Distilled Thompson Sampling: Practical and Efficient Thompson Sampling via Imitation Learning

Hongseok Namkoong[1*] Samuel Daulton[*2] Eytan Bakshy[3]

[1]Decision, Risk, and Operations Division, Columbia Business School
[2,3]Facebook Core Data Science

namkoong@gsb.columbia.edu, sdaulton@fb.com, ebakshy@fb.com

Abstract

Thompson sampling (TS) has emerged as a robust technique for contextual bandit problems. However, TS requires posterior inference and optimization for action generation, prohibiting its use in many real-world applications where latency and ease of deployment are of concern. We operationalize TS by proposing a novel imitation-learning-based algorithm that distills a TS policy into an explicit policy representation, allowing fast decision-making and easy deployment in mobile and server-based environments. Using batched data collected under the imitation policy, our algorithm iteratively performs offline updates to the TS policy, and learns a new explicit policy representation to imitate it. Our algorithm guarantees Bayes regret comparable to TS, up to the sum of single-step imitation errors. We show these imitation errors can be made small when unsupervised contexts are cheaply available, which is the case for most large-scale internet applications. Empirically, we demonstrate our imitation policy achieves performance comparable to TS, while allowing more than an order of magnitude reduction in decision-time latency. Buoyed by low latency and simplicity of implementation, our algorithm has been successfully deployed in a video upload system for a leading social networking service, and is reliably handling millions of uploads each day.

1 Introduction

In the past decade, Thompson sampling [77] has emerged as a powerful algorithm for contextual bandit problems. The underlying principle is simple: an action is chosen with probability proportional to it being optimal under the current posterior distribution. Driven by the algorithm's strong empirical performance [70, 21, 55], many authors have recently established rigorous performance guarantees [41, 9, 10, 34, 40, 66, 4]. Thompson sampling is increasingly being applied to a broad range of applications including revenue management [30], internet advertising [35, 7, 69], and recommender systems [42].

Despite its conceptual simplicity and strong performance, Thompson sampling can be difficult to deploy in practice. Concretely, Thompson sampling consists of two steps: *posterior sampling* and *optimization*. *Posterior sampling* requires evaluating a potentially large number of actions from a well-calibrated probabilistic model. Accurately calibrating uncertainty is crucial for optimally trading off exploration and exploitation, and has a high impact on practical performance [62]. Large-scale probabilistic machine learning models based on deep networks show much promise as they can adaptively learn good feature representations for uncertainty calibration [83]. However, sampling from these probabilistic models can be demanding in terms of computation and memory. While approximate inference methods with better runtime characteristics exist, they often produce poorly calibrated uncertainty estimates that lead to poorer empirical performance [62]. The second

[*]Equal contribution.

step, *optimization*, solves for a reward-optimizing action under the posterior sample. This can also be prohibitively expensive when the action space is large or continuous. For example, an advertising platform that matches advertisers to users at each time period has to solve combinatorial optimization problems real-time in order to run Thompson sampling [54].

When deploying Thompson sampling to large-scale internet services, the fact that the aforementioned computation is required *online* poses a substantive challenge as low latency—real-time computational performance—is critical for user satisfaction and retention for typical services. These challenges are especially pronounced in mobile applications, a ubiquitous modality for modern platforms. As of 2018, an estimated 52.2% of worldwide web traffic was generated by mobile devices [73]. Mobile applications require decisions to be made in a fast and memory-efficient manner, and on-device decision-making is critical to good user experience in domains such as adaptive video streaming [53] and social media ranking [59]. However, the majority of internet-connected mobile devices have limited memory, and utilize low-end processors that are orders of magnitude slower than server-grade devices [14, 88]. As affordable, compute-limited mobile devices are increasingly adopted in developing economies [61], the ability to deploy cutting-edge decision algorithms on diverse computing infrastructure is important for democratization of technology and long term business growth.

Software development cost is another core practical consideration when implementing contextual bandit algorithms in large-scale online platforms. The *online* nature of the computations required by Thompson sampling adds a substantial amount of system complexity. Under such complexity, long term software development costs—commonly referred to as tech debt—are often incurred when a suboptimal, myopic development plan is followed in lieu of one that requires (sometimes much) higher initial effort, but less future work [71]. Although avoiding tech debt is crucial to a reliable and scalable service, contextual bandit systems are highly complex since they require temporal feedback loops consisting of different pipelines on exploration, data logging, policy updates, and deployment [6]. Complex numerical routines required by Thompson sampling significantly exacerbate these practical difficulties: *real-time* posterior sampling and action optimization leads the overall system to be cumbersome and hard to debug, posing challenges to reliable software development.

We provide examples where latency and system complexity are of central concern.

Example 1 (Advertising on third party systems)**:** Every time a user arrives to a third party webpage (e.g. New York Times), the advertising platform (e.g. Google Ads) decides which ad to show in order to maximize conversion. Here, latency is crucial to good user experience [6], and curbing system complexity increases service reliability [71]. $\diamond$

Example 2 (Ranking)**:** When a user logs in, an internet service chooses a list of items to display to the user in order to maximize revenue or engagement. Concrete examples include ranking news articles (Microsoft Network, MSN), products (online marketplaces like Amazon and Airbnb), and content (Facebook and LinkedIn feed). In all of these cases, latency is central to user satisfaction, but edge devices and front-end servers are resource constrained [6]. For instance, Facebook performs secondary ranking on device to avoid server communication latency, and only display content that has been downloaded completely [59].$\diamond$

Example 3 (Personalized Pricing)**:** As a customer enters a virtual platform, the system generates a personalized price based on market conditions (or projections thereof), and user-specific contexts. Electronic commerce firms and airlines use price controls to manage revenue [76], and two-sided online marketplaces (e.g. Uber, Lyft, Airbnb) dynamically set prices on both sides of the market to

reduce supply-demand imbalance. In both cases, latency is critical to satisfactory user experience and engagement. ⋄

Motivated by aforementioned challenges in implementing and deploying Thompson sampling in real production systems, we develop and analyze a method that maintains an explicit policy representation designed to imitate Thompson sampling. In order to avoid computationally demanding routines *online*, our algorithm simulates and imitates a Thompson sampling policy *offline*. An explicit policy representation can efficiently generate actions real-time even in large action spaces, without requiring real-time posterior inference or numerical optimization. This allows leveraging state-of-the-art Bayesian models—such as Guassian processes parameterized by deep neural networks (Section 5.2.2)—and optimization solvers *offline*, while maintaining low latency on resource-constrained computing modalities such as low-end mobile devices.[1] During operation, actions can be generated efficiently from the distilled policy by sampling from a parameterized distribution, allowing fast concurrent and asynchronous interaction with users. If we use neural networks to parameterize our imitation policy, this corresponds to a single forward propagation. Striking recent advances in machine learning systems allow such computations to be exceedingly efficient: as of May 2021, a forward pass for an industrial-scale image recognition model—ResNet101 [38]—took 0.3880 milliseconds [24].

By performing posterior updates and mimicking the behavior of Thompson sampling offline, we are able to move complex numerical routines from resource-constrained mobile devices to backend servers, and reduce long-term software development costs (tech debt). Such offline procedures using batched observations can be easily implemented using modern industry machine learning pipelines [32, 31]. This allows leveraging the recent remarkable progress in machine learning software infrastructure, such as engineering best practices and tools for reliable testing & deployment[2]. Buoyed by these practical advantages, our algorithm has been implemented in a video transcoding system for a large social network service, and is reliably handling millions of video uploads per day.

Methodological & theoretical contributions Motivated by its strong practical advantages, we prove rigorous performance guarantees for our imitation algorithm. Since our updates to the Thompson sampling policy are based on observations generated by the imitation policy, our algorithm emulates an *off-policy* version of Thompson sampling which may substantially diverge from the true Thompson sampling policy. An uninformed, pessimistic view of our procedure states that any initially small deviation between the Thompson sampling policy may cascade across time. Our main theoretical results (Section 4-5) preclude such possibility, and ensure that small deviations between our imitation policy and Thompson sampling do not magnify over time. Specifically, our imitation policy enjoys Bayes regret comparable to that of true, on-policy Thompson sampling, up to the sum of single-step imitation errors.

For large-scale internet applications, unsupervised contexts—those without corresponding actions or rewards—are cheap and abundant. For example, the entire user database provides a wealth of such contexts. We note that solving the imitation problem, or equivalently finding the policy parameterization closest to TS, only requires unsupervised contexts. In Section 4, we prove that each

[1] More generally, optimization can be a challenge for non-Bayesian methods. Although outside of the scope of this paper, generalizing our imitation framework to other policies will likely yield fruit in separating optimization from online action-generation.

[2] As an example, a dedicated top peer-reviewed conference for ML systems `https://mlsys.org/` was recently established, and is undergoing rapid growth at the forefront of academia and industry. This community focuses on improving the efficiency of ML systems from an *operational* perspective.

single-period imitation error term can be controlled—with a sufficiently rich imitation model—at the rate $O_p(1/\sqrt{N})$, where N is the number of supervised and unsupervised contexts.

Combining this with our aforementioned regret bound (which we show in Section 5), our imitation algorithm achieves Bayes regret comparable to Thompson sampling up to $O_p(T/\sqrt{N})$-error. Despite the seemingly linear gap in Bayes regret, we often have $T \ll \sqrt{N}$ in internet applications where we can utilize the database of users / entities. Typically, N is in the order of hundreds of millions; as of 2020, Facebook had 2.7 billion monthly active users; in our motivating video transcoding application, the service receives millions of video upload requests *every day*, providing an effectively unlimited number of unsupervised contexts. In contrast, the number of model updates (horizon T) is relatively small, in tens or hundreds, due to complexities of policy deployment and nonstationary user behavior. In such practical problem instances, our imitation policy thus enjoys Bayes regret comparable to that of Thompson sampling, achieving optimal (gap-independent) regret in the wealth of examples where Thompson sampling is known to be optimal. Empirically, we evaluate our imitation algorithm on several benchmark problems and a real-world dataset for selecting optimal video transcoding configurations (Section 6). In all of our experiments, our imitation algorithm performs as well as Thompson sampling in terms of cumulative regret, while reducing decision-time latency by an order of magnitude.

2 Related work

There is a substantial body of work on Thompson sampling and its variants that use computationally efficient subroutines. We give a necessarily abridged overview of how our algorithm situates with respect to the extensive literature on bandits, approximate inference, and imitation learning.

A number of authors have showed that Thompson sampling achieves optimal regret for multi-armed bandits [8, 9, 41, 40]. We refer the reader to the recent tutorial by Russo et al. [68] and references therein for a comprehensive overview. Agrawal and Goyal [10], Abeille et al. [4] showed regret bounds for linear stochastic contextual bandits for a Thompson sampling algorithm with an uninformative Gaussian prior, and Gopalan et al. [34] studied finite parameter spaces. Russo and Van Roy [66] established Bayesian regret bounds for Thompson sampling with varying action sets (which includes, in particular, contextual bandits); Russo and Van Roy [67] provides an information-theoretic analysis that makes explicit the dependence on the prior (see also Bubeck and Eldan [19]). We build on the insights of Russo and Van Roy [66], and show that our imitation algorithm retains the advantageous properties of Thompson sampling, achieving (gap-independent) Bayes regret comparable to the *best* UCB algorithm.

Practical performance of Thompson sampling depends critically on having access to well-calibrated probabilistic predictions. Obtaining a balance between predictive accuracy, computational time, and memory requirements can be challenging in the context of large datasets with overparameterized models. Exact posterior sampling from even the simplest Gaussian linear models has a time complexity of $O(n^2)$, where n is the number of model parameters[3]. A common strategy used by some variational inference methods is to use a mean-field approach where parameters are assumed to be independent [17]. This assumption can decrease sampling costs from $O(n^2)$ to $O(n)$, where n is the number of parameters. However, Riquelme et al. [62] found that Thompson sampling using such approaches often leads to poor empirical performance.

[3]This assumes that the root decomposition of the covariance matrix has been computed and cached, which incurs a cost of $O(n^3)$.

When exact posterior inference is not possible, approximate inference methods can be used for posterior sampling. We refer the reader to Chapter 5 of Russo et al. [68]'s recent tutorial for a discussion of approximation methods in relation to Thompson sampling. Bootstrapping [29, 58, 51] is a simple heuristic procedure that maintains multiple models to approximate samples from the posterior distribution, although maintaining multiple models is often computationally expensive. MCMC-based methods for approximate inference, and Hamilton Monte Carlo (HMC) [57] in particular, are largely regarded as the "gold standard" for approximate Bayesian inference. HMC, and other MCMC-like approaches (e.g., Chen et al. [22], Welling and Teh [84]) generate an arbitrary number of posterior samples for all parameters. While such algorithms permit rapid evaluation of posterior samples (since the parameters are already sampled), they require substantial memory to store multiple samples of the parameters. Recent methods have also considered decomposing the covariance or precision matrix into a diagonal and low-rank component [90, 52]. While this reduces computational complexity and memory costs relative to using the full covariance, sampling still incurs a time complexity of $O((n+1)\rho)$ where ρ is the rank of the covariance (or precision matrix) and ρ copies of the weights must be stored.

By pre-computing and distilling Thompson sampling, our imitation learning framework allows the use of the most appropriate inferential procedure for the task at hand, rather than what is feasible to run in an online setting. In particular, the separation of online decision-making and offline computation allows the use of state-of-the-art Bayesian methods, such as those utilizing deep neural networks [83]. While we restrict discussion to Thompson sampling in this work, the basic idea of offline imitation learning can be used to learn a explicit policy representation of any complicated policy and allow operationalization at scale.

Imitation learning methods have received much attention recently, owing to their ability to learn complicated policies from expert demonstrations [3, 63, 39]. Our approach of minimizing the discrepancy between a parameterized policy and Thompson sampling can be viewed as an implementation of behavioral cloning [63, 75, 64]. Our imitation learning procedure resembles the "Bayesian dark knowledge" approach from Korattikara et al. [44], which uses a neural network to approximate Bayesian posterior distributions. While most works in the imitation learning literature study reinforcement learning problems, we focus on the more limited contextual bandit setting, which allows us to show strong theoretical guarantees. We anticipate the growing list of works on imitation learning to be crucial in generalizing our imitation framework to the reinforcement learning (RL) setting. To account for time dependencies in state evolutions, both inverse RL approaches that directly model the reward [3, 74], and the recent advances in generative adversarial imitation learning techniques [39, 49] show substantive promise in generalizing our imitation algorithm (behavioral cloning) to RL problems.

3 Distilled Thompson sampling

We consider a (Bayesian) contextual bandit problem where the agent / decision-maker observes a context, takes an action, and receives a reward. Let Θ be the parameter space, and let $\theta \sim P$ be a prior distribution on Θ. At each time t, we denote the context $S_t \overset{\text{iid}}{\sim} \mathbb{P}_S$, action $A_t \in \mathcal{A}$, and reward $R_t \in \mathbb{R}$. We consider a well-specified reward model class $\{f_\theta : \mathcal{A} \times \mathcal{S} \to \mathbb{R} \mid \theta \in \Theta\}$ such that

$$f_\theta(a, s) = \mathbb{E}[R_t \mid \theta, A_t = a, S_t = s] \quad \text{for all} \quad a \in \mathcal{A}, s \in \mathcal{S}.$$

Let $H_t = (S_1, A_1, R_1, \ldots, S_{t-1}, A_{t-1}, R_{t-1})$ be the history of previous observations at time t. Assume that regardless of H_t, the mean reward at time t is determined only by the context-action pair

$$\mathbb{E}[R_t \mid \theta, H_t, S_t = s, A_t = s] = f_\theta(a, s),$$

or equivalently, $R_t = f_\theta(A_t, S_t) + \epsilon_t$ where ϵ_t is a mean zero i.i.d. noise.

We use π_t to denote the policy at time t that generates the actions A_t, based on the history H_t: conditional on the history H_t, we have $A_t \mid S_t \sim \pi_t(\cdot \mid S_t)$, where we abuse notation to suppress the dependence of π_t on H_t. The agent's objective is to maximize the cumulative sum of rewards by sequentially updating the policy π_t based on observed context-action-reward tuples. The *regret* of the agent compares the agent's cumulative reward to the reward under the optimal action: for any fixed parameter value $\theta \in \Theta$, the (frequentist) regret for the set of policies $\{\pi_t\}_{t\in\mathbb{N}}$ is

$$\text{Regret}\left(T, \{\pi_t\}_{t\in\mathbb{N}}, \theta\right) := \sum_{t=1}^{T} \mathbb{E}\left[\max_{a\in\mathcal{A}} f_\theta(a, S_t) - f_\theta(A_t, S_t) \mid \theta\right].$$

For simplicity, we assume $\text{argmax}_{a\in\mathcal{A}} f_\theta(a, s)$ is nonempty almost surely.

We assume the agent's prior, P, is *well-specified*[4], a key (standard) assumption that drives our subsequent analysis. Under the prior P over $\theta \in \Theta$, the Bayes regret is simply the frequentist regret averaged over $\theta \sim P$

$$\text{BayesRegret}\left(T, \{\pi_t\}_{t\in\mathbb{N}}\right) := \mathbb{E}_{\theta\sim P}[\text{Regret}\left(T, \{\pi_t\}_{t\in\mathbb{N}}, \theta\right)] = \sum_{t=1}^{T} \mathbb{E}_{\theta\sim P}\left[\max_{a\in\mathcal{A}} f_\theta(a, S_t) - f_\theta(A_t, S_t)\right].$$

We ignore measurability issues in our subsequent discussion, although a careful use of outer measures can rigorously deal with such issues (e.g. see van der Vaart and Wellner [80, Chapter 1]).

Based on the history so far, Thompson sampling plays an action according to the posterior probability of the action being optimal. The posterior probabilities are computed based on the prior P and previously observed context-action-reward tuples. At time t, this is often implemented by sampling from the posterior $\theta_t \sim P(\theta \in \cdot \mid H_t, S_t)$, and setting $A_t^{\text{TS}} \in \text{argmax}_{a\in\mathcal{A}} f_{\theta_t}(a, S_t)$. By definition, Thompson sampling enjoys the optimality property $A_t^{\text{TS}} \mid H_t, S_t \overset{d}{=} A_t^\star \mid H_t, S_t$ where $A_t^\star \in \text{argmax}_{a\in\mathcal{A}} f_\theta(a, S_t)$ and θ is the true parameter drawn from the prior P.

Motivated by aforementioned challenges in implementing Thompson sampling real-time, we develop an imitation learning algorithm that separates *online* action generation from computationally intensive steps like posterior sampling and optimization. Our algorithm maintains an explicit policy representation that emulates the Thompson sampling policy by simulating its actions *offline*. At decision time, the algorithm generates an action simply by sampling from the current policy representation, which is straightforward to implement and computationally efficient to run real-time.

At each time t, our algorithm observes a context S_t, and plays an action drawn from its explicit policy representation. Formally, we parameterize our policy $\pi^m(a \mid s)$ with a model class $m \in \mathcal{M}$.

[4]When the prior is misspecified so that the Thompson sampling policy uses Q instead of P, we have the equivalence as noted by Russo and Van Roy [66]

$$\mathbb{E}_{\theta\sim P}[\text{Regret}\left(T, \{\pi_t\}_{t\in\mathbb{N}}, \theta\right)] \le \left\|\frac{dP}{dQ}\right\|_{L^\infty(\mathcal{X})} \mathbb{E}_{\theta\sim Q}[\text{Regret}\left(T, \{\pi_t\}_{t\in\mathbb{N}}, \theta\right)],$$

where dP/dQ is the Radon-Nikodym derivative of P with respect to Q. While misspecified priors can incur substantially higher regret [50] in the worst-case, empirical evidence suggests Thompson sampling is a strong algorithm in practice [70, 36, 21, 55, 30, 35, 7, 42, 69, 6].

For example, $\mathcal{M}$ can be a neural network that takes as input a context and outputs a distribution over actions. We generate actions by sampling from the current policy $A_t \sim \pi_t^m(\cdot \mid S_t)$, which can be easily implemented to run with low latency on resource-constrained computing infrastructure such as mobile devices. Upon receiving a corresponding reward R_t, the agent uses the context-action-reward tuple to update its posterior on the parameter $\theta \in \Theta$ *offline*. Although this step requires posterior inference that may be too burdensome to run real-time, our method allows running it offline on a different computing node, so that it does not affect latency. Using the updated posterior $\theta_t \sim \mathbb{P}(\cdot \mid H_t)$, the agent then simulates actions drawn by the Thompson sampling policy by computing the maximizer $A_t^{\mathrm{TS}}(s) \in \operatorname{argmax}_{a \in \mathcal{A}} f_{\theta_t}(a, s)$, for a range of values $s \in \mathcal{S}$. Using these simulated context-action pairs, we learn an explicit policy representation that *imitates* the observed actions of the Thompson sampling policy.

We summarize an idealized form of our method in Algorithm 1, where conditional on the history H_t generated by the imitation policy, we denote the off-policy Thompson sampling policy at time t as $\pi_t^{\mathrm{TS}}(a \mid s)$. This policy is different from the true, on-policy Thompson sampler since the imitation policy generates actions based on which rewards are observed.

Algorithm 1 Imitating Thompson Sampling

1: Input: prior P on parameter space Θ, reward model class $\{f_\theta(\cdot, \cdot)\}$, imitation policy model class $\{\pi^m : m \in \mathcal{M}\}$, notion of distance D for probabilities
2: Initialize $m \leftarrow \operatorname{argmin}_{m \in \mathcal{M}} \mathbb{E}_{S \sim \mathbb{P}_S}[D\left(\pi_0^{TS}, \pi^m \mid S\right)]$
3: **for** $t = 1$ **to** T **do**
4: Observe S_t, sample $A_t \sim \pi_t^m(\cdot \mid S_t)$, receive R_t
5: Fit $m \leftarrow \operatorname{argmin}_{m \in \mathcal{M}} \mathbb{E}_{S \sim \mathbb{P}_S}[D\left(\pi_t^{TS}, \pi^m \mid S\right)]$
6: **end for**

Dropping the t subscript to simplify notation, the imitation learning problem

$$\underset{m \in \mathcal{M}}{\text{minimize}}\, \mathbb{E}_{S \sim \mathbb{P}_S}\left[D\left(\pi^{\mathrm{TS}}, \pi^m \mid S\right)\right]. \tag{1}$$

learns a model $m \in \mathcal{M}$ minimizing a measure of discrepancy $D\left(\cdot, \cdot \mid S\right)$ between the two distributions on $\mathcal{A}$, conditional on the context S. This imitation objective (1) cannot be computed analytically, and we provide efficient approximation algorithms below.

To instantiate Algorithm 1, we fix Kullback-Leibler (KL) divergence as the notion of discrepancy between probabilities and present finite-sample approximations based on observed contexts and simulated actions from the off-policy Thompson sampling policy π_t^{TS}. For probabilities q^1 and q^2 on $\mathcal{A}$ such that $q^1, q^2 \ll \nu$ for some σ-finite measure ν on $\mathcal{A}$, the KL divergence between q^1 and q^2 is

$$D_{\mathrm{kl}}\left(q^1 \| q^2\right) := \int_{\mathcal{A}} \log \frac{dq^1/d\nu}{dq^2/d\nu}(a) d\nu(a),$$

where we use $\frac{dq^1}{d\nu}$ and $\frac{dq^2}{d\nu}$ to denote Radon-Nikodym derivatives of q^1 and q^2 with respect to ν. For two policies π^1 and π^2, we define

$$D_{\mathrm{kl}}\left(\pi^1, \pi^2 \mid S\right) := D_{\mathrm{kl}}\left(\pi^1(\cdot \mid S) \| \pi^2(\cdot \mid S)\right),$$

where we use π^1, π^2 to also denote their conditional densities over $\mathcal{A}$.

The imitation problem (1) with $D\left(\cdot, \cdot \mid S\right) = D_{\mathrm{kl}}\left(\cdot, \cdot \mid S\right)$ is equivalent to a maximum log

likelihood problem

$$\underset{m \in \mathcal{M}}{\text{maximize}}\, \mathbb{E}_{S \sim \mathbb{P}_S, A^{\text{TS}} \sim \pi^{\text{TS}}(\cdot|S)}[\log \pi^m(A^{\text{TS}} \mid S)]. \tag{2}$$

In the following, we write $\mathbb{E}[\cdot] = \mathbb{E}_{S \sim \mathbb{P}_S, A^{\text{TS}} \sim \pi^{\text{TS}}(\cdot|S)}[\cdot]$ for simplicity. In the maximum likelihood estimation (MLE) problem (2), the data comprises of context-action pairs; contexts are generated under the marginal distribution $S \sim \mathbb{P}_S$ independent of everything else, and conditional on the context, actions are simulated from the Thompson sampling policy $A^{\text{TS}} \sim \pi^{\text{TS}}(\cdot \mid S)$. The MLE problem (2) finds models maximizing the log likelihood of actions observed under π_t^{TS}.

The imitation objective $m \mapsto \mathbb{E}[\log \pi^m(A^{\text{TS}} \mid S)]$ involves an expectation over the unknown marginal distribution of contexts $\mathbb{P}_S$ and actions generated by the Thompson sampling policy $\pi^{\text{TS}}(\cdot \mid S)$. Although the expectation over $S \sim \mathbb{P}_S$ involves a potentially high-dimensional integral over an unknown distribution, sampling from this distribution is usually very cheap since the observations $S \sim \mathbb{P}_S$ can be "unsupervised" in the sense that no corresponding action/reward are necessary. For example, it is common for internet services to maintain a database of features S for all of its customers. Using these contexts, we can solve the MLE problem (2) efficiently via stochastic gradient descent methods [47, 28].

Accommodating typical application scenarios, Algorithm 1 and its empirical approximation extends to settings where the agent has the ability to concurrently interact with inputs (users) in an asynchronous manner. This is a practically important feature of the algorithm, since most applications require the agent to concurrently generate actions as user requests come in asynchronously. Using the imitation policy $\pi^m(a \mid s)$, it is trivial to parallelize action generation over multiple computing nodes, even on mobile devices.

In Section 4, we show that it is easy to solve the imitation problem (1) to high accuracy by using cheap unsupervised contexts. In Section 5, we show that our imitation algorithm enjoys Bayes regret comparable to that of the true Thompson sampling algorithm, up to the sum of single step imitation errors. Although our theoretical developments focus on the non-concurrent case, in our experiments (Section 6) we present large batch concurrent scenarios to illustrate typical application scenarios. For continuous action spaces with a notion of geometry, it is sometimes natural to allow imitation policies to have slightly different support than the Thompson sampling policy. In this scenario, we can instantiate the abstract form of Algorithm 1 with Wasserstein distances as our notion of discrepancy $D(\cdot, \cdot \mid s)$. The subsequent theoretical development for KL divergences has its analogue for Wasserstein distances, which we outline in Section A

4 Generalization guarantees for imitation learning

Given i.i.d. observations of (potentially unsupervised contexts) $S_i \overset{\text{iid}}{\sim} P_S$, we solve the empirical approximation to the imitation problem (2)

$$\widehat{m} \in \underset{m \in \mathcal{M}}{\text{argmax}} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{N_a} \sum_{j=1}^{N_a} \log \pi^m(A_{ij}^{\text{TS}} \mid S_i), \tag{3}$$

where we simulate actions from the (off-policy) Thompson sampler $A_{ij}^{\text{TS}} \sim \pi_t^{TS}(\cdot \mid S_i) \;\; j = 1, \ldots, N_a$ for each context S_i. Since actions can be simulated *offline* in a parallel manner, we can efficiently generate a large number of actions N_a.

In what follows, we assume that our imitation model class is *well-specified*, so that there exists $m^\star \in \mathcal{M}$ satisfying $\pi^{\text{TS}} = \pi^{m^\star}$, where we omitted the subscript t and denote $\pi^{\text{TS}} = \pi_t^{\text{TS}}$ to ease.

For example, by considering expressive model classes such as neural networks or nonparametric models, this is often a reasonable assumption. For well-specified imitation models, we prove that with probability at least $1-\delta$,

$$\mathbb{E}_{S\sim\mathbb{P}_s}[D_{\mathrm{kl}}\left(\pi^{\mathrm{TS}}, \pi^{\widehat{m}} \mid S\right)] \lesssim \frac{\mathfrak{Comp}_N}{N}\log\frac{1}{\delta} + \frac{1}{\sqrt{NN_a}}\mathfrak{Comp}_{N,N_a}\log\frac{1}{\delta}, \tag{4}$$

for some complexity measures $\mathfrak{Comp}_N$ and $\mathfrak{Comp}_{N,N_a}$ associated with the imitation model class $\mathcal{M}$. Here, the notation $\lesssim$ denotes inequality up to a universal constant. In typical internet applications, the number of unsupervised contexts N is exceedingly large, and the imitation error (4) can be made vanishingly small.

The key challenges to showing the preceding result are twofold: 1) the empirical procedure (3) employs non-i.i.d. samples $(S_i, A_{ij}^{\mathrm{TS}})$, so standard concentration results do not apply as-is, and 2) the bound (4) scales at the "fast rate" $1/N$, rather than the canonical parametric rate $1/\sqrt{N}$. To overcome the first challenge, our proof carefully derives concentration inequalities for the two-step sampling process where nature generates $S_i \overset{\mathrm{iid}}{\sim} P_s$, and for each S_i, we simulate $A_{ij} \overset{\mathrm{iid}}{\sim} \pi^{\mathrm{TS}}(\cdot \mid S_i)$ via posterior sampling. To prove the fast rate of convergence $1/N$, we use a somewhat sophisticated approach to generalization bounds based on localization [11]; our proof leverages the fact that complexity of the function class $(s,a) \mapsto \log \pi^m(a \mid s)$ may be substantially smaller on a neighborhood of the optimum $m^\star$, than over the entire space $m \in \mathcal{M}$.

To formalize our arguments, recall first the standard notion of Rademacher complexity: for a fixed $\xi_1, \ldots, \xi_n$ and i.i.d. random signs (Rademacher variables) $\varepsilon_i \in \{-1, 1\}$ that are independent of the ξ_i's, the empirical Rademacher complexity of the class of functions $\mathcal{G} \subseteq \{g : \Xi \to \mathbb{R}\}$ is

$$\mathfrak{R}_n(\mathcal{G}) := \mathbb{E}_\varepsilon\left[\sup_{g\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i g(\xi_i)\right].$$

A function $\psi : \mathbb{R}_+ \to \mathbb{R}_+$ is *sub-root* [11] if it is nonnegative, nondecreasing, and $r \mapsto \psi(r)/\sqrt{r}$ is nonincreasing for all $r > 0$. This analytic notion guarantees that any non-constant sub-root function ψ is continuous, and has a unique positive fixed point $r^\star = \psi(r^\star)$, where $r \geq \psi(r)$ for all $r \geq r^\star$. Let $\psi_n : \mathbb{R}_+ \to \mathbb{R}_+$ be a sub-root upper bound on the localized Rademacher complexity

$$\psi_n(r) \geq \mathbb{E}[\mathfrak{R}_n(\{g \in \mathcal{G} : \mathbb{E}[g^2] \leq r\})]. \tag{5}$$

(The localized Rademacher complexity itself is sub-root.) Fixed points of ψ_n characterize uniform concentration guarantees; see Bartlett et al. [11] and Koltchinskii [43] for a detailed analysis of localized Rademacher complexities.

The Rademacher complexity of the following set of functions controls generalization performance of the empirical imitation model (3)

$$\begin{aligned}
\mathcal{G}_1 &:= \left\{ s \mapsto \mathbb{E}_{A^{\mathrm{TS}}\sim\pi^{\mathrm{TS}}(\cdot|s)}\left[\log\frac{\pi^{\mathrm{TS}}(A^{\mathrm{TS}} \mid s)}{\pi^m(A^{\mathrm{TS}} \mid s)}\right] : m \in \mathcal{M}\right\} \\
\mathcal{G}_2(s) &:= \{a \mapsto \log \pi^m(a \mid s) : m \in \mathcal{M}\} \\
\mathcal{G}_3 &:= \{(a,s) \mapsto \log \pi^m(a \mid s) : m \in \mathcal{M}\}.
\end{aligned}$$

We let $r_N^\star$ be the fixed point of the sub-root function ψ_n satisfying the bound (5) for $\mathcal{G} = \mathcal{G}_1$. For

any fixed context $s \in \mathcal{S}$, using i.i.d. random signs ε_j, we write

$$\mathfrak{R}_N \mathcal{G}_2(s) := \mathbb{E}_\epsilon \left[\sup_{m \in \mathcal{M}} \frac{1}{N_a} \sum_{j=1}^{N_a} \varepsilon_j \log \pi^m(A_j^{\mathrm{TS}} \mid s) \right].$$

For $\mathcal{G}_3$, using i.i.d. random signs ε_{ij} we still write

$$\mathfrak{R}_{NN_a} \mathcal{G}_3 := \mathbb{E}_\epsilon \left[\sup_{m \in \mathcal{M}} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{N_a} \sum_{j=1}^{N_a} \varepsilon_{ij} \log \pi^m(A_{ij}^{\mathrm{TS}} \mid S_i) \right].$$

Our main result in this section shows that the empirical solution (3) generalizes at the rate $O_p\left(N^{-1} + N^{-1/2} N_a^{-1/2}\right)$. See Section B.1 for the proof.

Theorem 1. *Let there exist a $m^\star \in \mathcal{M}$ such that $\pi^{\mathrm{TS}} = \pi^m$. Assume $|\log \pi^m(a \mid s)| \le M$ for all $a \in \mathcal{A}, s \in \mathcal{S}, m \in \mathcal{M}$. There is a numerical constant $C > 0$ such that with probability at least $1 - 2e^{-t}$*

$$\mathbb{E}\left[D_{\mathrm{kl}}\left(\pi^{\mathrm{TS}}, \pi^{\widehat{m}} \mid S \right) \right] \le C \left(\frac{1}{M} r_N^\star + \frac{Mt}{N} + \sqrt{\frac{t}{N}} \sup_{s \in \mathcal{S}} \mathbb{E}_{A_j^{\mathrm{TS}} \overset{\mathrm{iid}}{\sim} \pi^{\mathrm{TS}}(\cdot \mid s)} [\mathfrak{R}_{N_a}(\mathcal{G}_2(s))] + \mathbb{E}[\mathfrak{R}_{NN_a}(\mathcal{G}_3)] \right).$$

For finite-dimensional model classes with bounded VC-dimension, standard arguments bound the Rademacher complexity terms in the above theorem [80, Ch 2.6]. Denoting by $\mathsf{VC}(\cdot)$ the VC-dimension, we have

$$\sup_{s \in \mathcal{S}} \mathbb{E}_{A_j^{\mathrm{TS}} \overset{\mathrm{iid}}{\sim} \pi^{\mathrm{TS}}(\cdot \mid s)} [\mathfrak{R}_{N_a}(\mathcal{G}_2(s))] \le M \sqrt{\frac{\sup_{s \in \mathcal{S}} \mathsf{VC}(\mathcal{G}_2(s))}{N_a}} \quad \text{and} \quad \mathbb{E}[\mathfrak{R}_{NN_a}(\mathcal{G}_3)] \le M \sqrt{\frac{\mathsf{VC}(\mathcal{G}_3)}{NN_a}}.$$

Moreover, Corollary 3.7 of Bartlett et al. [11] implies that $r_N^\star \asymp \frac{M \mathsf{VC}(\mathcal{G}_1) \log(N/\mathsf{VC}(\mathcal{G}_1))}{N}$. Plugging these bounds in Theorem 1, we obtain the previously claimed convergence rate (4).

Our bounds based on the localized Rademacher complexity can provide guarantees for substantially larger and more expressive *nonparametric* model classes. We consider a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ defined over a kernel $k : \Xi \times \Xi \to \mathbb{R}_+$ [13]. For such nonparametric models, standard covering number bounds are loose [46], while localized arguments can still provide fast concentration [56]. Consider a RKHS with norm $\|\cdot\|_{\mathcal{H}}$ and evaluation kernel $k(\cdot, \cdot)$. Mercer's theorem [25] states that the integral operator $T_k : L^2(\Xi, P) \to L^2(\Xi, P), T_k(h)(\xi) = \int h(\xi') K(\xi, \xi') dP(\xi')$ is compact, and we have the eigenbasis expansion $k(\xi, \xi') = \sum_{j=1}^{\infty} \lambda_j \phi_j(\xi) \phi_j(\xi')$ where λ_j are eigenvalues of T sorted in decreasing order and ϕ_j give an orthonormal decomposition in $L^2(\mathcal{Z}, P)$.

Let $k_{\mathcal{S}} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_+$ and $k_{\mathcal{A}} : \mathcal{A} \times \mathcal{A} \to \mathbb{R}_+$ be kernels on $\mathcal{S}$ and $\mathcal{A}$ respectively, and let us denote by $\mathbb{B}_{\mathcal{S}}$ and $\mathbb{B}_{\mathcal{A}}$ the unit ball in the respective RKHS's. The kernels $k_{\mathcal{S}}$ and $k_{\mathcal{A}}$ induce a RKHS over functions on $\mathcal{S} \times \mathcal{A}$ formed with the kernel $k((s, a), (s', a')) = k_{\mathcal{S}}(s, s') + k_{\mathcal{A}}(a, a')$; we denote the unit ball in this space by $\mathbb{B}_{\mathcal{S} \times \mathcal{A}}$. For simplicity, we assume that the function classes $\mathcal{G}_1, \mathcal{G}_2(s)$, and $\mathcal{G}_3$ belong in a unit ball in appropriately defined RKHS's

$$\mathcal{G}_1 \in \mathbb{B}_{\mathcal{S}}, \quad \mathcal{G}_2(s) \in \mathbb{B}_{\mathcal{A}} \text{ for all } s \in \mathcal{S}, \quad \mathcal{G}_3 \in \mathbb{B}_{\mathcal{S} \times \mathcal{A}}.$$

In what follows, we show that the rate of decay of the eigenvalues of $T_{k_{\mathcal{S}}}$ controls the rate of convergence in Theorem 1. For example, eigenvalues of the popular Gaussian kernel $k(\xi, \xi') =$

$\exp(-\frac{1}{2}\|\xi - \xi'\|_2^2)$ decay exponentially fast $\lambda_j \lesssim e^{-j^2}$ [56]. Eigenvalues of kernel operators T_k for Sobolev spaces [16, 37] decay polynomially fast $\lambda_j \lesssim j^{-2\beta}$, where $\beta > \frac{1}{2}$ is the smoothness level. e.g., in 1-dimension, the first-order Sobolev kernel $k(\xi, \xi') = 1 + \min\{\xi, \xi'\}$ where $\beta = 1$ generates RKHS of Lipschitz functions. We prove the below corollary in Section B.2.

Corollary 1. *Assume* $\sup_{s\in\mathcal{S}} k_{\mathcal{S}}(s,s) + \sup_{a\in\mathcal{A}} k_{\mathcal{A}}(a,a) \le B$ *for some* $B > 0$. *If the eigenvalues of* $T_{k_{\mathcal{S}}}$ *decay as* $\lambda_j \lesssim e^{-j^2}$, *for a numerical constant* $C > 0$, *with probability at least* $1 - 2e^{-t}$

$$\mathbb{E}\left[D_{\mathrm{kl}}\left(\pi^{\mathrm{TS}}, \pi^{\widehat{m}} \mid S\right)\right] \le C\frac{Mt + \sqrt{\log N}}{N} + CMB\sqrt{\frac{(t+1)}{NN_a}}.$$

If the eigenvalues of $T_{k_{\mathcal{S}}}$ *decay as* $\lambda_j \lesssim j^{-2\beta}$ *for some* $\beta > 1/2$, *then there is another numerical constant* $C > 0$ *such that with probability at least* $1 - 2e^{-t}$

$$\mathbb{E}\left[D_{\mathrm{kl}}\left(\pi^{\mathrm{TS}}, \pi^{\widehat{m}} \mid S\right)\right] \le C\left(\frac{Mt}{N} + N^{\frac{-2\beta}{2\beta+1}} + MB\sqrt{\frac{t+1}{NN_a}}\right).$$

5 Imitation controls regret

We now show that minimizing the KL divergence (1) between Thompson sampling and the imitation policy allows controlling the Bayes regret of the imitation algorithm. In this sense, the imitation learning loss (1) is a valid objective where better imitation translates to gains in decision-making performance. By controlling the imitation objective (1), our results prove Algorithm 1 and its empirical approximation enjoys similar optimality guarantees as Thompson sampling.

Since the imitation policy is responsible for generating actions in Algorithm 1, the observations used to update the Thompson sampling policy are different from what the Thompson sampling policy would have generated. In this sense, our imitation algorithm does not emulate the *true* Thompson sampling policy, but rather simply mimics its *off-policy* variant, where the posterior updates are performed based on the history generated by the imitation policy. Our analysis shows that such off-policy imitation is sufficient to achieve similar Bayes regret bounds available for *on-policy* Thompson sampling [66], up to the sum of single-step imitation errors. In particular, our results guard against potential compounding of errors resulting from imitating the off-policy variant of Thompson sampling.

In Section 5.1, we relate performance of our imitation policy with that of the off-policy Thompson sampler π^{TS}, and show that the off-policy version π^{TS} admits a Bayes regret decomposition similar to its on-policy counterpart (true Thompson sampling). Our imitation algorithm matches the performance of π^{TS} up to the sum of single-period imitation errors, which can be made small by using results shown in Section 4. As we prove in Section 5.2, this allows proving Bayes regret bounds for the imitation policy by utilizing existing proofs for bounding the regret of a UCB algorithm.

5.1 Regret decomposition

Since our imitation learning problem (1) approximates *off-policy* Thompson sampling, a pessimistic view is that any small deviation between the imitation and Thompson sampling policy can exacerbate over time. A suboptimal sequence of actions taken by the imitation policy may deteriorate the performance of the off-policy Thompson sampling policy π^{TS} updated based on this data, when

compared to the update based on data collected by itself. Since the imitation policy again mimics this off-policy Thompson sampler, this may lead to a negative feedback loop in the worst-case.

Our analysis precludes such negative cascades when outcomes are averaged over the prior P: the Bayes regret of the imitation policy is comparable to that of the best UCB algorithm, up to only the sum of expected discrepancy between the off-policy Thompson sampling policy and the imitation learner at each period. Imitation error at each period does not affect the Bayes regret linearly in T as our worst-case intuition suggests, but rather only as a one-time approximation cost. To achieve near-optimal regret, it thus suffices to control the imitation objective (1) using cheap unsupervised contexts as demonstrated in Section 4.

Before giving a formal result, we first summarize our approach which builds on the insights of Russo and Van Roy [66]. We connect the performance of our imitation policy to that of the off-policy Thompson sampler and in turn relate the latter method's Bayes regret to that of the *best* UCB algorithm. Let $U_t(\cdot; H_t, S_t) : \mathcal{A} \to \mathbb{R}$ be a sequence of upper confidence bounds (UCB), and let A_t^{UCB} be the action taken by the UCB policy

$$A_t^{\mathrm{UCB}} \in \operatorname*{argmax}_{a \in \mathcal{A}} \; U_t(a; H_t, S_t).$$

Again denoting the optimal action by $A_t^\star \in \operatorname{argmax}_{a \in \mathcal{A}} f_\theta(a, S_t)$, a typical argument for bounding the regret of a UCB algorithm proceeds by noting that since $U_t(A_t^{\mathrm{UCB}}; H_t, S_t) \geq U_t(A_t^\star; H_t, S_t)$,

$$f_\theta(A_t^\star, S_t) - f_\theta(A_t^{\mathrm{UCB}}, S_t) \leq f_\theta(A_t^\star, S_t) - U_t(A_t^\star; H_t, S_t) + U_t(A_t^{\mathrm{UCB}}; H_t, S_t) - f_\theta(A_t^{\mathrm{UCB}}, S_t).$$

Taking expectations and summing over $t = 1, \ldots, T$, $\mathrm{BayesRegret}\left(T, \{\pi_t^{\mathrm{UCB}}\}_{t \in \mathbb{N}}\right)$ is bounded by

$$\sum_{t=1}^{T} \mathbb{E}[f_\theta(A_t^\star, S_t) - U_t(A_t^\star; H_t, S_t)] + \sum_{t=1}^{T} \mathbb{E}[U_t(A_t^{\mathrm{UCB}}; H_t, S_t) - f_\theta(A_t^{\mathrm{UCB}}, S_t)].$$

If the upper confidence bound property holds uniformly over the actions so that $U_t(a; H_t, S_t) \geq f_\theta(a, S_t)$ for all $a \in \mathcal{A}$ with high probability, the first term in the above regret decomposition can be seen to be nonpositive. To bound the second term, a canonical proof notes each upper confidence bound is not too far away from the population mean $f_\theta(A_t^{\mathrm{UCB}}, S_t)$. Russo and Van Roy [66]'s key insight was that a Thompson sampling algorithm admits an analagous Bayes regret decomposition as above, but with respect to *any* UCB sequence. This allows leveraging arguments that bound the (frequentist) regret of a UCB algorithm to bound the Bayes regret of Thompson sampling. Since the Bayes regret decomposition for Thompson sampling holds for *any* UCB sequence, the performance of Thompson sampling enjoys Bayes regret guarantees of the best UCB algorithm.

We show a similar Bayes regret decomposition for our imitation policy for any UCB sequence; the Bayes regret of the imitation policy enjoys a UCB regret decomposition similar to Thompson sampling, up to the cumulative sum of single-period approximation errors. Recall that we denote $A_t^{\mathrm{TS}} \sim \pi_t^{\mathrm{TS}}(\cdot \mid S_t)$, the action generated by the off-policy Thompson sampler. See Section C.1 for the proof of the following result.

Lemma 1. *Let $\{\pi_t\}_{t \in \mathbb{N}}$ be any sequence of policies (adapted to the history H_t), and let $U_t(\cdot; H_t, S_t) : \mathcal{A} \to \mathbb{R}$ be any upper confidence bound sequence based on (H_t, S_t). Let there be a sequence $M_t(H_t, S_t)$ and $L > 0$ such that $\sqrt{\mathbb{E}[M_t(H_t, S_t)^2]} \leq L$, and*

$$\sup_{a \in \mathcal{A}} |U_t(a; H_t, S_t)| \leq M_t(H_t, S_t). \tag{6}$$

Then, for all $T \in \mathbb{N}$

$$\text{BayesRegret}\left(T, \{\pi_t\}_{t\in\mathbb{N}}\right) \leq \underbrace{\sum_{t=1}^{T} \mathbb{E}[f_\theta(A_t^\star, S_t) - U_t(A_t^\star; H_t, S_t)] + \sum_{t=1}^{T} \mathbb{E}[U_t(A_t; H_t, S_t) - f_\theta(A_t, S_t)]}_{\textit{(b): regret decomposition for any UCB algorithm}}$$
$$+ \underbrace{L \sum_{t=1}^{T} \sqrt{\frac{1}{2}\mathbb{E}\left[D_{\text{kl}}\left(\pi_t^{\text{TS}}, \pi_t \mid S_t\right)\right]}}_{\textit{(a): imitation error}}. \tag{7}$$

The Bayes regret decomposition (7) shows that performance analysis of any UCB algorithm can characterize the regret of our imitation policy. In this sense, the imitation policy achieves regret comparable to the *optimal* UCB algorithm, up to the sum of single-period imitation errors. As outlined above, term (a) can be bounded using canonical UCB proofs following the approach of Russo and Van Roy [66]. We detail such arguments in two general modeling scenarios in the next subsection, where our results draw from regret guarantees for UCB algorithms [1, 2, 72]. Term (b) can be controlled by our imitation learning algorithm (Algorithm 1) and its empirical approximation as seen in Section 4. Although this term seemingly scales linearly in T at initial glance, each summand can be made as small as $O_p(1/\sqrt{N})$. In large-scale internet applications, the number of unsupervised contexts N is very large as they can simply be read off of a database of user information ($N \approx$ tens to hundreds of millions). The number of policy updates T is often orders of magnitude smaller (tens to hundreds) in a typical product lifecycle due to operational challenges in deploying a policy (e.g. software reliability). Thus, term (b) can often be made arbitrarily small using big datasets and powerful overparameterized imitation models.

The fact that we are studying Bayes regret, as opposed to the frequentist regret, plays an important role in the above decomposition. We conjecture that in the worst-case, even initially small imitation error (and consequently suboptimal exploration) can each linearly compound over time (leading to a prohibitive quadratic dependence on T). It remains open whether certain structures of the problem can provably preclude these negative feedback loops uniformly over θ, which is necessary for obtaining frequentist regret bounds for the imitation algorithm.

5.2 Regret bounds

We now show concrete regret guarantees for our imitation algorithm by bounding term (a) in the decomposition (7). Our Bayes regret bounds are instance-independent (gap-independent), and shows that our imitation policy achieves optimal regret (or best known bounds thereof) up to the sum of imitation error terms that can be controlled. We present two modeling scenarios, focusing first on the setting where the mean reward function $(a, s) \mapsto f_\theta(a, s)$ can be represented as a generalized linear model. Then, we turn to a general setting where the mean reward function $(a, s) \mapsto f_\theta(a, s)$ can be modeled as a Gaussian process in Section 5.2.2; in this nonparametric setting, our regret bounds scale with the maximal information gain possible over T rounds.

5.2.1 Generalized linear models

Consider the setting where the mean reward function $f_\theta(a, s)$ can be modeled as a generalized linear model with some link function. Specifically, we assume that $\Theta \subseteq \mathbb{R}^d$, and that there exists a feature

vector $\phi : \mathcal{A} \times \mathcal{S} \to \mathbb{R}^d$ and a link function $g : \mathbb{R} \to \mathbb{R}$ satisfying

$$f_\theta(a, s) = g(\theta^\top \phi(a, s)) \quad \text{for all} \quad \theta \in \Theta, a \in \mathcal{A}, s \in \mathcal{S}.$$

Armed with the regret decomposition in Lemma 1, we obtain regret bounds by following an eluder dimension argument pioneered by Russo and Van Roy [66]. We give its proof in Section C.2 for completeness.

Theorem 2. *Let $g : \mathbb{R} \to \mathbb{R}$ and $\phi : \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ such that $f_\theta(a, s) = g(\phi(a, s)^\top \theta)$ for all $\theta \in \Theta$, where g is an increasing, differentiable, 1-Lipschitz function. Let $c_1, c_2, \sigma > 0$ be such that*

$$\sup_{\theta \in \Theta} \|\theta\|_2 \leq c_1, \quad \text{and} \quad \sup_{a \in \mathcal{A}, s \in \mathcal{S}} \|\phi(a, s)\|_2 \leq c_2,$$

and assume that $R_t - f_\theta(A_t, S_t)$ is σ-sub-Gaussian conditional on (θ, H_t, S_t, A_t). If r is the maximal ratio of the slope of g

$$r := \frac{\sup_{\theta \in \Theta, a \in \mathcal{A}, s \in \mathcal{S}} g'(\phi(a, s)^\top \theta)}{\inf_{\theta \in \Theta, a \in \mathcal{A}, s \in \mathcal{S}} g'(\phi(a, s)^\top \theta)},$$

then there is a constant C that depends on c_1, c_2 such that

$$\text{BayesRegret}\left(T, \{\pi_t\}_{t \in \mathbb{N}}\right) \leq C(\sigma + 1) r d \sqrt{T} \log rT + c_1 c_2 \sum_{t=1}^{T} \sqrt{2\mathbb{E}\left[D_{\text{kl}}\left(\pi_t^{\text{TS}}, \pi_t \mid S_t\right)\right]}. \tag{8}$$

For linear contextual bandits $g(x) = x$, our upper bound on the Bayes regret is tight up to a factor of $\log T$ and the cumulative sum of the imitation errors [65]. For generalized linear models, the first two terms in the bound (8) are the tightest bounds on the regret available in the literature. By controlling the imitation error using the empirical approximation (3), we conclude that the imitation policy enjoy good Bayes regret bounds.

In the case of linear bandits, we can use a more direct argument that leverage the rich analysis of UCB algorithms provided by previous authors [26, 1, 2], instead of the eluder dimension argument used to show Theorem 2. See Section C.3 for such direct analysis.

5.2.2 Contextual Gaussian processes

In this section, we consider the setting where the mean reward function is nonparametric and model it as a sample path of a Gaussian process. Formally, we assume that $(a, s) \mapsto f_\theta(a, s)$ is sampled from a Gaussian process on $\mathcal{A} \times \mathcal{S}$ with mean function $\mu(a, s)$ and covariance function (kernel)

$$\Sigma((a, s), (a', s')) := \mathbb{E}[(f_\theta(a, s) - \mu(a, s))(f_\theta(a', s') - \mu(a', s'))].$$

We assume that the decision maker observes rewards $R_t = f_\theta(A_t, S_t) + \epsilon_t$, where the noise $\epsilon_t \overset{\text{iid}}{\sim} N(0, \sigma^2)$ are independent of everything else. Given these rewards, we are interested in optimizing the function $a \mapsto f_\theta(a, S_t)$ for each observed context S_t at time t. Modeling mean rewards as a Gaussian process is advantageous since we can utilize analytic formulae to update the posterior at each step. For large-scale applications, we can parameterize our kernels by a neural network and leverage the recently developed interpolations techniques to perform efficient posterior updates [85, 86, 87].

We build on the UCB regret bound due to Srinivas et al. [72] and bound the first two terms in the Bayes regret decomposition (7). In particular, we show that they can be controlled by the maximal amount of information on the optimal action that can be gained after T time steps. Recall

the definition of mutual information between two random vectors: $I(Z,Y) := D_{\mathrm{kl}}\left(P_{Z,Y} \| P_Z \times P_Y\right)$. We define the maximal possible information gain after T time steps as

$$\gamma_T := \sup_{\mathcal{X} \subseteq \mathcal{A} \times \mathcal{S} : |\mathcal{X}| = T} I(r_{\mathcal{X}}, f_{\mathcal{X}})$$

where $r_{\mathcal{X}} = \{f_\theta(x) + \epsilon_x\}_{x \in \mathcal{X}}$ and $f_{\mathcal{X}} = \{f_\theta(x)\}_{x \in \mathcal{X}}$. For popular Gaussian and Matern kernels, Srinivas et al. [72] has shown that the maximal information gain can be bounded explicitly; we summarize these bounds shortly.

Letting $\mathcal{A} \subseteq [0,r]^d$ for some $r > 0$, we show that the first two terms in the decomposition (7) can be bounded by $O\left(\sqrt{d\gamma_t T (\log T)^d}\right)$, thus bounding the Bayes regret up to the sum of imitation error terms. In the following, we use L_f to denote the (random) Lipschitz constant of the map $a \mapsto f_\theta(a,s)$

$$L_f := \sup_{s \in \mathcal{S}} \sup_{a, a' \in \mathcal{A}} \frac{|f_\theta(a,s) - f_\theta(a',s)|}{\|a - a'\|_1}.$$

Theorem 3. *Let $\mathcal{A} \subseteq [0,r]^d$ for some $r > 0$. Assume that*

$$c_1 := \sup_{a \in \mathcal{A}, s \in \mathcal{S}} |\mu(a,s)| < \infty, \qquad c_2 := \sup_{a,a' \in \mathcal{A}, s,s' \in \mathcal{S}} \Sigma(a,a') < \infty,$$

and let $c_3 := \left\| \sup_{a \in \mathcal{A}, s \in \mathcal{S}} |f_\theta(a,s)| \right\|_{2,P}$. If $\mathbb{E}[L_f^2] < \infty$, there is a universal constant $C > 1$ such that

$$\begin{aligned}\mathrm{BayesRegret}\left(T, \pi\right) \leq C\mathbb{E}[L_f] + Cc_2 + Cd\log(rd)\left(c_1\sqrt{\mathbb{E}[L_f]} + c_3\sqrt{\mathbb{E}[L_f^2]}\right) \\ + \left(T\gamma_T \frac{d\log T + d\log rd}{\log(1+\sigma^{-2})}\right)^{1/2} + \sum_{t=1}^{T} (c_3 + Cc_2 d\log(rdt))\sqrt{2\mathbb{E}\left[D_{\mathrm{kl}}\left(\pi_t^{\mathrm{TS}}, \pi_t \mid S_t\right)\right]}.\end{aligned}$$

See Section C.4 for the proof.

To instantiate Theorem 3, it remains to bound the smoothness of the reward function $\mathbb{E}[L_f^2]$, and the maximal information gain γ_T. Standard arguments from Gaussian process theory show $\mathbb{E}[L_f^2] < \infty$ holds whenever the mean and covariance function (kernel) is smooth, which holds for commonly used kernels.

Lemma 2 (Theorem 5, Ghosal et al. [33])**.** *If $\mu(\cdot)$ and $\Sigma(\cdot,\cdot)$ are 4 times continuously differentiable, then $(a,s) \mapsto f_\theta(a,s)$ is continuously differentiable and follows a Gaussian process again. In particular, $\mathbb{E}[L_f^2] < \infty$.*

To obtain concrete bounds on the maximal information gain γ_T, we use the results of Srinivas et al. [72], focusing on the popular Gaussian and Matern kernels

$$\begin{aligned}\Sigma_g(x,x') &:= \exp\left(-\frac{\|x - x'\|^2}{2l^2}\right), \\ \Sigma_m(x,x') &:= \frac{2^{1-\nu}}{\Gamma(\nu)} r^\nu B_\nu(r) \quad \text{where } r = \frac{\sqrt{2\nu}}{l}\left\|x - x'\right\|,\end{aligned}$$

where we used $B(\cdot)$ and $\Gamma(\cdot)$ to denote the Besel and Gamma functions respectively. To ease notation, we let κ denote the dimension of the underlying space, and define

$$\mathfrak{M}(\Sigma_g, T) := (\log T)^{\kappa+1} \quad \text{and} \quad \mathfrak{M}(\Sigma_m, T) := T^{\frac{\kappa^2+\kappa}{\kappa^2+\kappa+2\nu}} \log T.$$

We have the following bound on γ_T for Gaussian and Matern kernels; the bound is a direct consequence of Theorem 2, Krause and Ong [45] and Theorem 5, Srinivas et al. [72].

Lemma 3. *Let $\mathcal{A} \subseteq \mathbb{R}^d$ and $\mathcal{S} \subseteq \mathbb{R}^{d'}$ be convex and compact. Let the kernel Σ be given by the sum of two kernels Σ_A and Σ_S on $\mathcal{A}$ and $\mathcal{S}$ respectively*

$$\Sigma((a,s),(a',s')) = \Sigma_A(a,a') + \Sigma_S(s,s').$$

If Σ_A and Σ_S are either the Gaussian kernel Σ_g or the Matern kernel Σ_m with $\nu > 1$, then

$$\gamma_T = O\left(\mathfrak{M}(\Sigma_A, T) + \mathfrak{M}(\Sigma_S, T) + \log T\right).$$

For example, taking $\Sigma_A = \Sigma_g$ and $\Sigma_S = \Sigma_g$, we conclude

$$\text{BayesRegret}\left(T, \{\pi_t\}_{t\in\mathbb{N}}\right) = O\left(\sqrt{dT(\log T)^{\max\{d,d'\}+1}} + d\sum_{t=1}^{T}\log(rdt)\sqrt{\mathbb{E}\left[D_{\text{kl}}\left(\pi_t^{\text{TS}}, \pi_t \mid S_t\right)\right]}\right).$$

6 Empirical evaluation

We study the performance of our imitation learning algorithm in terms of cumulative regret / reward and decision-time latency in a number of datasets. Our imitation learning algorithm achieves a significant reduction in latency on all problems and enjoys regret comparable to that of TS, avoiding compounding of imitation error over time. Our experiments include a real-world video upload transcoding application for an internet service receiving millions of video upload requests per day.

Datasets We compare our imitation algorithm alongside an array of benchmark methods on four problem scenarios. For our first experiment, we study the **wheel bandit problem**, a synthetic problem constructed to require significant exploration [62]. In this two-dimensional problem, there are 5 actions and rarely seen contexts yield high rewards under one context-dependent action. We sample $T = 10,000$ contexts for each trial. Specifically, two-dimensional contexts are sampled in the unit sphere with uniform probability. The first action always has a mean reward of $\mathbb{E}[r(\boldsymbol{s}, a_1)] = 1.2$ independent of the context, and the mean rewards of the other actions depend on the context. If $||\boldsymbol{s}||_2 \leq \delta$, then the remaining four actions are non-optimal with a mean reward of 1. If $||\boldsymbol{s}||_2 > \delta$, then one of the remaining actions is optimal—and determined by the sign of the two dimensions of $\boldsymbol{s}$ —with a mean reward of 50. The remaining three actions all have a mean reward of 1. All rewards are observed with zero-mean additive Gaussian noise with standard deviation $\sigma = 0.01$. We set $\delta = 0.95$, which means the probability of sampling a context on the perimeter ($||\boldsymbol{s}||_2 \geq \delta$) where one action yields a large reward is $1 - (0.95)^2 = 0.0975 \approx 10\%$.

For our second problem, we design a contextual bandit problem from a supervised classification task. The **Mushroom UCI Dataset** contains 8,124 examples with 22 categorical features about the mushroom and labels indicating if the mushroom is poisonous or not. At each time step, the forager decides whether to eat the mushroom or not and receives a small positive reward for eating a safe mushroom, and a large negative reward for eating an unsafe mushroom. With equal probability, eating a poisonous mushroom lead to illness ($r = -35$) or it may not harm the consumer ($r = 5$), while a nonpoisonous mushroom always yields a positive reward ($r = 5$). The reward for abstaining is always 0. We sample $T = 50,000$ contexts for each trial.

Next, we turn our attention to a more realistic healthcare scenario, **pharamcological dosage optimization**, where we wish to learn a good dosing policy for Warfarin. Warfarin is one of the

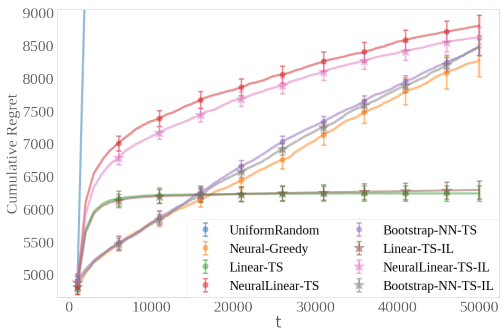
most common anticoagulants (blood thinner), often prescribed to patients with atrial fibrillation to prevent strokes [89]. The optimal dosage varies considerably across genetic, demographic, and clinical differences [12]. The Warfarin dataset [89] contains the optimal dosage of Warfarin for $4,788$ patients, which were found via trial and error by physicians. Using a 17-dimensional context vector on patient-specific demographics, medical history, and genetic markers, we construct a contextual bandit benchmark where the action space is a uniformly discretized dosage levels, and rewards are given by absolute deviation from the optimal dosage. We present results for 20 discretized dosage levels, but as we shown in Section D, we observe even bigger latency gains for 50 discretized dosage levels. We present results where we reshuffle contexts for each trial, but again find similar results when $T = 50,000$ contexts are re-sampled each trial.

Finally, we focus on a real-world **video upload transcoding application**, where we study a video upload system for a leading social network platform receiving millions of upload requests on *mobile devices*. Video is an increasingly popular medium on social networks, but uploading video is still a technically challenging problem, where limited bandwidth and compute capacity—particularly problematic on mobile devices—leads to unsuccessful uploads. Video needs to be optimally transcoded considering quality, and success of file upload. It is preferable to upload videos at a high quality because it can lead to a better viewer experience (if the viewer has a sufficiently good network connection). However, higher quality videos have larger file sizes, making it more likely to fail to upload: larger files take longer time to upload, increasing the likelihood that the network connection to fail, or the user to grow frustrated and cancel the upload.

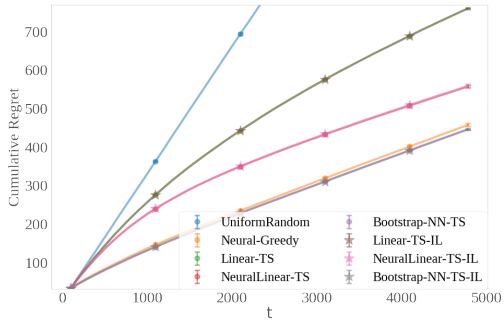
We are interested in a web service who wish to make contextual decisions about how to optimally transcode a video at upload time. Although transcoding decisions needs to be made quickly in order to be responsive and keep the user engaged, most upload requests come from resource-constrained mobile devices. We have access to a 38-dimensional context representing information about the video file (e.g. the raw bitrate, resolution, and file size) and the network connection (e.g. connection type, download bandwidth, country). There are 7 actions corresponding to a unique (resolution, bitrate) pairs. The actions are ranked ordered in terms of quality: action i yields a video with higher quality than action j if and only if $i \geq j$. If successful, the reward for a successful upload is a positive and monotonically increasing function of the action. The reward for a failed upload is 0.

We evaluate the performance of different contextual bandit algorithms using the unbiased, offline, policy evaluation technique proposed by Li et al. [48]. The method evaluates a contextual bandit algorithm by performing rejection sampling on a stream of logged observation tuples of the form (S_t, A_t, R_t) collected under a uniform random policy. Specifically, the observed tuple is rejected if the logged action does not match the action selected by the algorithm being evaluated. Our dataset contains 8 million observations logged under a uniform random policy. We evaluate each algorithm using the stream of logged data until each algorithm has "observed" $T = 50,000$ *valid* time steps.

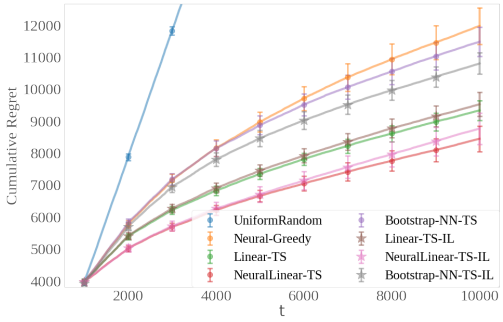
Algorithms and evaluation For all experiments, we consider models previously found to perform the best in a broad range of benchmark problems, as reported by Riquelme et al. [62] in their extensive empirical experiments. **Linear-TS** uses an exact Bayesian linear regression to model the reward distribution for each action a independently. This policy evaluates the exact posterior under the assumption that the data for action a were generated from the linear function: $r_a = \boldsymbol{s}^T \boldsymbol{\theta}_a + \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, \sigma_a^2)$. For each action, we independently model the joint distribution, $P(\boldsymbol{\theta}, \sigma^2) = P(\boldsymbol{\theta}|\sigma^2)P(\sigma^2)$ as a normal-inverse-gamma distribution which allows for tractable posterior inference (see Appendix D for closed form expressions). **NeuralLinear-TS** models rewards using a neural



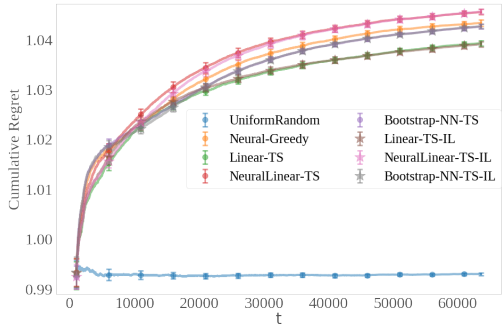
(a) Cumulative regret on Mushroom dataset



(b) Cumulative regret on Warfarin dataset



(c) Cumulative regret on Wheel bandit



(d) Running average of rewards for video transcoding

Figure 1. We report mean cumulative regret (or running average of rewards for video transcoding), alongside two standard errors over 50 trials (100 trials for the Wheel bandit, due to rarity of large rewards).

network with two 100-unit hidden layers and ReLU activations, but discards the last linear layer and uses the last hidden layer $\boldsymbol{\phi}(\boldsymbol{s})$ as the feature representation for a Linear-TS policy. The neural network takes the context as input, and predicts the reward for each action. The parameters of the neural network are shared for all actions and are learned independently of the Bayesian linear models. **Bootstrap-NN-TS** trains multiple neural networks on bootstrapped observations and randomly samples a single network to use for each decision. For all of the aforementioned TS policies, **TS-IL** denotes their imitated counterpart. We use a fully-connected neural network to parameterize the policy π^m in the imitation learning problem (1). The policy representation has two hidden layers with 100 units each, hyperbolic tangent activations on the hidden layers, and a soft-max activation on the output layer to predict a the conditional distribution $P(a|\boldsymbol{s})$ for all $a \in \mathcal{A}$. We compare Thompson sampling and its imitation counterparts against two additional benchmarks: a random policy (**UniformRandom**) and a greedy policy that uses a feed-froward neural network to model rewards (**Neural-Greedy**).

Policies are updated every 1000 time steps (except for the Warfarin problem, where we use update policies every 100 time steps due to the small size of the dataset) and are initialized using a uniform random policy before the first batch update. We detail our hyperparameter choices in Section D: following extensive evaluations by Riquelme et al. [62], we use their proposed settings for Thompson sampling.

In Figure 1, we show that each TS-IL method achieves performance comparable to its corresponding vanilla TS algorithm on all benchmark problems. We evaluate the cumulative performance at

Table 1. Decision-making latency in milliseconds. All latency measurements were made on a Intel Xeon E5-2680 v4 @ 2.40GHz CPU with 32-bit floating point precision. For each latency measurement, action generation is repeated 100K times and the mean latency and its 2-standard errors are reported.

	Mushroom	Wheel	Video Transcode	Warfarin
UniformRandom	0.040 (±0.000)	0.039 (±0.000)	0.040 (±0.000)	0.040 (±0.000)
Neural-Greedy	0.242 (±0.001)	0.228 (±0.001)	0.231 (±0.001)	0.232 (±0.000)
Linear-TS	0.715 (±0.001)	1.142 (±0.001)	1.575 (±0.002)	3.963 (±0.002)
NeuralLinear-TS	0.826 (±0.001)	1.492 (±0.001)	1.931 (±0.002)	4.814 (±0.004)
Bootstrap-NN-TS	0.235 (±0.001)	0.235 (±0.001)	0.236 (±0.001)	0.226 (±0.001)
Linear-TS-IL	0.184 (±0.001)	0.178 (±0.000)	0.169 (±0.000)	0.175 (±0.000)
NeuralLinear-TS-IL	0.186 (±0.000)	0.179 (±0.001)	0.169 (±0.000)	0.175 (±0.000)
Bootstrap-NN-TS-IL	0.190 (±0.001)	0.178 (±0.000)	0.175 (±0.000)	0.179 (±0.001)

time steps along the entire learning curve, and observe that each TS-IL policy consistently matches its corresponding TS policy over time.

(Approximate) Bayesian inference often requires a substantial amount of compute and memory. We evaluate decision-time latency and time complexity for the specific models being considered, but note that the latency and complexity may be even greater under inference schemes not considered here. We define *decision time latency* as the time required for a policy to select an action when it is queried. While Bootstrap-NN-TS achieves low prediction latency, it requires storing many replicates of the neural network and can significantly increase the memory footprint. On low-end mobile devices, such memory requirements can be prohibitive, limiting the applicability of methods based on bootstrapping; our imitation methods offer a practical and effective alternative.

Table 1 shows that the imitation policies (TS-IL) have significantly lower decision time latency compared to TS algorithms, often by *over an order of magnitude* on problems with larger action spaces (Warfarin and video upload transcoding). This is because generating an action under the vanilla TS policies requires drawing a sample from the joint posterior $P(\boldsymbol{\theta}_a, \sigma_a^2)$ for each of the actions a, which is quadratic with respect to the context dimension for Linear-TS or the size of the last hidden layer for NeuralLinear-TS. On the other hand, TS-IL simply requires a forward propagation through the policy network and a sample from multinomial sample, both of which are exceedingly cheap. In Section E, we provide a detailed discussion of time and space complexity, including those for alternative model choices.

7 Discussion

In this paper, we used imitation learning to operationalize Thompson sampling, allowing it to scale to applications where latency and software complexity are of core concern. We demonstrated that imitation learning provides a simple, practical, and efficient method with desirable regret properties. By distilling the Thompson sampling policy into easy-to-deploy explicit policy representations (e.g. neural networks), we allow state-of-the-art Bayesian approaches to be used in contextual bandit problems. We hope that this work facilitates applications of modern deep Bayesian approaches to large-scale contextual bandit problems.

While we have empirically evaluated two types of Bayesian models, our framework is compatible with any type of probabilistic model. For example, practitioners may utilize domain knowledge to develop grey-box models (see e.g., Schwartz et al. [69]). Such models, while simple to implement in

probabilistic programming languages [20, 15, 78], would otherwise be challenging and inefficient to deploy; our imitation framework can allow ease of deployment for these models, while maintaining a comparable level of performance.

In internet applications, platforms interact with users asynchronously, and policies are updated using a large batch of observations. A careful theoretical understanding of our algorithm in this setting, including adjusting exploration behavior to depend on the batch size, is an open area of research. To our knowledge, in contextual scenarios such understanding remains to be developed even for Thompson sampling. Nonstationarity and satisficing are other important areas of research, where a distillation approach may provide practical advantages.

While we restricted attention to contextual bandits problems, an interesting research direction is to extend these methods to combinatorial ranking problems [23, 27], where computational savings of distillation may be even larger. Extending our imitation framework to reinforcement learning problems will also likely yield fruit.

References

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

[2] Y. Abbasi-Yadkori, D. Pal, and C. Szepesvari. Online-to-confidence-set conversions and application to sparse stochastic bandits. In *Artificial Intelligence and Statistics*, pages 1–9, 2012.

[3] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, page 1, 2004.

[4] M. Abeille, A. Lazaric, et al. Linear thompson sampling revisited. *Electronic Journal of Statistics*, 11(2):5165–5197, 2017.

[5] R. J. Adler and J. E. Taylor. *Random fields and geometry*, volume 115. Springer, 2009.

[6] A. Agarwal, S. Bird, M. Cozowicz, L. Hoang, J. Langford, S. Lee, J. Li, D. Melamed, G. Oshri, O. Ribas, et al. Making contextual decisions with low technical debt. *arXiv preprint arXiv:1606.03966*, 2016.

[7] D. Agarwal, B. Long, J. Traupman, D. Xin, and L. Zhang. Laser: A scalable response prediction platform for online advertising. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 173–182. ACM, 2014.

[8] S. Agrawal and N. Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In *Proceedings of the Twenty Fifth Annual Conference on Computational Learning Theory*, 2012.

[9] S. Agrawal and N. Goyal. Further optimal regret bounds for thompson sampling. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, pages 99–107, 2013.

[10] S. Agrawal and N. Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning*, pages 127–135, 2013.

[11] P. L. Bartlett, O. Bousquet, and S. Mendelson. Local Rademacher complexities. *Annals of Statistics*, 33(4):1497–1537, 2005.

[12] H. Bastani and M. Bayati. Online decision-making with high-dimensional covariates. *SSRN Electronic Journal*, 01 2015. doi: 10.2139/ssrn.2661896.

[13] A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, 2004.

[14] K. Bhardwaj, C.-Y. Lin, A. Sartor, and R. Marculescu. Memory- and communication-aware model compression for distributed deep learning inference on iot. *ACM Trans. Embed. Comput. Syst.*, 18(5s), Oct. 2019. doi: 10.1145/3358205.

[15] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*, 2018.

[16] M. Birman and M. Solomjak. Piecewise-polynomial approximations of functions of the classes W_p^α. *Sbornik: Mathematics*, 2(3):295–317, 1967.

[17] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622, 2015.

[18] J. F. Bonnans and A. Shapiro. *Perturbation Analysis of Optimization Problems*. Springer, 2000.

[19] S. Bubeck and R. Eldan. Multi-scale exploration of convex functions and bandit convex optimization. In *Conference on Learning Theory*, pages 583–589, 2016.

[20] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

[21] O. Chapelle and L. Li. An empirical evaluation of thompson sampling. In *Advances in Neural Information Processing Systems 24*, pages 2249–2257, 2011.

[22] T. Chen, E. Fox, and C. Guestrin. Stochastic gradient hamiltonian monte carlo. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, 2014.

[23] W. C. Cheung, V. Y. F. Tan, and Z. Zhong. Thompson sampling for cascading bandits. *CoRR*, 2018.

[24] C. Coleman, D. Narayanan, D. Kang, T. Zhao, J. Zhang, L. Nardi, P. Bailis, K. Olukotun, C. Ré, and M. Zaharia. Dawnbench: An end-to-end deep learning benchmark and competition. *Training*, 100(101):102, 2017.

[25] N. Cristianini and J. Shawe-Taylor. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[26] V. Dani, T. Hayes, and S. Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the Twenty First Annual Conference on Computational Learning Theory*, 2008.

[27] M. Dimakopoulou, N. Vlassis, and T. Jebara. Marginal posterior sampling for slate bandits. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 2019.

[28] J. C. Duchi. Introductory lectures on stochastic convex optimization. In *The Mathematics of Data*, IAS/Park City Mathematics Series. American Mathematical Society, 2018.

[29] D. Eckles and M. Kaptein. Thompson sampling with the online bootstrap. *arXiv preprint arXiv:1410.4009*, 2014.

[30] K. J. Ferreira, D. Simchi-Levi, and H. Wang. Online network revenue management using thompson sampling. *Operations Research*, 66(6):1586–1602, 2018.

[31] S. Fujimoto, D. Meger, and D. Precup. Off-policy deep reinforcement learning without exploration. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2052–2062, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[32] J. Gauci, E. Conti, Y. Liang, K. Virochsiri, Y. He, Z. Kaden, V. Narayanan, X. Ye, Z. Chen, and S. Fujimoto. Horizon: Facebook's open source applied reinforcement learning platform. *arXiv:1811.00260 [cs.LG]*, 2018.

[33] S. Ghosal, A. Roy, et al. Posterior consistency of gaussian process prior for nonparametric binary regression. *Annals of Statistics*, 34(5):2413–2429, 2006.

[34] A. Gopalan, S. Mannor, and Y. Mansour. Thompson sampling for complex online problems. In *Proceedings of the 31st International Conference on Machine Learning*, pages 100–108, 2014.

[35] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[36] O.-C. Granmo. Solving two-armed bernoulli bandit problems using a bayesian learning automaton. *International Journal of Intelligent Computing and Cybernetics*, 3(2):207–234, 2010.

[37] C. Gu. *Smoothing spline ANOVA models*. Springer, 2002.

[38] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645, 2016.

[39] J. Ho and S. Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems 29*, pages 4565–4573, 2016.

[40] J. Honda and A. Takemura. Optimality of thompson sampling for gaussian bandits depends on priors. In *Artificial Intelligence and Statistics*, pages 375–383, 2014.

[41] E. Kaufmann, N. Korda, and R. Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *International Conference on Algorithmic Learning Theory*, pages 199–213. Springer, 2012.

[42] J. Kawale, H. H. Bui, B. Kveton, L. Tran-Thanh, and S. Chawla. Efficient thompson sampling for online matrix factorization recommendation. In *Advances in Neural Information Processing Systems 15*, pages 1297–1305, 2015.

[43] V. Koltchinskii. Local Rademacher complexities and oracle inequalities in risk minimization. *Annals of Statistics*, 34(6):2593–2656, 2006.

[44] A. Korattikara, V. Rathod, K. Murphy, and M. Welling. Bayesian dark knowledge. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 3438–3446, Cambridge, MA, USA, 2015. MIT Press.

[45] A. Krause and C. S. Ong. Contextual gaussian process bandit optimization. In *Advances in Neural Information Processing Systems 24*, pages 2447–2455, 2011.

[46] T. Kühn. Covering numbers of Gaussian reproducing kernel Hilbert spaces. *Journal of Complexity*, 27(5):489–499, 2011.

[47] H. J. Kushner and G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, second edition, 2003.

[48] L. Li, W. Chu, J. Langford, and X. Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, page 297–306. Association for

Computing Machinery, 2011. doi: 10.1145/1935826.1935878.

[49] Y. Li, J. Song, and S. Ermon. Infogail: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems 30*, pages 3812–3822, 2017.

[50] C.-Y. Liu and L. Li. On the prior sensitivity of thompson sampling. In *International Conference on Algorithmic Learning Theory*, pages 321–336. Springer, 2016.

[51] X. Lu and B. Van Roy. Ensemble sampling. In *Advances in neural information processing systems*, pages 3258–3266, 2017.

[52] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems 32*, 2019.

[53] H. Mao, S. Chen, D. Dimmery, S. Singh, D. Blaisdell, Y. Tian, M. Alizadeh, and E. Bakshy. Real-world video adaptation with reinforcement learning. In *Neural Information Processing Systems Workshop on RL for Real Life*, 2019.

[54] A. Mas-Colell, M. D. Whinston, J. R. Green, et al. *Microeconomic theory*, volume 1. Oxford university press New York, 1995.

[55] B. C. May and D. S. Leslie. Simulation studies in optimistic bayesian sampling in contextual-bandit problems. *Technical Report, Statistics Group, Department of Mathematics, University of Bristol*, 11:02, 2011.

[56] S. Mendelson. On the performance of kernel classes. *Journal of Machine Learning Research*, 4 (Oct):759–771, 2003.

[57] R. M. Neal. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2, 2011.

[58] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy. Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems*, pages 4026–4034, 2016.

[59] A. Petrescu and S. Tas. Client side ranking to more efficiently show people stories in feed, Oct 2016. URL `https://engineering.fb.com/networking-traffic/client-side-ranking-to-more-efficiently-show-people-stories-in-feed/`.

[60] G. Peyré, M. Cuturi, et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[61] V. Ricciardi. Reddit comments, 2019. URL `https://blogs.worldbank.org/opendata/are-cell-phones-becoming-more-popular-toilets#:~:text=The%20percentage%20of%20people%20with,trend%20consistent%20across%20world%20regions`.

[62] C. Riquelme, G. Tucker, and J. Snoek. Deep bayesian bandits showdown. In *International Conference on Learning Representations*, 2018.

[63] S. Ross and D. Bagnell. Efficient reductions for imitation learning. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pages 661–668, 2010.

[64] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 627–635, 2011.

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

[66] D. Russo and B. Van Roy. Learning to optimize via posterior sampling. *Mathematics of*

Operations Research, 39(4):1221–1243, 2014.

[67] D. Russo and B. Van Roy. An information-theoretic analysis of thompson sampling. *Journal of Machine Learning Research*, 17(1):2442–2471, 2016.

[68] D. J. Russo, B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018.

[69] E. M. Schwartz, E. T. Bradlow, and P. S. Fader. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science*, 36(4):500–522, 2017.

[70] S. L. Scott. A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658, 2010.

[71] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems 28*, pages 2503–2511, 2015.

[72] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[73] Statistica. Percentage of all global web pages served to mobile phones from 2009 to 2018, 2020. URL `https://www-statista-com/statistics/241462/global-mobile-phone-website-traffic-share/`.

[74] U. Syed and R. E. Schapire. A game-theoretic approach to apprenticeship learning. In *Advances in Neural Information Processing Systems 22*, pages 1449–1456, 2008.

[75] U. Syed and R. E. Schapire. A reduction from apprenticeship learning to classification. In *Advances in Neural Information Processing Systems 23*, pages 2253–2261, 2010.

[76] K. T. Talluri and G. Van Ryzin. *The theory and practice of revenue management*, volume 1. Springer, 2004.

[77] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

[78] D. Tran, M. W. Hoffman, D. Moore, C. Suter, S. Vasudevan, and A. Radul. Simple, distributed, and accelerated probabilistic programming. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7598–7609. Curran Associates, Inc., 2018. URL `http://papers.nips.cc/paper/7987-simple-distributed-and-accelerated-probabilistic-programming.pdf`.

[79] A. B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

[80] A. W. van der Vaart and J. A. Wellner. *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer, New York, 1996.

[81] C. Villani. *Optimal Transport: Old and New*. Springer, 2009.

[82] M. J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019.

[83] H. Wang and D.-Y. Yeung. A survey on bayesian deep learning. *ACM Computing Surveys (CSUR)*, 53(5):1–37, 2020.

[84] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

[85] A. Wilson and H. Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning*, pages 1775–1784, 2015.

[86] A. G. Wilson, C. Dann, and H. Nickisch. Thoughts on massively scalable gaussian processes. *arXiv:1511.01870 [cs.LG]*, 2015.

[87] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378, 2016.

[88] C.-J. Wu, D. Brooks, K. Chen, D. Chen, S. Choudhury, M. Dukhan, K. Hazelwood, E. Isaac, Y. Jia, B. Jia, T. Leyvand, H. Lu, Y. Lu, L. Qiao, B. Reagen, J. Spisak, F. Sun, A. Tulloch, P. Vajda, X. Wang, Y. Wang, B. Wasti, Y. Wu, R. Xian, S. Yoo, and P. Zhang. Machine learning at facebook: Understanding inference at the edge. In *Proceedings - 25th IEEE International Symposium on High Performance Computer Architecture, HPCA 2019*, pages 331–344, 3 2019. doi: 10.1109/HPCA.2019.00048.

[89] H. Xiao. Online learning to estimate warfarin dose with contextual linear bandits. *CoRR*, 2019.

[90] G. Zhang, S. Sun, D. Duvenaud, and R. Grosse. Noisy natural gradient as variational inference. In *International Conference on Machine Learning*, pages 5852–5861, 2018.

A Imitation learning with Wasserstein distances

When actions can be naturally embedded in a continuous space, we may want to measure closeness between the imitation and TS policy by incorporating the geometry of the action space $\mathcal{A}$. In this section, we provide an alternative instantiation of the abstract form of Algorithm 1 by using Wasserstein distances as the notion of discrepancy $D(\cdot, \cdot \mid s)$. Our previous theoretical development for KL divergences has direct analogues in this setting, which we now briefly outline.

Given a metric $d(\cdot, \cdot)$ on $\mathcal{A}$, the Wasserstein distance between two distributions q^1 and q^2 on $\mathcal{A}$ is defined by the optimal transport problem

$$D_{\mathrm{w}}\left(q^1, q^2\right) = \inf_{\eta \in L(q^1, q^2)} \mathbb{E}_\eta[d(A, A')]$$

where $\eta(q^1, q^2)$ denotes the collection of all probabilities on $\mathcal{A} \times \mathcal{A}$ with marginals q^1 and q^2 (i.e., couplings). Intuitively, $D_{\mathrm{w}}\left(q^1, q^2\right)$ measures how much cost $d(A, A')$ is incurred by moving mass away from $A \sim q^1$ to $A' \sim q^2$ in an optimal fashion[5]. Wasserstein distances encode the geometry of the underlying space $\mathcal{A}$ via the distance d. Unlike the KL divergence $D_{\mathrm{kl}}\left(q^1 \| q^2\right)$ that take value ∞ whenever q^1 has support not contained in q^2, Wasserstein distance allows imitation policies to have different support than the Thompson sampling policy, which is more appropriate in continuous action spaces. To simplify notation, for two policies π^1 and π^2, we let

$$D_{\mathrm{w}}\left(\pi^1, \pi^2 \mid S\right) := D_{\mathrm{w}}\left(\pi^1(\cdot \mid S), \pi^2(\cdot \mid S)\right).$$

When Algorithm 1 is instantiated with the Wasserstein distance as its notion of discrepancy $D(\cdot, \cdot \mid S) := D_{\mathrm{w}}(\cdot, \cdot \mid S)$, the imitation learning problem (1) becomes

$$\underset{m \in \mathcal{M}}{\text{minimize}}\, \mathbb{E}_{S \sim \mathbb{P}_S}\left[D_{\mathrm{w}}\left(\pi^{\mathrm{TS}}, \pi^m \mid S\right)\right]. \tag{9}$$

To solve the above stochastic optimization problem, we can again use stochastic gradient descent methods, where the stochastic gradient $\nabla_m D_{\mathrm{w}}\left(\pi_t^{\mathrm{TS}}, \pi^m \mid S\right)$ can be computed by solving an optimal transport problem. From Kantorovich-Rubinstein duality (see, for example, [81]), we have

$$\begin{aligned} &D_{\mathrm{w}}\left(\pi_t^{\mathrm{TS}}, \pi^m \mid s\right) \\ &= \sup_{g: \mathcal{A} \to \mathbb{R}} \left\{\mathbb{E}_{A \sim \pi^{\mathrm{TS}}(\cdot \mid s)} g(a) - \mathbb{E}_{A \sim \pi^m(\cdot \mid s)} g(a) :\ g(a) - g(a') \le d(a, a') \text{ for all } a, a' \in \mathcal{A}\right\}, \end{aligned} \tag{10}$$

where $d(\cdot, \cdot)$ is the metric on $\mathcal{A}$ used to define $D_{\mathrm{w}}(\cdot, \cdot)$. For discrete action spaces, the maximization problem (10) is a linear program with $O(|\mathcal{A}|)$ variables and constraints; for continuous action spaces, we can solve the problem over empirical distributions to approximate the optimal transport problem. We refer the interested reader to Peyré et al. [60] for a comprehensive introduction to computational methods for solving optimal transport problems.

Letting $g^\star$ denote the optimal solution to the dual problem (10), the envelope theorem (or Danskin's theorem; Bonnans and Shapiro [18, Theorem 4.13]) implies that under simple regularity conditions

$$\nabla_m D_{\mathrm{w}}\left(\pi_t^{\mathrm{TS}}, \pi^m \mid s\right) = -\nabla_m \mathbb{E}_{A \sim \pi^m(\cdot \mid s)}[g^\star(a)].$$

Assuming that an appropriate change of gradient and expectation is justified, we can use the policy gradient trick to arrive at

$$-\nabla_m \mathbb{E}_{A \sim \pi^m(\cdot \mid s)}[g^\star(A)] = -\mathbb{E}_{A \sim \pi^m(\cdot \mid s)}[g^\star(A) \nabla_m \log \pi^m(A \mid s)].$$

[5]For a discrete action space, $D_{\mathrm{w}}(\cdot, \cdot)$ can be defined with any symmetric matrix $d(a_i, a_j)$ satisfying $d(a_i, a_j) \ge 0$ with 0 iff $a_i = a_j$, and $d(a_i, a_j) \le d(a_i, a_k) + d(a_k, a_j)$ for any $a_i, a_j, a_k \in \mathcal{A}$.

We conclude that for $A \sim \pi^m(\cdot \mid S_i)$,

$$-g^\star(A)\nabla_m \log \pi^m(A \mid S_i) \tag{11}$$

is a stochastic gradient for the imitation problem (9). As before, we can get lower variance estimates of the gradient by averaging the above estimator over many actions $A \sim \pi^m(\cdot \mid S_i)$. Using these stochastic gradients (11), we can solve the imitation problem (9) efficiently.

We now show that the resulting imitation policy admits a regret decomposition similar to Lemma 1 for KL divergences. As a direct consequence of this decomposition, the regret bounds in Section 5.2 have their natural analogues with Wasserstein distances replacing KL divergences as the notion of discrepancy, though we omit them for brevity.

Lemma 4. *Let $\pi = \{\pi_t\}_{t\in\mathbb{N}}$ be any set of policies, and let $U_t(\cdot; H_t, S_t) : \mathcal{A} \to \mathbb{R}$ be any upper confidence bound sequence that is measurable with respect to $\sigma(H_t, S_t, A_t)$. For some sequence $M_t(H_t, S_t)$ and a constant $L > 0$, let U_t satisfy*

$$|U_t(a; H_t, S_t) - U_t(a'; H_t, S_t)| \le L d(a, a') \text{ for all } a, a' \in \mathcal{A} \text{ almost surely.} \tag{12}$$

Then for all $T \in \mathbb{N}$,

$$\begin{aligned}\text{BayesRegret}\left(T, \{\pi_t\}_{t\in\mathbb{N}}\right) \le &\sum_{t=1}^{T} \mathbb{E}[f_\theta(A_t^\star, S_t) - U_t(A_t^\star; H_t, S_t)] + \sum_{t=1}^{T} \mathbb{E}[U_t(A_t; H_t, S_t) - f_\theta(A_t, S_t)] \\ &+ L \sum_{t=1}^{T} \mathbb{E}\left[D_{\mathrm{w}}\left(\pi_t^{\mathrm{TS}}, \pi_t \mid S_t\right)\right]. \end{aligned} \tag{13}$$

where $D_{\mathrm{w}}(\cdot, \cdot \mid \cdot)$ is the Wasserstein distance defined with the metric d in the condition (12).

Proof The proof mirrors that of Lemma 1, but bound the differences (16) by $D_{\mathrm{w}}\left(\pi_t^{\mathrm{TS}}, \pi_t \mid S_t\right)$. By the Kantorovich dual representation (10), we have

$$\mathbb{E}[|U_t(A_t^{\mathrm{TS}}; H_t, S_t) - U_t(A_t; H_t, S_t)| \mid H_t, S_t] \le M_t(H_t, S_t) D_{\mathrm{w}}\left(\pi_t^{\mathrm{TS}}, \pi_t \mid S_t\right).$$

Applying this bound in the decomposition (15), and taking expectation over (H_t, S_t) on both sides and summing $t = 1, \ldots, T$, we get the desired bound. $\diamond$

B Proof of generalization results

B.1 Proof of Theorem 1

We abuse notation and use $C > 0$ to denote a numerical constant that changes value line to line. We use the following concentration guarantee using localized Rademacher averages.

Lemma 5 (Bartlett et al. [11, Theorem 3.3])**.** *For a class of functions $\mathcal{G}$ with range $[0, M]$, let $r_n^\star$ be the unique positive fixed point of the sub-root function ψ_n satisfying the bound* (5)*. Then, for i.i.d. observations $\xi \overset{\text{iid}}{\sim} \mathbb{P}$, there is a numerical constant $C > 0$ such that*

$$\mathbb{E}[g] \le \left(1 + \frac{1}{\eta}\right) \frac{1}{n} \sum_{i=1}^{n} g(\xi_i) + C(1+\eta)\left(\frac{1}{M} r_n^\star + \frac{Mt}{n}\right) + \frac{CMt}{n} \quad \text{for all } g \in \mathcal{G} \text{ and } \eta \ge 0$$

with probability at least $1 - e^{-t}$.

Notice that by Jensen inequality, we have

$$\mathbb{E}_{A^{\mathrm{TS}}\sim\pi^{\mathrm{TS}}(\cdot|S_i)}\left[\log\frac{\pi^{\mathrm{TS}}(A^{\mathrm{TS}}\mid S_i)}{\pi^{\widehat{m}}(A^{\mathrm{TS}}\mid S_i)}\right]\geq 0 \ \text{ almost surely.}$$

Applying Lemma 5 with the function class $\mathcal{G}_1$ and $\eta = 1/2$, we have

$$\mathbb{E}\left[D_{\mathrm{kl}}\left(\pi^{\mathrm{TS}},\pi^{\widehat{m}}\mid S\right)\right]\leq\frac{3}{2}\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{A^{\mathrm{TS}}\sim\pi^{\mathrm{TS}}(\cdot|S_i)}\left[\log\frac{\pi^{\mathrm{TS}}(A^{\mathrm{TS}}\mid S_i)}{\pi^{\widehat{m}}(A^{\mathrm{TS}}\mid S_i)}\right]+\frac{C}{M}r_N^\star+\frac{CMt}{N}. \tag{14}$$

In the rest of the proof, we bound the interim (uniform) approximation error

$$Z_{N,N_a}:=\sup_{m\in\mathcal{M}}\left\{\frac{1}{N}\sum_{i=1}^{N}\left(\mathbb{E}_{A^{\mathrm{TS}}\sim\pi^{\mathrm{TS}}(\cdot|S_i)}\left[\log\frac{\pi^{\mathrm{TS}}(A^{\mathrm{TS}}\mid S_i)}{\pi^{m}(A^{\mathrm{TS}}\mid S_i)}\right]-\frac{1}{N_a}\sum_{j=1}^{N_a}\log\frac{\pi^{\mathrm{TS}}(A_{ij}^{\mathrm{TS}}\mid S_i)}{\pi^{m}(A_{ij}^{\mathrm{TS}}\mid S_i)}\right)\right\}.$$

This is indeed sufficient for our purposes since the bound (14) implies

$$\mathbb{E}\left[D_{\mathrm{kl}}\left(\pi^{\mathrm{TS}},\pi^{\widehat{m}}\mid S\right)\right]\leq\frac{3}{2}\frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_a}\log\frac{\pi^{\mathrm{TS}}(A_{ij}^{\mathrm{TS}}\mid S_i)}{\pi^{\widehat{m}}(A_{ij}^{\mathrm{TS}}\mid S_i)}+C\left(Z_{N,N_a}+\frac{1}{M}r_N^\star+\frac{Mt}{N}\right).$$

By the definition (3) of the empirical solution $\widehat{m}$ and by virtue of having a well-specified model class $\mathcal{M}$, the first term in the preceding bound is nonpositive

$$\frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_a}\log\frac{\pi^{\mathrm{TS}}(A_{ij}^{\mathrm{TS}}\mid S_i)}{\pi^{\widehat{m}}(A_{ij}^{\mathrm{TS}}\mid S_i)}\leq\frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_a}\log\frac{\pi^{\mathrm{TS}}(A_{ij}^{\mathrm{TS}}\mid S_i)}{\pi^{m^\star}(A_{ij}^{\mathrm{TS}}\mid S_i)}=0.$$

Consider the Doob martingale $M_0=\mathbb{E}[Z_{N,N_a}]$, and

$$M_k:=\mathbb{E}[Z_{N,N_a}\mid S_1,\ldots,S_k]\quad\text{for}\quad 1\leq k\leq N,$$

a martingale adapted to the filtration $\mathcal{F}_k:=\sigma(S_1,\ldots,S_k)$. Denote the martingale difference sequence $D_k=M_k-M_{k-1}$ for $k\geq 1$. Let $\bar{S}_k$ be an independent copy of S_k that is independent of all S_i, and let $\bar{A}_{kj}^{\mathrm{TS}}\sim\pi^{\mathrm{TS}}(\cdot\mid\bar{S}_k)$ independent of everything other than $\bar{S}_k$. We can write

$$\begin{aligned}D_k=\mathbb{E}&\left[\sup_{m\in\mathcal{M}}\left\{\frac{1}{N}\sum_{i=1}^{N}\left(\mathbb{E}_{A^{\mathrm{TS}}\sim\pi^{\mathrm{TS}}(\cdot|S_i)}\left[\log\frac{\pi^{\mathrm{TS}}(A^{\mathrm{TS}}\mid S_i)}{\pi^{m}(A^{\mathrm{TS}}\mid S_i)}\right]-\frac{1}{N_a}\sum_{j=1}^{N_a}\log\frac{\pi^{\mathrm{TS}}(A_{ij}^{\mathrm{TS}}\mid S_i)}{\pi^{m}(A_{ij}^{\mathrm{TS}}\mid S_i)}\right)\right\}\mid S_1,\ldots,S_k\right]\\
&-\mathbb{E}\left[\sup_{m\in\mathcal{M}}\left\{\frac{1}{N}\sum_{i\neq k}\left(\mathbb{E}_{A^{\mathrm{TS}}\sim\pi^{\mathrm{TS}}(\cdot|S_i)}\left[\log\frac{\pi^{\mathrm{TS}}(A^{\mathrm{TS}}\mid S_i)}{\pi^{m}(A^{\mathrm{TS}}\mid S_i)}\right]-\frac{1}{N_a}\sum_{j=1}^{N_a}\log\frac{\pi^{\mathrm{TS}}(A_{ij}^{\mathrm{TS}}\mid S_i)}{\pi^{m}(A_{ij}^{\mathrm{TS}}\mid S_i)}\right)\right.\right.\\
&\qquad\left.\left.+\frac{1}{N}\left(\mathbb{E}_{\bar{A}^{\mathrm{TS}}\sim\pi^{\mathrm{TS}}(\cdot|\bar{S}_k)}\left[\log\frac{\pi^{\mathrm{TS}}(\bar{A}^{\mathrm{TS}}\mid\bar{S}_k)}{\pi^{m}(\bar{A}^{\mathrm{TS}}\mid\bar{S}_k)}\right]-\frac{1}{N_a}\sum_{j=1}^{N_a}\log\frac{\pi^{\mathrm{TS}}(\bar{A}_{ij}^{\mathrm{TS}}\mid\bar{S}_k)}{\pi^{m}(\bar{A}_{ij}^{\mathrm{TS}}\mid\bar{S}_k)}\right)\right\}\right|S_1,\ldots,S_k\right].\end{aligned}$$

Thus, we arrive at the bound independence of S_i's yields

$$|D_k| \le \frac{1}{N}\mathbb{E}\Bigg[\sup_{m\in\mathcal{M}}\Bigg|\frac{1}{N_a}\sum_{j=1}^{N_a}\mathbb{E}_{A^{\mathrm{TS}}\sim\pi^{\mathrm{TS}}(\cdot|S_k)}\left[\log\frac{\pi^{\mathrm{TS}}(A^{\mathrm{TS}}\mid S_k)}{\pi^m(A^{\mathrm{TS}}\mid S_k)}\right]-\log\frac{\pi^{\mathrm{TS}}(A_{ij}^{\mathrm{TS}}\mid S_k)}{\pi^m(A_{ij}^{\mathrm{TS}}\mid S_k)}$$
$$-\mathbb{E}_{\bar{A}^{\mathrm{TS}}\sim\pi^{\mathrm{TS}}(\cdot|\bar{S}_k)}\left[\log\frac{\pi^{\mathrm{TS}}(\bar{A}^{\mathrm{TS}}\mid \bar{S}_k)}{\pi^m(\bar{A}^{\mathrm{TS}}\mid \bar{S}_k)}\right]+\log\frac{\pi^{\mathrm{TS}}(\bar{A}_{ij}^{\mathrm{TS}}\mid \bar{S}_k)}{\pi^m(\bar{A}_{ij}^{\mathrm{TS}}\mid \bar{S}_k)}\Bigg|\mid S_k\Bigg]$$
$$\le \frac{2}{N}\sup_{s\in\mathcal{S}}\mathbb{E}_{A_j^{\mathrm{TS}}\overset{\mathrm{iid}}{\sim}\pi^{\mathrm{TS}}(\cdot|s)}\left[\sup_{m\in\mathcal{M}}\left|\frac{1}{N_a}\sum_{j=1}^{N_a}\mathbb{E}_{A^{\mathrm{TS}}\sim\pi^{\mathrm{TS}}(\cdot|s)}\left[\log\frac{\pi^{\mathrm{TS}}(A^{\mathrm{TS}}\mid s)}{\pi^m(A^{\mathrm{TS}}\mid s)}\right]-\log\frac{\pi^{\mathrm{TS}}(A_j^{\mathrm{TS}}\mid s)}{\pi^m(A_j^{\mathrm{TS}}\mid s)}\right|\right].$$

Next, we use a standard symmetrization result to bound the last display; see, for example, Chapter 2.3, van der Vaart and Wellner [80] for a comprehensive treatment.

Lemma 6. *If $\xi_i \overset{\mathrm{iid}}{\sim} P$, we have*

$$\mathbb{E}\left[\sup_{g\in\mathcal{G}}\left|\frac{1}{n}\sum_{i=1}^{n}(g(\xi_i)-\mathbb{E}[g(\xi)])\right|\right]\le 4\mathbb{E}[\mathfrak{R}_n(\mathcal{G})]$$

Applying Lemma 6 to the bound on $|D_k|$. we conclude $|D_k|\le\frac{8}{N}\sup_{s\in\mathcal{S}}\mathbb{E}_{A_j^{\mathrm{TS}}\overset{\mathrm{iid}}{\sim}\pi^{\mathrm{TS}}(\cdot|s)}[\mathfrak{R}_{N_a}(\mathcal{G}_2'(s))]$, where $\mathcal{G}_2'(s)$ is the function class

$$\mathcal{G}_2'(s):=\left\{a\mapsto\log\frac{\pi^{\mathrm{TS}}(a\mid s)}{\pi^m(a\mid s)}:m\in\mathcal{M}\right\}.$$

Note that $\mathfrak{R}_{N_a}(\mathcal{G}_2'(s))=\mathfrak{R}_{N_a}(\mathcal{G}_2(s))$. Then, Azuma-Hoeffding bound (Corollary 2.20, Wainwright [82]) yields

$$Z_{N,N_a}\le\mathbb{E}[Z_{N,N_a}]+\sqrt{\frac{32t}{N}}\sup_{s\in\mathcal{S}}\mathbb{E}_{A_j^{\mathrm{TS}}\overset{\mathrm{iid}}{\sim}\pi^{\mathrm{TS}}(\cdot|s)}[\mathfrak{R}_{N_a}(\mathcal{G}_2(s))]$$

with probability at least $1-e^{-t}$.

It now remains to bound $\mathbb{E}[Z_{N,N_a}]$, for which we use a symmetrization argument. Although $(S_i,A_{ij}^{\mathrm{TS}})$ are not i.i.d., a standard argument still applies, which we outline for completeness. Denoting by $(\bar{S}_i,\bar{A}_{ij}^{\mathrm{TS}})$ independent copies of $(S_i,A_{ij}^{\mathrm{TS}})$ again, we have

$$\begin{aligned}
\mathbb{E}[Z_{N,N_a}] &= \mathbb{E}\left[\sup_{m\in\mathcal{M}}\left|\frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_a}\log\frac{\pi^{\mathrm{TS}}(A_{ij}^{\mathrm{TS}}\mid S_i)}{\pi^m(A_{ij}^{\mathrm{TS}}\mid S_i)}-\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_a}\log\frac{\pi^{\mathrm{TS}}(\bar{A}_{ij}^{\mathrm{TS}}\mid \bar{S}_i)}{\pi^m(\bar{A}_{ij}^{\mathrm{TS}}\mid \bar{S}_i)}\right]\right|\right]\\
&\le \mathbb{E}\left[\sup_{m\in\mathcal{M}}\left|\frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_a}\log\frac{\pi^{\mathrm{TS}}(A_{ij}^{\mathrm{TS}}\mid S_i)}{\pi^m(A_{ij}^{\mathrm{TS}}\mid S_i)}-\log\frac{\pi^{\mathrm{TS}}(\bar{A}_{ij}^{\mathrm{TS}}\mid \bar{S}_i)}{\pi^m(\bar{A}_{ij}^{\mathrm{TS}}\mid \bar{S}_i)}\right|\right]\\
&= \mathbb{E}\left[\sup_{m\in\mathcal{M}}\left|\frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_a}\epsilon_{ij}\left(\log\frac{\pi^{\mathrm{TS}}(A_{ij}^{\mathrm{TS}}\mid S_i)}{\pi^m(A_{ij}^{\mathrm{TS}}\mid S_i)}-\log\frac{\pi^{\mathrm{TS}}(\bar{A}_{ij}^{\mathrm{TS}}\mid \bar{S}_i)}{\pi^m(\bar{A}_{ij}^{\mathrm{TS}}\mid \bar{S}_i)}\right)\right|\right]\\
&\le 2\mathbb{E}[\mathfrak{R}_{NN_a}(\mathcal{G}_3)].
\end{aligned}$$

Collecting these bounds, we obtain the desired result.

B.2 Proof of Corollary 1

We use the following standard result that bound the Rademacher complexity of kernel models. Let k be a reproducing kernel on Ξ, and let $\mathbb{B}$ be the unit ball in the RKHS $\mathcal{H}$.

Claim 7. *Let* $\sup_{\xi\in\Xi} k(\xi,\xi) = B < \infty$. *Then,* $\mathfrak{R}_n(\mathbb{B}) \le \frac{B}{\sqrt{n}}$.

Proof of Claim 7 For any fixed $\xi_1, \dots, \xi_n$,

$$\mathfrak{R}_n(\mathbb{B}) = \frac{1}{n}\mathbb{E}_\varepsilon\left[\sup_{h\in\mathbb{B}}\left\langle h, \sum_{i=1}^n \varepsilon_i k(\cdot,\xi_i)\right\rangle\right] = \frac{1}{n}\mathbb{E}_\varepsilon\left[\left\|\sum_{i=1}^n \varepsilon_i k(\cdot,\xi_i)\right\|_{\mathcal{H}}\right]$$
$$\le \frac{1}{n}\left(\mathbb{E}_\varepsilon\left[\left\|\sum_{i=1}^n \varepsilon_i k(\cdot,\xi_i)\right\|_{\mathcal{H}}^2\right]\right)^{\frac{1}{2}} = \frac{1}{n}\sqrt{\sum_{i=1}^n \|k(\cdot,\xi_i)\|_{\mathcal{H}}^2} \le \frac{B}{\sqrt{n}}. \quad \square$$

Applying the claim to $\mathbb{B}_{\mathcal{A}}$ and and $\mathbb{B}_{\mathcal{S}\times\mathcal{A}}$, we get

$$\sup_{s\in\mathcal{S}} \mathfrak{R}_{N_a}(\mathcal{G}_2(s)) \le \frac{B}{\sqrt{N_a}} \quad \text{and} \quad \mathfrak{R}_{NN_a}(\mathcal{G}_3) \le \frac{B}{\sqrt{NN_a}}.$$

To bound $r_N^\star$, we use the following result due to Mendelson [56].

Lemma 8 (Mendelson [56, Theorem 2.1])**.** *If* $\lambda_1 \ge 1/N$, *then for all* $r \ge 1/N$

$$\mathbb{E}\left[\mathfrak{R}_N\left\{h\in\mathbb{B} : \mathbb{E}[h(S)^2] \le r\right\}\right] \lesssim \left(\frac{1}{N}\sum_{j=1}^\infty \min\{\lambda_j, r\}\right)^{\frac{1}{2}}.$$

Consider the case where the spectrum of $T_{k_{\mathcal{S}}}$ decay exponentially

$$\left(\frac{1}{N}\sum_{j=1}^\infty \min\left\{e^{-j^2}, \frac{\sqrt{\log N}}{N}\right\}\right)^{\frac{1}{2}} \lesssim \left(\frac{1}{N}\sum_{j=1}^{\sqrt{\log N}} \frac{\sqrt{\log N}}{N} + \frac{1}{N}\int_{\sqrt{\log N}}^\infty e^{-t^2}dt\right)^{\frac{1}{2}} \lesssim \frac{\sqrt{\log N}}{N},$$

where we use $\lesssim$ to denote inequality up to a numerical constant. We conclude $r_N^\star \lesssim M\frac{\sqrt{\log N}}{N}$. For polynomially decaying spectrum $\lambda_j \lesssim j^{-2\beta}$,

$$\sum_{j=1}^\infty \min\{j^{-2\beta}, r\} \approx r^{\frac{2\beta-1}{2\beta}} + \int_{r^{-1/2\beta}}^\infty t^{-2\beta}dt \asymp r^{\frac{2\beta-1}{2\beta}}.$$

Solving for the fixed point, we get $r_N^\star \asymp Mn^{-\frac{2\beta}{2\beta+1}}$.

Collecting these bounds and plugging them into Theorem 1, we obtain the desired result.

C Proof of regret bounds

C.1 Proof of regret decomposition (Lemma 1)

Conditional on (H_t, S_t), A_t^{TS} has the same distribution as $A_t^\star$. Since $U_t(a; H_t, S_t)$ is a deterministic function conditional on (H_t, S_t), we have

$$\mathbb{E}[U_t(A_t^{\mathrm{TS}}; H_t, S_t) \mid H_t, S_t] = \mathbb{E}[U_t(A_t^\star; H_t, S_t) \mid H_t, S_t].$$

We can rewrite the (conditional) instantenous regret as

$$\begin{aligned}
&\mathbb{E}[f_\theta(A_t^\star, S_t) - f_\theta(A_t, S_t) \mid H_t, S_t] \\
&= \mathbb{E}[f_\theta(A_t^\star, S_t) - U_t(A_t^\star; H_t, S_t) \mid H_t, S_t] + \mathbb{E}[U_t(A_t^{\mathrm{TS}}; H_t, S_t) - f_\theta(A_t, S_t) \mid H_t, S_t] \\
&= \mathbb{E}[f_\theta(A_t^\star, S_t) - U_t(A_t^\star; H_t, S_t) \mid H_t, S_t] + \mathbb{E}[U_t(A_t; H_t, S_t) - f_\theta(A_t, S_t) \mid H_t, S_t] \\
&\quad + \mathbb{E}[U_t(A_t^{\mathrm{TS}}; H_t, S_t) - U_t(A_t; H_t, S_t) \mid H_t, S_t]. \qquad (15)
\end{aligned}$$

We proceed by bounding the gap

$$\mathbb{E}[U_t(A_t^{\mathrm{TS}}; H_t, S_t) - U_t(A_t; H_t, S_t) \mid H_t, S_t] \qquad (16)$$

by the KL divergence between π_t^{TS} and π_t. Recall Pinsker's inequality [79]

$$\|P - Q\|_{\mathrm{TV}} := \sup_{g:\mathcal{A}\to[-1,1]} |\mathbb{E}_P[g(A)] - \mathbb{E}_Q[g(A)]| \le \sqrt{\frac{1}{2} D_{\mathrm{kl}}\left(P \| Q\right)}.$$

From the hypothesis, Pinsker's inequality implies

$$\begin{aligned}
\mathbb{E}[|U_t(A_t^{\mathrm{TS}}; H_t, S_t) - U_t(A_t; H_t, S_t)| \mid H_t, S_t] &\le M_t(H_t, S_t) \left\|\pi_t^{\mathrm{TS}}(\cdot \mid S_t) - \pi_t(\cdot \mid S_t)\right\|_{\mathrm{TV}} \\
&\le M_t(H_t, S_t) \sqrt{\frac{1}{2} D_{\mathrm{kl}}\left(\pi_t^{\mathrm{TS}}, \pi_t \mid S_t\right)}.
\end{aligned}$$

Applying this bound in the decomposition (15), and taking expectation over (H_t, S_t) on both sides and summing $t = 1, \ldots, T$, we get

$$\begin{aligned}
\mathrm{BayesRegret}\left(T, \pi\right) \le &\sum_{t=1}^{T} \mathbb{E}[f_\theta(A_t^\star, S_t) - U_t(A_t^\star; H_t, S_t)] + \sum_{t=1}^{T} \mathbb{E}[U_t(A_t; H_t, S_t) - f_\theta(A_t, S_t)] \\
&+ \sum_{t=1}^{T} \mathbb{E}\left[M_t(H_t, S_t) \sqrt{\frac{1}{2} D_{\mathrm{kl}}\left(\pi_t^{\mathrm{TS}}, \pi_t \mid S_t\right)}\right].
\end{aligned}$$

Applying Cauchy-Schwarz inequality and noting that $\sqrt{\mathbb{E}[M_t(H_t, S_t)^2]} \le L$, we obtain the final decomposition.

C.2 Proof of Theorem 2

We begin by defining a few requisite concepts. Recall that a collection $v_1, \ldots, v_N$ is an *ϵ-cover* of a set V in norm $\|\cdot\|$ if for each $v \in \mathcal{V}$, there exists v_i such that $\|v - v_i\| \le \epsilon$. The *covering number* is

$$N(V, \epsilon, \|\cdot\|) := \inf\left\{N \in \mathbb{N} \mid \text{there is an } \epsilon\text{-cover of } V \text{ with respect to } \|\cdot\|\right\}.$$

For a class of functions $\mathcal{H} \subset \{f : \mathcal{A} \times \mathcal{S} \to \mathbb{R}\}$, we consider the sup-norm $\|h\|_{L^\infty(\mathcal{X})} := \sup_{a\in\mathcal{A}, s\in\mathcal{S}} |h(a, s)|$.

We use the notion of *eluder dimension* proposed by Russo and Van Roy [66], which quantifies the difficulty of sequential decision making with the reward function class $\mathcal{F} = \{f_\theta(\cdot, \cdot) : \theta \in \Theta\}$.

Definition 1. *An action-state pair $(a, s) \in (\mathcal{A}, \mathcal{S})$ is ϵ-dependent on $\{(a_1, s_1), \ldots, (a_n, s_n)\} \subset \mathcal{A} \times \mathcal{S}$ with respect to $\mathcal{F}$ if for any $f, f' \in \mathcal{F}$*

$$\left(\sum_{i=1}^{n} (f(a_i, s_i) - f'(a_i, s_i))^2\right)^{\frac{1}{2}} \le \epsilon \quad \textit{implies} \quad f(a, s) - f'(a, s) \le \epsilon.$$

We say that $(a, s) \in \mathcal{A} \times \mathcal{S}$ is ϵ-independent of $\{(a_1, s_1), \ldots, (a_n, s_n)\}$ with respect to $\mathcal{F}$ if (a, s) is not ϵ-dependent on $\{(a_1, s_1), \ldots, (a_n, s_n)\}$.

Definition 2. *The* eluder dimension $d_{\mathrm{E}}(\mathcal{F}, \epsilon)$ *of* $\mathcal{F}$ *is the length of the longest sequence in* $\mathcal{A} \times \mathcal{S}$ *such that for some* $\epsilon' \geq \epsilon$*, every element in the sequence is* ϵ'*-independent of its predecessors.*

The eluder dimension bounds the Bayes regret decomposition given in Lemma 1.

Lemma 9 (Russo and Van Roy [66])**.** *Let* $\pi = \{\pi_t\}_{t \geq 1}$ *be any policy, and* $\mathcal{F} = \{(a, s) \mapsto f_\theta(a, s) : \theta \in \Theta\}$*. Assume* $f_\theta(a, s) \in [-M, M]$ *for all* $\theta \in \Theta, a \in \mathcal{A}, s \in \mathcal{S}$*, and* $R_t - f_\theta(A_t, S_t)$ *is* σ *sub-Gaussian conditional on* (θ, H_t, S_t, A_t)*. When* $\sup_{a \in \mathcal{A}} |U_t(a; H_t, S_t)| \leq M_t(H_t, S_t)$ *holds, we have*

$$\begin{aligned}\mathrm{BayesRegret}\,(T, \pi) \leq & C d_{\mathrm{E}}(\mathcal{F}, T^{-1}) + \sigma \sqrt{T d_{\mathrm{E}}(\mathcal{F}, T^{-1})(\log T + \log N(\mathcal{F}, T^{-1}, \|\cdot\|_{L^\infty(\mathcal{X})}))} \\ & + L \sum_{t=1}^{T} \sqrt{\frac{1}{2} \mathbb{E}\left[D_{\mathrm{kl}}\left(\pi_t^{\mathrm{TS}}, \pi_t \mid S_t\right)\right]},\end{aligned}$$

and when condition (12) *holds, we have*

$$\begin{aligned}\mathrm{BayesRegret}\,(T, \pi) \leq & C d_{\mathrm{E}}(\mathcal{F}, T^{-1}) + \sigma \sqrt{T d_{\mathrm{E}}(\mathcal{F}, T^{-1})(\log T + \log N(\mathcal{F}, T^{-1}, \|\cdot\|_{L^\infty(\mathcal{X})}))} \\ & + L \sum_{t=1}^{T} \mathbb{E}\left[D_{\mathrm{w}}\left(\pi_t^{\mathrm{TS}}, \pi_t \mid S_t\right)\right].\end{aligned}$$

for some constant $C > 0$ *that only depends on* M*.*

From Lemma 9, it suffices to bound the covering number and the eluder dimension of the linear model class

$$\mathcal{F} = \{(a, s) \mapsto g(\langle \phi(a, s), \theta \rangle) : \theta \in \Theta\}.$$

Since $\theta \mapsto g(\langle \phi(a, s), \theta \rangle)$ is c_2-Lipschitz with respect to $\|\cdot\|_2$, a standard covering argument (e.g. see Chapter 2.7.4 of van der Vaart and Wellner [80]) gives

$$N\left(\mathcal{H}, \epsilon, \|\cdot\|_{L^\infty(\mathcal{X})}\right) \leq N\left(\Theta, \frac{\epsilon}{c_2}, \|\cdot\|\right) \leq \left(1 + \frac{2 c_1 c_2}{\epsilon}\right)^d.$$

Proposition 11, Russo and Van Roy [66] shows that

$$d_{\mathrm{E}}(\mathcal{F}, T^{-1}) \leq C d r^2 \log rT$$

for some constant C that depends only on c_1 and c_2. Using these bounds in Lemma 9, we obtain the result.

C.3 Explicit regret bounds for linear bandits

Instead of bounding the eluder dimension, we can directly bound the upper confidence bounds in the decomposition in Lemma 1. By using the regret analysis of Dani et al. [26], Abbasi-Yadkori et al. [1, 2] for UCB algorithms, we obtaint he following result for linear contextual bandits.

Lemma 10. *Let* $\phi : \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ *such that* $f_\theta(a, s) = \phi(a, s)^\top \theta$ *for all* $\theta \in \Theta$*. Let* $c_1, c_2, \sigma > 0$ *be such that*

$$\sup_{\theta \in \Theta} \|\theta\|_2 \leq c_1, \qquad \sup_{a \in \mathcal{A}, s \in \mathcal{S}} \|\phi(a, s)\|_2 \leq c_2,$$

and assume that $R_t - f_\theta(A_t, S_t)$ is σ-sub-Gaussian conditional on (θ, H_t, S_t, A_t). Then, there exists a constant C that depends on c_1, c_2, σ such that

$$\begin{aligned}\text{BayesRegret}\left(T, \{\pi_t\}_{t\in\mathbb{N}}\right) \leq & 2\left((c_1+1)c_1 + \sigma\sqrt{d + \log\sqrt{T}\left(1+\frac{c_2^2 T}{\lambda}\right)}\right)\sqrt{2Td\log\left(\lambda + \frac{Tc_2^2}{d}\right)} \\ & + 4c_1c_2\sqrt{T} + c_1c_2\sum_{t=1}^{T}\sqrt{\frac{1}{2}\mathbb{E}\left[D_{\text{kl}}\left(\pi_t^{\text{TS}}, \pi_t \mid S_t\right)\right]} \end{aligned} \tag{17}$$

Furthermore, if $a \mapsto \phi(a, s)$ is L-Lipschitz with respect to a metric d, then the same bound holds with $L\sum_{t=1}^{T}\mathbb{E}\left[D_{\text{w}}\left(\pi_t^{\text{TS}}, \pi_t \mid S_t\right)\right]$ replacing the last sum, where $D_{\text{w}}(\cdot,\cdot\mid\cdot)$ is the Wasserstein distance defined with the metric d.

Although we omit it for brevity, the above $O(\sqrt{dT}\log T)$ regret bound can be improved to $\tilde{O}(\mathbb{E}[\sqrt{\|\theta\|_0\, dT}])$ by using a similar argument as below (see Proposition 3, [66] and [2]).

Proof

Lemma 10 follows from a direct consequence of Lemma 1, and Dani et al. [26], Abbasi-Yadkori et al. [1]; we detail the proof below for completeness. To show the bound (17), let $L_t(a; H_t, S_t)$ be an arbitrary sequence of measurable functions denoting lower confidence bounds. The Bayes regret decomposition (7) implies

$$\begin{aligned}\text{BayesRegret}\left(T, \pi\right) \leq & \sum_{t=1}^{T}\mathbb{E}[U_t(A_t; H_t, S_t) - L_t(A_t; H_t, S_t)] \\ & + 2c_1c_2\sum_{t=1}^{T}\left\{\mathbb{P}(f_\theta(A_t, S_t) \leq L_t(A_t; H_t, S_t)) + \mathbb{P}(f_\theta(A_t^\star, S_t) \geq U_t(A_t^\star; H_t, S_t))\right\} \\ & + L\sum_{t=1}^{T}\sqrt{2\mathbb{E}\left[D_{\text{kl}}\left(\pi_t^{\text{TS}}, \pi_t \mid S_t\right)\right]}. \end{aligned} \tag{18}$$

We proceed by bounding the first and second sum in the above inequality.

To ease notation, for a fixed $\lambda \geq 1 \vee c_2^2$ define

$$X_t := \begin{bmatrix} \phi(A_1, S_1)^\top \\ t\vdots \\ \phi(A_t, S_t)^\top \end{bmatrix}, \quad Y_t := \begin{bmatrix} R_1 \\ \vdots \\ R_t \end{bmatrix}, \quad V_t := \lambda I + \sum_{k=1}^{t}\phi(A_k, S_k)\phi(A_k, S_k)^\top$$

for all $t \in \mathbb{N}$, and we let $V_0 := \lambda I$. We use the following key result due to Dani et al. [26], Abbasi-Yadkori et al. [1].

Lemma 11 (Theorem 2, Abbasi-Yadkori et al. [1])**.** *Under the conditions of the proposition, for any $\delta > 0$*

$$\mathbb{P}\left(\left\|\theta - \widehat{\theta}_t\right\|_{V_t} \leq \sqrt{\lambda}c_1 + \sigma\sqrt{d\left(\log\frac{1}{\delta} + \log\left(1+\frac{c_2^2 t}{\lambda}\right)\right)} =: \beta_t(\delta) \text{ for all } t \geq 0 \;\middle|\; \theta\right) \geq 1-\delta$$

where we used $\|\theta\|_A := \sqrt{\theta^\top A\theta}$.

To instantiate the decomposition (18), we let

$$U_t(a; H_t, S_t) := \sup_{\theta' : \left\|\theta' - \widehat{\theta}_{t-1}\right\|_{V_{t-1}} \le \beta_{t-1}(\delta)} \phi(a, S_t)^\top \theta',$$
$$L_t(a; H_t, S_t) := \inf_{\theta' : \left\|\theta' - \widehat{\Theta}_{t-1}\right\|_{V_{t-1}} \le \beta_{t-1}(\delta)} \phi(a, S_t)^\top \theta'.$$

We are now ready to bound the second term in the decomposition (18). On the event

$$\mathcal{E} := \left\{ \left\|\theta - \widehat{\theta}_t\right\|_{V_t} \le \beta_t(\delta) \text{ for all } t \ge 0 \right\},$$

we have $f_\theta(A_t, S_t) \ge L_t(A_t; H_t, S_t)$ and $f_\theta(A_t^\star, S_t) \le U_t(A_t^\star; H_t, S_t)$ by definition. Since Lemma 11 states $\mathbb{P}(\mathcal{E} \mid \theta) \ge 1 - \delta$, we conclude that the second sum in the decomposition (18) is bounded by $4c_1c_2T\delta$.

To bound the first sum in the decomposition (18), we use the following bound on the norm of feature vectors.

Lemma 12 (Lemma 11, Abbasi-Yadkori et al. [1])**.** *If $\lambda \ge c_2^2 \vee 1$, for any sequence of a_t, s_t for $t \ge 1$, and corresponding $A_t := \lambda I + \sum_{k=1}^t \phi(a_k, s_k)\phi(a_k, s_k)^\top$, we have*

$$\sum_{t=1}^T \|\phi(A_t, S_t)\|^2_{V_{t-1}^{-1}} \le 2d \log\left(\lambda + \frac{Tc_2^2}{d}\right).$$

Noting that by definition

$$U_t(A_t; H_t, S_t) - L_t(A; H_t, S_t) \le 2 \left\|\phi(A_t, S_t)\right\|_{V_{t-1}^{-1}} \beta_{t-1}(\delta),$$

we obtain

$$\begin{aligned}
\sum_{t=1}^T U_t(A_t; H_t, S_t) - L_t(A; H_t, S_t) &\le 2 \sum_{t=1}^T \left\|\phi(A_t, S_t)\right\|_{V_{t-1}^{-1}} \beta_{t-1}(\delta) \\
&\overset{(a)}{\le} 2\beta_T(\delta) \sum_{t=1}^T \left\|\phi(A_t, S_t)\right\|_{V_{t-1}^{-1}} \\
&\overset{(b)}{\le} 2\beta_T(\delta) \sqrt{T \sum_{t=1}^T \|\phi(A_t, S_t)\|^2_{V_{t-1}^{-1}}} \\
&\overset{(c)}{\le} 2\beta_T(\delta) \sqrt{2Td \log\left(\lambda + \frac{Tc_2^2}{d}\right)}
\end{aligned}$$

where we used monotonicity of $t \mapsto \beta_t(\delta)$ in step (a), Cauchy-Schwarz inequality in step (b), and Lemma 12 in step (c).

Collecting these bounds, we conclude

$$\begin{aligned}
\text{BayesRegret}\,(T, \pi) \le & 2\beta_T(\delta) \sqrt{2Td \log\left(\lambda + \frac{Tc_2^2}{d}\right)} + 4c_1c_2T\delta \\
& + L \sum_{t=1}^T \sqrt{\frac{1}{2} \mathbb{E}\left[D_{\text{kl}}\left(\pi_t^{\text{TS}}, \pi_t \mid S_t\right)\right]}.
\end{aligned}$$

Setting $\delta = 1/\sqrt{T}$, we obtain the first result. The second result is immediate by starting with the decomposition (13) and using an identical argument.

C.4 Proof of Theorem 3

In what follows, we abuse notation and let C be a universal constant that changes line by line. Since $f_\theta(a, s)$ follows a Gaussian process, its posterior mean and variance is given by

$$\mu_t(a,s) := \mathbb{E}[f_\theta(a,s) \mid H_t] = \Sigma_t(a,s)^\top (K_t + \sigma^2 I)^{-1} y_t,$$
$$\sigma_t^2(a,s) := \mathrm{Var}(f_\theta(a,s) \mid H_t) = \Sigma((a,s),(a,s)) - \Sigma_t(a,s)^\top (K_t + \sigma^2 I)^{-1} \Sigma_t(a,s)$$

where $\Sigma_t(a,s) := [\Sigma((A_j,S_j),(a,s))]_{1\le j\le t}$, $K_t := [k((A_i,S_i),(A_j,S_j))]_{1\le i,j\le t}$ and $y_t = [r_j]_{1\le j\le t}$. Define the upper confidence bound

$$U_t(a; H_t, s) := \mu_t(a,s) + \sqrt{\beta_t}\sigma_t(a,s)$$

where $\beta_t = 2\log((t^4 r d)^d t^2)$. Noting that

$$|U_t(a;H_t,s)| \le \sup_{a\in\mathcal{A}, s\in\mathcal{S}} |\mathbb{E}\left[f_\theta(a,s) \mid H_t\right]| + \sqrt{\beta_t}\Sigma((a,s),(a,s)) \le \mathbb{E}\left[\sup_{a\in\mathcal{A},s\in\mathcal{S}} |f_\theta(a,s)| \;\middle|\; H_t\right] + \sqrt{\beta_t}c_2,$$

a minor modification to the proof of Lemma 1 yields

$$\begin{aligned}\mathrm{BayesRegret}\,(T,\pi) \le &\sum_{t=1}^T \mathbb{E}[f_\theta(A_t^\star, S_t) - U_t(A_t^\star; H_t, S_t)] + \sum_{t=1}^T \mathbb{E}[U_t(A_t; H_t, S_t) - f_\theta(A_t, S_t)] \\ &+ \sum_{t=1}^T \left(\left\| \mathbb{E}\left[\sup_{a\in\mathcal{A},s\in\mathcal{S}} |f_\theta(a,s)| \;\middle|\; H_t\right]\right\|_{2,P} + \sqrt{\beta}_t c_2 \right) \sqrt{2\mathbb{E}\left[D_{\mathrm{kl}}\left(\pi_t^{\mathrm{TS}}, \pi_t \mid S_t\right)\right]}.\end{aligned} \tag{19}$$

From Jensen's inequality and the tower property,

$$\left\| \mathbb{E}\left[\sup_{a\in\mathcal{A},s\in\mathcal{S}} |f_\theta(a,s)| \;\middle|\; H_t\right]\right\|_{2,P} \le \left\| \sup_{a\in\mathcal{A},s\in\mathcal{S}} |f_\theta(a,s)| \right\|_{2,P} = c_3.$$

From Borel-TIS inequality (e.g., see [5]), we have $c_3 < \infty$.

We now proceed by bounding the first two terms in the regret decomposition (19). Let $\mathcal{A}_t$ be a $(1/t^4)$-cover of $\mathcal{A}$, so that for any $a \in \mathcal{A}$, there exists $[a]_t \in \mathcal{A}_t$ such that $\|a - [a]_t\|_1 \le 1/t^4$. Since $|\mathcal{A}_t| \le (t^4 r d)^d$, we have $2\log(|\mathcal{A}_t| t^2) \le \beta_t$. We begin by decomposing the first term in the bound (7).

$$\begin{aligned}\sum_{t=1}^T f_\theta(A_t^\star, S_t) - U_t(A_t^\star; H_t, S_t) = &\underbrace{\sum_{t=1}^T \mathbb{E}[f_\theta(A_t^\star, S_t) - f_\theta([A_t^\star]_t, S_t)]}_{(a)} + \underbrace{\sum_{t=1}^T \mathbb{E}[f_\theta([A_t^\star]_t, S_t) - U_t([A_t^\star]_t; H_t, S_t)]}_{(b)} \\ &+ \underbrace{\sum_{t=1}^T \mathbb{E}[U_t([A_t^\star]_t; H_t, S_t) - U_t(A_t^\star; H_t, S_t)]}_{(c)}.\end{aligned}$$

Using the definition of L_f, the first term (a) in the above equality is bounded by

$$\sum_{t=1}^{T} \mathbb{E}[f_\theta(A_t^\star, S_t) - f_\theta([A_t^\star]_t, S_t)] \le \mathbb{E}[L_f] \sum_{t=1}^{T} \|A_t^\star - [A_t^\star]_t\|_1 \le \mathbb{E}[L_f] \sum_{t=1}^{\infty} \frac{1}{t^4} \le C\mathbb{E}[L_f]$$

where we used the fact that $\mathcal{A}_t$ is a $1/t^4$-cover of $\mathcal{A}$. To bound the second term (b), note that since $f_\theta(a,s) \mid H_t \sim N(\mu_t(a,s), \sigma_t^2(a,s))$, we have

$$\mathbb{E}[f_\theta(a,s) - U_t(a; H_t, s) \mid H_t] \le \mathbb{E}[(f_\theta(a,s) - U_t(a; H_t, s))_+ \mid H_t] = \frac{\sigma_t(a,s)}{\sqrt{2\pi}} e^{-\frac{\beta_t}{2}} \le \frac{c_2}{\sqrt{2\pi} t^2 |\mathcal{A}_t|}. \tag{20}$$

Hence, we obtain the bound

$$\sum_{t=1}^{T} \mathbb{E}[f_\theta([A_t^\star]_t, S_t) - U_t([A_t^\star]_t; H_t, S_t)] \le \sum_{t=1}^{T} \sum_{a \in \mathcal{A}_t} \mathbb{E}[f_\theta(a, S_t) - U_t(a; H_t, S_t)] \le \sum_{t=1}^{\infty} \frac{c_2}{\sqrt{2\pi} t^2} \le C c_2$$

where we used the independence of S_t and H_t, and the bound (20).

To bound the third term (c), we show the claim

$$\begin{aligned} |U_t(a; H_t, s) - U_t(a'; H_t, s)| &\le \mathbb{E}[L_f \mid H_t] \left\|a - a'\right\|_1 \\ &\quad + \sqrt{\beta_t} \left(2\mathbb{E}\left[L_f \left(\sup_{a \in \mathcal{A}, s \in \mathcal{S}} \mu(a,s)^2 + \sup_{a \in \mathcal{A}, s \in \mathcal{S}} f_\theta(a,s)^2 \right) \mid H_t \right] \right)^{\frac{1}{2}} \left\|a - a'\right\|_1^{\frac{1}{2}}. \end{aligned} \tag{21}$$

From the above claimed bound, it follows that

$$\begin{aligned} \sum_{t=1}^{T} \mathbb{E}[U_t([A_t^\star]_t; H_t, S_t) - U_t(A_t^\star; H_t, S_t)] &\le \sum_{t=1}^{T} \frac{\mathbb{E}[L_f]}{t^4} + \sum_{t=1}^{T} \sqrt{2\beta_t} \frac{c_1 \sqrt{\mathbb{E}[L_f]} + c_3 \sqrt{\mathbb{E}[L_f^2]}}{t^2} \\ &\le C\mathbb{E}[L_f] + Cd \log(rd) \left(c_1 \sqrt{\mathbb{E}[L_f]} + c_3 \sqrt{\mathbb{E}[L_f^2]} \right). \end{aligned}$$

To show the bound (21), first note that $a \mapsto \mathbb{E}[f_\theta(a,s) \mid H_t]$ and $a \mapsto \mathbb{E}[f_\theta(a,s)^2 \mid H_t]$ is $\mathbb{E}[L_f \mid H_t]$- and $\mathbb{E}[2L_f \sup_{a \in \mathcal{A}, s \in \mathcal{S}} |f_\theta(a,s)| \mid H_t]$- Lipschitz respectively, for all $s \in \mathcal{S}$. Hence, $a \mapsto \sigma_t^2(a,s)$ is $\mathbb{E}[2L_f(c_1^2 + \sup_{a \in \mathcal{A}, s \in \mathcal{S}} |f_\theta(a,s)|^2) \mid H_t]$-Lipschitz. Noting that

$$|\sigma_t(a,s) - \sigma_t(a',s)| = \left| \frac{\sigma_t^2(a,s) - \sigma_t^2(a',s)}{\sigma_t(a,s) + \sigma_t(a',s)} \right| \le \frac{1}{c} |\sigma_t^2(a,s) - \sigma_t^2(a',s)| + c$$

for any $c > 0$, taking the infimum over $c > 0$ on the right hand side yields

$$\begin{aligned} |\sigma_t(a,s) - \sigma_t(a',s)| &\le \sqrt{2|\sigma_t^2(a,s) - \sigma_t^2(a',s)|} \\ &\le \left(2\mathbb{E}\left[L_f \left(c_1^2 + \sup_{a \in \mathcal{A}, s \in \mathcal{S}} f_\theta(a,s)^2 \right) \mid H_t \right] \right)^{\frac{1}{2}} \left\|a - a'\right\|_1^{\frac{1}{2}} \end{aligned}$$

which shows the bound (21).

Collecting these bounds, we have shown that

$$\sum_{t=1}^{T} \mathbb{E}[f_\theta(A_t^\star, S_t) - U_t(A_t^\star; H_t, S_t)] \le C\mathbb{E}[L_f] + Cc_2 + Cd \log(rd) \left(c_1 \sqrt{\mathbb{E}[L_f]} + c_3 \sqrt{\mathbb{E}[L_f^2]} \right). \tag{22}$$

To bound the second term in the Bayes regret decomposition (19), we use the following lemma due to Srinivas et al. [72].

Lemma 13 (Lemma 5.3 Srinivas et al. [72]). *For any sequence of A_t and S_t,*

$$\mathbb{E}\left(\sum_{t=1}^{T}\sigma_t(A_t,S_t)^2\right)^{\frac{1}{2}} \leq \sqrt{\frac{2\gamma_T}{\log(1+\sigma^{-2})}}$$

Using the lemma, we have

$$\sum_{t=1}^{T}\mathbb{E}[U_t(A_t;H_t,S_t)-f_\theta(A_t,S_t)] = \sum_{t=1}^{T}\sqrt{\beta_t}\mathbb{E}[\sigma_t(A_t,S_t)] \leq \sqrt{T\beta_T}\sqrt{\frac{2\gamma_T}{\log(1+\sigma^{-2})}}.$$

Combining this with the bound (22), we obtain our result.

D Experiment Details

Hyperparameters We use hyperparameters from Riquelme et al. [62] as follows. The NeuralGreedy, NeuralLinearTS methods use a fully-connected neural network with two hidden layers of containing 100 rectified linear units. The networks are multi-output, where each output corresponds for predicted reward under each action. The networks are trained using 100 mini-batch updates at each period to minimize the mean-squared error via RMSProp with an initial learning rate of 0.01. The learning rate is decayed after each mini-batch update according to an inverse time decay schedule with a decay rate of 0.55 and the learning rate is reset the initial learning rate each update period. For Bootstrap-NN-TS, we use 10 replicates and train each replicate with all observations as in Riquelme et al. [62].

The Bayesian linear regression models used on the last linear layer for NeuralLinear-TS use the normal inverse gamma prior $\text{NIG}(\mu_a=\mathbf{0},\alpha_a=3,\beta_a=3,\Lambda_a=0.25I_d)$. Linear-TS uses a $\text{NIG}(\mu_a=\mathbf{0},\alpha_a=6,\beta_a=6,\Lambda_a=0.25I_d)$ prior distribution.

The imitation models used by the IL methods are fully-connected neural networks with two hidden layers of 100 units and hyperbolic tangent activations. The networks use a Softmax function on the outputs to predict the probability of selecting each action. The networks are trained using 2000 mini-batch updates via RMSProp to minimize the KL-divergence between the predicted probabilities and the approximate propensity scores of the Thompson sampling policy π^{TS}. For each observed context S_i, we approximate the propensity scores of the Thompson sampling policy $\pi^{TS}(\cdot|S_i)$ using $N_a = 2048$ Monte Carlo samples: $\hat{\pi}^{TS}(a|S_i) = \frac{1}{N_a}\sum_{j=1}^{N_a}\mathbf{1}\{A_{ij}=a\}$ where $A_{ij}\sim\pi^{TS}(\cdot|S_i)$. We use an initial learning rate of 0.001. learning rate is decayed every 100 mini-batches according to an inverse time decay schedule with a decay rate of 0.05. In practice, the hyperparameters of the imitation model can be optimized or adjusted at each update period by minimizing the KL-divergence on a held-out subset of the observed data, which may lead to better regret performance. We do not use inverse propensity-weighting on the observations, but we suspect that may it may further improve performance.

We normalize all numeric features to be in [0,1] and one-hot encode all categorical features. For the Warfarin dataset, we also normalize the rewards to be in [0,1].

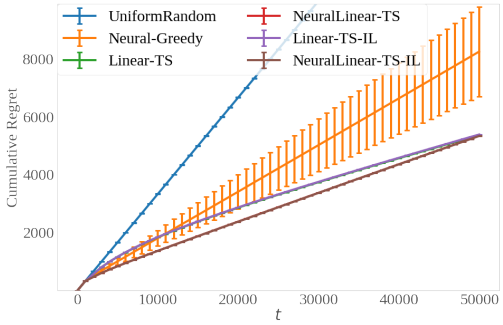


Figure 2: Cumulative regret on the Warfarin problem with 50 actions

Posterior Inference for Bayesian Linear Regression **Linear-TS**: For each action, We assume the data for action a were generated from the linear function: $r_a = \boldsymbol{s}^T\boldsymbol{\theta}_a + \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, \sigma_a^2)$.

$$\sigma_a^2 \sim \text{IG}(\alpha_a, \beta_a), \quad \boldsymbol{\theta_a}|\sigma_a^2 \sim \mathcal{N}(\boldsymbol{\mu}_a, \sigma_a^2\Sigma_a),$$

where the prior distribution is given by $\text{NIG}(\boldsymbol{\mu}_a, \Lambda_a, \alpha_a, \beta_a)$ and $\Lambda_a = \Sigma_a^{-1}$ is the precision matrix. After n_a observations of contexts $X_a \in \mathbb{R}^{n_a \times (d+1)}$ and rewards $\boldsymbol{y}_a \in \mathbb{R}^{n_a \times 1}$, we denote the joint posterior by $P(\boldsymbol{\theta}_a, \sigma_a^2) \sim \text{NIG}(\bar{\boldsymbol{\mu}}_a, \bar{\Lambda}_a, \bar{\alpha}_a, \bar{\beta}_a)$, where

$$\bar{\Lambda} = X_a^T X_a + \Lambda_a, \quad \bar{\boldsymbol{\mu}}_a = \bar{\Lambda}_a^{-1}(\Lambda_a\boldsymbol{\mu}_a + X_a^T\boldsymbol{y}_a)$$
$$\bar{\alpha}_a = \alpha + \frac{n_a}{2}, \quad \bar{\beta}_a = \beta + \frac{1}{2}(\boldsymbol{y}_a^T\boldsymbol{y}_a + \boldsymbol{\mu}_a^T\Lambda_a\boldsymbol{\mu}_a - \bar{\boldsymbol{\mu}}_a^T\bar{\Lambda}_a\bar{\boldsymbol{\mu}}_a).$$

Additional Results **Warfarin - 50 Actions** Figure 2 shows the cumulative regret on Warfarin using 50 actions. The imitation learning methods match the cumulative regret of the vanilla Thompson sampling methods.

E Time and Space Complexity

E.1 Complexity of Evaluated Methods

Table 2 shows the decision-making time complexity for the methods used in our empirical analysis. The time complexity is equivalent to the space complexity for all evaluated methods.

NeuralGreedy The time complexity of NeuralGreedy is the sum of matrix-vector multiplications involved in a forward pass.

Linear-TS The time complexity of Linear-TS is dominated by sampling from the joint posterior, which requires sampling from a multivariate normal with dimension d. To draw a sample from the joint posterior $P(\boldsymbol{\theta}, \sigma)$ at decision time, we first sample the noise level $\tilde{\sigma}^2 \sim \text{IG}(\alpha, \beta)$ and then sample $\tilde{\boldsymbol{\theta}}|\tilde{\sigma}^2 \sim \mathcal{N}\big(\boldsymbol{\mu}, \tilde{\sigma}^2\Lambda^{-1}\big)$. Rather than inverting the precision matrix $\tilde{\Sigma} = \tilde{\sigma}^2\Lambda^{-1}$, we compute root decomposition (e.g. a Cholesky decomposition) of the $d \times d$ precision matrix $\Lambda = LL^T$. The root decomposition can be computed once, with cost $O(d^3)$, after an offline batch update and cached until the next batch update. Given L^T, we sample directly by computing $\tilde{\boldsymbol{\theta}} = \boldsymbol{\mu} + \boldsymbol{z}$, where

$$\frac{1}{\tilde{\sigma}}L^T\boldsymbol{z} = \boldsymbol{\zeta} \tag{23}$$

and $\boldsymbol{\zeta} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$. Since L^T is upper triangular, Eqn. (23) can be solved using a backward substitution in quadratic time: $O(d^2)$.[6]

NeuralLinear-TS The time complexity of NeuralLinear-TS is the sum of a forward pass up to the last hidden layer and sampling from a multivariate normal with dimension h_M, where h_M is the size of the last hidden layer.

Imitation Learning The IL methods have the same time complexity as NeuralGreedy, ignoring the cost of sampling from multinomial with k categories.

E.2 Complexity Using Embedded Actions

An alternative modeling approach for the non-imitation methods is to embed the action with the context as input to the reward model.

NeuralGreedy Using an embedded action, the time complexity for a forward pass up to the last layer is $O_{\text{last-layer}} = O\big(kd_a h_1 + k\sum_{m=1}^{M-1} h_m h_{m+1}\big)$ because the input at decision time is a $k \times d_a$ matrix where the context is embedded with each of the k actions and the each context-action vector has dimension d_a. The time complexity of computing the output layer remains $O(kh_M)$. The space complexity remains linear in the number of parameters, but it also requires computing temporary intermediate tensors of size $k \times h_m$ for $m = 1...M$: $O\big(d_a h_1 + \sum_{m=1}^{M-1} h_m h_{m+1} + \sum_{m=1}^{M} k h_m\big)$.

Linear-TS Linear-TS with an embedded action only requires using a single sample of the parameters, which yields a complexity of to $O(d_a^2 + kd_a)$ for Linear-TS. The space complexity is also $O(d_a^2 + kd_a)$.

NeuralLinear-TS For NeuralLinear-TS the time complexity of computing the outputs given the last hidden layer is $O(h_M^2 + kh_M)$, since only a single sample of h_M parameters is required for computed the reward for all actions. The space complexity for NeuralLinear-TS the sum the space complexities of NeuralGreedy and Linear-TS.

Imitation Learning The computatiuonal cost of the IL methods would be unchanged.

We choose to empirically evaluate models *without* embedded actions because linear methods using embedded actions cannot model reward functions that involve non-linear interactions between the contexts and actions, whereas modeling each action independently allows for more flexibility. Riquelme et al. [62] find that Thompson sampling using disjoint, exact linear bayesian regressions are a strong baseline in many applications. Furthermore, Riquelme et al. [62] observe that it is important to model the noise levels independently for each action.

E.3 Complexity of Alternative Methods

Alternative Thompson sampling methods including mean-field approaches, the low-rank approximations of the covariance matrix, and bootstrapping can also decrease the computational cost of posterior sampling. Mean-field approaches can reduce time complexity of sampling parameters from the posterior from quadratic $O(n^2)$ to linear $O(n)$ in the number of parameters n.[7] However, assuming independence among parameters has been observed to result in worse performance in some settings [62]. Low-rank approximations of the covariance matrix allow for sampling parameters in

[6] The alternative approach of inverting the precision matrix to compute the covariance matrix $\Sigma = \Lambda^{-1}$, computing and caching its root decomposition $\Sigma = L_\Sigma L_\Sigma^T$, and sampling $\tilde{\boldsymbol{\theta}}$ as $\tilde{\boldsymbol{\theta}} = \boldsymbol{\mu} + L_\Sigma \boldsymbol{\zeta}$, where $\boldsymbol{\zeta} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$ also has a time complexity of $O(d^2)$ from the matrix-vector multiplication $L_\Sigma \boldsymbol{\zeta}$.

[7] We describe space complexity in terms of the number of parameters n, so that we do not make assumptions about the underlying model.

Table 2. Decision-making time complexity and space complexity for each method . For methods relying on fully-connected neural networks, the time complexity of a forward pass to the last hidden layer is $C_{\text{last-layer}} = dh_1 + \sum_{m=1}^{M-1} h_m h_{m+1}$, where d is the dimension of the context and h_m is the number of units in hidden layer m. For BOOTSTRAP-NN-TS, B denotes the number of bootstrap replicates.

METHOD	TIME COMPLEXITY	SPACE COMPLEXITY
NEURALGREEDY	$O(C_{\text{LAST-LAYER}}) + O(kh_M)$	$O(C_{\text{LAST-LAYER}}) + O(kh_M)$
LINEAR-TS	$O(kd^2)$	$O(kd^2)$
NEURALLINEAR-TS	$O(C_{\text{LAST-LAYER}}) + O\big(kh_M^2\big)$	$O(C_{\text{LAST-LAYER}}) + O\big(kh_M^2\big)$
BOOTSTRAP-NN-TS	$O(C_{\text{LAST-LAYER}}) + O(kh_M)$	$O(C_{\text{LAST-LAYER}} \cdot B) + O(kh_M B))$
IL	$O(C_{\text{LAST-LAYER}}) + O(kh_M)$	

$O((n+1)\rho)$, where ρ is the rank of the approximate covariance, but such methods have a space complexity of $O(\rho n)$ since they require storing ρ copies of the parameters [90, 52]. Bootstrapping also requires storing multiple copies of the parameters, so the space is $O(bn)$ where b is the number of bootstrap replicates. However, bootstrapping simply requires a multinomial draw to select one set of bootstrapped parameters. All these methods require a forward pass using the sampled parameters, and the time complexity is the sum of the time complexities of sampling parameters and the forward pass.