Dynamic Balancing for Model Selection in Bandits and RL

Ashok Cutkosky [*1] Christoph Dann [*2] Abhimanyu Das [*3] Claudio Gentile [*2] Aldo Pacchiano [*4]
Manish Purohit [*3]

Abstract

We propose a framework for model selection by combining base algorithms in stochastic bandits and reinforcement learning. We require a candidate regret bound for each base algorithm that may or may not hold. We select base algorithms to play in each round using a "balancing condition" on the candidate regret bounds. Our approach simultaneously recovers previous worst-case regret bounds, while also obtaining much smaller regret in natural scenarios when some base learners significantly exceed their candidate bounds. Our framework is relevant in many settings, including linear bandits and MDPs with nested function classes, linear bandits with unknown misspecification, and tuning confidence parameters of algorithms such as LinUCB. Moreover, unlike recent efforts in model selection for linear stochastic bandits, our approach can be extended to consider adversarial rather than stochastic contexts.

1. Introduction

Multi-armed bandits are a sequential learning framework whereby a learning agent repeatedly interacts with an unknown environment across a sequence of T *rounds*. During each round, the learner picks an action from a set of available actions (possibly after observing some *context* information for that round), and the environment generates a feedback signal in the form of a *reward* value, associated with the chosen action for that context. Given a class of policies, the goal of the learning agent is to accumulate a total reward during the T rounds which is not much smaller than that of the best policy in hindsight within the class.

This problem has been extensively studied under diverse assumptions about the class of policies, the source generating reward signals, the shape of the action space, etc. (e.g. see Auer et al., 2002; Langford & Zhang, 2008; Beygelzimer et al., 2011; Lattimore & Szepesvári, 2020; Agarwal et al., 2014). Many of these algorithms have different behaviors in different environments; for instance, one algorithm might do much better if the average reward is a linear function of the context, while another might do better when reward and context are independent. This plethora of prior algorithms necessitates "meta-decisions": If the environment is not known in advance, which algorithm should be used for the task at hand? This is especially important for industrial deployment, where the complexity and diversity of the available solutions typically require being able to select among several alternatives, like selecting the best within a pool of bandit algorithms, or even alternative configurations of the same algorithm (as in, e.g., hyper-parameter optimization).

We model the above meta-decision by assuming we have access to a pool of M base bandit algorithms, and our goal is to design a bandit meta-algorithm, whose actions are the base algorithms themselves, such the total regret experienced by the meta-algorithm is comparable to that of the best base algorithm in hindsight for the environment at hand. In each round t, the meta-algorithm needs to choose one of the base algorithms, and plays the action suggested by that algorithm. Since we do not know in advance which base algorithm will perform best, we need to address this problem in an online fashion. We call this problem online *model selection* for bandit algorithms. In this paper, we focus on this problem for *stochastic* environments.

The bandit model selection problem has received a lot of recent attention, as witnessed by a flurry of recent works (e.g., Foster et al., 2019; Abbasi-Yadkori et al., 2020; Pacchiano et al., 2020b; Arora et al., 2021; Ghosh et al., 2021; Chatterji et al., 2020; Bibaut et al., 2020; Foster et al., 2020a; Lee et al., 2021). A pioneering prior work in this domain has considered the *adversarial* setting (Agarwal et al., 2017) and utilizes an adversarial meta algorithm based on mirror descent. However their algorithm (CORRAL) needs some technical stability conditions for the base learners, thus requiring each base algorithm to be individually modified. Recently, Pacchiano et al. (2020b) extended CORRAL for the stochastic setting, while dispensing with the stability

[*] Equal contribution [1] Boston University, Boston, Massachussetts, USA [2] Google Research, New York, NY, USA [3] Google Research, Mountain View, California, USA [4] University of California, Berkeley, California, USA. Correspondence to: Ashok Cutkosky <ashok@cutkosky.com>, Christoph Dann <cdann@cdann.net>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

conditions on the base algorithms.

In this paper, we propose a general-purpose meta-algorithm that can be used in combination with any set of stochastic base bandit algorithms (without requiring any modifications or stability conditions on the base algorithms). Our algorithm is based on the principle of *dynamic regret balancing*, which generalizes an approach by Abbasi-Yadkori et al. (2020). We require only a putative (or candidate) regret bound for each base learner that may or may not hold. In each round, the algorithm maintains a set of "active" base-learners based on a misspecification criterion, and then chooses a base learner among them so as to make all putative regret bounds (evaluated at the number of rounds that the respective base learner was played) to be roughly equal, up to carefully-chosen bias and scale factors. Under the assumption that at least one of the base learners' putative regret bound is indeed valid, we show that our algorithm's total regret, in many settings, is only a multiplicative factor of the regret of the best base algorithm in hindsight. Besides, the parameters of our algorithm are solely derived from the putative regret bounds, so that once these putative bounds are available, no further parameters have to be tuned.

Our technique is both different and simpler than previous model selection techniques such as those by Pacchiano et al. (2020b), while still recovering (and in some settings, improving on) their regret guarantees. For example, when each base learner comes with a putative regret bound of $d_i\sqrt{T}$, we obtain regret guarantees that recover the regret of the best learner up to a multiplicative factor of $\sqrt{B}d_\star$ where B is the number of misspecified base learners, and $\star$ is the best well-specified base learner. This improves on the $\sqrt{M}d_\star$ result of Pacchiano et al. (2020b) when there are only a few misspecified base learners. The simplicity of our algorithm also comes with a much smaller memory requirement ($O(M)$ compared to $O(TM)$ of Pacchiano et al. (2020b)).

Furthermore, our algorithm can simultaneously provide *gap-dependent* regret bounds under various suitable "gap" assumptions on the base learners, that can avoid the $\sqrt{T}$ limit of the previous adversarial meta-algorithm approaches. This lets us provide overall regret bounds that depend only on a multiplicative factor of the best learner, even for the case when the base learners have $o(\sqrt{T})$ putative regret (including the case of $O(\log(T))$ base learners). Arora et al. (2021) showed that gap-dependent results are possible in the related problem of corralling stochastic multi-armed bandit algorithms. Furthermore, Lee et al. (2021) recently also obtained gap-dependent regret bounds for model-selection in reinforcement learning – however, their algorithm has a weaker gap-independent regret bound with a $T^{2/3}$ dependence.

Unlike most prior model-selection work, our regret guarantees can also be made non-uniform over the base learners by utilizing the user-specified biases over the base algorithms in our regret balancing algorithm. This lets us capture some notion of "prior knowledge" over the base algorithms and allows us to obtain more delicate trade-offs between their performances in a manner reminiscent of (Lattimore, 2015). Additionally, unlike previous work, our approach also extends to the case when contexts are generated from an adversarial environment, rather than a stochastic one.

Our results can be specialized to various model-selection applications such as linear bandits and MDPs with nested function classes (Foster et al., 2019), linear bandits with unknown misspecification, and confidence-parameter tuning for contextual linear bandits. For the case of linear bandits with nested model classes, we show that using our gap-dependent bounds, we can recover the optimal $d_*\sqrt{T}$ regret dependence in the infinite action space setting.

To summarize, our contributions significantly advance the state of the art, especially when compared to Abbasi-Yadkori et al. (2020); Pacchiano et al. (2020b), which are the references closest to this work. In particular:

- Our worst-case regret $\tilde{O}(d_\star^2\sqrt{BT})$ improves the best known rate of $\tilde{O}(d_\star^2\sqrt{MT})$ by Pacchiano et al. (2020b).
- Unlike Abbasi-Yadkori et al. (2020); Pacchiano et al. (2020b), we provide simultaneous $\sqrt{T}$ worst-case and $\ln(T)$ gap-dependent regret bounds under general conditions.
- Our gap-dependent guarantees allow us to prove model selection bounds in a number of relevant settings which strongly outperform existing results, e.g., our $d^*\sqrt{T}$ bound in Section 5.1 for nested model classes.
- We have several other new results that cannot be obtained by prior work, e.g., adversarial contexts (Section 6) and demonstrate the usefulness of our approach empirically in Section 7.

2. Setup and Assumptions

We consider contextual sequential decision making problems described by a context space $\mathcal{X}$, an action space $\mathcal{A}$, and a policy space $\Pi = \{\pi : \mathcal{X} \to \mathcal{A}\}$. At each round t, a context $x_t \in \mathcal{X}$ is drawn[1] i.i.d. from some distribution, the learner observes this context, picks a policy $\pi_t \in \Pi$, thereby playing action $a_t = \pi_t(x_t) \in \mathcal{A}$, and receives an associated *reward* $r_t \in [0, 1]$ drawn from some fixed distribution that may depend on the current action and context. The expected reward of the optimal policy at the context x_t at round t will be denoted by $\mu_t^\star = \max_{\pi' \in \Pi} \mathbb{E}[r|\pi'(x_t), x_t]$and, when contexts are stochastic, the expectation of $\mu_t^\star$ over contexts

[1] This assumption will actually be relaxed in Section 6.

simply as $\mu^\star = \mathbb{E}_x\left[\mu^\star_t\right]$ which is a fixed quantity and independent of the round t.

Base learners. Our learning policy in fact relies on base learners which are in turn learning algorithms operating in the same problem $\langle \mathcal{X}, \mathcal{A}, \Pi \rangle$. Specifically, there are M base learners which we index by $i \in [M] = \{1, \ldots, M\}$.[2] In each round t, we select one of the base learners to play, and receive the reward associated with the action played by the policy deployed by that base learner in that round. Let us denote by $T_i(t) \subseteq \mathbb{N}$ the set of rounds in which learner i was selected up to time $t \in \mathbb{N}$. Then the pseudo-regret Reg_i our algorithm incurs over rounds $k \in T_i(t)$ due to the selection of base learner i is

$$\mathsf{Reg}_i(t) = \sum_{k \in T_i(t)} \left(\mu^\star_k - \mathbb{E}[r_k | \pi_k(x_k), x_k]\right) ,$$

and the total pseudo-regret Reg of our algorithm is then $\mathsf{Reg}(t) = \sum_{i=1}^M \mathsf{Reg}_i(t)$. Similarly, we denote the total reward accumulated by base learner i after a total of t rounds as $U_i(t) = \sum_{k \in T_i(t)} r_k$.

Putative regret bounds. Each base learner i comes with a *putative* (or *candidate*) regret (upper) bound $R_i \colon \mathbb{N} \to \mathbb{R}_+$, which is a function of the number of rounds this base learner has been played. This bound is typically known a-priori to us, and can also be a random variable, as long as its current value is observable, that is, we assume $R_i(n_i(t))$ is observable for all $i \in [M]$ and $t \in \mathbb{N}$, where $n_i(t) = |T_i(t)|$ is the number of rounds learner i was played after t total rounds. Our notion of regret bound measures regret comparing to $\mu^\star_t$, the best expected reward overall.

Well- and misspecified learners. We call learner i *well-specified* if $\mathsf{Reg}_i(t) \leq R_i(n_i(t))$ for all $t \in [T]$, with high probability over the involved random variables (see later sections for more details and examples), and otherwise *misspecified* (or *bad*). A well-specified base learner i is then one for which the candidate regret bound $R_i(\cdot)$ is a reliable upper bound on the actual regret of that learner.

For a given set of base learners with candidate regret upper bounds, we denote the set of bad learners by $\mathcal{B} \subseteq [M]$, and the set of good (well-specified) ones by

$$\mathcal{G} = \{i \in [M] \colon \forall t \in [T]\ \mathsf{Reg}_i(t) \leq R_i(n_i(t))\} = [M] \setminus \mathcal{B}.$$

Notice that sets $\mathcal{G}$ and $\mathcal{B}$ are random sets. As a matter of fact, these sets do also depend on the time horizon T, but we leave this implicit in our notation. We assume in our regret-analysis that there is always a well-specified learner, that is $\mathcal{G} \neq \varnothing$. We will show that in the applications we consider, this happens with high probability. The index $i_\star \in \mathcal{G}$ (or just $\star$ for short) will be used for any well-specified learner.

Problem statement. Our goal is to devise sequential decision making algorithms that have access to base learners as subroutines and associated candidate regret bounds $R_i(\cdot)$, and are guaranteed to have regret that is comparable to the smallest regret bound among all well-specified base learners in $\mathcal{G}$, without knowing a-priori $\mathcal{G}$ and $\mathcal{B}$.

3. Dynamic Balancing

In this section, we describe the intuition behind our algorithm. There are two main conceptual components: a *balancing* scheme, and a *de-activation* scheme. For the balancing part, in each round, the algorithm chooses a base learner with minimum value of $R_i(n_i(t))$. To see why this is a good idea, assume for now that all base learners are well-specified. Then, because the regret of each base learner is at most its candidate regret bound, and these regret bounds are approximately equal, the total regret our algorithm incurs is at most M times worse than had we only played the algorithm with the best putative regret bound:

$$\begin{aligned}\mathsf{Reg}(T) &= \sum_{i=1}^M \mathsf{Reg}_i(T) \leq \sum_{i=1}^M R_i(n_i(T)) \\ &\approx M \min_{i \in [M]} R_i(n_i(T)) \leq M \min_{i \in [M]} R_i(T) .\end{aligned}$$

Yet, the above only works if all base learners are well specified, which may not be the case. Besides, if all base learners were well specified, we could simply single out at the beginning the learner whose regret bound is lowest at time T, and select that learner from beginning to end. In order to handle the situation where some putative regret bounds may not hold, we pair the above regret bound balancing principle with a test to identify misspecified base learners. This test compares the time-average rewards $U_i(t)/n_i(t)$ and $U_j(t)/n_j(t)$ achieved by two base learners i and j. Up to martingale concentration terms, a well-specified learner should satisfy $\frac{U_i + R_i(n_i(t))}{n_i(t)} \geq \mu_\star$, because

$$\mathsf{Reg}_i(n_i(t)) = n_i(t)\mu_\star - \mathbb{E}[U_i(t)] \leq R_i(n_i(t)) ,$$

while for all learners, we have $\mathbb{E}[U_i(t)/n_i(t)] \leq \mu_\star$. Therefore, we should be "suspicious" of any learner for which $\frac{U_i(t) + R_i(n_i(t))}{n_i(t)} \lesssim \max_{j \in [M]} \frac{U_j(t)}{n_j(t)}$.

Our approach is to (temporarily) de-activate learners that do not satisfy this "misspecification test", and only balance the regret bounds among the currently active learners.

However, in order to obtain more refined bounds, we will add two twists to this procedure. First, we introduce some *bias* functions $b_i(t)$. The $b_i(t)$ represents a "budget" of

[2] A learner may choose to internally work on a smaller policy class / only use a subset of the context.

Algorithm 1: The Dynamic Balancing Algorithm

```
   input : $M$ base learners
           Candidate regret bound $R_i$ for each learner
           Confidence parameter $\delta \in (0,1)$
           Reward bias $b_i(\cdot)$ and scaling coefficient $v_i$
1  $U_i(0) = n_i(0) = 0$ for all $i \in [M]$
2  Active set: $\mathcal{I}_1 \leftarrow [M]$
3  for round $t = 1, 2, \ldots$ do
4      Select learner from active set as
           $i_t \in \operatorname{argmin}_{i \in \mathcal{I}_t} v_i R_i(n_i(t-1))$
5      Play action $a_t$ of learner $i_t$ and receive reward $r_t$
6      Update learner $i_t$ with $r_t$
7      Update $n_i(\cdot)$ and $U_i(\cdot)$ :
           $U_{i_t}(t) \leftarrow U_{i_t}(t-1) + r_t$
           $n_{i_t}(t) \leftarrow n_{i_t}(t-1) + 1$
8      foreach learner $i \in [M]$ do
9          Compute adjusted avg. reward:
               $\eta_i(t) \leftarrow \frac{U_i(t)}{n_i(t)} - b_i(t)$
10         Compute confidence band:
               $\gamma_i(t) \leftarrow c\sqrt{\frac{\ln(M \ln n_i(t)/\delta)}{n_i(t)}}$
11     Set active learners $\mathcal{I}_{t+1}$ as all $i \in [M]$ that satisfy
           $\eta_i(t) + \gamma_i(t) + \frac{R_i(n_i(t))}{n_i(t)} \geq \max_{j \in [M]} \eta_j(t) + \gamma_j(t)$
```

extra regret over learner i's regret that we are willing to experience in the event that learner i is indeed the optimal well-specified learner. Intuitively, a learner with a very large putative regret bound can be safely ignored during early iterations, perhaps at the cost of a constant factor more regret. Thus, we will give such learners larger values for $b_i(t)$, and utilize the following misspecification test:

$$\frac{U_i(t) + R_i(n_i(t))}{n_i(t)} - b_i(t) \geq \max_{j \in [M]} \frac{U_j(t)}{n_j(t)} - b_j(t)\,. \quad (1)$$

This test forces the algorithm to de-activate learners for whom the regret bound is not higher than the bias. The formal test is provided in line 11 of Algorithm 1, including an extra term arising from martingale concentration bounds.

The second twist is the use of *scaling coefficients* v_i. Specifically, instead of playing the active learner with minimum regret bound, we instead play the active learner with minimum *scaled* regret bound $v_i R_i(n_i(t))$. This will cause the values for $v_i R_i(n_i(t))$ to remain roughly balanced among all active learners. By decreasing v_i, we play algorithm i more frequently. Together, these twists allow us to improve dependencies on M in the regret.

4. Regret Guarantees for Dynamic Balancing

We now give general regret bound guarantees for Algorithm 1 that hold in the presence of a well-specified learner in the pool of base learners. We separate between *gap-independent* (or *worst case*) guarantees, which do not depend on how much the misspecified learners violate their candidate regret bounds ("gap" of the learner) from the *gap-dependent* guarantees where, when misspecified, the base learners exceed their candidate bounds by a significant amount. We emphasize that the same algorithm (with the same parameter settings) *simultaneously* obtains both the worst-case and gap-dependent guarantees, reminiscent of a best-of-both-worlds guarantee. For simplicity of presentation, we significantly abbreviate results here. The full bound in can be found in the supplementary material (Corollary 23 for the gap independent result, Theorem 30 for the gap-dependent result).

Theorem 1 (General Regret Bound)**.** *Let $R_i(n) = Cd_i n^\beta$, where $1 = d_1 \leq \ldots \leq d_M$, $\beta \in (0,1)$, and C is some positive constant independent of n and i. Let $b_i(t) = \max\left[2Cd_i^{\frac{1}{\beta}} \cdot i^{1-\beta} \cdot t^{\beta-1},\ \frac{c\sqrt{M \ln \frac{M \ln t}{\delta}}}{\sqrt{t}}\right]$ and $v_i = \frac{i^{\beta^2}}{d_i}$ denote the bias and scaling coefficients for each learner, where c is an absolute constant.*[3] *Then the regret $\mathsf{Reg}(T)$ of Algorithm 1 is bounded with probability at least $1-\delta$ for all rounds $T \in \mathbb{N}$ as*

$$\tilde{O}\left(\sqrt{MT} + (B^{1-\beta} d_\star^{1/\beta} + d_\star + M^{1-\beta})\, CT^\beta\right)\,.$$

Further, if we assume that all misspecified learners $i \in \mathcal{B}$ have regret $\mathsf{Reg}_i(t)$ bounded from below as $\mathsf{Reg}_i(t) \geq \Delta_i n_i(t)$, for some constants $\Delta_i > 0$. Then the regret $\mathsf{Reg}(T)$ of Algorithm 1 is bounded with probability at least $1-\delta$ for all rounds $T \in \mathbb{N}$ as

$$\tilde{O}\left(MCd_\star T^\beta + \sqrt{MT} + \sum_{i \in \mathcal{B}} \frac{B\, C^{\frac{1}{1-\beta}}\, d_\star^{\frac{1}{\beta-\beta^2}}}{\Delta_i^{\frac{\beta}{1-\beta}}} + \frac{1}{\Delta_i}\right).$$

Here $\star \in \mathcal{G}$ is the smallest well-specified learner, $B = |\mathcal{B}|$ is the number of misspecified learners and the $\tilde{O}$ notation hides any log factors.

Theorem 1 shows that for any $\beta \geq 1/2$, our worst-case regret bound recovers the optimal T^β rate. On the other hand, when $\beta < 1/2$, our bound scales sub-optimally as $\sqrt{T}$. This is not surprising since the lower bound by Pacchiano et al. (2020b) indicates a $\Omega(\sqrt{T})$ barrier for model-selection based on observed rewards without additional assumptions. The second part of the theorem yields gap-dependent guarantees, where the term *gap* refers to a property of the base learners, rather than the underlying action space. Comparing the two regret guarantees, we see that in the latter the

[3] Constant c is a known constant stemming from our analysis – see the appendix for details.

multiplicative factor in front of the best well-specified regret bound is only $O(M)$, as compared to the presence of extra $d_\star$ factors without a gap-assumption. Further, while the extra additive term in the gap-dependent bound may have a dependency on a potentially large d_i, this term only scales with T as $\ln\ln T$ (see the precise statement in the appendix), and is thus virtually constant.

Importantly, through a slightly different analysis of the exact same algorithm we can also achieve the optimal scaling in T even when $\beta < \frac{1}{2}$ so that the additional $\sqrt{T}$-term occurring in the corresponding gap-independent bound can be avoided (see Theorem 31). This result is in contrast with existing approaches such as Pacchiano et al. (2020b), where the adoption of an adversarial aggregation algorithm makes the $\sqrt{T}$ dependence inevitable.

Remark 1. *It is worth emphasizing that the statement of Theorem 1 (as well as the statement of Corollary 2 below), does not imply extra parameters to tune in Algorithm 1: The choice of $b_i(t)$ and v_i is solely dictated by the shape of the putative regret bounds $R_i(n)$. For instance, in a standard contextual bandit scenario like the one considered in Section 5.1, $R_i(n) = Cd_i\sqrt{n}$, where $d_1 < \ldots < d_M$ are known input dimensionalities each base learner operates with. The regret upper bounds $R_i(n)$ are those provided by a LinUCB-like analysis, so that also constant C is known in advance. Thus, once the putative bounds $R_i(\cdot)$ are known, there are no constants whatsoever to tune in Algorithm 1, Theorem 1 or Corollary 2 below.*

For the typical case of $\beta = 1/2$, the following corollary shows that we obtain the regret bound of the best well-specified learner up to a multiplicative factor of only $\sqrt{B}d_\star$.

Corollary 2. *Suppose the candidate regret bounds are given by $R_i(n) = Cd_i\sqrt{n}$ where $1 = d_1 \leq \ldots \leq d_M$ and C is some positive constant independent of n and i. Let $b_i(t) = \frac{1}{\sqrt{t}}\max\left\{2Cd_i^2\sqrt{i}\,,\, c\sqrt{M\ln\frac{M\ln t}{\delta}}\right\}$ and $v_i = \frac{i^{1/4}}{d_i}$ denote the bias and scaling coefficients for each learner, where c is an absolute constant. Then with probability at least $1-\delta$, the regret of Algorithm 1 is bounded as follows*

$$\mathrm{Reg}(T) = \tilde{O}\left(\sqrt{MT} + (\sqrt{B}d_\star^2 + d_\star + \sqrt{M})\,C\sqrt{T}\right)\,.$$

If we further assume that all misspecified learners $i \in \mathcal{B}$ have regret $\mathrm{Reg}_i(t) \geq \Delta_i n_i(t)$, for some constants $\Delta_i > 0$, then with probability at least $1-\delta$, the regret of Algorithm 1 is bounded as follows

$$\mathrm{Reg}(T) = \tilde{O}\left(MCd_\star\sqrt{T} + \sqrt{MT} + B\,C^2 d_\star^4 \sum_{i\in\mathcal{B}} \frac{1}{\Delta_i}\right)\,.$$

Comparison to prior results. The CORRAL algorithms of Agarwal et al. (2017); Pacchiano et al. (2020b) each involve a learning rate parameter η. In general, they obtain regret:

$$\sqrt{MT} + \frac{M}{\eta} + T\eta + T(Cd_\star)^{\frac{1}{\beta}}\eta^{\frac{1-\beta}{\beta}} \qquad (2)$$

after which various values of η are deployed to obtain useful bounds. With an appropriate setting for $b_i(t)$ and v_i depending on η, we recover exactly this bound (Corollary 24), so our algorithm is at least as powerful as these prior works.

In Lee et al. (2021) the authors consider an episodic MDP setting with a nested sequence of policy classes, where each base learner operates on one class. Each base learner is assumed to be well-specified w.r.t. its own policy class (that is, it satisfies its candidate regret bound R_i w.r.t. to the policy class it operates on). In our notation, their regret bound reads as

$$Cd_\star\sqrt{T} + M\sqrt{T} + M^2 d_\star^4 \sum_{i<\star} \frac{1}{\Delta_i^3}\,, \qquad (3)$$

whereas ours from Corollary 2 is of the form

$$Cd_\star\sqrt{T}M + \sqrt{MT} + M\,d_\star^4 \sum_{i<\star} \frac{1}{\Delta_i}\,. \qquad (4)$$

The two bounds are incomparable for a number of reasons: (i) Eq. (3) has constant one in front of the regret $Cd_\star\sqrt{T}$ of the best learner, while (4) has constant M; on the other hand, the dependence on the gaps in (3) is far worse than the one in (4); (ii) Eq. (3) only holds under the restricting assumption of well-specification for all base learners w.r.t. their policy class, and only applies to nested policy classes, which is not the case for Eq. (4).

In Arora et al. (2021), the authors study the special case of a K-armed bandits problem where each base learner is an instance of UCB restricted to some subset S_i of the arms. Assuming only one algorithm is in command of the optimal arm (i.e. the existence of only one well-specified algorithm), they study two algorithms that achieve *logarithmic* expected regret bounds of the form $\log(T)\mathbb{E}[\mathrm{Reg}_\star(T)] + O\left(\sum_{i\in\mathcal{B}} \frac{|S_i|\log(T)^2}{\Delta_i}\right)$ and $\mathbb{E}[\mathrm{Reg}_\star(T)]+O\left(\sum_{i\in\mathcal{B}} \frac{|S_i|\log(T)^5}{\Delta_i}\right)$ respectively. $\mathrm{Reg}_\star(T)$ is the actual regret rather than the regret bound $R_\star(T)$. Thus, since UCB obtains logarithmic regret, the overall regret is logarithmic. From Theorem 1, such logarithmic bounds appear out of reach due to the $O(\sqrt{T})$ term. Fortunately, a more refined analysis of Dynamic Balancing is possible in this case. Using the *worst-case* bounds $R_i(n) = \tilde{O}(\sqrt{|S_i|n})$, by Theorem 31 in the appendix, Algorithm 1 has a high-probability bound

$$\mathrm{Reg}(T) \leq \mathrm{Reg}_\star(T) + \tilde{O}\left(\sum_{i\in\mathcal{B}} \frac{M|S_\star|^2}{\Delta_i}\right)\,.$$

Thus, we can recover logarithmic regret bounds in this setting as well. That said, our bound is generally incomparable to the two bounds of Arora et al. (2021). While the M and $|S_i|$ factors multiplied by $\frac{1}{\Delta_i}$ in our result are worse, we avoid the $\log(T)$ scaling on $\mathrm{Reg}_\star(T)$ of their first bound and have a better $\log(T)$ dependence of at most $\log(T)^4$ in the $\frac{1}{\Delta_i}$ terms compared $\log(T)^5$ in their second bound.

Trading-off regret guarantees. Theorem 1 provides results for a particularly attractive setting for the parameters $b_i(t)$ and v_i, but we could use different settings to achieve other tradeoffs between the base learners. For example, suppose we have a "guess" that some learner j will be perform best. In this case, we would like to decrease v_j and $b_j(\cdot)$ to encourage Algorithm 1 to choose learner j more often, thus reducing the model-selection overhead when our guess is correct. In particular, with $v_j = 1/d_j^{3/2}$ and $b_i(t) = \max(\sqrt{M}, d_j)/\sqrt{t}$, Corollary 26 shows that we can obtain the bound:

$$\mathrm{Reg}(T) \le (\mathbf{1}\{\star \neq j\}\sqrt{B}d_\star^2 + d_j + d_\star + \sqrt{M})C\sqrt{T}\,.$$

So that we no longer suffer the $d_\star^2$ term if $\star = j$, but in payment we must always suffer a d_j term in the regret. This might be particularly useful if many of the base learners in fact have the *same* candidate regret bound, and the difficulty lies in detecting the misspecified learners. In fact, even more subtle tradeoffs are possible. In Corollary 25, we show that we are able to recover the Pareto frontier of regret bounds for multi-armed bandit by setting v_i and $b_i(t)$ appropriately.

A number of consequences of Theorem 1 and Corollary 2 applied to linear bandits and MDPs are spelled out in the next section. Further results are contained in the appendix.

5. Applications

5.1. Linear Bandits with Nested Model Classes

An important area of application of our Dynamic Balancing approach are contextual linear bandits. In this setting, the context x_t determines the set of actions $\mathcal{A}_t \subseteq \mathcal{A}$ that can be played at time t and the policies we consider are of the form $\pi_\theta(x_t) = \arg\max_{a\in\mathcal{A}_t}\langle a, \theta\rangle$, for some $\theta \in \mathbb{R}^d$. The class of policies Π can thus be identified with a class d-dimensional vectors: $\Pi \subseteq \mathbb{R}^d$. Moreover, rewards are generated according to a noisy linear function: $r_t = \langle a_t, \theta_\star\rangle + \xi_t$, where $\theta_\star \in \Pi$ is unknown, and ξ_t is a conditionally zero mean $\sigma-$subgaussian random variable. We denote the optimal action at time t as $a_t^\star = \mathrm{argmax}_{a\in\mathcal{A}_t}\langle a, \theta_\star\rangle$. The learner's objective is to control its pseudo-regret $\mathrm{Reg}(T) = \sum_{t=1}^T \langle a_t^\star, \theta_\star\rangle - \langle a_t, \theta_\star\rangle$. When the dimensionality $d_\star$ is known a-priori, the OFUL (Abbasi-Yadkori et al., 2011) algorithm achieves regret $O(d_\star\sqrt{T})$ (we provide a brief review of this algorithm and the precise regret bound in Appendix D.1).

We can apply our Dynamic Balancing approach to contextual linear bandits where the true dimensionality $d_\star$ of the model $\theta_\star$ is unknown a-priori. In this standard scenario, considered by many recent papers in the model selection literature for bandit algorithms (e.g. Foster et al., 2019; Pacchiano et al., 2020b), the learner chooses among actions $\mathcal{A}_t \subseteq \mathbb{R}^{d_{\max}}$ of dimension $d_{\max}$ but only the first $d_\star$ dimensions are relevant (that is, $(\theta_\star)_i = 0$ for $i > d_\star$).

In Appendix D.2, we show that a variant of the OFUL algorithm of dimensionality d can be combined with a misspecification test to obtain a regret bound that satisfies one of two possibilities - either (i) the regret is bounded by $\mathrm{Reg}(t) \le O(d\sqrt{t})$, or (ii) the algorithm suffers linear regret $\mathrm{Reg}(t) \ge \Delta t$ for some constant Δ for sufficiently large t. Further, whenever the algorithm is well-specified, the former regret upper bound applies. Equipped with such a suitably modified OFUL algorithm, one can perform model selection in this setting as follows: We use $\log_2 d_{\max}$ instances of modified OFUL (Appendix D.2) as base learners[4]. Each instance i first truncates the actions to dimension $d_i = 2^i$ and only then applies the OFUL update. Based on the OFUL regret guarantees (as described in Appendix D.1 and D.2), we use $R_i(n) = d_i C\sqrt{\ln(n)n}$, with suitable constant C. Although our previously discussed results technically do not cover log factors, it is relatively straightforward to modify the arguments to obtain the same bound multiplied by a log factor (e.g. see Theorem 33 in Appendix for the formal analog of Theorem 30 with log factors in the regret bounds).

By the regret guarantee of OFUL, with probability at least $1 - M\delta$, any base learner i such that $d_i \ge d_\star$ will be well specified, while the others may be misspecified. That is, we have $M = O(\ln d_{\max})$ total base learners, out of which at most $B = O(\ln d_\star)$ are misspecified. Let us consider the case that each of the misspecified learners experiences some gap Δ, which is an intuitively reasonable situation to occur. Then applying Corollary 2 (extracting the $\log(T)$ factors from the more detailed result in Theorem 33) yields

$$\begin{aligned}&\mathrm{Reg}(T)\\&= \tilde{O}\left(\ln(d_{\max})d_\star C\sqrt{T} + \ln(d_\star)\frac{C^2 d_\star^4}{\Delta}\right)\\&\approx O\left(\ln(d_{\max})d_\star\sqrt{T}\ln T + \frac{d_\star^4 \ln d_\star}{\Delta}(\ln T)^2\sqrt{\ln\ln T}\right).\end{aligned}$$

This bound is appealing because the dominant T-dependent term is $\tilde{O}(d_\star\sqrt{T})$, which matches the best base learner. However, this bound only holds under our gap assumption, so we would like to have a more generic result. To accomplish this, we will deploy our modified OFUL algorithm

[4] Here we assume $d_\star$ and $d_{\max}$ are powers of 2 for convenience but our results also hold generally up to a constant factor of 2.

with misspecification test (D.2), which is constructed so as to guarantee that all misspecified learners suffer asymptotically linear regret: for large enough t, $\frac{\mathsf{Reg}_i(t)}{n_i(t)} \geq \Delta_i$ for some fixed $\Delta_i > 0$. In other words, this algorithm essentially ensures that a gap will exist, although it makes no guarantees about how big that gap will be. Employing this modified OFUL as our base algorithms, for large enough T we have $\mathsf{Reg}(T) = O(\ln(d_{\max}) d_\star C \sqrt{T} \ln(T))$, so that *asymptotically* we are able to recover the desired model-selection guarantee. Note that this need *not* be the ideal model selection bound because we do not have any bounds on the Δ_i: for any fixed finite T there may be a linear bandit instance for which some Δ_i is $O(1/T)$, so that our gap-based regret bound is vacuous. However, since Δ_i is solely a function of the problem instance, which is fixed at the first time step, we obtain the desired result asymptotically by allowing the time horizon to grow without bound.

A standard goal in model selection is to obtain sub-linear regret bounds even in the case where the model complexity of the target class is allowed to grow sub-linearly with T, e.g., as bound of the form $\sqrt{d_\star T}$ (Foster et al., 2020b). However, such goals are stated for finite action spaces. We are dealing with *infinite* action spaces, and the best one can hope for in this case is indeed $d_\star \sqrt{T}$ (see e.g. Rusmevichientong & Tsitsiklis, 2010, Section 2).

Comparison to prior results. Ghosh et al. (2021) also recover the optimal model selection guarantee up to additive terms but require a specialized algorithm for this setting. In their case, the additive terms depend on the magnitude of the smallest non-zero component of $\theta_\star$. Their approach combines OFUL with phases of uniform exploration which (implicitly) requires the action-set to be well-conditioned (e.g. by assuming the action space to be the unit sphere). Similarly, the specialized MODCB approach by Foster et al. (2019) also interleaves phases of uniform exploration with a bandit algorithm, EXP-IX in their case of finite action spaces. They make the conditioning of the action space explicit in their guarantees through a dependence on the smallest eigenvalue of the feature covariance matrix. While we avoid such a dependence, our bounds are generally incomparable due to the dependence on gaps.

5.2. Confidence Parameter Tuning in OFUL

A common problem that arises in the practical deployment of contextual bandit algorithms like OFUL is that they are extremely sensitive to the tuning of their upper-confidence parameter that rules the actual trade-off between exploration and exploitation. The choice of confidence parameters suggested by theory (see e.g. Lemma 34 in the appendix) is often too conservative in practice. This is due to approximations in the derivation of such bounds but may also be the case when the actual noise variance is smaller than the assumed σ^2 variance. While there are concentration results (empirical Bernstein bounds) that can adapt to favorable low-variance noise for scalar parameters (e.g., in unstructured multi-armed bandits), such adaptive bounds are still unavailable for least-squares estimators. Scaling down the confidence radii $\beta_1, \dots, \beta_T$ used in OFUL by a factor $\kappa < 1$ can often achieve significantly better empirical performance but comes at the cost of losing any theoretical guarantee. Our model-selection framework can be used to tune the confidence parameter online and simultaneously achieve a regret guarantee. Specifically, we look at ways to compete against the instance of the OFUL algorithm which is equipped with the optimal scaling of its upper-confidence value, in the sense of the following definition:

Definition 3. *Denote by $\bar{\beta}_t$ the standard confidence-parameter choice (see Appendix D.1) and let $\kappa \in \mathbb{R}_+$ be a scaling factor. Further, let $\hat{\theta}_S(\kappa)$ and $\Sigma_S(\kappa)$ be the iterates of least squares estimator and covariance matrix obtained by running* OFUL *with scaled confidence parameters $(\kappa \bar{\beta}_t)_{t \in \mathbb{N}}$ on a subset of rounds $S \subseteq [T]$. Then, for a given range $[\kappa_{\min}, 1]$, the optimal confidence parameter scaling for* OFUL *is defined as*

$$\kappa_\star = \min_{\kappa \in [\kappa_{\min}, 1]} \max_{S \subseteq [T]} \frac{\|\hat{\theta}_S(\kappa) - \theta_\star\|_{\Sigma_S(\kappa)^{-1}}}{\bar{\beta}_{|S|}} .$$

In words, the optimal $\kappa_\star$ is the smallest scaling factor of confidence parameters that ensures that no matter to what subset of rounds we would apply OFUL to, the optimal parameter $\theta_\star$ is always contained in the confidence ellipsoid. Observe that $\kappa_\star$ is a random quantity, i.e., $\kappa_\star$ is the best scaling factor for the given realizations in hindsight. While $\mathbb{P}(\kappa_\star \leq 1) \geq 1 - \delta$, empirical observations suggest that $\kappa_\star$ is much smaller in many events and bandit instances.

OFUL with confidence parameters $\kappa \bar{\beta}_t$ admits a regret bound of the form[5] $\mathsf{Reg}(n) \lesssim \kappa d \sqrt{n} \ln(n)$ if $\kappa \geq \kappa_\star$ (see Appendix D.1). Since $\kappa_\star$ is unknown, we run Algorithm 1 with $M = (1 + \log_2 \frac{1}{\kappa_{\min}})$ instances of OFUL as base learners, each with a scaling factor $\kappa_i = 2^{1-i}$, $i = 1, \dots, M$, and putative regret bound $R_i(n) \approx \kappa_i d \ln(T) \sqrt{n}$. Then, by Corollary 2 (with $C = d \ln(T) \kappa_{\min}$ and $d_i = \frac{\kappa_i}{\kappa_{\min}} \geq 1$), with probability at least $1 - \delta$ we have:

$$\begin{aligned} &\mathsf{Reg}(T) \\ &\lesssim \left(\sqrt{M} \kappa_{\min} + \kappa_\star + \sqrt{B} \frac{\kappa_\star^2}{\kappa_{\min}} \right) d \ln(T) \sqrt{T} \\ &= \tilde{O}\left(\left(\frac{\kappa_{\min}}{\kappa_\star} \sqrt{\ln \frac{1}{\kappa_{\min}}} + \frac{\kappa_\star}{\kappa_{\min}} \sqrt{\ln \frac{\kappa_\star}{\kappa_{\min}}} \right) \kappa_\star d \sqrt{T} \right) . \end{aligned}$$

Note that this is a problem-dependent bound because it depends on $\kappa_\star$. In cases where $\kappa_\star \lesssim \frac{\sqrt{\kappa_{\min}}}{\ln(1/\kappa_{\min})^{1/4}}$, this bound

[5] For simplicity of presentation, we set here $\lambda = 1$ and disregarded the dependence on other parameters like L, S, and σ.

strictly improves on the standard OFUL bound relying on $\kappa = 1$, which is often way too conservative in practice. In Section 7, we empirically demonstrate the performance of our model selection approach in this setting.

5.3. Further Applications

Our results can also be specialized to other applications in bandits and reinforcement learning. These include:

- Reinforcement learning in linear MDPs with nested model classes (see Appendix D.3);
- ϵ-approximate linear bandits with unknown approximation error ϵ (see Appendix D.4).

6. Adversarial Contexts for Linear Bandits

In this section, we show that the dynamic regret balancing principle can also be used for model selection in linear stochastic bandits when the contexts x_t are generated in an adversarial manner. Our technique can be easily adapted to the various applications discussed in Section 5 but, for the sake of concreteness, we present our extension for the setting of nested linear models described in Section 5.1. We lift the assumption that contexts are drawn i.i.d., and consider instead the one where contexts x_t (corresponding to the action set $\mathcal{A}_t$ at round t) are generated adversarially.

Algorithm 1, which assumes stochastic contexts, compares the sum of rewards from learners that were executed on two disjoint subsets of rounds to determine misspecification. This strategy no longer works with adversarial contexts, since the optimal rewards that an algorithm could have achieved depends on the contexts in the rounds that the algorithm was played. To address this challenge, we modify the basic form of Algorithm 1 in two ways: (1) we randomize the learner's choice for regret balancing, and (2) we change the activation condition to compare upper and lower confidence bounds on the optimal policy value of *all* rounds played so far. The resulting algorithm (for details and pseudocode see Appendix E) operates in epochs.

In each epoch, there is a set of active learners $\mathcal{I}$ whose candidate regret bounds are balanced via a randomized procedure. Specifically, in each round of the epoch a learner is picked from $\mathcal{I}$ by sampling an index $i_t \sim \text{Categorical}(p)$ from a categorical distribution with probabilities

$$p_i = \frac{1/z_i}{\sum_{j \in \mathcal{I}} 1/z_j}\,, \quad \text{where} \quad z_i \approx d_i^2\,. \tag{5}$$

An epoch ends whenever the algorithm detects that there is a misspecified learner in the active set $\mathcal{I}$. This happens when the following condition is triggered:

$$\sum_{i \in \mathcal{I}} [U_i(t) + R_i(n_i(t))] + c\sqrt{t \ln \frac{\ln(t)}{\delta}} < \max_{i \in \mathcal{I}} \sum_{k=1}^{t} B_{k,i}\,,$$

where $B_{k,i}$ is a lower-confidence bound from learner i on the expected reward of the action it would have played in round k had it been selected. This test bears some similarity with the test in (1) for the stochastic case but instead of comparing the rewards and regret bound of a single algorithm to the rewards of another algorithm, we here compare the sum of rewards and regret bounds across all active learners with the lower-confidence bounds on optimal reward obtained from each learner. For details, see Appendix E. We prove the following regret guarantee:

Theorem 4. *Assume that Algorithm 5 in Appendix E is run with M instances of OFUL as base learners that use different dimensions d_i and norm bounds L_i, S_i with $2z_i \leq z_{i+1}$ (see Eq. (5)). Then, with probability at least $1 - \delta$, the regret is bounded for all rounds T as*

$$\mathsf{Reg}(T) = \tilde{O}\left(\left(d_\star + \sqrt{d_\star} S_\star + M\right)\sqrt{B} R_\star(T)\right),$$

where $\star$ is the index of the smallest well-specified base learner and $R_\star(T) \approx d_\star\sqrt{T}$ is its regret bound.

7. Experiments

To investigate the practical usefulness of our Dynamic Balancing approach and compare it against existing methods, we conducted experiments on synthetic linear bandit instances with 100 actions of dimension 10 each. We use the application described in Section 5.2 and optimize over exploration-exploitation trade-off. Specifically, we use 10 instances of OFUL as base learners with confidence scaling parameters on a geometric grid in $[\frac{1}{100}, 1]$, where a scaling of 1 makes OFUL explore more compared to scaling $\frac{1}{100}$.

To test the versatility of our model selection approach, we evaluate it on three bandit instances with reward noise of standard deviation $\sigma = 1$, $\sigma = 0.3$ and $\sigma = 0.05$ each. We found that when running each base learner individually, the best confidence scaling is $\kappa \approx 0.36$, $\kappa \approx 0.13$ and $\kappa \approx 0.07$, respectively, which are all significantly smaller than $\kappa = \sigma$ required by the OFUL theoretical regret analysis.

We compare three model-selection algorithms: Corral (Agarwal et al., 2017), Stochastic Corral (Pacchiano et al., 2020b) and a basic version of Algorithm 1 without biases and scaling factors ($b_i(t) = 0$, $v_i = 1$). This version of dynamic balancing is discussed and analyzed in detail in Appendix B. For putative regret bounds of the form $R_i(n) = Cd_i\sqrt{n}$ (as in Corollary 2), this version achieves a regret of

$$\mathsf{Reg}(T) = \tilde{O}((M + d_\star\sqrt{B})Cd_\star\sqrt{T})$$

as well as gap-dependent regret bounds comparable to Corollary 2. Thus, this basic version recovers the theoretical guarantees of Algorithm 1 up to factors of M and B that are typically small in practice ($\leq$ 10 here). When this is

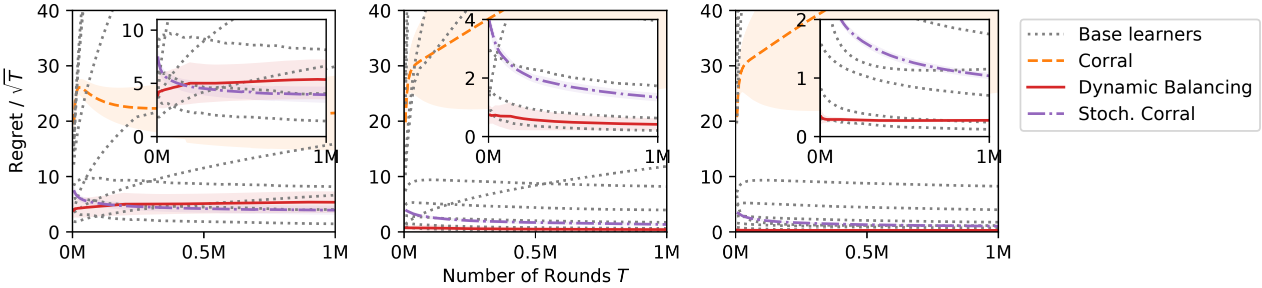

Figure 1. Empirical comparison of Dynamic Balancing: We select among 10 OFUL instances (dotted grey) with different confidence parameters in $[\frac{1}{100}, 1]$ on 3 linear bandit problems; Left: reward noise $\sigma = 1$, middle: $\sigma = 0.3$, right: $\sigma = 0.05$. Results are averages of 10 independent runs with shaded areas representing 95% confidence bands for model selection algorithms. For details, see Appendix F.

the case, we expect the basic algorithm with $b_i(t) = 0$ and $v_i = 1$ to perform empirically better, due to fewer conservative overestimates in the computation of $b_i(\cdot)$, which is why we chose it for our experiments.

Both Corral and Stochastic Corral require a learning rate which we set to $\frac{\xi}{\sqrt{T}}$ where ξ was picked as the value from $\{1, 10, 100, 1000\}$ that was most competitive for each algorithm across the 3 problem instances. Note that a more extensive learning rate optimization would defeat the purpose of model-selection in practice: we did not perform any parameter tuning at all for Dynamic Balancing. As putative regret bound, we used the sum of confidence widths in OFUL of all past rounds where the base learner was played. This has the same $\sqrt{T}$ rate as the analytical regret bound of OFUL but possibly tighter constants, and is consistent with theory. Finally, for Stochastic Corral and Dynamic Balancing, we updated all base learners with each observation (Corral requires an update with importance-weighted rewards instead). We emphasize that our setup is consistent with the assumptions needed for theoretical guarantees to hold.[6]

Figure 1 shows our experimental results for the three bandit instances. The y-axis is cumulative regret divided by $\sqrt{T}$, so good learners should have a flat line with small offset. Across all instances, Corral performs much worse than the other methods due to the high variance of the importance-sampling estimator used in the updates of the base learners. While the performance of Stochastic Corral and Dynamic Balancing is similar for the large reward noise instance, Dynamic Balancing significantly outperforms Stochastic Corral on the other two instances. Importantly, the regret of Dynamic Balancing is close to the second best base learner, which demonstrates that it can be a highly effective tool for tuning confidence parameters and making algorithms adaptive to the typically unknown noise level in linear bandits.

8. Conclusion

We have presented a simple but powerful dynamic balancing technique for model selection in stochastic bandit and reinforcement learning tasks. Our algorithm's total regret is bounded by the regret of the best base learner times a multiplicative factor. Using our framework, we not only recover the best previously known model selection regret guarantees, but also obtain stronger gap-dependent regret bounds that also apply to base learners with $o(\sqrt{T})$ candidate regret. Our approach can be instantiated for a number of relevant applications ranging from nested model classes in (linear) contextual bandits and MDPs to mis-specified models to hyperparameter tuning of contextual bandit algorithms. The flexibility of our approach is also witnessed by our ability to extend our linear bandit analysis to adversarial contexts.

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pp. 2312–2320, 2011.

Abbasi-Yadkori, Y., Pacchiano, A., and Phan, M. Regret balancing for bandit and RL model selection. *arXiv preprint arXiv:2006.05491*, 2020.

Agarwal, A., Hsu, D., Kale, S., Langford, J., Li, L., and Schapire, R. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pp. 1638–1646, 2014.

Agarwal, A., Luo, H., Neyshabur, B., and Schapire, R. E. Corralling a band of bandit algorithms. In *Conference on Learning Theory*, pp. 12–38. PMLR, 2017.

Arora, R., Marinov, T. V., and Mohri, M. Corralling stochastic bandit algorithms. In *International Conference on Artificial Intelligence and Statistics*, pp. 2116–2124. PMLR, 2021.

[6] For the regret guarantees of this particular version of Dynamic Balancing, see Appendix B.5.

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

Beygelzimer, A., Langford, J., Li, L., Reyzin, L., and Schapire, R. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 19–26, 2011.

Bibaut, A. F., Chambaz, A., and van der Laan, M. J. Rate-adaptive model selection over a collection of black-box contextual bandit algorithms. *arXiv preprint arXiv:2006.03632*, 2020.

Chatterji, N., Muthukumar, V., and Bartlett, P. Osom: A simultaneously optimal algorithm for multi-armed and linear contextual bandits. In *International Conference on Artificial Intelligence and Statistics*, pp. 1844–1854, 2020.

Cutkosky, A., Das, A., and Purohit, M. Upper confidence bounds for combining stochastic bandits. *arXiv preprint arXiv:2012.13115*, 2020.

Foster, D., Gentile, C., Mohri, M., and Zimmert, J. Adapting to misspecification in contextual bandits. In *Advances in Neural Information Processing Systems*, 2020a.

Foster, D. J., Krishnamurthy, A., and Luo, H. Model selection for contextual bandits. In *Advances in Neural Information Processing Systems*, pp. 14741–14752, 2019.

Foster, D. J., Krishnamurthy, A., and Luo, H. Open problem: Model selection for contextual bandits. In *Conference on Learning Theory*, pp. 3842–3846. PMLR, 2020b.

Ghosh, A., Sankararaman, A., and Kannan, R. Problem-complexity adaptive model selection for stochastic linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pp. 1396–1404. PMLR, 2021.

Howard, S. R., Ramdas, A., McAuliffe, J., and Sekhon, J. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2):1055–1080, 2021.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pp. 2137–2143, 2020.

Langford, J. and Zhang, T. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems*, pp. 817–824, 2008.

Lattimore, T. The pareto regret frontier for bandits. *Advances in Neural Information Processing Systems*, 28: 208–216, 2015.

Lattimore, T. and Szepesvári, C. *Bandit algorithms*. Cambridge University Press, 2020.

Lee, J., Pacchiano, A., Muthukumar, V., Kong, W., and Brunskill, E. Online model selection for reinforcement learning with function approximation. In *International Conference on Artificial Intelligence and Statistics*, pp. 3340–3348. PMLR, 2021.

Pacchiano, A., Dann, C., Gentile, C., and Bartlett, P. Regret bound balancing and elimination for model selection in bandits and RL. *arXiv preprint arXiv:2012.13045*, 2020a.

Pacchiano, A., Phan, M., Abbasi Yadkori, Y., Rao, A., Zimmert, J., Lattimore, T., and Szepesvari, C. Model selection in contextual stochastic bandit problems. In *Advances in Neural Information Processing Systems*, volume 33, pp. 10328–10337, 2020b.

Rusmevichientong, P. and Tsitsiklis, J. N. Linearly parameterized bandits. *Mathematics of Operations Research*, 35 (2):395–411, 2010.

Zanette, A., Lazaric, A., Kochenderfer, M., and Brunskill, E. Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning*, pp. 10978–10989. PMLR, 2020.