top y^{t,x,m} & \text{otherwise} \end{cases}.$$

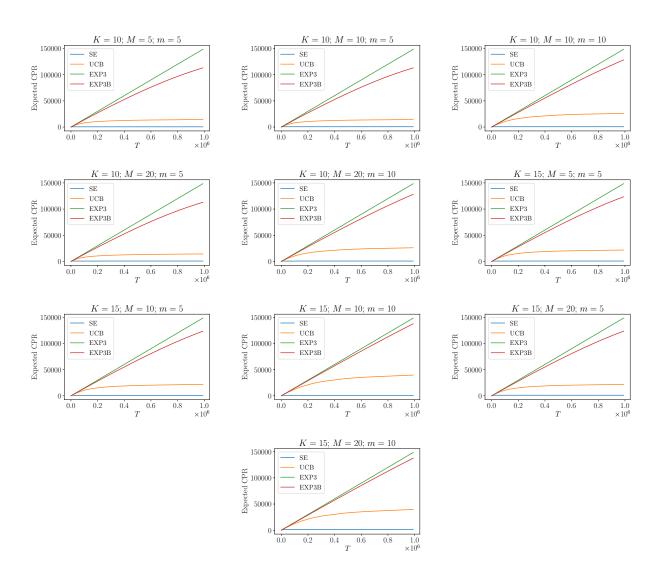Hence, this weighted tallying bandit problem satisfies 0-REO.

Figure 3: We plot the expected CPR of each algorithm, when deployed on the unweighted tallying bandit problem described in Section 4.1, with varying values of m, K, M. In all plots, each datapoint is obtained by averaging over 20 problem instances, and the shaded region depicts ± 1 standard error around the mean.
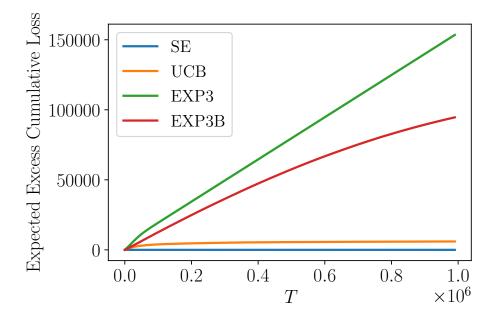
Figure 4: We plot as a function of time the expected cumulative loss of each algorithm in excess of that of SE, on the unweighted tallying bandit instance described in Appendix D.1, with $m = 4, K = 5, M = 4$. Note this instance satisfies α-REO with $\alpha = 0.1125$. Each datapoint is obtained by averaging over 20 problem instances, and the shaded region depicts ± 1 standard error around the mean.