# Corralling Stochastic Bandit Algorithms

**Raman Arora**
Johns Hopkins University
arora@cs.jhu.edu

**Teodor V. Marinov**
Johns Hopkins University
tmarino2@jhu.edu

**Mehryar Mohri**
Courant Institute and Google Research
mohri@google.com

## Abstract

We study the problem of corralling stochastic bandit algorithms, that is combining multiple bandit algorithms designed for a stochastic environment, with the goal of devising a corralling algorithm that performs almost as well as the best base algorithm. We give two general algorithms for this setting, which we show benefit from favorable regret guarantees. We show that the regret of the corralling algorithms is no worse than that of the best algorithm containing the arm with the highest reward, and depends on the gap between the highest reward and other rewards.

## 1 Introduction

We study the problem of *corralling* multi-armed bandit algorithms in a stochastic environment. This consists of selecting, at each round, one out of a fixed collection of bandit algorithms and playing the action returned by that algorithm. Note that the corralling algorithm does not directly select an arm, but only a base algorithm. It never requires knowledge of the action set of each base algorithm. The objective of the corralling algorithm is to achieve a large cumulative reward or a small pseudo-regret, over the course of its interactions with the environment. This problem was first introduced and studied by Agarwal et al. (2016). Here, we are guided by the same motivation but consider the stochastic setting and seek more favorable guarantees. Thus, we assume that the reward, for each arm, is drawn from an unknown distribution.

In the simplest setting of our study, we assume that each base bandit algorithm has access to a distinct set of arms. This scenario appears in several applications. As an example, consider the online contractual display ads allocation problem (BasuMallick, 2020): when users visit a website, say some page of the online site of a national newspaper, an ads allocation algorithm chooses an ad to display at each specific slot with the goal of achieving the largest value. This could be an ad for a clothing item, which could be meant for the banner of the online front page of that newspaper. To do so, the ads allocation algorithm chooses one out of a large set of advertisers, each a clothing brand or company in this case, which have signed a contract with the ads allocation company. Each clothing company has its own marketing strategy and thus its own bandit algorithm with its own separate set of clothing items or arms. There is no sharing of information between these companies which are typically competitors. Furthermore, the ads allocation algorithm is not provided with any detailed information about the base bandits algorithms of these companies, since that is proprietary information private to each company. The allocation algorithm cannot choose a specific arm or clothing item, it can only choose a base advertiser. The number of ads or arms can be very large. The number of advertisers can also be relatively large in practice, depending on the domain. The number of times the ads allocation is run is in the order of millions or even billions per day, depending on the category of items.

A similar problem arises with online mortgage broker companies offering loans to new applicants. The mortgage broker algorithm must choose a bank, each with different mortgage products. The broker brings a new application exclusively to one of the banks, as part of the contract, which also entitles them to incentives. The bank's online algorithm can be a bandit algorithm proposing a product, and the details of the algorithm are not accessible to the broker; for instance, the bank's credit rate and incentives may depend on the financial and credit history of the applicant. The number of mortgage products is typically fairly large, and the number of online loan requests per day is in the order of several thousands. Other instances of this problem appear when an algorithm can only select one of multiple bandit algorithms and, for privacy or reg-

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

ulatory reasons, it cannot directly select an arm or receive detailed information about the base algorithms.

In the most general setting we study, there may be an arbitrary sharing of arms between the bandit algorithms. We will only assume that only one algorithm has access to the arm with maximal expected reward, which implies a positive gap between the expected reward of the best arm of any algorithm and that of the best algorithm. This is because we seek to devise a corralling algorithm with favorable gap-dependent pseudo-regret guarantees.

**Related work.** The previous work most closely related to this study is the seminal contribution by Agarwal et al. (2016) who initiated the general problem of corralling bandit algorithms. The authors gave a general algorithm for this problem, which is an instance of the generic Mirror Descent algorithm with an appropriate mirror map (LOG-BARRIER-OMD), (Foster et al., 2016; Wei and Luo, 2018), and which includes a carefully constructed non-decreasing step-size schedule, also used by Bubeck et al. (2017). The algorithm of Agarwal et al. (2016), however, cannot in general achieve regret bounds better than $\tilde{O}(\sqrt{T})$ in the time horizon, unless optimistic instance-dependent regret bounds are known for the corralled algorithms. Prior to their work, Arora et al. (2012) presented an algorithm for learning deterministic Markov decision processes (MDPs) with adversarial rewards, using an algorithm for corralling bandit linear optimization algorithms. In an even earlier work, Maillard and Munos (2011) attempted to corral EXP3 algorithms (Auer et al., 2002b) with a top algorithm that is a slightly modified version of EXP4. The resulting regret bounds are in $\tilde{O}(T^{2/3})$.

Our work can also be viewed as selecting the best algorithm for a given unknown environment and, in this way, is similar in spirit to the literature solving the *best of both worlds* problem (Audibert and Bubeck, 2009; Bubeck and Slivkins, 2012; Seldin and Slivkins, 2014; Auer and Chiang, 2016; Seldin and Lugosi, 2017; Wei and Luo, 2018; Zimmert and Seldin, 2018; Zimmert et al., 2019) and the model selection problem for linear bandit (Foster et al., 2019; Chatterji et al., 2019).

Very recently, Pacchiano et al. (2020) also considered the problem of corralling stochastic bandit algorithms. The authors seek to treat the problem of model selection, where multiple algorithms might share the best arm. More precisely, the authors consider a setting in which there are $K$ stochastic contextual bandit algorithms and try to minimize the regret with respect to the best overall policy belonging to any of the bandit algorithms. They propose two corralling algorithms, one based on the work of (Agarwal et al., 2016) and one based on EXP3.P (Auer et al., 2002b). The main novelty in their work is a smoothing technique for each of the base algorithms, which avoids having to restart the base algorithms throughout the $T$ rounds, as was proposed in (Agarwal et al., 2016). The proposed regret bounds are of the order $\tilde{\Theta}(\sqrt{T})$. We expect that the smoothing technique is also applicable to one of the corralling algorithms we propose. Since Pacchiano et al. (2020) allow for algorithms with shared best arms, their main results do not discuss the optimistic setting in which there is a gap between the reward of the optimal policy and all other competing policies, and do not achieve the *optimistic guarantees* we provide. Further, they show a min-max lower bound which states that even if one of the base algorithms is optimistic and contains the best arm, there is still no hope to achieve regret better than $\tilde{\Omega}(\sqrt{T})$ if the best arm is shared by an algorithm with regret $\tilde{\Omega}(\sqrt{T})$. We view their contributions as complementary to ours.

In general, some caution is needed when designing a corralling algorithm, since aggressive strategies may discard or disregard a base learner that admits an arm with the best mean reward if it performs poorly in the initial rounds. Furthermore, as noted by Agarwal et al. (2016), additional assumptions are required on each of the base learners if one hopes to achieve non-trivial corralling guarantees.

**Contributions.** We first motivate our key assumption that all of the corralled algorithms must have favorable regret guarantees during all rounds. To do so, in Section 3, we show that if one does not assume anytime regret guarantees, then even when corralling simple stochastic bandit algorithms, each with $o(\sqrt{T})$ regret, any corralling strategy will have to incur $\Omega(\sqrt{T})$ regret. Therefore, for the rest of the paper we assume that each base learner admits anytime guarantees. In Section 4 and Section 5, we present two general corralling algorithms whose pseudo-regret guarantees admit a dependency on the gaps between base learners, that is their best arms, and only poly-logarithmic dependence on time horizon. These bounds are syntactically similar to the instance-dependent guarantees for the stochastic multi-armed bandit problem (Auer et al., 2002a). Thus, our corralling algorithm performs almost as well as the best base learner, if it were to be used on its own, modulo gap-dependent terms and logarithmic factors. The algorithm in Section 4 uses the standard UCB ideas combined with a boosting technique, which runs multiple copies of the same base learner. In Section 4.1, we show that simply using UCB-style corralling without boosting can incur linear regret. If, additionally, we assume that each of the base learners satisfy the stability condition adopted in (Agarwal et al., 2016), then, in Section 5 we show that it suffices to run a single copy of each base learner by using a corralling approach based on OMD. We show that UCB-I (Auer et al., 2002a) can be made to satisfy the stability condition, as long as the confidence bound is rescaled and changed by an additive factor. In Section 6, to further examine the properties of our algorithms, we report the results of experiments with our algorithms for synthetic datasets. Finally, while our main motivation is not model selection, in Section 7, we briefly discuss some related matters and show that our algorithms can help recover several known results in that area.

## 2 Preliminaries

We consider the problem of corralling $K$ stochastic multi-armed bandit algorithms $\mathcal{A}_1, \ldots, \mathcal{A}_K$, which we often refer to as *base algorithms* (base learners). At each round $t$, a *corralling algorithm* selects a base algorithm $\mathcal{A}_{i_t}$, which plays action $a_{i_t,j_t}$. The corralling algorithm is not informed of the identity of this action but it does observe its reward $r_t(a_{i_t,j_t})$. The top algorithm then updates its decision rule and provides feedback to each of the base learners $\mathcal{A}_i$. We note that the feedback may be just the empty set, in which case the base learners do not update their state. We will also assume access to the parameters controlling the behavior of each $\mathcal{A}_i$ such as the step size for mirror descent-type algorithms, or the confidence bounds for UCB-type algorithms. Our goal is to minimize the cumulative pseudo-regret of the corralling algorithm as defined in Equation 1:[1]

$$\mathbb{E}[R(T)] = T\mu_{1,1} - \mathbb{E}\left[\sum_{t=1}^{T} r_t(a_{i_t,j_t})\right], \qquad (1)$$

where $\mu_{1,1}$ is the mean reward of the best arm.

**Notation.** We denote by $\mathrm{e}_i$ the $i$th standard basis vector, by $\mathbf{0}_K \in \mathbb{R}^K$ the vector of all 0s, and by $\mathbf{1}_K \in \mathbb{R}^K$ the vector of all 1s. For two vectors $x, y \in \mathbb{R}^K$, $x \odot y$ denotes their Hadamard product. We also denote the line segment between $x$ and $y$ as $[x, y]$. $w_{t,i}$ denotes the $i$-th entry of a vector $w_t \in \mathbb{R}^K$. $\Delta^{K-1}$ denotes the probability simplex in $\mathbb{R}^K$, $D_\Psi(x, y)$ the Bregman divergence induced by the potential $\Psi$, whose conjugate function we denote by $\Psi^*$. We use $\mathrm{I}_C$ to denote the indicator function of a set $C$. For any $k \in \mathbb{N}$, we use the shorthand $[k] := \{1, 2, \ldots, k\}$.

For the base algorithms $\mathcal{A}_1, \ldots, \mathcal{A}_K$, let $T_i(t)$ be the number of times algorithm $\mathcal{A}_i$ has been played until time $t$. Let $T_{i,j}(t)$ be the number of times action $j$ has been proposed by algorithm $\mathcal{A}_i$ until time $t$. Let $[k_i]$ denote the set of arms or action set of algorithm $\mathcal{A}_i$. We denote the reward of arm $j$ in the action set of algorithm $i$ at time $t$ as $r_t(a_{i,j})$ and denote its mean reward by $\mu_{i,j}$. We also use $a_{i,j_t}$ to denote the arm proposed by algorithm $\mathcal{A}_i$ during time $t$. Further, the algorithm played at time $t$ is denoted as $i_t$, its action played at time $t$ is $a_{i_t,j_t}$ and the reward for that action is $r_t(a_{i_t,j_t})$ with mean $\mu_{i_t,j_t}$. Let $i^*$ denote the index of the base algorithm that contains the arm with the highest mean reward. Without loss of generality, we will assume that $i^* = 1$. Similarly, we assume that $a_{i,1}$ is the arm with highest reward in algorithm $\mathcal{A}_i$. We assume that the best arm of the best algorithm has a gap to the best arm of every other algorithm. We denote the gap between the best arm of $\mathcal{A}_1$ and the best arm of $\mathcal{A}_i$ as $\Delta_i$: $\Delta_i = \mu_{i^*,1} - \mu_{i,1} > 0$ for $i \neq i^*$. Further, we denote the intra-algorithm gaps by $\Delta_{i,j} = \mu_{i,1} - \mu_{i,j}$. We denote by $\bar{R}_i(t)$ an upper bound on the regret of algorithm $\mathcal{A}_i$ at time $t$ and by $R_i(t)$ the actual regret of $\mathcal{A}_i$, so that $\mathbb{E}[R_i(t)]$ is the expected regret of algorithm $\mathcal{A}_i$ at time $t$. The asymptotic notations $\tilde{\Omega}$ and $\tilde{O}$ are equal to $\Omega$ and $O$ up to poly-logarithmic factors.

[1]For conciseness, from now on, we will simply write *regret* instead of *pseudo-regret*.

## 3 Lower bounds without anytime regret guarantees

We begin by showing a simple and yet instructive lower bound that helps guide our intuition regarding the information needed from the base algorithms $\{\mathcal{A}_i\}_{i=1}^K$ in the design of a corralling algorithm. Our lower bound is based on corralling base algorithms that only admit a fixed-time horizon regret bound and do not enjoy anytime regret guarantees. We further assume that the corralling strategy cannot simulate anytime regret guarantees on the base algorithms, say by using the so-called doubling trick. This result suggests that the base algorithms must admit a strong regret guarantee during every round of the game.

The key idea behind our construction is the following. Suppose one of the corralled algorithms, $\mathcal{A}_i$, incurs a linear regret over the first $R_i(T)$ rounds. In that case, the corralling algorithm is unable to distinguish between $\mathcal{A}_i$ and an another algorithm that mimics the linear regret behavior of $\mathcal{A}_i$ throughout all $T$ rounds, unless the corralling algorithm plays $\mathcal{A}_i$ at least $R_i(T)$ times. The successive elimination algorithm (Even-Dar et al., 2002) benefits from gap-dependent bounds and can have the behavior just described for a base algorithm. Thus, our lower bound is presented for successive elimination base algorithms, all with regret $O(T^{1/4})$. It shows that, with constant probability, no corralling strategy can achieve a more favorable regret than $\tilde{\Omega}(\sqrt{T})$ in that case.

**Theorem 3.1.** *Let the corralled algorithms be instances of successive elimination defined by a parameter $\alpha$. With probability $1/4$ over the random sampling of $\alpha$, any corralling strategy will incur regret at least $\tilde{\Omega}(\sqrt{T})$, while the gap, $\Delta$, between the best and second best reward is such that $\Delta > \omega(T^{-1/4})$ and all algorithms have a regret bound of $\tilde{O}(1/\Delta)$.*

This theorem shows that, even when corralling natural algorithms that benefit from asymptotically better regret bounds, corralling can incur $\tilde{\Omega}(\sqrt{T})$ regret. It can be further proven (Theorem B.3, Appendix B) that, even if the worst case upper bounds on the regret of the base algorithms were known, achieving an optimistic regret guarantee for corralling would not be possible, unless some additional assumptions were made.

## 4 UCB-style corralling algorithm

The negative result of Section 3 hinge on the fact that the base algorithms do not admit anytime regret guarantees. Therefore, we assume, for the rest of the paper, that the base algorithms, $\{\mathcal{A}_i\}$, satisfy the following:

$$\mathbb{E}\left[t\mu_{i,1} - \sum_{s=1}^{t} r_s(a_{i_s,j_s})\right] \leq \bar{R}_i(t), \tag{2}$$

for any time $t \in [T]$. For UCB-type algorithms, such bounds can be derived from the fact that the expected number of pulls, $T_{i,j}(t)$, of a suboptimal arm $j$, is bounded as $\mathbb{E}[T_{i,j}(t)] \leq c\frac{\log(t)}{(\Delta_{i,j})^2}$, for some time and gap-independent constant $c$ (e.g., Bubeck (2010)), and take the following form, $\bar{R}_i(t) \leq c'\sqrt{k_i t \log(t)}$, for some constant $c'$.

Suppose that the bound in Equation 2 holds with probability $1 - \delta_t$. Note that such bounds are available for some UCB-type algorithms (Audibert et al., 2009). We can then adopt the optimism in the face of uncertainty principle for each $\mu_{i,1}$ by overestimating it with $\frac{1}{t}\sum_{s=1}^{t} r_s(a_{i,j_s}) + \frac{1}{t}\bar{R}_i(t)$. As long as this occurs with high enough probability, we can construct an upper confidence bound for $\mu_{i,1}$ and use it in a UCB-type algorithm. Unfortunately, the upper confidence bounds required for UCB-type algorithms to work need to hold with high enough probability, which is not readily available from Equation 2 or from probabilistic bounds on the pseudo-regret of anytime stochastic bandit algorithms. In fact, as discussed in Section 4.1, we expect it to be impossible to corral any-time stochastic MAB algorithms with a standard UCB-type strategy. However, a simple boosting technique, in which we run $2\log(1/\delta)$ copies of each algorithm $\mathcal{A}_i$, gives the following high probability version of the bound in Equation 2.

**Algorithm 1** UCB-C

```
Input: Stochastic bandit algorithms $\mathcal{A}_1, \ldots, \mathcal{A}_K$
Output: Sequence of algorithms $(i_t)_{t=1}^{T}$.
 1: $t = 1$
 2: for $i = 1, \ldots, K$
 3:    $\mathbb{A}_i = \emptyset$ % contains all copies of $\mathcal{A}_i$
 4:    for $s = 1, \ldots, \lceil 2\log(T) \rceil$
 5:       Initialize $\mathcal{A}_i(s)$ as a copy of $\mathcal{A}_i$, $\widehat{\mu}_i(s) = 0$
 6:       Append $(\mathcal{A}_i(s), \widehat{\mu}_i(s))$ to $\mathbb{A}_i$
 7: for $i = 1, \ldots, K$
 8:    Foreach $(\mathcal{A}_i(s), \widehat{\mu}_i(s)) \in \mathbb{A}_i$, play $\mathcal{A}_i(s)$, update
       empirical mean $\widehat{\mu}_i(s)$, $t = t + 2\log(T)$
 9:    $\widehat{\mu}_{med_i} = \texttt{Median}(\{\widehat{\mu}_i(s)\}_{s=1}^{\lceil 2\log(T) \rceil})$
10: while $t \leq T$
11:    $b_\ell(t) = \frac{\sqrt{2\bar{R}_{med_\ell}(T_{med_\ell}(t))} + \sqrt{2T_{med_\ell}(t)\log(t)}}{T_{med_\ell}(t)}, \forall \ell \in [K]$
12:    $i = \text{argmax}_{\ell \in [K]} \{\widehat{\mu}_{med_\ell} + b_\ell(t)\}$
13:    Foreach $(\mathcal{A}_i(s), \widehat{\mu}_i(s)) \in \mathbb{A}_i$, play $\mathcal{A}_i(s)$, update em-
       pirical mean $\widehat{\mu}_i(s)$, $t = t + 2\log(T)$
14:    $\widehat{\mu}_{med_i} = \texttt{Median}(\{\widehat{\mu}_i(s)\}_{s=1}^{\lceil 2\log(T) \rceil})$
```

**Lemma 4.1.** *Suppose we run $2\log(1/\delta)$ copies of algorithm $\mathcal{A}_i$ which satisfies Equation 2. If $\mathcal{A}_{med_i}$ is the algorithm with median cumulative reward at time $t$, then $\mathbb{P}[t\mu_{i,1} - \sum_{s=1}^{t} r_s(a_{med_i,j_s}) \geq 2\bar{R}_i(t)] \leq \delta$.*

We consider the following variant of the standard UCB algorithm for corralling. We initialize $2\log(T)$ copies of each base algorithm $\mathcal{A}_i$. Each $\mathcal{A}_i$ is associated with the median empirical average reward of its copies. At each round, the corralling algorithm picks the $\mathcal{A}_i$ with the highest sum of median empirical average reward and an upper confidence bound based on Lemma 4.1. The pseudocode is given in Algorithm 1. The algorithm admits the following regret guarantees.

**Theorem 4.2.** *Suppose that algorithms $\mathcal{A}_1, \ldots, \mathcal{A}_K$ satisfy the following regret bound $\mathbb{E}[R_i(t)] \leq \sqrt{\alpha k_i t \log(t)}$, respectively for $i \in [K]$. Algorithm 1 selects a sequence of algorithms $i_1, \ldots, i_T$ which take actions $a_{i_1,j_1}, \ldots, a_{i_T,j_T}$, respectively, such that*

$$\mathbb{E}[R(T)] \leq O\left(\sum_{i \neq i^*} \frac{k_i \log(T)^2}{\Delta_i} + \log(T)\,\mathbb{E}[R_{i^*}(T)]\right),$$

$$\mathbb{E}[R(T)] \leq O\left(\log(T)\sqrt{KT\log(T)\max_{i \in [K]}(k_i)}\right).$$

We note that both the optimistic and the worst case regret bounds above involve an additional factor that depends on the number of arms, $k_i$, of the base algorithm $\mathcal{A}_i$. This dependence reflects the complexity of the decision space of algorithm $\mathcal{A}_i$. We conjecture that a complexity-free bound is not possible, in general. To see this, consider a setting where each

$\mathcal{A}_i$, for $i \neq i^*$, only plays arms with equal means $\mu_i = \mu_{1,1} - \Delta_i$. Standard stochastic bandit regret lower bounds, e.g. (Garivier et al., 2018b), state that any strategy on the combined set of arms of all algorithms will incur regret at least $\Omega(\sum_{i \neq i^*} k_i \log(T)/\Delta_i)$. The $\log(T)$ factor in front of the regret of the best algorithm comes from the fact that we are running $\Omega(\log(T))$ copies of it.

## 4.1 Discussion regarding tightness of bounds

A natural question is if it is possible to achieve bounds that do not have a $\log(T)^2$ scaling. After all, for the simpler stochastic MAB problem, regret upper bounds only scale as $O(\log(T))$ in terms of the time horizon. As already mentioned, the extra logarithmic factor comes from the boosting technique, or, more precisely, the need for exponentially fast concentration of the true regret to its expected value, when using a UCB-type corralling strategy. We now show that, in the absence of such strong concentration guarantees, if only a single copy of each of the base algorithms in Algorithm 1 is run, then linear regret is unavoidable.

**Theorem 4.3.** *There exist instances $\mathcal{A}_1$ and $\mathcal{A}_2$ of UCB-I and a reward distribution, such that, if Algorithm 1 runs a single copy of $\mathcal{A}_1$ and $\mathcal{A}_2$, then $\mathbb{E}[R(T)] \geq \tilde{\Omega}(\Delta_2 T)$.*

*Further, for any algorithm $\mathcal{A}_1$ such that $\mathbb{P}\left[R_1(t) \geq \frac{1}{2}\Delta_{1,2}\tau\right] \geq \frac{1}{\tau^c}$, there exists a reward distribution such that if Algorithm 1 runs a single copy of $\mathcal{A}_1$ and $\mathcal{A}_2$, then $\mathbb{E}[R(T)] \geq \tilde{\Omega}((\Delta_{1,2})^c \Delta_2 T)$.*

The proof of the above theorem and further discussion can be found in Appendix C.1. The requirement that the regret of the best algorithm satisfies $\mathbb{P}[R_1(t) \geq \frac{1}{2}\Delta_{1,2}\tau] \geq \frac{1}{\tau^c}$ in Theorem 4.3 is equivalent to the condition that the regret of the base algorithms admit only a polynomial concentration. Results in (Salomon and Audibert, 2011) suggest that there cannot be a tighter bound on the tail of the regret for anytime algorithms. It is therefore unclear if the $\log(T)^2$ rate can be improved upon or if there exists a matching information-theoretic lower bound.

## 5 Corralling using Tsallis-INF

In this section, we consider an alternative approach, based on the work of Agarwal et al. (2016), which avoids running multiple copies of base algorithms. Since the approach is based on the OMD framework, which is naturally suited to losses instead of rewards, for the rest of the section we switch to losses.

We design a corralling algorithm that maintains a probability distribution $w \in \Delta^{K-1}$ over the base algorithms, $\{\mathcal{A}_i\}_{i=1}^K$. At each round, the corralling algorithm samples $i_t \sim w$. Next, $\mathcal{A}_{i_t}$ plays $a_{i_t,j_t}$ and the corralling algorithm observes the loss $\ell_t(a_{i_t,j_t})$. The corralling algorithm updates its distribution over the base algorithms using the observed loss and provides an unbiased estimate $\widehat{\ell}_t(a_{i,j_t})$ of $\ell_t(a_{i,j_t})$ to algorithm $\mathcal{A}_i$: the feedback provided to $\mathcal{A}_i$ is $\widehat{\ell}_t(a_{i_t,j_t}) = \frac{\ell_t(a_{i_t,j_t})}{w_{t,i_t}}$, and for all $a_{i,j_t} \neq a_{i_t,j_t}$, $\widehat{\ell}_t(a_{i,j_t}) = 0$. Notice that $\widehat{\ell}_t \in \mathbb{R}^K$, as opposed to $\ell_t \in [0,1]^{\prod_i k_i}$. Essentially, the loss fed to $\mathcal{A}_i$, with probability $w_{t,i}$, is the true loss rescaled by the probability $w_{t,i}$ to observe the loss, and is equal to 0 with probability $1 - w_{t,i}$.

The change of environment induced by the rescaling of the observed losses is analyzed in Agarwal et al. (2016). Following Agarwal et al. (2016), we denote the environment of the original losses $(\ell_t)_t$ as $\mathcal{E}$ and that of the rescaled losses $(\widehat{\ell}_t)_t$ as $\mathcal{E}'$. Therefore, in environment $\mathcal{E}$, algorithm $\mathcal{A}_i$ observes $\ell_t(a_{i,j_t})$ and in environment $\mathcal{E}'$, $\mathcal{A}_i$ observes $\widehat{\ell}_t(a_{i_t,j_t})$. A few important remarks are in order. As in (Agarwal et al., 2016), we need to assume that the base algorithms admit a *stability property* under the change of environment. In particular, if $w_{s,i} \geq \frac{1}{\rho_t}$ for all $s \leq t$ and some $\rho_t \in \mathbb{R}$, then $\mathbb{E}[R_i(t)]$ under environment $\mathcal{E}'$ is bounded by $\mathbb{E}[\sqrt{\rho_t} R_i(t)]$. For completeness, we provide the definition of stability by Agarwal et al. (2016).

**Definition 5.1.** *Let $\gamma \in (0,1]$ and let $R\colon \mathbb{N} \to \mathbb{R}_+$ be a non-decreasing function. An algorithm $\mathcal{A}$ with action space $\mathrm{A}$ is $(\gamma, R(\cdot))$-stable with respect to an environment $\mathcal{E}$ if its regret under $\mathcal{E}$ is $R(T)$ and its regret under $\mathcal{E}'$ induced by the importance weighting is $\max_{a \in \mathrm{A}} \mathbb{E}\left[\sum_{t=1}^T \widehat{\ell}_t(a_{i_t,j_t}) - \ell_t(a)\right] \leq \mathbb{E}[(\rho_T)^\gamma R(T)]$.*

We show that UCB-I (Auer et al., 2002a) satisfies the stability property above with $\gamma = \frac{1}{2}$. The techniques used in the proof are also applicable to other UCB-type algorithms. Other algorithms for stochastic bandits like Thompson sampling and OMD/FTRL variants have been shown to be 1/2-stable in (Agarwal et al., 2016).

The corralling algorithm of Agarwal et al. (2016) is based on Online Mirror Descent (OMD), where a key idea is to increase the step size whenever the probability of selecting some algorithm $\mathcal{A}_i$ becomes smaller than some threshold. This induces a negative regret term which, coupled with a careful choice of step size (dependent on regret upper bounds of the base algorithms), provides regret bounds that scale as a function of the regret of the best base algorithm.

Unfortunately, the analysis of the corralling algorithm always leads to at least a regret bound of $\tilde{\Omega}(\sqrt{T})$ and also requires knowledge of the regret bound of the best algorithm. Since our goal is to obtain instance-dependent regret bounds, we cannot appeal to this type of OMD approach. Instead, we draw inspiration from the recent work of Zimmert and Seldin (2018), who use a Follow-the-Regularized-Leader (FTRL) type of algorithm to design an algorithm that is simultaneously optimal for both stochastic and adversarially generated losses, without requiring knowledge of instance-dependent parameters such as the sub-optimality gaps to the loss of the best arm. The overall intuition for our algorithm is as follows. We use the FTRL-type algorithm proposed by Zimmert and Seldin (2018) until the probability to sample some arm falls below a threshold. Next, we run an OMD step with an increasing step size schedule which contributes a negative regret term. After the OMD step, we resume the normal step size schedule and updates from the FTRL algorithm. After carefully choosing the initial step size rate, which can be done in an instance-independent way, the accumulated negative regret terms are enough to compensate for the increased regret due to the change of environment.

## 5.1 Algorithm and the main result

We now describe our corralling algorithm in more detail. The potential function $\Psi_t$ used in all of the updates is defined by $\Psi_t(w) = -4\sum_{i\in[K]}\frac{1}{\eta_{t,i}}\left(\sqrt{w_i}-\frac{1}{2}w_i\right)$, where $\eta_t = \left[\eta_{t,1},\eta_{t,2},\ldots,\eta_{t,K}\right]$ is the step-size schedule during time $t$. The algorithm proceeds in epochs and begins by running each base algorithm for $\log(T)+1$ rounds. Each epoch is twice as large as the preceding, so that the number of epochs is bounded by $\log_2(T)$, and the step size schedule remains non-increasing throughout the epochs, except when an OMD step is taken. The algorithm also maintains a set of thresholds, $\rho_1,\rho_2,\ldots,\rho_n$, where $n = O(\log(T))$. These thresholds are used to determine if the algorithm executes an OMD step, while increasing the step size:

$$\begin{aligned}
w_{t+1} &= \operatorname*{argmin}_{w\in\Delta^{K-1}}\langle\widehat{\ell}_t, w\rangle + D_{\Psi_t}(w,w_t),\\
\eta_{t+1,i} &= \beta\eta_{t,i}\ (\text{for } i:\ w_{t,i}\le 1/\rho_{s_i}),\\
w_{t+2} &= \operatorname*{argmin}_{w\in\Delta^{K-1}}\langle\widehat{\ell}_{t+1}, w\rangle + D_{\Psi_{t+1}}(w,w_{t+1}), \rho_{s_i} = 2\rho_{s_i}
\end{aligned} \tag{3}$$

or the algorithm takes an FTRL step

$$w_{t+1} = \operatorname*{argmin}_{w\in\Delta^{K-1}}\langle\widehat{L}_t, w\rangle + \Psi_{t+1}(w), \tag{4}$$

where $\widehat{L}_t = \widehat{L}_{t-1}+\widehat{\ell}_t$, unless otherwise specified by the algorithm. We note that the algorithm can only increase the step size during the OMD step. For technical reasons, we require an FTRL step after each OMD step. Further, we require that the second step of each epoch be an OMD step if there exists at least one $w_{t,i}\le\frac{1}{\rho_1}$. The algorithm also can enter an OMD step during an epoch if at least one $w_{t,i}$ becomes smaller than a threshold $\frac{1}{\rho_{s_i}}$ which has not been exceeded so far.

**Algorithm 2** Corralling with Tsallis-INF

```
Input: Mult. constant $\beta$, thresholds $\{\rho_i\}_{i=1}^n$, initial
    step size $\eta$, epochs $\{\tau_i\}_{i=1}^m$, algorithms $\{\mathcal{A}_i\}_{i=1}^K$.
Output: Algorithm selection sequence $(i_t)_{t=1}^T$.
 1: Initialize $t=1$, $w_1 = Unif(\Delta^{K-1})$, $\eta_1=\eta$
 2: Initialize current threshold list $\theta\in[n]^K$ to $\mathbf{1}$
 3: while $t\le K\log(T)+K$
 4:   for $i\in[K]$
 5:     $\mathcal{A}_i$ plays $a_{i,j_t}$, $\widehat{L}_{1,i}+=\ell_t(a_{i,j_t}), t+=1$
 6: $t=2, w_2=\nabla\Phi_2(-\widehat{L}_1), 1/\eta_{t+1}^2=1/\eta_t^2+1$
 7: while $j\le m$
 8:   for $t\in\tau_j$
 9:     $\mathcal{R}_t=\emptyset$,$\widehat{\ell}_t=$ PLAY-ROUND$(w_t)$
10:     if $t$ is first round of $\tau_j$ and $\exists w_{t,i}\le\frac{1}{\rho_1}$
11:       for $i$: $w_{t,i}\le\frac{1}{\rho_1}$
12:         $\theta_i=\min\{s\in[n]: w_{t,i}>\frac{1}{\rho_s}\}$, $\mathcal{R}_t=$
            $\mathcal{R}_t\bigcup\{i\}$.
13:       $(w_{t+3},\widehat{L}_{t+2})=$ NRS$(w_t,\widehat{\ell}_t,\eta_t,\mathcal{R}_t,\widehat{L}_{t-1})$, $t=$
          $t+2$, $\widehat{\ell}_t=$ PLAY-ROUND$(w_t)$
14:     if $\exists i$: $w_{t,i}\le\frac{1}{\rho_{\theta_i}}$ and prior step was not NRS
15:       for $i$: $w_{t,i}\le\frac{1}{\rho_{\theta_i}}$
16:         $\theta_i+=1$, $\mathcal{R}_t=\mathcal{R}_t\bigcup\{i\}$.
17:       $(w_{t+3},\widehat{L}_{t+2})=$ NRS$(w_t,\widehat{\ell}_t,\eta_t,\mathcal{R}_t,\widehat{L}_{t-1})$, $t=$
          $t+2$, $\widehat{\ell}_t=$ PLAY-ROUND$(w_t)$
18:     else
19:       $1/\eta_{t+1}^2=1/\eta_t^2+1$, $w_{t+1}=\nabla\Phi_{t+1}(-\widehat{L}_t)$
```

We set the probability thresholds so that $\rho_1=O(1)$, $\rho_j=2\rho_{j-1}$ and $\frac{1}{\rho_n}\ge\frac{1}{T}$, so that $n\le\log_2(T)$. In the beginning of each epoch, except for the first epoch, we check if $w_{t,i}<\frac{1}{\rho_1}$. If it is, we increase the step size as $\eta_{t+1,i}=\beta\eta_{t,i}$ and run the OMD step. The pseudocode for the algorithm is given in Algorithm 2. The routines OMD-STEP and PLAY-ROUND can be found in Algorithm 6 and Algorithm 7 (Appendix D) respectively. OMD-STEP essentially does the update described in Equation 3 and PLAY-ROUND samples and plays an algorithm, after which constructs an unbiased estimator of the losses and feeds these back to all of the sub-algorithms. We show the following regret bound for the corralling algorithm.

**Theorem 5.2.** *Let $\bar{R}_i(\cdot)$ be a function upper bounding the expected regret, $\mathbb{E}[R_i(\cdot)]$, of $\mathcal{A}_i$ for all $i\in[K]$. For $\beta=e^{1/\log(T)^2}$ and for $\eta$ such that for all $i\in[K]$, $\eta_{1,i}\le\min_{t\in[T]}\frac{\left(1-\exp\left(-\frac{1}{\log(T)^2}\right)\right)\sqrt{t}}{50\bar{R}_i(t)}$, the expected*

**Algorithm 3** NEG-REG-STEP(NRS)

**Input:** Prior iterate $w_t$, loss $\widehat{\ell}_t$, step size $\eta_t$, set of rescaled step-sizes $\mathcal{R}_t$, cumulative loss $\widehat{L}_{t-1}$
**Output:** Plays two rounds of the game and returns distribution $w_{t+3}$ and cumulative loss $\widehat{L}_{t+2}$

```
1: $(w_{t+1}, \widehat{L}_t) =$ OMD-STEP$(w_t, \widehat{\ell}_t, \eta_t, \mathcal{R}_t, \widehat{L}_{t-1})$
2: $\widehat{\ell}_{t+1} =$ PLAY-ROUND$(w_{t+1})$, $\widehat{L}_{t+1} = \widehat{L}_t + \widehat{\ell}_{t+1}$
3: for all $i$ such that $w_{t,i} \leq \frac{1}{\rho_1}$
4:     $\eta_{t+2,i} = \beta\eta_{t,i}$, $\mathcal{R}_t = \mathcal{R}_t \cup \{i\}$ and restart $\mathcal{A}_i$ with updated environment $\theta_i = \frac{1}{2w_{t,i}}$
5: $w_{t+2} = \nabla\Phi_{t+2}(-\widehat{L}_{t+1})$
6: $\widehat{\ell}_{t+2} =$ PLAY-ROUND$(w_{t+2})$
7: $\widehat{L}_{t+2} = \widehat{L}_{t+1} + \widehat{\ell}_{t+2}, \eta_{t+3} = \eta_{t+2}, t = t + 2$
8: $w_{t+1} = \nabla\Phi_{t+1}(-\widehat{L}_t), t = t + 1$
```

*regret of Algorithm 2 is bounded as follows:* $\mathbb{E}\left[R(T)\right] \leq O\left(\sum_{i\neq i^*} \frac{\log(T)}{\eta_{1,i}^2\Delta_i} + \mathbb{E}[R_{i^*}(T)]\right)$.

To parse the bound above, suppose $\{\mathcal{A}_i\}_{i\in[K]}$ are standard stochastic bandit algorithms such as UCB-I. In Theorem 5.4, we show that UCB-I is indeed $\frac{1}{2}$-stable as long as we are allowed to rescale and introduce an additive factor to the confidence bounds. In this case, a worst-case upper bound on the regret of any $\mathcal{A}_i$ is $\mathbb{E}[R_i(t)] \leq c\sqrt{k_i \log{(t)}\, t}$ for all $t \in [T]$ and some universal constant $c$. We note that the min-max regret bound for the stochastic multi-armed bandit problem is $\Theta(\sqrt{KT})$ and most known any-time algorithms solving the problem achieve this bound up to poly-logarithmic factors. Further we note that $\left(1 - \exp\left(-\frac{1}{\log(T)^2}\right)\right) > \frac{1}{e\log(T)^2}$. This suggests that the bound in Theorem 5.2 on the regret of the corralling algorithm is at most $O\big(\sum_{i\neq i^*} \frac{k_i \log(T)^5}{\Delta_i} + \mathbb{E}[R_{i^*}(T)]\big)$. In particular, if we instantiate $\mathbb{E}[R_{i^*}(T)]$ to the instance-dependent bound of $O\left(\sum_{j\neq 1} \frac{\log(T)}{\Delta_{i^*,j}}\right)$, the regret of Algorithm 2 is bounded by $O\left(\sum_{i\neq i^*} \frac{k_i\log(T)^5}{\Delta_i} + \sum_{j\neq 1} \frac{\log(T)}{\Delta_{i^*,j}}\right)$. In general we cannot exactly compare the current bound with that of UCB-C (Algorithm 1), as the regret bound in Theorem 5.2 has worse scaling in the time horizon on the gap-dependent terms, compared to the regret bound in Theorem 4.2, but has no additional scaling in front of the $\mathbb{E}[R_{i^*}(T)]$ term. In practice we observe that Algorithm 2 outperforms Algorithm 1.

Since essentially all stochastic multi-armed bandit algorithms enjoy a regret bound, in time horizon, of the order $\tilde{O}(\sqrt{T})$, we are guaranteed that $1/\eta_{t,i}^2$ scales only poly-logarithmically with the time horizon. What happens, however, if algorithm $\mathcal{A}_i$ has a worst case regret bound of the order $\omega(\sqrt{T})$? For the next part of the discussion, we only focus on time horizon dependence. As a simple example, suppose that $\mathcal{A}_i$ has worst case regret of $T^{2/3}$ and that $\mathcal{A}_{i^*}$ has a worst case regret of $\sqrt{T}$. In this case, Theorem 5.2 tells us that we should set $\eta_{1,i} = \tilde{O}(1/T^{1/6})$ and hence the regret bound scales at least as $\Omega(T^{1/3}/\Delta_i + \mathbb{E}[R_{i^*}(T)])$. In general, if the worst case regret bound of $\mathcal{A}_i$ is in the order of $T^\alpha$ we have a regret bound scaling at least as $T^{2\alpha-1}/\Delta_i$. This is not unique to Algorithm 2 and a similar scaling of the regret would occur in the bound for Algorithm 1 due to the scaling of confidence intervals.

**Corralling in an adversarial environment.** Because Algorithm 2 is based on a best of both worlds algorithm, we can further handle the case when the losses/rewards are generated adversarially or whenever the best overall arm is shared across multiple algorithms, similarly to the settings studied by Agarwal et al. (2016); Pacchiano et al. (2020).

**Theorem 5.3.** *Let $\bar{R}_{i^*}(\cdot)$ be a function upper bounding the expected regret of $\mathcal{A}_{i^*}$, $\mathbb{E}[R_{i^*}(\cdot)]$. For any $\eta_{1,i^*} \leq \min_{t\in[T]} \frac{\left(1-\exp\left(-\frac{1}{\log(T)^2}\right)\right)\sqrt{t}}{50\bar{R}_{i^*}(t)}$ and $\beta = e^{1/\log(T)^2}$ it holds that the expected regret of Algorithm 2 is bounded as follows:* $\mathbb{E}\left[R(T)\right] \leq O\left(\max_{w\in\Delta^{K-1}} \sqrt{T}\sum_{i=1}^K \frac{\sqrt{w_i}}{\eta_{1,i}} + \mathbb{E}[R_{i^*}(T)]\right)$.

The bound in Theorem 5.3 essentially evaluates to $O(\max(\sqrt{TK}, \max_{i\in[K]} \bar{R}_i(T)) + \mathbb{E}[R_{i^*}(T)])$. Unfortunately, this is not quite enough to recover the results in (Agarwal et al., 2016; Pacchiano et al., 2020). This is attributed to the fact that we use the $\frac{1}{2}$-Tsallis entropy as the regularizer instead of the log-barrier function. It is possible to improve the above bound for algorithms with stability $\gamma < 1/2$, however, because model selection is not the primary focus of this work, we will not present such results here.

**Stability of UCB-I.** We now briefly discuss how the regret bounds of UCB-I and similar algorithms change whenever the variance of the stochastic losses is rescaled by Algorithm 2. Let us focus on base learner $\mathcal{A}_i$ during epoch $\tau_j$. During epoch $\tau_j$, there is some largest threshold $\rho_{s_i}$ which is never exceeded by the inverse probabilities, i.e., $\min_{t\in\tau_j} w_{t,i} \geq 1/\rho_{s_i}$. This implies that the rescaled losses are in $[0, \rho_{s_i}]$. Further, their variance is bounded by $\mathbb{E}[\widehat{\ell}_t(i)^2] = \mathbb{E}[\ell_t(a_{i,j_t})^2/w_{t,i}] \leq \rho_{s_i}$. Using a version of Freedman's inequality (Freedman, 1975), we show the following.

**Theorem 5.4** (Informal)**.** *Suppose that during epoch $\tau_j$ of size $\mathcal{T}_j$, UCB-I (Auer et al., 2002a) uses an upper confidence bound $\sqrt{\frac{4\rho_{s_i}\log(t)}{T_{i,j}(t)}} + \frac{4\rho_{s_i}\log(t)}{3T_{i,j}(t)}$ for arm $j$ at time $t$. Then, the expected regret of $\mathcal{A}_i$ under the rescaled rewards is at most* $\mathbb{E}[R_i(\mathcal{T}_j)] \leq \sqrt{8\rho_{s_i}k_i\mathcal{T}_j\log{(\mathcal{T}_j)}}$.

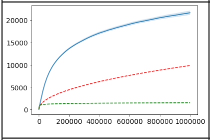
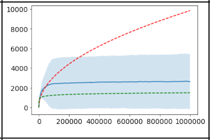
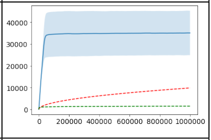
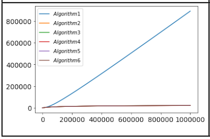
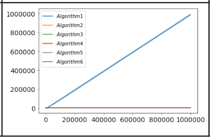
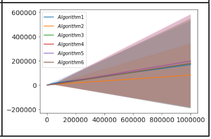
| Corral | Algorithm 2 | Algorithm 1 |
|---|---|---|
|  |  |  |
|  |  |  |

Table 1: Regret for corralling when $\Delta_i = 0.2$

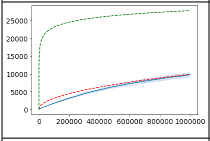
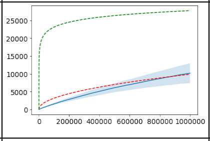
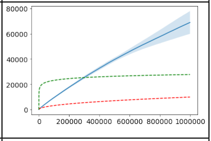
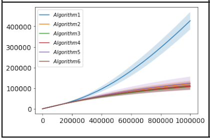
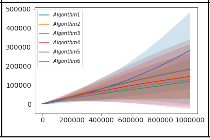
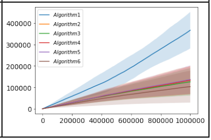
| Corral | Algorithm 2 | Algorithm 1 |
|---|---|---|
|  |  |  |
|  |  |  |

Table 2: Regret for corralling when $\Delta_i = 0.02$

We expect that other UCB-type algorithms (Audibert et al., 2009; Garivier and Cappé, 2011; Bubeck et al., 2013; Garivier et al., 2018a) should also be $\frac{1}{2}$-stable.

## 6 Empirical results

In this section, we further examine the empirical properties of our algorithms via experiments on synthetically generated datasets. We compare Algorithm 1 and Algorithm 2 to the Corral algorithm (Agarwal et al., 2016)[Algorithm 1], which is also used in (Pacchiano et al., 2020). We note that Pacchiano et al. (2020) also use Exp3.P as a corralling algorithm. Recent work (Lee et al., 2020) suggests that Corral exhibits similar high probability regret guarantees as Exp3.P and that Corral would completely outperform Exp3.P.

**Experimental setup.** The algorithms that we corral are UCB-I, Thompson sampling (TS), and FTRL with $\frac{1}{2}$-Tsallis entropy reguralizer (Tsallis-INF). When implementing Algorithm 2 and Corral, we make an important deviation from what theory prescribes: we *never* restart the corralled algorithms and run them with their default parameters. In all our experiments, we corral two instances of UCB-I, TS, and Tsallis-INF for a total of six algorithms. The algorithm containing the best arm plays over 10 arms. Every other algorithm plays over 5 arms. The rewards for each base algorithm are Bernoulli random variables with expectations set so that for all $i > 2$ and $j > 1$, $\Delta_{i,j} = 0.01$. We run two sets of experiments with $\Delta_i$ equal to either 0.2 or 0.02. This setting implies that Algorithm 1 always contains the best arm and that the best arm of each base algorithm is arm one. Even though $\Delta_{i,j} = 0.01$ implies large regret for all sub-optimal algorithms, it also reduces the variance of the total reward for these algorithms thereby making the corralling problem harder. Finally, the time horizon is set to $T = 10^6$. For a more extensive discussion, about our choice of algorithms and parameters for the experimental setup we refer the reader to Appendix A.

**Large gap experiments.** Table 1 reports the regret (top) and number of plays of each algorithm found in our experiments when $\Delta_i = 0.2$. The plots represent the average regret, in blue, and the average number of pulls of each algorithm (color according to the legend) over 75 runs of each experiment. The standard deviation is represented by the shaded blue region. The algorithm that contains the optimal arm is $\mathcal{A}_1$ and is an instance of UCB-I. The red dotted line in the top plots is given by $4\sqrt{KT}+\mathbb{E}[R_1(T)]$, and the green dotted line is given by $4\sum_{i\neq 1}\frac{k_i\log(T)}{\Delta_i}+\mathbb{E}[R_1(T)]$. These lines serve as a reference across experiments and we believe they are more accurate upper bounds for the regret of the proposed and existing algorithms. As expected, we see that, in the large gap regime, the Corral algorithm exhibits $\Omega(\sqrt{T})$ regret, while the regret of Algorithm 2 remains bounded in $O(\log(T))$. Algorithm 1 admits two regret phases. In the initial phase, its regret is linear, while in the second phase it is logarithmic. This is typical of UCB strategies in the stochastic MAB problem (Garivier et al., 2018b).

**Small gap experiments** Table 2 reports the results of our experiments for $\Delta_i = 0.02$. The setting of the experiments is the same as in the large gap case. We observe that both Corral and Algorithm 2 behave according to the $O(\sqrt{T})$ bounds. This is expected since, when $\Delta_i = 0.02$, the optimistic bound dominates the $\sqrt{T}$-bound. The result for Algorithm 1 might be somewhat surprising, as its regret exceeds both the green and red lines. We emphasize that this experiment does not contradict Theorem 4.2. Indeed, if we were to plot the green and red lines according to the bounds of Theorem 4.2, the regret would remain below both lines.

Our experiments suggest that Algorithm 2 is the best corralling algorithm. A tighter analysis would potentially yield optimistic regret bounds in the order of $O\left(\sum_{i\neq i^*}\frac{k_i\log(T)}{\Delta_i}+\mathbb{E}[R_{i^*}(T)]\right)$. Furthermore, we expect that the bounds of Theorem 4.2 are tight. For more detailed experiments, we refer the reader to Appendix A.

## 7 Model selection for linear bandits

While the main focus of the paper is corralling MAB base learners when there exists a best overall base algorithm, we now demonstrate that several known model selection results can be recovered using Algorithm 2.

We begin by recalling the model selection problem for linear bandits. The learner is given access to a set of loss functions $\mathcal{F}\colon \mathcal{X}\times\mathcal{A}\to\mathbb{R}$ mapping from contexts $\mathcal{X}$ and actions $\mathcal{A}$ to losses. In the linear bandits setting, $\mathcal{F}$ is structured as a nested sequence of classes $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \ldots \subseteq \mathcal{F}_K = \mathcal{F}$, where each $\mathcal{F}_i$ is defined as

$$\mathcal{F}_i = \{(x,a) \to \langle \beta_i, \phi_i(x,a)\rangle : \beta_i \in \mathbb{R}^{d_i}\},$$

for some feature embedding $\phi_i\colon \mathcal{X}\times\mathcal{A}\to\mathbb{R}^{d_i}$. It is assumed that each feature embedding $\phi_i$ contains $\phi_{i-1}$ as its first $d_{i-1}$ coordinates. It is further assumed that there exists a smallest $i^* \leq K$ to which the optimal parameter $\beta^*$ belongs, that is the observed losses for each context-action pair $(x,a)$ satisfy $\ell_t(x,a) = \mathbb{E}\left[\langle \beta^*, \phi_{d_{i^*}}(x,a)\rangle\right]$. The goal in the model selection problem is to identify $i^*$ and compete against the smallest loss for the $t$-th context in $\mathbb{R}^{d_{i^*}}$ by minimizing the regret:

$$R_{i^*}(T) = \sum_{t=1}^{T}\Big(\mathbb{E}[\langle \beta^*, \phi_{i^*}(x_t,a_t)\rangle] - \min_{a\in\mathcal{A}}\mathbb{E}[\langle \beta^*, \phi_{i^*}(x_t,a)\rangle]\Big),$$

where the expectation is with respect to all randomness in the sampling of the contexts $x_t \sim \mathcal{D}$, actions and additional noise. We adopt the standard assumption that, given $x_t$, the observed loss for any $a$ can be expressed as follows: $\langle \beta, \phi_i(x_t,a)\rangle + \xi_t$, where $\xi_t$ is a zero-mean, sub-Gaussian random variable with variance proxy 1 and for each of the context-action pairs it holds that $\langle \beta, \phi_i(x_t,a)\rangle \in [0,1]$.

### 7.1 Algorithm and main result

We assume that there are $K$ base learners $\{\mathcal{A}_i\}_{i=1}^K$ such that the regret of $\mathcal{A}_i$, for $i \geq i^*$, is bounded by $\tilde{O}(d_i^\alpha\sqrt{T})$. That is, whenever the model is correctly specified, the $i$-th algorithm admits a meaningful regret guarantee. In the setting of Foster et al. (2019), $\mathcal{A}_i$ can be instantiated as LinUCB and in that case $\alpha = 1/2$. Further, in the setting of infinite arms, $\mathcal{A}_i$ can be instantiated as OFUL (Abbasi-Yadkori et al., 2011), in which case $\alpha = 1$. Both $\alpha = 1/2$ and $\alpha = 1$ govern the min-max optimal rates in the respective settings. Our algorithm is now a simple modification of Algorithm 2. At every time-step $t$, we update $\widehat{L}_t = \widehat{L}_{t-1} + \widehat{\ell}_t + \mathbf{d}$, where $\mathbf{d}_i = \frac{d_i^{2\alpha}}{\sqrt{T}}$. Intuitively, our modification creates a gap between the losses of $\mathcal{A}_{i^*}$ and any $\mathcal{A}_i$ for $i > i^*$ of the order $d_i^{2\alpha}$. On the other hand for any $i < i^*$, perturbing the loss can result in at most additional $d_{i^*}^{2\alpha}\sqrt{T}$ regret. With the above observations, the bound guaranteed by Theorem 5.2 implies that the modified algorithm should incur at most $\tilde{O}(d_{i^*}^{2\alpha}\sqrt{T})$ regret. In Appendix F, we show the following regret bound.

**Theorem 7.1.** *Assume that every base learner $\mathcal{A}_i$, $i \geq i^*$, admits a $\tilde{O}(d_i^\alpha\sqrt{T})$ regret. Then, there exists a corralling strategy with expected regret bounded by $\tilde{O}(d_{i^*}^{2\alpha}\sqrt{T} + K\sqrt{T})$. Moreover, under the additional assumption that the following holds for any $i < i^*$, for all $(x,a) \in \mathcal{X}\times\mathcal{A}$*

$$\mathbb{E}[\langle \beta_i, \phi_i(x,a)\rangle] - \min_{a\in\mathcal{A}}\mathbb{E}[\langle \beta^*, \phi_{i^*}(x,a)\rangle] \geq 2\frac{d_{i^*}^{2\alpha} - d_i^{2\alpha}}{\sqrt{T}},$$

*the expected regret of the same strategy is bounded as $\tilde{O}(d_{i^*}^{\alpha}\sqrt{T} + K\sqrt{T})$.*

Typically, we have $K = O(\log(T))$ and thus Theorem 7.1 guarantees a regret of at most $\tilde{O}(d_{i^*}^{2\alpha}\sqrt{T})$. Furthermore, under a gap-assumption, which implies that the value of the smallest loss for the optimal embedding $i^*$ is sufficiently smaller compared to the value of any sub-optimal embedding $i < i^*$, we can actually achieve a corralling regret of the order $R_{i^*}(T)$. In particular, for the setting of Foster et al. (2019), our strategy yields the desired $\tilde{O}(\sqrt{d_{i^*}T})$ regret bound. Notice that the regret guarantees are only meaningful as long as $d_{i^*} = o(T^{1/(2\alpha)})$. In such a case, the second assumption on the gap is that the gap is lower bounded by $o(1)$. This is a completely problem-dependent assumption and in general we expect that it cannot be satisfied.

## 8 Conclusion

We presented an extensive analysis of the problem of corralling stochastic bandits. Our algorithms are applicable to a number of different contexts where this problem arises. There are also several natural extensions and related questions relevant to our study. One natural extension is the case where the set of arms accessible to the base algorithms admit some overlap and where the reward observed by one algorithm could serve as side-information to another algorithm. Another extension is the scenario of corralling online learning algorithms with feedback graphs. In addition to these and many other interesting extensions, our analysis may have some connection with the study of other problems such as model selection in contextual bandits (Foster et al., 2019) or active learning.

## Acknowledgements

This research was supported in part by NSF BIGDATA awards IIS-1546482, IIS-1838139, NSF CAREER award IIS-1943251, and by NSF CCF-1535987, NSF IIS-1618662, and a Google Research Award. RA would like to acknowledge support provided by Institute for Advanced Study and the Johns Hopkins Institute for Assured Autonomy. We warmly thank Julian Zimmert for insightful discussions regarding the Tsallis-INF approach.

## References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *NIPS*, volume 11, pages 2312–2320, 2011.

Alekh Agarwal, Haipeng Luo, Behnam Neyshabur, and Robert E Schapire. Corralling a band of bandit algorithms. *arXiv preprint arXiv:1612.06246*, 2016.

Raman Arora, Ofer Dekel, and Ambuj Tewari. Deterministic MDPs with adversarial rewards and bandit feedback. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 93–101, 2012.

Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *COLT*, pages 217–226, 2009.

Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

Peter Auer and Chao-Kai Chiang. An algorithm with nearly optimal pseudo-regret for both stochastic and adversarial bandits. In *Conference on Learning Theory*, pages 116–120, 2016.

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

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

Peter L Bartlett, Varsha Dani, Thomas Hayes, Sham Kakade, Alexander Rakhlin, and Ambuj Tewari. High-probability regret bounds for bandit online linear optimization. In *Conference on Learning Theory*, 2008.

Chiradeep BasuMallick. What is display advertising? definition, targeting process, management, network, types, and examples. https://marketing.toolbox.com/articles/what-is-display-advertising-definition-targeting-process-management-network-types-and-examples, 2020.

Haim Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.

Sébastien Bubeck. *Bandits games and clustering foundations*. PhD thesis, INRIA Nord Europe (Lille, France), 2010.

Sébastien Bubeck and Aleksandrs Slivkins. The best of both worlds: Stochastic and adversarial bandits. In *Conference on Learning Theory*, pages 42–1, 2012.

Sébastien Bubeck, Nicolo Cesa-Bianchi, and Gábor Lugosi. Bandits with heavy tail. *IEEE Transactions on Information Theory*, 59(11):7711–7717, 2013.

Sébastien Bubeck, Yin Tat Lee, and Ronen Eldan. Kernel-based methods for bandit convex optimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 72–85, 2017.

Niladri S Chatterji, Vidya Muthukumar, and Peter L Bartlett. Osom: A simultaneously optimal algorithm for multi-armed and linear contextual bandits. *arXiv preprint arXiv:1905.10040*, 2019.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Pac bounds for multi-armed bandit and markov decision processes. In *International Conference on Computational Learning Theory*, pages 255–270. Springer, 2002.

Dylan J. Foster, Zhiyuan Li, Thodoris Lykouris, Karthik Sridharan, and Éva Tardos. Learning in games: Robustness of fast convergence. In *Proceedings of NIPS*, pages 4727–4735, 2016.

Dylan J Foster, Akshay Krishnamurthy, and Haipeng Luo. Model selection for contextual bandits. In *Advances in Neural Information Processing Systems*, pages 14714–14725, 2019.

David A Freedman. On tail probabilities for martingales. *the Annals of Probability*, pages 100–118, 1975.

Aurélien Garivier and Olivier Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual conference on learning theory*, pages 359–376, 2011.

Aurélien Garivier, Hédi Hadiji, Pierre Menard, and Gilles Stoltz. Kl-ucb-switch: optimal regret bounds for stochastic bandits from both a distribution-dependent and a distribution-free viewpoints. *arXiv preprint arXiv:1805.05071*, 2018a.

Aurélien Garivier, Pierre Ménard, and Gilles Stoltz. Explore first, exploit next: The true shape of regret in bandit problems. *Mathematics of Operations Research*, 44(2):377–399, 2018b.

Chung-Wei Lee, Haipeng Luo, Chen-Yu Wei, and Mengxiao Zhang. Bias no more: high-probability data-dependent regret bounds for adversarial bandits and mdps. *arXiv preprint arXiv:2006.08040*, 2020.

Odalric-Ambrym Maillard and Rémi Munos. Adaptive bandits: Towards the best history-dependent strategy. In *Proceedings of AISTATS*, pages 570–578, 2011.

Aldo Pacchiano, My Phan, Yasin Abbasi-Yadkori, Anup Rao, Julian Zimmert, Tor Lattimore, and Csaba Szepesvari. Model selection in contextual stochastic bandit problems. *arXiv preprint arXiv:2003.01704*, 2020.

Antoine Salomon and Jean-Yves Audibert. Deviations of stochastic bandit regret. In *International Conference on Algorithmic Learning Theory*, pages 159–173. Springer, 2011.

Yevgeny Seldin and Gábor Lugosi. An improved parametrization and analysis of the exp3++ algorithm for stochastic and adversarial bandits. In *The 30th Annual Conference on Learning Theory (COLT) Conference on Learning Theory*, pages 1743–1759. Proceedings of Machine Learning Research, 2017.

Yevgeny Seldin and Aleksandrs Slivkins. One practical algorithm for both stochastic and adversarial bandits. In *ICML*, pages 1287–1295, 2014.

Chen-Yu Wei and Haipeng Luo. More adaptive algorithms for adversarial bandits. In *Proceedings of COLT 2018*, pages 1263–1291, 2018.

Julian Zimmert and Yevgeny Seldin. An optimal algorithm for stochastic and adversarial bandits. *arXiv preprint arXiv:1807.07623*, 2018.

Julian Zimmert, Haipeng Luo, and Chen-Yu Wei. Beating stochastic and adversarial semi-bandits optimally and simultaneously. In *Proceedings of ICML*, pages 7683–7692, 2019.