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

Hongseok Namkoong[*]
Columbia Business School
namkoong@gsb.columbia.edu

Samuel Daulton[*]
Facebook
sdaulton@fb.com

Eytan Bakshy
Facebook
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 internet applications where latency and ease of deployment are of concern. We propose a novel imitation-learning-based algorithm that distills a TS policy into an explicit policy representation by performing posterior inference and optimization offline. The explicit policy representation enables fast online decision-making and easy deployment in mobile and server-based environments. Our algorithm iteratively performs offline batch updates to the TS policy and learns a new imitation policy. Since we update the TS policy with observations collected under the imitation policy, our algorithm emulates an off-policy version of TS. Our imitation algorithm guarantees Bayes regret comparable to TS, up to the sum of single-step imitation errors. We show these imitation errors can be made arbitrarily small when unlabeled contexts are cheaply available, which is the case for most large-scale internet applications. Empirically, we show that our imitation policy achieves comparable regret to TS, while reducing decision-time latency by over an order of magnitude.

1 Introduction

In the past decade, Thompson sampling [54] 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 [50, 15, 39] and rigorous performance guarantees [30, 7, 8, 25, 28, 47, 3], Thompson sampling has gained popularity in a broad range of applications including revenue management [21], internet advertising [26, 5, 49], and recommender systems [31].

Despite its conceptual simplicity and strong performance, Thompson sampling can be difficult to deploy. Concretely, choosing an action with 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. Uncertainty calibration is crucial for empirical performance [45], since a well-calibrated model allows optimally trading off exploration and exploitation. In this regard, large-scale (probabilistic) machine learning models based on deep networks show much promise as they can adaptively learn good feature representations. However, sampling from these probabilistic models can be demanding in terms of both computation and memory. While approximate inference methods with better runtime characteristics exist, they tend to produce poorly calibrated uncertainty estimates that translate into poorer empirical performance on bandit tasks [45]. The second step, *optimization*, solves for a reward-optimizing action under the posterior sample. This can be prohibitively computationally expensive when the action space is large or continuous. For

[*] Equal contribution

Offline Reinforcement Learning Workshop at Neural Information Processing Systems, 2020.

example, an advertising platform deciding to match advertisers to users has to solve combinatorial optimization problems in real-time in order to run Thompson sampling [38].

These challenges are pronounced in mobile applications; as of 2018, an estimated 52.2% of web traffic was generated by mobile devices [52]. 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 [37] and social media ranking [42]. However, the majority of internet-connected mobile devices have limited memory, and rely on low-end processors that are orders of magnitude slower than server-grade devices [10, 64]. Another practical consideration is the technical debt of implementing contextual bandit algorithms in large-scale internet services. The online nature of the complex routines required by Thompson sampling leads the overall system to be cumbersome and hard to debug, making reliable software development challenging.

Motivated by 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, which allows efficient action generation without real-time posterior inference or numerical optimization. In order to avoid complex and computationally demanding routines *online*, our algorithm simulates a Thompson sampling policy *offline* and learns an explicit policy parameterization that mimics the behavior of Thompson sampling. As in many production systems, we consider the setting where the policy is updated offline with batches of new data [27]. At each period, the imitation learning problem can be efficiently solved using stochastic gradient-based methods, as it is equivalent to a log likelihood maximization (MLE) problem where the goal is to find a policy parameterization maximizing the likelihood of observing the actions generated by Thompson sampling. With this distilled policy in hand, actions can be generated efficiently by sampling from this parameterized distribution, conditional on the observed context; if we use neural networks to parameterize our imitation policy, this corresponds to a single forward pass. By virtue of moving complex numerical routines offline, our imitation procedure allows arbitrary, state-of-the-art Bayesian models and optimization solvers to be used even in latency and memory sensitive environments, and can be easily implemented using modern machine learning pipelines [23, 22].

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. Our main theoretical result (Section 4) establishes that our imitation algorithm maintains performance comparable to the true on-policy Thompson sampling policy, up to the sum of single-step imitation errors. In particular, our results preclude the possibility of small deviations between our imitation policy and Thompson sampling magnifying over time. We prove that each 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 (potentially unlabeled, meaning those without corresponding actions or rewards) available contexts. Combining this with our regret bound, our imitation algorithm achieves performance comparable to Thompson sampling up to $O_p(T/\sqrt{N})$-error. Despite the seemingly linear regret, we often have $T \ll \sqrt{N}$ in internet applications where databases with entity features provide abundant unlabeled contexts, often in the order of hundreds of millions. In contrast, the number of model updates (horizon T) is relatively small, in tens or hundreds, due to complexities of reliable software deployment and nonstationary user behavior. In practical problem settings where contexts are abundant, our results show that the imitation policy 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.

We empirically evaluate our imitation algorithm on several benchmark problems and a real-world dataset for selecting optimal video transcoding configurations (Section 5). In all of our experiments, we find that that our imitation algorithm performs as well as Thompson sampling in terms of cumulative regret, while reducing decision-time latency by an order of magnitude.

Related work Many authors have showed regret bounds for TS [6, 7, 30, 25, 28, 47, 3, 48]; we refer the reader to the recent tutorial [48] for a comprehensive overview. We build on the insights of Russo and Van Roy [47], and show that our imitation algorithm retains the advantageous properties of TS, achieving (gap-independent) Bayes regret comparable to the *best* UCB algorithm.

The performance of explore-exploit algorithms like TS depend on having access to well-calibrated probabilistic predictions. Obtaining a balance between predictive accuracy, time, and space can be challenging in the context of large datasets with over-parameterized 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; this assumes that the root decomposition of the covariance matrix has been computed and cached, which incurs a cost of $O(n^3)$. A common strategy used by some variational inference methods is to use a mean-field approach where parameters are assumed to be independent [11]. This assumption can decrease sampling costs from $O(n^2)$ to $O(n)$, where n is the number of parameters. However, Riquelme et al. [45] found that TS using such approaches often leads to poorer empirical performance.

When exact inference is not possible, approximate inference methods can be used for posterior sampling. See Russo et al. [48] for a discussion of elementary approximation methods in relation to TS (e.g. Laplace approximation for unimodal distributions). Bootstrapping [20, 41, 35] is a simple procedure that fits multiple models to resampled versions of the dataset to approximate samples from the posterior distribution; the computational burden of maintaining multiple models can be prohibitive. MCMC-based methods for approximate inference, and Hamilton Monte Carlo (HMC) [40] in particular, are largely regarded as the "gold standard" for approximate Bayesian inference. HMC, and other MCMC-like approaches (e.g., Chen et al. [16], Welling and Teh [59]) 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 [66, 36]. 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 TS, 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 methods from the active research on deep Bayesian methods.

In contrast with previous works that consider one-shot, offline policy optimization [53, 29], our method explicitly balances exploration and exploitation across multiple horizons, while using large batch updates.

2 Thompson Sampling (TS)

We consider a (Bayesian) contextual bandit problem where the agent (decision-maker) observes a context, takes an action, and receives a reward. Let P be a prior distribution over a parameter space 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 $f_\theta(a, s) = \mathbb{E}[R_t \mid \theta, A_t = a, S_t = s]$ for all $a \in \mathcal{A}, s \in \mathcal{S}$. 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, and assume 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's based on observed context-action-reward tuples. The *regret* of the agent compares the agent's cumulative rewards to the rewards under the optimal actions; for any fixed parameter value $\theta \in \Theta$, the (frequentist) regret for the set of policies $\{\pi_t\}_{t\in\mathbb{N}}$ is

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

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*, 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 $\theta \sim P$:

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

Based on the history so far, a TS algorithm 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

Algorithm 1 Imitating Thompson Sampling

```
Input: prior $P$ on parameter space $\Theta$, reward model class $\{f_\theta(\cdot,\cdot)\}$, imitation policy model class $\{\pi^m : m \in \mathcal{M}\}$, notion of distance $D$ for probabilities
for $t = 1$ to $T$ do
    Update $\pi_t^{TS} | H_t$ and $\pi_t^m$, where $m \leftarrow \mathrm{argmin}_{m \in \mathcal{M}}\, \mathbb{E}_{S \sim \mathbb{P}_S}[D\left(\pi_t^{TS}, \pi^m \mid S\right)]$
    Observe $S_t$, sample $A_t \sim \pi_t^m(\cdot \mid S_t)$, receive $R_t$
end for
```

from the posterior $\theta_t \sim P(\theta \in \cdot \mid H_t, S_t)$, and setting $A_t^{\mathrm{TS}} \in \mathrm{argmax}_{a \in \mathcal{A}}\, f_{\theta_t}(a, S_t)$. By definition, TS enjoys the optimality property $A_t^{\mathrm{TS}} \mid H_t, S_t \overset{d}{=} A_t^\star \mid H_t, S_t$ where $A_t^\star \in \mathrm{argmax}_{a \in \mathcal{A}}\, f_\theta(a, S_t)$ and θ is the true parameter drawn from the prior P.

3 Imitation Learning

Motivated by challenges in implementing Thompson sampling real-time in production systems, we develop an imitation learning algorithm that separates action generation from computationally intensive steps like posterior sampling and optimization. Our algorithm maintains an explicit policy representation that emulates the TS 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 $\mathcal{M}$ (e.g. a neural network that takes as input a context and outputs a distribution over actions) and generate actions by sampling from the current policy $A_t \sim \pi_t^m(\cdot \mid S_t)$. 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 TS policy by computing the maximizer $A_t^{\mathrm{TS}}(s) \in \mathrm{argmax}_{a \in \mathcal{A}}\, f_{\theta_t}(a, s)$, 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 TS 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 TS policy at time t as $\pi_t^{\mathrm{TS}}(a \mid s)$. This policy is different from the true, on-policy TS since the imitation policy generates actions. Dropping the subscript t to simplify notation, the imitation learning problem that updates the agent's policy is

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

Imitation problem (1) finds a model $m \in \mathcal{M}$ that minimizes a measure of discrepancy $D(\cdot, \cdot \mid S)$ 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 a notion of discrepancy between probabilities and present finite-sample approximations based on observed contexts and simulated actions from the TS policy $\pi_t^{\mathrm{TS}}(a \mid s)$. 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 $D_{\mathrm{kl}}\left(q^1 \| q^2\right) := \int_{\mathcal{A}} \log \frac{dq^1/d\nu}{dq^2/d\nu}(a) dP(a)$, where we use $dq^1/d\nu$ and $dq^2/d\nu$ to denote Radon-Nikodym derivatives of q^1 and q^2 with respect to ν. For policies π^1 and π^2, let $D_{\mathrm{kl}}\left(\pi^1, \pi^2 \mid S\right) := D_{\mathrm{kl}}\left(\pi^1(\cdot \mid S) \| \pi^2(\cdot \mid S)\right)$, where we use π^1, π^2 to also denote their densities over $\mathcal{A}$. From the preceding display, the imitation problem (1) with $D(\cdot, \cdot \mid S) = D_{\mathrm{kl}}(\cdot, \cdot \mid S)$ is equivalent to a maximum log likelihood problem

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

We write $\mathbb{E}[\cdot] = \mathbb{E}_{S \sim \mathbb{P}_S, A^{\mathrm{TS}} \sim \pi^{\mathrm{TS}}(\cdot \mid S)}[\cdot]$ for simplicity. The data comprises of context-action pairs; contexts are generated under the marginal distribution $S \sim \mathbb{P}_S$ independent of everything else, and actions are generated/simulated from the TS policy $A^{\mathrm{TS}} \sim \pi^{\mathrm{TS}}(\cdot \mid S)$. The MLE problem (2) finds the model $m \in \mathcal{M}$ that maximizes the log likelihood of observing these actions.

The imitation objective $m \mapsto \mathbb{E}[\log \pi^m(A^{\rm TS} \mid S)]$ involves an expectation over the unknown marginal distribution of contexts $\mathbb{P}_S$ and actions generated by the TS policy $\pi^{\rm TS}(\cdot \mid S)$. 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 "unlabeled" 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 [33, 19].

Accommodating typical application scenarios, Algorithm 1 and its empirical approximation applies to settings where the agent has the ability to interact with the system concurrently. This is a practically important feature of the algorithm, since most applications require the agent to concurrently generate actions, often with large batch sizes. Using the imitation policy $\pi^m(a \mid s)$, it is trivial to parallelize action generation over multiple computing nodes, even on mobile devices. Although our theoretical developments focus on the non-concurrent case for ease of exposition, our regret bounds can be extended to the batch setting; our experiments in Section 5 present large batch scenarios to illustrate typical application scenarios. While we restrict discussion to TS in this work, offline imitation can learn a explicit policy representation of any complicated policy and allow operationalization at scale.

For continuous actions with geometry, it may be natural to allow imitation policies to have slightly different support than the Thompson sampling policy. We can instantiate Algorithm 1 with Wasserstein distances as our notion of discrepancy $D\left(\cdot, \cdot \mid s\right)$. The subsequent theoretical development for KL divergences has its analogue for Wasserstein distances, which we outline in Section A.

4 Imitation Controls Regret

In this section, we show that minimizing the KL divergence (1) between TS and the imitation policy allows control on 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 show that Algorithm 1 and its empirical approximation enjoys the similar optimality guarantees as TS. 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 TS policy would have generated. In this sense, our imitation algorithm does not emulate the *true* TS 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 off-policy imitation is sufficient to achieve the same optimal regret bounds available for *on-policy* TS [47]. In particular, our results guard against potential compounding of errors resulting from imitating the off-policy variant of TS.

We relate the performance of our imitation policy with that of the *off-policy* TS, $\pi^{\rm TS}$, and show that $\pi^{\rm TS}$ admits a Bayes regret decomposition that allows us to show that it still achieves same optimal Bayes regret as the (on-policy) TS policy. This allows proving Bayes regret bounds for the imitation policy by utilizing existing proofs for bounding regret of UCB algorithms. We build on the flexible approach of Russo and Van Roy [47], and extend it to imitation policies that emulate off-policy TS.

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 TS policy can exacerbate over time. A suboptimal sequence of actions taken by the imitation policy may deteriorate the performance of the off-policy TS policy $\pi^{\rm TS}$ updated based on this data. Since the imitation policy again mimics the 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 the sum of expected discrepancy between the off-policy TS policy and the imitation learner at each period. Our bounds imply that each single-period approximation error does not affect the Bayes regret linearly in T as our worst-case intuition suggests, but only as a one-time approximation cost. To achieve near-optimal regret, it thus suffices at each period to control the imitation objective (1) to the off-policy TS policy $\pi_t^{\rm TS}$; in Section C, we show that the imitation objective can be efficiently optimized using approximations based on unlabeled contexts.

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). Russo and Van Roy [47] showed that a

TS algorithm admits a Bayes regret decomposition with respect to *any* UCB sequence. This allows leveraging arguments that bound the regret of a UCB algorithm to bound the Bayes regret of TS. Since the Bayes regret decomposition for TS holds for *any* UCB sequence, the performance of TS enjoys Bayes regret guarantees of the best UCB algorithm. We can 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 TS, 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 D.1 for the proof of the following result.

Lemma 1. *Let $\{\pi_t\}_{t\in\mathbb{N}}$ be any sequence of policies (adapted to history), 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 (H_t, S_t). Let there be a sequence $M_t(H_t, S_t)$ and $L > 0$ such that $\sqrt{\mathbb{E}[M_t(H_t, S_t)^2]} \le L$, and $\sup_{a\in\mathcal{A}} |U_t(a; H_t, S_t)| \le M_t(H_t, S_t)$. Then, for all $T \in \mathbb{N}$,* $\mathrm{BayesRegret}\,(T, \{\pi_t\}_{t\in\mathbb{N}})$ *is bounded by*

$$\sum_{t=1}^{T} \mathbb{E}[f_\theta(A_t^\star, S_t) - U_t(A_t^\star; H_t, S_t)] + \sum_{t=1}^{T} \mathbb{E}[U_t(A_t; H_t, S_t) - f_\theta(A_t, S_t)] + L \sum_{t=1}^{T} \sqrt{2\mathbb{E}\left[D_{\mathrm{kl}}\left(\pi_t^{\mathrm{TS}}, \pi_t \mid S_t\right)\right]}.$$

The above Bayes regret decomposition 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. The first two terms in the regret decompositions can be bounded using canonical UCB proofs following the approach of Russo and Van Roy [47]. We detail such arguments in concrete modeling scenarios in the next subsection, where our results draw from regret guarantees for UCB algorithms [1, 2, 51]. The last term in the decomposition is controlled by our imitation learning algorithm (Algorithm 1) and its empirical approximation; we show in Section C that the cheap availability of unlabeled contexts allows tight control of imitation errors, with a near-optimal dimension dependence.

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 compound over time in a linear manner. It remains open whether certain problem structures can provably preclude these negative feedback cycles uniformly over θ, which is necessary for obtaining frequentist regret bounds under imitation.

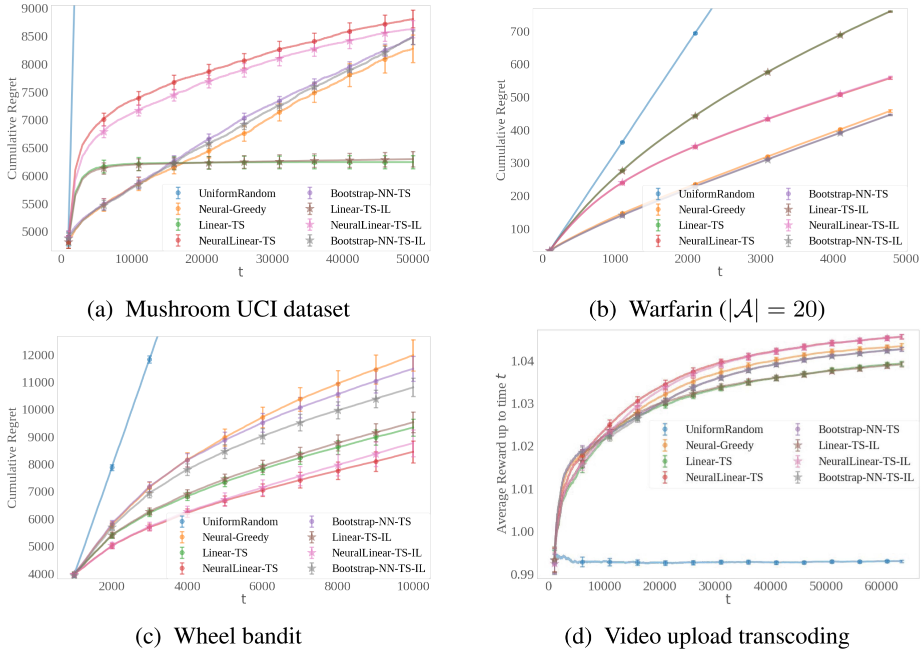
Regret bounds Building on the Bayes regret decomposition, we now show concrete regret guarantees for our imitation algorithm. By leveraging analysis of UCB algorithms, we proceed by bounding the first two sums in the decomposition. Our bounds on the Bayes regret are instance-independent (gap-independent),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

Let the mean reward function $f_\theta(a, s)$ be modeled as a generalized linear model. We assume $\Theta \subseteq \mathbb{R}^d$, 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 $f_\theta(a, s) = g(\theta^\top \phi(a, s))$ for all $\theta \in \Theta, a \in \mathcal{A}, s \in \mathcal{S}$. Armed with the regret decomposition in Lemma 1, we obtain regret bounds by following an eluder dimension argument pioneered by Russo and Van Roy [47]. We give its proof in Section D.2.

Theorem 1. *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 $\sup_{\theta\in\Theta} \|\theta\|_2 \le c_1$, and $\sup_{a\in\mathcal{A}, s\in\mathcal{S}} \|\phi(a, s)\|_2 \le c_2$, 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*

$$\mathrm{BayesRegret}\,(T, \{\pi_t\}_{t\in\mathbb{N}}) \le C(\sigma + 1) r d \sqrt{T} \log rT + c_1 c_2 \textstyle\sum_{t=1}^{T} \sqrt{2\mathbb{E}\left[D_{\mathrm{kl}}\left(\pi_t^{\mathrm{TS}}, \pi_t \mid S_t\right)\right]}. \quad (3)$$

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 [46]; for generalized linear models, the first two terms are the tightest bounds on the regret available. We conclude that controlling the imitation error directly controls Bayes regret. As we present in Section C, solving the empirical approximation to the imitation problem (1) with a sufficiently rich imitation model class $m \in \mathcal{M}$ guarantees low imitation errors in typical application scenarios.



(a) Mushroom UCI dataset

(b) Warfarin ($|\mathcal{A}| = 20$)

(c) Wheel bandit

(d) Video upload transcoding

Figure 1. For all benchmark problems, we report the mean cumulative regret (or running average regret up to time t for the video upload transcoding application) and two standard errors of the mean over 50 trials (except for the Wheel which we run for 100 trials due the rare large rewards).

	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)

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.

5 Empirical Evaluation and Discussion

We now empirically demonstrate our imitation algorithm on real and simulated problems. As our main real-world application, we consider a internet service receiving millions of video upload requests per day. We learn policies that decide how to transcode a video at upload time, where latency of decisions is important to the quality of service and keeping the user engaged. We observe that our IL algorithm achieves regret comparable to that of TS, while significantly reducing latency on all problems. We begin by providing a brief description of benchmark problems, deferring their details to Section F.3.

MUSHROOM is a UCI dataset [18] containing 8,124 examples with 22 categorical features about the mushroom and binary labels (poisonous or not). The agent decides whether to eat the mushroom or not and receives a stochastic reward: with equal probability, eating a poisonous mushroom lead to illness ($R_t = -35$) or no harm ($R_t = 5$), while a nonpoisonous mushroom is always harmless ($R_t = 5$). The reward for abstaining is always $R_t = 0$. We use $50,000$ contexts for each trial. **WARFARIN** is a pharmacological dosage optimization example, where we learn optimal personalized dosages for Warfarin, the most widely used anticoagulant in the US. The dataset [60, 65] contains optimal dosages for $4,788$ patients, alongside 17-dimensional genetic and demographic features. We construct a CB problem with 20 discrete dosage levels as actions, where the reward is the difference between the patient's prescribed and optimal dosage. We reshuffle contexts for each trial; similar results hold when $50,000$ contexts are re-sampled each trial and 50 actions are used (Appendix

F). As our third benchmark, to evaluate our methods on examples where exploration matters, we consider the **Wheel** problem, a 2-dimensional synthetic problem specifically designed to emphasize exploration [45]. There are 5 actions and rarely seen contexts yield high rewards under one context-dependent action. We sample $10,000$ contexts for each trial. As our last experiment, we consider a real-world video upload transcoding optimization problem (**VIDEO TRANSCODING**). This logged dataset contains 8M observations of video upload transcoding decisions collected by Facebook under a uniform random policy. The 38-dimensional contexts contain information about the video and network connection, and 7 actions corresponding to different video codec qualities. The rewards for successful uploads are positive and are monotonically increasing with quality. The reward is zero if the upload fails. We evaluate the performance of different contextual bandit algorithms using the rejection sampling technique proposed by Li et al. [34]. We sample $50,000$ contexts for each trial.

We consider a variety of models among those reported by Riquelme et al. [45] to perform the best in a broad range of benchmark problems. The hyperparameters for the Thompson sampling algorithms are from Riquelme et al. [45], which we detail in Appendix F.2 along with those for the imitation algorithm. Policies are updated every 1000 contexts (except for on the Warfarin problem, where we update policies every 100 contexts due to the small size of the dataset) and are initialized using a uniform random policy before the first batch update. **LINEAR-TS** uses a exact Bayesian linear regression to model the reward distribution for each action a. Exact posterior samples are used 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 F.2 for details). **NEURALLINEAR-TS** models rewards using a neural network with two 100-unit hidden layers and ReLU activations, 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 has an output for each action corresponding to the reward under 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. **TS-IL** imitates a TS policy using a fully-connected neural network as its policy representation. The policy network has two 100-unit hidden layers and tanh activations, and a final soft-max layer. We compare TS and its imitation counterparts against two additional methods: a random policy (**UNIFORMRANDOM**) and a greedy policy that uses a feed-froward neural network (**NEURAL-GREEDY**).

Figure 1 shows that each TS-IL method achieves comparable performance to its corresponding vanilla TS algorithm on each respective benchmark problem. In terms of cumulative performance, each TS-IL policy consistently matches its corresponding TS policy over time. For all methods, we also evaluate decision-time latency—the time required for a policy to select an action when it is queried—and time complexity for online action-generation. While BOOTSTRAP-NN-TS achieves low prediction latency, it requires storing many replicates of the neural network and can significantly increase the memory footprint. 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. TS-IL simply requires a forward pass through policy network and a cheap multinomial sample. Latency and complexity may be even greater under inference schemes not considered here (see Appendix G for a discussion).

Discussion In this paper, we demonstrated that Thompson sampling via imitation learning is a simple, practical, and efficient method for batch Thompson sampling with desirable regret properties. By distilling the Thompson sampling policy into easy-to-deploy explicit policy representations (e.g. small 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 Bayesian approaches in large-scale contextual bandit problems.

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

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[10] 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.

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

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

[13] S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification: a survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.

[14] S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: a Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

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

[16] 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.

[17] 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.

[18] D. Dua and C. Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

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

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

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

[22] 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.

[23] 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.

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

[25] 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.

[26] 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.

[27] D. N. Hill, H. Nassif, Y. Liu, A. Iyer, and S. Vishwanathan. An efficient bandit algorithm for realtime multivariate optimization. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug 2017. doi: 10.1145/3097983.3098184. URL `http://dx.doi.org/10.1145/3097983.3098184`.

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

[29] T. Joachims, A. Swaminathan, and M. de Rijke. Deep learning with logged bandit feedback. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=SJaP_-xAb`.

[30] 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.

[31] 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.

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

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

[34] 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.

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

[36] 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.

[37] 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.

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

[39] 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.

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

[41] 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.

[42] 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/`.

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

[44] D. Pollard. Empirical processes: theory and applications. In *NSF-CBMS regional conference series in probability and statistics*, pages i–86. JSTOR, 1990.

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

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

[47] D. Russo and B. Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

[48] 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.

[49] 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.

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

[51] 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.

[52] 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/`.

[53] A. Swaminathan and T. Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *Journal of Machine Learning Research*, 16(52):1731–1755, 2015. URL `http://jmlr.org/papers/v16/swaminathan15a.html`.

[54] 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.

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

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

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

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

[59] 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.

[60] M. Whirl-Carrillo, E. McDonagh, J. Hebert, l. Gong, K. Sangkuhl, C. Thorn, R. Altman, and T. Klein. Pharmacogenomics knowledge for personalized medicine. *Clinical pharmacology and therapeutics*, 92:414–7, 10 2012. doi: 10.1038/clpt.2012.96.

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

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

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

[64] 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.

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

[66] 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.

Appendix to:

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

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 actions taken by the respective policies. In this section, we provide an alternative instantiation of the abstract form of Algorithm 1 that uses Wasserstein distances as the notion of discrepancy $D\left(\cdot, \cdot \mid s\right)$ instead of the KL divergence. Our previous theoretical development for KL divergences has its analogues here, 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

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

where $\Gamma(q^1, q^2)$ denotes the collection of all probabilities on $\mathcal{A} \times \mathcal{A}$ with marginals q^1 and q^2 (i.e., couplings). Intuitively, $D_{\rm w}\left(q^1, q^2\right)$ 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. Wasserstein distances encode the geometry of the underlying space $\mathcal{A}$ via the distance d. Unlike the KL divergence $D_{\rm kl}\left(q^1 \| q^2\right)$ that take value ∞ whenever q^1 has support not contained in q^2, the Wasserstein distance allows the imitation policy to have slightly different support than the Thompson sampling policy. For a discrete action space, $D_{\rm w}\left(\cdot, \cdot\right)$ can be defined with any symmetric matrix $d(a_i, a_j)$ satisfying $d(a_i, a_j) \geq 0$ with 0 iff $a_i = a_j$, and $d(a_i, a_j) \leq d(a_i, a_k) + d(a_k, a_j)$ for any $a_i, a_j, a_k \in \mathcal{A}$. As before, to simplify notation, we let

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

for two policies π^1 and π^2.

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

$$\underset{m \in \mathcal{M}}{\text{minimize}}\, \mathbb{E}_{S \sim \mathbb{P}_S}\left[D_{\rm w}\left(\pi^{\rm TS}, \pi^m \mid S\right)\right]. \tag{4}$$

To solve the above stochastic optimization problem, we can again use stochastic gradient descent methods, where the stochastic gradient $\nabla_m D_{\rm w}\left(\pi_t^{\rm TS}, \pi^m \mid S\right)$ can be computed by solving an optimal transport problem. From Kantorovich-Rubinstein duality (see, for example, [57]), we have

$$\begin{aligned} &D_{\rm w}\left(\pi_t^{\rm TS}, \pi^m \mid s\right) \\ &= \sup_{g: \mathcal{A} \to \mathbb{R}} \left\{\mathbb{E}_{A \sim \pi^{\rm TS}(\cdot \mid s)} g(a) - \mathbb{E}_{A \sim \pi^m(\cdot \mid s)} g(a) :\ g(a) - g(a') \leq d(a, a') \text{ for all } a, a' \in \mathcal{A}\right\}, \end{aligned} \tag{5}$$

where $d(\cdot, \cdot)$ is the metric on $\mathcal{A}$ used to define $D_{\rm w}\left(\cdot, \cdot\right)$. For discrete action spaces, the maximization problem (5) 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. [43] for a comprehensive introduction to computational methods for solving optimal transport problems.

Letting $g^\star$ denote the optimal solution to the dual problem (5), the envelope theorem (or Danskin's theorem)—see Bonnans and Shapiro [12, Theorem 4.13]—implies that under simple regularity conditions

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

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

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

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

$$-g^\star(A)\nabla_m \log \pi^m(A \mid S_i) \tag{6}$$

is a stochastic gradient for the imitation problem (4). As before, we can get lower variance estimates by average the above estimator over many actions $A \sim \pi^m(\cdot \mid S_i)$.

Using stochastic gradients (6), we can solve the imitation problem (4) 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 4 have their natural analogues with Wasserstein distances replacing KL divergences as the notion of discrepancy, though we omit them for brevity.

Lemma 2. *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*

$$|U_t(a; H_t, S_t) - U_t(a'; H_t, S_t)| \le Ld(a, a') \text{ for all } a, a' \in \mathcal{A} \text{ almost surely.} \tag{7}$$

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

$$\begin{aligned}\text{BayesRegret}\,(T, \{\pi_t\}_{t\in\mathbb{N}}) \le \sum_{t=1}^{T} \mathbb{E}[f_\theta(A_t^\star, S_t) - U_t(A_t^\star; H_t, S_t)] + \sum_{t=1}^{T} \mathbb{E}[U_t(A_t; H_t, S_t) - f_\theta(A_t, S_t)] \\ + L\sum_{t=1}^{T} \mathbb{E}\left[D_{\mathrm{w}}\left(\pi_t^{\mathrm{TS}}, \pi_t \mid S_t\right)\right]. \end{aligned} \tag{8}$$

where $D_{\mathrm{w}}(\cdot, \cdot \mid \cdot)$ is the Wasserstein distance defined with the metric d in the condition (7).

Proof Proof. The proof mirrors that of Lemma 2, but bound the differences (13) by $D_{\mathrm{w}}\left(\pi_t^{\mathrm{TS}}, \pi_t \mid S_t\right)$. By the Kantorovich dual representation (5), 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 (12), and taking expectation over (H_t, S_t) on both sides and summing $t = 1, \ldots, T$, we get the desired bound. □ □

B 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)

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

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 [61, 62, 63].

As before, we measure performance by using the Bayes regret, averaging outcomes over the prior P. We build on the UCB regret bound due to Srinivas et al. [51] and bound the first two terms in the Bayes regret decomposition (Lemma 1). 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: $I(X,Y) := D_{\text{kl}}\left(P_{X,Y} \| P_X \times P_Y\right)$. We define the maximal possible information gain after T time steps as

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

where $y_X = \{f_\theta(x)\}_{x \in X}$ and $f_X = \{f_\theta(x)\}_{x \in X}$. For popular Gaussian and Matern kernels, Srinivas et al. [51] 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 in Lemma 1 can be bounded by $O(d\gamma_t T \log T)$, 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 2. *Let $\mathcal{A} \subseteq [0,r]^d$ for some $r > 0$. Assume that*

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

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

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

See Section D.4 for the proof.

To instantiate Theorem 2, it remains to bound 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 3 (Theorem 5, Ghosal et al. [24])**.** *If $\mu(\cdot)$ and $k(\cdot,\cdot)$ 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. [51], focusing on the popular Gaussian and Matern kernels

$$k_g(x,x') := \exp\left(-\frac{\|x - x'\|^2}{2l^2}\right),$$
$$k_m(x,x') := \frac{2^{1-\nu}}{\Gamma(\nu)} r^\nu B_\nu(r) \;\text{ where } r = \frac{\sqrt{2\nu}}{l}\|x - x'\|,$$

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

$$\mathfrak{M}(k_g,T) := (\log T)^{\kappa+1} \;\text{ and }\; \mathfrak{M}(k_m,T) := T^{\frac{\kappa^2+\kappa}{\kappa^2+\kappa+2\nu}} \log T.$$

We have the following bound on γ_T for Gaussian and Matern kernels; the bound is a direct consequence of Theorem 2, [32] and Theorem 5, [51].

Lemma 4. *Let $\mathcal{A} \subseteq \mathbb{R}^d$ and $\mathcal{S} \subseteq \mathbb{R}^{d'}$ be convex and compact. Let the kernel k be given by the sum of two kernels k_A and k_S on $\mathcal{A}$ and $\mathcal{S}$ respectively*

$$k((a,s),(a',s')) = k_A(a,a') + k_S(s,s').$$

If k_A and k_S are either the Gaussian kernel k_g or the Matern kernel k_m with $\nu > 1$, then

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

For example, taking $k_A = k_g$ and $k_S = k_g$, we conclude

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

C Generalization guarantees for imitation learning

So far, we showed that in order to achieve good regret, it suffices to control the KL-divergence between the imitation and (off-policy) Thompson sampling policy in order. We now show that each of these terms can be optimized efficiently using finite-sample approximations; we are interested in how well the model learned from an empirical approximation of the imitation problem (1) performs with respect to the true imitation objective (KL divergence). Since we consider the problem for any fixed time step t, we omit the subscript t and denote $\pi^{\text{TS}} = \pi_t^{\text{TS}}$. Recalling that N denotes the number of observed "unlabeled" contexts $S_1, \ldots, S_N$, we simulate N_a number of 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 the imitation learning objective $\mathbb{E}[D_{\text{kl}}\left(\pi^{\text{TS}}, \pi^m \mid S\right)]$ is proportional to $-\mathbb{E}_{S,A^{\text{TS}}\sim\pi^{\text{TS}}(\cdot|S)}[\log \pi^m(A^{\text{TS}} \mid S)]$, we are interested in solving the following empirical approximation to the population problem (1)

$$\widehat{m}_{N,N_a} \in \operatorname*{argmax}_{m\in\mathcal{M}} \frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_a}\log \pi^m(A_{ij}^{\text{TS}} \mid S_i). \tag{9}$$

"Unlabeled" contexts without corresponding action-reward information are often cheap and abundant in internet applications, and we can take N to be very large. For any observed context S_i, the actions $A^{\text{TS}} \sim \pi^{\text{TS}}(\cdot \mid S_i)$ can be generated by posterior sampling. Since this can be done *offline*, and is trivial to parallelize per S_i, we can generate many actions; hence we usually have very large N_a as well.

To make our results concrete, we rely on standard notions of complexity to measure the size of the imitation model class $\mathcal{M}$, using familiar notions based on Rademacher averages. For a sample $\xi_1, \ldots, \xi_n$ 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

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

For example, when $g \in \mathcal{G}$ takes values in $[-M, M]$ with VC-subgraph dimension d, a standard bound is $\mathbb{E}[\mathfrak{R}_n(\mathcal{G})] \lesssim M\sqrt{\frac{d}{n}}$; see Chapter 2 of van der Vaart and Wellner [56] and Bartlett and Mendelson [9] for a comprehensive treatment.

In what follows, we show that the empirical minimizer $\widehat{m}_{N,N_a}$ achieves good performance optimum with respect to the population problem (1). More concretely, we show that

$$\mathbb{E}\left[D_{\text{kl}}\left(\pi^{\text{TS}}, \pi^{\widehat{m}_{N,N_a}} \mid S\right)\right] \le \inf_{m\in\mathcal{M}}\mathbb{E}\left[D_{\text{kl}}\left(\pi^{\text{TS}}, \pi^m \mid S\right)\right] + N^{-1/2}\left(\mathfrak{T}_1(\mathcal{M}) + N_a^{-1/2}\mathfrak{T}_2(\mathcal{M})\right)$$

where $\mathfrak{T}_1(\mathcal{M})$ and $\mathfrak{T}_1(\mathcal{M})$ are problem-dependent constants that measure the complexity of the imitation model class $\mathcal{M}$. We show that the dominating dimension-dependent constant $\mathfrak{T}_1(\mathcal{M})$ term is in a sense the best one can hope for, matching the generalization guarantee available for the idealized scenario where KL-divergence $D_{\text{kl}}\left(\pi^{\text{TS}}, \pi^m \mid S_i\right)$ can be computed and optimized exactly for each S_i, $i = 1, \ldots, N$.

We begin by first illustrating this "best-case scenario", where we can generate an infinite number of actions (i.e. $N_a = \infty$). We consider the solution $\widehat{m}_{N,\infty}$ to the idealized empirical imitation learning problem where the KL divergence between the imitation policy and the Thompson sampler can be computed (and optimized) exactly. Formally, we let

$$\widehat{m}_{N,\infty} \in \operatorname*{argmin}_{m\in\mathcal{M}}\frac{1}{N}\sum_{i=1}^{N}D_{\text{kl}}\left(\pi^{\text{TS}}, \pi^m \mid S_i\right) = \operatorname*{argmax}_{m\in\mathcal{M}}\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{A^{\text{TS}}\sim\pi^{\text{TS}}(\cdot|S_i)}[\log \pi^m(A^{\text{TS}} \mid S)].$$

The Rademacher complexity of the following set of functions controls generalization performance of $\widehat{m}_{N,\infty}$

$$\mathcal{G}_1 := \left\{ s \mapsto \mathbb{E}_{A^{\text{TS}} \sim \pi^{\text{TS}}(\cdot|s)}[\log \pi^m(A^{\text{TS}} \mid s)] : m \in \mathcal{M} \right\}.$$

Lemma 5. *Let $\widehat{m}_{N,\infty}$ be defined as above. If $|\log \pi^m(a \mid s)| \leq M$ for all $a \in \mathcal{A}, s \in \mathcal{S}, m \in \mathcal{M}$, then with probability at least $1 - 2e^{-t}$*

$$\mathbb{E}\left[D_{\text{kl}}\left(\pi^{\text{TS}}, \pi^{\widehat{m}_{N,\infty}} \mid S \right) \right] \leq \inf_{m \in \mathcal{M}} \mathbb{E}\left[D_{\text{kl}}\left(\pi^{\text{TS}}, \pi^m \mid S \right) \right] + 4\mathbb{E}[\mathfrak{R}_N(\mathcal{G}_1)] + 2M\sqrt{\frac{2t}{N}}.$$

This lemma follows from a standard concentration argument, which we present in Section E.1 for completeness.

We now show that the empirical approximation (9) enjoys a similar generalization performance as $\widehat{m}_{N,\infty}$, so long as N_a is moderately large. To give our result, we define two additional sets of functions

$$\begin{aligned} \mathcal{G}_2(s) &:= \{ a \mapsto \log \pi^m(a \mid s) : m \in \mathcal{M} \} \\ \mathcal{G}_3 &:= \{ (a, s) \mapsto \log \pi^m(a \mid s) : m \in \mathcal{M} \}. \end{aligned}$$

For $\mathcal{G}_3$, we abuse notation slightly and write

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

for i.i.d. random signs ε_{ij}. The following lemma, whose proof we give in Section E.2, shows that the empirical solution (9) generalizes at a rate comparable to the idealized model $\widehat{m}_{N,\infty}$.

Theorem 3. *Let $|\log \pi^m(a \mid s)| \in M$ for all $a \in \mathcal{A}, s \in \mathcal{S}, m \in \mathcal{M}$. Then, with probability at least $1 - 3e^{-t}$,*

$$\begin{aligned} \mathbb{E}\left[D_{\text{kl}}\left(\pi^{\text{TS}}, \pi^{\widehat{m}_{N,N_a}} \mid S \right) \right] \leq \inf_{m \in \mathcal{M}} \mathbb{E}\left[D_{\text{kl}}\left(\pi^{\text{TS}}, \pi^m \mid S \right) \right] + 4\mathbb{E}[\mathfrak{R}_N(\mathcal{G}_1)] + 2M\sqrt{\frac{2t}{N}} \\ + \sqrt{\frac{32t}{N}} \sup_{s \in \mathcal{S}} \mathbb{E}_{A_j^{\text{TS}} \overset{\text{iid}}{\sim} \pi^{\text{TS}}(\cdot|s)}[\mathfrak{R}_{N_a}(\mathcal{G}_2(s))] + 8\mathbb{E}[\mathfrak{R}_{N N_a}(\mathcal{G}_3)] \end{aligned}$$

Recalling the standard scaling $\mathbb{E}[\mathfrak{R}_n(\mathcal{G})] \lesssim M\sqrt{\frac{d}{n}}$, we see that $\widehat{m}_{N,N_a}$ achieves performance comparable to the idealized solution $\widehat{m}_{N,\infty}$, up to an $O(N^{-1/2} N_a^{-1/2})$-error term. Although we omit it for brevity, boundedness of $\log \pi^m(a \mid s)$ can be relaxed to sub-Gaussianity by using standard arguments (see, for example, Chapter 2.14 [56]).

We now provide an application of the theorem. **Example 1:** Let $\mathcal{V}$ be a vector space, and $V \subset \mathcal{V}$ be any collection of vectors in $\mathcal{V}$. Let $\|\cdot\|$ be a (semi)norm on $\mathcal{V}$. A collection $v_1, \ldots, v_N \subset \mathcal{V}$ is an ϵ-*cover* of V if for each $v \in V$, there exists v_i such that $\|v - v_i\| \leq \epsilon$. The *covering number* of V with respect to $\|\cdot\|$ is then

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

Letting $\mathcal{G}$ be a collection of functions $g : \mathcal{X} \to \mathbb{R}$, a standard argument due to Pollard [44] yields

$$\mathbb{E}_\varepsilon\left[\frac{1}{n} \sup_{g \in \mathcal{G}} \sum_{i=1}^{n} \varepsilon_i g(\xi_i) \right] \lesssim \inf_{\delta \geq 0} \left\{ \delta + \frac{1}{\sqrt{n}} \sqrt{\log N(\mathcal{G}, \delta, \|\cdot\|_{L^2(P_n)})} \right\} \qquad (10)$$

where P_n denotes the point masses on $\xi_1, \ldots, \xi_n$ and $\|\cdot\|_{L^2(P_n)}$ is the empirical L^2-norm on functions $g : \Xi \to [-M, M]$.

Let $\mathcal{M} \subset \mathbb{R}^{d_m}$ and assume that $m \mapsto \log \pi^m(a \mid s)$ is C-Lipschitz with respect to the ℓ_2-norm for all $a \in \mathcal{A}, s \in \mathcal{S}$ so that

$$|\log \pi^m(a \mid s) - \log \pi^{m'}(a \mid s)| \leq C \|m - m'\|_2.$$

Any ϵ-covering $\{m_1, \ldots, m_N\}$ of $\mathcal{M}$ in ℓ_2-norm yields $\min_i |\log \pi^m(a \mid s) - \log \pi^{m_i}(a \mid s)| \le C\epsilon$ for all $m \in \mathcal{M}, a \in \mathcal{A}, s \in \mathcal{S}$. This implies that ℓ_2-covering numbers of $\mathcal{M}$ control L^∞-covering numbers of the set of functions $\mathcal{G}_1, \mathcal{G}_2(s), \mathcal{G}_3$:

$$\max\left\{N(\mathcal{G}_1, \epsilon, L^\infty), \sup_{s\in\mathcal{S}} N(\mathcal{G}_2(s), \epsilon, L^\infty), N(\mathcal{G}_3, \epsilon, L^\infty)\right\} \le N(\mathcal{M}, \epsilon/C, \|\cdot\|_2) \le \left(1 + \frac{\mathrm{diam}(\mathcal{M})C}{\epsilon}\right)^{d_m},$$

where $\mathrm{diam}(\mathcal{M}) = \sup_{m,m\in\mathcal{M}} \|m - m'\|_2$. In Pollard's discretization-based bound (10), setting $\delta = \mathrm{diam}(\mathcal{M})CN^{-1}$ yields

$$\mathbb{E}[\mathfrak{R}_N(\mathcal{G}_1)] \lesssim \sqrt{\frac{d_m}{N}} + \frac{\mathrm{diam}(\mathcal{M})C}{N}.$$

Plugging this bound in Lemma 5, the idealized empirical model $\widehat{m}_{N,\infty}$ achieves

$$\mathbb{E}\left[D_{\mathrm{kl}}\left(\pi^{\mathrm{TS}}, \pi^{\widehat{m}_{N,\infty}} \mid S\right)\right] \lesssim \inf_{m\in\mathcal{M}} \mathbb{E}\left[D_{\mathrm{kl}}\left(\pi^{\mathrm{TS}}, \pi^m \mid S\right)\right] + \sqrt{\frac{d_m}{N}} + \frac{\mathrm{diam}(\mathcal{M})C}{N} \tag{11}$$

with probability at least $1 - 2e^{-t}$, where $\lesssim$ denotes an inequality up to some universal constant.

We now show that $\widehat{m}_{N,N_a}$ achieves a similar generalization guarantee as the idealized model $\widehat{m}_{N,\infty}$. Using the bound (10), we again get

$$\begin{aligned}
&\sup_{s\in\mathcal{S}} \mathbb{E}_{A_j^{\mathrm{TS}} \overset{\mathrm{iid}}{\sim} \pi^{\mathrm{TS}}(\cdot|s)}[\mathfrak{R}_{N_a}(\mathcal{G}_2(s))] \lesssim \sqrt{\frac{d_m}{N_a}} + \frac{\mathrm{diam}(\mathcal{M})C}{N_a}, \\
&\mathbb{E}[\mathfrak{R}_{NN_a}(\mathcal{G}_3)] \lesssim \sqrt{\frac{d_m}{NN_a}} + \frac{\mathrm{diam}(\mathcal{M})C}{NN_a}.
\end{aligned}$$

Applying these bounds to Theorem 3, we see that $\widehat{m}_{N,N_a}$ enjoys the same guarantee (11) as the "best-case" idealized empirical solution $\widehat{m}_{N,\infty}$ (up to constants). $\diamond$

D Proof of regret bounds

D.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

$$\begin{aligned}
&\mathbb{E}[f_\theta(A_t^\star, S_t) - f_\theta(A_t, S_t) \mid H_t, S_t] \\
&= \mathbb{E}[f_\theta(A_t^\star, S_t) - U_t(A_t^\star; H_t, S_t) \mid H_t, S_t] + \mathbb{E}[U_t(A_t^{\mathrm{TS}}; H_t, S_t) - f_\theta(A_t, S_t) \mid H_t, S_t] \\
&= \mathbb{E}[f_\theta(A_t^\star, S_t) - U_t(A_t^\star; H_t, S_t) \mid H_t, S_t] + \mathbb{E}[U_t(A_t; H_t, S_t) - f_\theta(A_t, S_t) \mid H_t, S_t] \\
&\quad + \mathbb{E}[U_t(A_t^{\mathrm{TS}}; H_t, S_t) - U_t(A_t; H_t, S_t) \mid H_t, S_t].
\end{aligned} \tag{12}$$

We proceed by bounding the gap

$$\mathbb{E}[U_t(A_t^{\mathrm{TS}}; H_t, S_t) - U_t(A_t; H_t, S_t) \mid H_t, S_t] \tag{13}$$

by the KL divergence between π_t^{TS} and π_t. Recall Pinsker's inequality [55]

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

From the hypothesis, Pinsker's inequality implies

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

Applying this bound in the decomposition (12), and taking expectation over (H_t, S_t) on both sides and summing $t = 1, \ldots, T$, we get

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

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

D.2 Proof of Theorem 1

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 $\|v - v_i\| \leq \epsilon$. The *covering number* is

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

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 [47], which quantifies the size of the function class $\mathcal{F} = \{f_\theta(\cdot, \cdot) : \theta \in \Theta\}$ for sequential decision making problems.

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 $f, f' \in \mathcal{F}$*

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

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 6 (Russo and Van Roy [47])**.** *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*

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

and when condition (7) *holds, we have*

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

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

From Lemma 6, 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 [56]) gives

$$N\left(\mathcal{H}, \epsilon, \|\cdot\|_{L^\infty(\mathcal{X})}\right) \le N\left(\Theta, \frac{\epsilon}{c_2}, \|\cdot\|\right) \le \left(1 + \frac{2c_1c_2}{\epsilon}\right)^d.$$

Proposition 11, Russo and Van Roy [47] shows that

$$d_{\mathrm{E}}(\mathcal{F}, T^{-1}) \le Cdr^2 \log rT$$

for some constant C that depends only on c_1 and c_2. Using these bounds in Lemma 6, we obtain the result.

D.3 Explicit regret bounds for linear bandits

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 [17, 1, 2], instead of the eluder dimension argument used to show Theorem 1.

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. [17], Abbasi-Yadkori et al. [1, 2] for UCB algorithms, we obtaint he following result for linear contextual bandits.

Lemma 7. *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*

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

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

$$\begin{aligned}\mathrm{BayesRegret}\left(T, \{\pi_t\}_{t \in \mathbb{N}}\right) \le & 2\left((c_1 + 1)c_1 + \sigma\sqrt{d + \log\sqrt{T}\left(1 + \frac{c_2^2 T}{\lambda}\right)}\right)\sqrt{2Td\log\left(\lambda + \frac{Tc_2^2}{d}\right)} \\ & + 4c_1c_2\sqrt{T} + c_1c_2\sum_{t=1}^T \sqrt{2\mathbb{E}\left[D_{\mathrm{kl}}\left(\pi_t^{\mathrm{TS}}, \pi_t \mid S_t\right)\right]} \qquad (14)\end{aligned}$$

Furthermore, if $a \mapsto \phi(a, s)$ is L-Lipschitz with respect to a metric d, then the same bound holds with $L\sum_{t=1}^T \mathbb{E}\left[D_{\mathrm{w}}\left(\pi_t^{\mathrm{TS}}, \pi_t \mid S_t\right)\right]$ replacing the last sum, where $D_{\mathrm{w}}(\cdot, \cdot \mid \cdot)$ 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, [47] and [2]).

Proof

Lemma 7 follows from a direct consequence of Lemma 1, and Dani et al. [17], Abbasi-Yadkori et al. [1]; we detail it below for completeness. We first show the bound (14). Letting $L_t(a; H_t, S_t)$ be an arbitrary sequence of measurable functions denoting lower confidence bounds, the Bayes regret decomposition in Lemma 1 implies

$$\begin{aligned}\mathrm{BayesRegret}\left(T, \pi\right) \le & \sum_{t=1}^T \mathbb{E}[U_t(A_t; H_t, S_t) - L_t(A_t; H_t, S_t)] \\ & + 2c_1c_2\sum_{t=1}^T \Big\{\mathbb{P}(f_\theta(A_t, S_t) \le L_t(A_t; H_t, S_t)) + \mathbb{P}(f_\theta(A_t^\star, S_t) \ge U_t(A_t^\star; H_t, S_t))\Big\} \\ & + L\sum_{t=1}^T \sqrt{2\mathbb{E}\left[D_{\mathrm{kl}}\left(\pi_t^{\mathrm{TS}}, \pi_t \mid S_t\right)\right]}. \qquad (15)\end{aligned}$$

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. [17], Abbasi-Yadkori et al. [1].

Lemma 8 (Theorem 2, Abbasi-Yadkori et al. [1]). *Under the conditions of the proposition, for any $\delta > 0$*

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

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

To instantiate the decomposition (15), we let

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

We are now ready to bound the second term in the decomposition (15). On the event

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

we have $f_\theta(A_t, S_t) \geq L_t(A_t; H_t, S_t)$ and $f_\theta(A_t^\star, S_t) \leq U_t(A_t^\star; H_t, S_t)$ by definition. Since Lemma 8 states $\mathbb{P}(\mathcal{E} \mid \theta) \geq 1 - \delta$, we conclude that the second sum in the decomposition (15) is bounded by $4 c_1 c_2 T \delta$.

To bound the first sum in the decomposition (15), we use the following bound on the norm of feature vectors.

Lemma 9 (Lemma 11, Abbasi-Yadkori et al. [1]). *If $\lambda \geq c_2^2 \vee 1$, for any sequence of a_t, s_t for $t \geq 1$, and corresponding $A_t := \lambda I + \sum_{k=1}^{t} \phi(a_k, s_k)\phi(a_k, s_k)^\top$, we have*

$$\sum_{t=1}^{T} \|\phi(A_t, S_t)\|_{V_{t-1}^{-1}}^2 \leq 2d \log \left(\lambda + \frac{T c_2^2}{d} \right).$$

Noting that by definition

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

we obtain

$$\begin{aligned} \sum_{t=1}^{T} U_t(A_t; H_t, S_t) - L_t(A; H_t, S_t) &\leq 2 \sum_{t=1}^{T} \|\phi(A_t, S_t)\|_{V_{t-1}^{-1}} \beta_{t-1}(\delta) \\ &\overset{(a)}{\leq} 2\beta_T(\delta) \sum_{t=1}^{T} \|\phi(A_t, S_t)\|_{V_{t-1}^{-1}} \\ &\overset{(b)}{\leq} 2\beta_T(\delta) \sqrt{T \sum_{t=1}^{T} \|\phi(A_t, S_t)\|_{V_{t-1}^{-1}}^2} \\ &\overset{(c)}{\leq} 2\beta_T(\delta) \sqrt{2Td \log \left(\lambda + \frac{T c_2^2}{d} \right)} \end{aligned}$$

where we used monotonicity of $t \mapsto \beta_t(\delta)$ in step (a), Cauchy-Schwarz inequality in step (b), and Lemma 9 in step (c).

Collecting these bounds, we conclude

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

Setting $\delta = 1/\sqrt{T}$, we obtain the first result. The second result is immediate by starting with the decomposition (8) and using an identical argument.

D.4 Proof of Theorem 2

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

$$\begin{aligned}\mu_t(a, s) &:= \mathbb{E}[f_\theta(a, s) \mid H_t] = k_t(a, s)^\top(K_t + \sigma^2 I)^{-1}y_t, \\ \sigma_t^2(a, s) &:= \text{Var}(f_\theta(a, s) \mid H_t) = k((a, s), (a, s)) - k_t(a, s)^\top(K_t + \sigma^2 I)^{-1}k_t(a, s)\end{aligned}$$

where $k_t(a, s) := [k((A_j, S_j), (a, s))]_{1\le j\le t}$, $K_t := [k((A_i, S_i), (A_j, S_j))]_{1\le i,j\le t}$ 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 $\beta_t = 2\log((t^4 rd)^d t^2)$. Noting that

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

a minor modification to the proof of Lemma 1 yields

$$\begin{aligned}\text{BayesRegret}\,(T, \pi) \le & \sum_{t=1}^{T}\mathbb{E}[f_\theta(A_t^\star, S_t) - U_t(A_t^\star; H_t, S_t)] + \sum_{t=1}^{T}\mathbb{E}[U_t(A_t; H_t, S_t) - f_\theta(A_t, S_t)] \\ & + \sum_{t=1}^{T}\left(\left\|\mathbb{E}\left[\sup_{a\in\mathcal{A}, s\in\mathcal{S}} |f_\theta(a, s)| \,\Big|\, H_t\right]\right\|_{2,P} + \sqrt{\beta_t}c_2\right)\sqrt{2\mathbb{E}\left[D_{\text{kl}}\left(\pi_t^{\text{TS}}, \pi_t \mid S_t\right)\right]}.\end{aligned} \tag{16}$$

From Jensen's inequality and the tower property,

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

From Borell-TIS inequality (e.g., see [4]), we have $c_3 < \infty$.

We now proceed by bounding the first two terms in the regret decomposition (16). 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 $|\mathcal{A}_t| \le (t^4 rd)^d$, we have $2\log(|\mathcal{A}_t|t^2) \le \beta_t$. We begin by decomposing the first term in the decomposition.

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

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

$$\mathbb{E}[f_\theta(a, s) - U_t(a; H_t, s) \mid H_t] \le \mathbb{E}[(f_\theta(a, s) - U_t(a; H_t, s))_+ \mid H_t] = \frac{\sigma_t(a, s)}{\sqrt{2\pi}} e^{-\frac{\beta_t}{2}} \le \frac{c_2}{\sqrt{2\pi} t^2 |\mathcal{A}_t|}. \tag{17}$$

Hence, we obtain the bound

$$\sum_{t=1}^{T} \mathbb{E}[f_\theta([A_t^\star]_t, S_t) - U_t([A_t^\star]_t; H_t, S_t)] \le \sum_{t=1}^{T} \sum_{a \in \mathcal{A}_t} \mathbb{E}[f_\theta(a, S_t) - U_t(a; H_t, S_t)] \le \sum_{t=1}^{\infty} \frac{c_2}{\sqrt{2\pi} t^2} \le Cc_2$$

where we used the independence of S_t and H_t, and the bound (17).

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

$$\begin{aligned} |U_t(a; H_t, s) - U_t(a'; H_t, s)| \le \mathbb{E}[L_f \mid H_t] \, \|a - a'\|_1 \\ + \sqrt{\beta_t} \left(2\mathbb{E}\left[L_f \left(\sup_{a \in \mathcal{A}, s \in \mathcal{S}} \mu(a, s)^2 + \sup_{a \in \mathcal{A}, s \in \mathcal{S}} f_\theta(a, s)^2 \right) \mid H_t \right] \right)^{\frac{1}{2}} \|a - a'\|_1^{\frac{1}{2}}. \end{aligned} \tag{18}$$

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 (18), 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

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

which shows the bound (18).

Collecting these bounds, we have shown that

$$\sum_{t=1}^{T} \mathbb{E}[f_\theta(A_t^\star, S_t) - U_t(A_t^\star; H_t, S_t)] \le C\mathbb{E}[L_f] + Cc_2 + Cd \log(rd) \left(c_1 \sqrt{\mathbb{E}[L_f]} + c_3 \sqrt{\mathbb{E}[L_f^2]} \right). \tag{19}$$

To bound the second term in the Bayes regret decomposition (16), we use the following lemma due to Srinivas et al. [51].

Lemma 10 (Lemma 5.3 Srinivas et al. [51])**.** *For any sequence of A_t and S_t,*

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

Using the lemma, we have

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

Combining this with the bound (19), we obtain our result.

E Proof of generalization results

E.1 Proof of Lemma 5

We use the following standard concentration result based on the bounded differences inequality and a symmetrization argument; see, for example, [13, 58, 14]. We denote by $\widehat{P}_n$ the empirical distribution constructed from any i.i.d. sample $X_i \sim P$.

Lemma 11. *If $|g| \le M$ for all $g \in \mathcal{G}$, then with probability at least* $1 - 2e^{-t}$

$$\sup_{g\in\mathcal{G}} |\mathbb{E}[g(X)] - \mathbb{E}_{\widehat{P}_n}[g(X)]| \le 2\mathbb{E}[\mathfrak{R}_n(\mathcal{G})] + M\sqrt{\frac{2t}{n}}.$$

Noting that for any $m \in \mathcal{M}$

$$\begin{aligned}
&\mathbb{E}\left[D_{\rm kl}\left(\pi^{\rm TS}, \pi^{\widehat{m}_{N,\infty}} \mid S\right)\right] - \mathbb{E}[D_{\rm kl}\left(\pi^{\rm TS}, \pi^{m} \mid S\right)] \\
&= \mathbb{E}[\log \pi^m(A^{\rm TS} \mid S)] - \mathbb{E}[\log \pi^{\widehat{m}_{N,\infty}}(A^{\rm TS} \mid S)] \\
&= \mathbb{E}[\log \pi^m(A^{\rm TS} \mid S)] - \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{A^{\rm TS}\sim\pi^{\rm TS}(\cdot|S_i)}[\log \pi^m(A^{\rm TS} \mid S_i)] \\
&\qquad + \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{A^{\rm TS}\sim\pi^{\rm TS}(\cdot|S_i)}[\log \pi^m(A^{\rm TS} \mid S_i)] - \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{A^{\rm TS}\sim\pi^{\rm TS}(\cdot|S_i)}[\log \pi^{\widehat{m}_{N,\infty}}(A^{\rm TS} \mid S_i)] \\
&\qquad + \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{A^{\rm TS}\sim\pi^{\rm TS}(\cdot|S_i)}[\log \pi^{\widehat{m}_{N,\infty}}(A^{\rm TS} \mid S_i)] - \mathbb{E}[\log \pi^{\widehat{m}_{N,\infty}}(A^{\rm TS} \mid S)] \\
&\le 2\sup_{m\in\mathcal{M}} \left| \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{A^{\rm TS}\sim\pi^{\rm TS}(\cdot|S_i)}[\log \pi^m(A^{\rm TS} \mid S_i)] - \mathbb{E}[\log \pi^m(A^{\rm TS} \mid S)] \right|
\end{aligned}$$

where we used the fact that $\widehat{m}_{N,\infty}$ maximizes $\frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{A^{\rm TS}\sim\pi^{\rm TS}(\cdot|S_i)}[\log \pi^m(A^{\rm TS} \mid S_i)]$ in the last inequality. Applying Lemma 11 with $\mathcal{G} = \mathcal{G}_1$ and taking the infimum over $m \in \mathcal{M}$, we obtain the result.

E.2 Proof of Theorem 3

We begin by noting that

$$\begin{aligned}
&\mathbb{E}\left[D_{\rm kl}\left(\pi^{\rm TS}, \pi^{\widehat{m}_{N,N_a}} \mid S\right)\right] - \mathbb{E}[D_{\rm kl}\left(\pi^{\rm TS}, \pi^{m} \mid S\right)] = \mathbb{E}[\log \pi^m(A^{\rm TS} \mid S)] - \mathbb{E}[\log \pi^{\widehat{m}_{N,N_a}}(A^{\rm TS} \mid S)] \\
&= \mathbb{E}[\log \pi^m(A^{\rm TS} \mid S)] - \frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_A} \log \pi^m(A^{\rm TS}_{ij} \mid S_i) \\
&\qquad + \frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_A} \log \pi^m(A^{\rm TS}_{ij} \mid S_i) - \frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_A} \log \pi^{\widehat{m}_{N,N_a}}(A^{\rm TS}_{ij} \mid S_i) \\
&\qquad + \frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_A} \log \pi^{\widehat{m}_{N,N_a}}(A^{\rm TS}_{ij} \mid S_i) - \mathbb{E}[\log \pi^{\widehat{m}_{N,N_a}}(A^{\rm TS} \mid S)].
\end{aligned}$$

Since $\widehat{m}_{N,N_a}$ maximizes $\frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_A}\log\pi^m(A_{ij}^{\mathrm{TS}} \mid S_i)$, the preceeding display can be bounded by

$$\begin{aligned}
&2\sup_{m\in\mathcal{M}}\left|\frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_a}\log\pi^m(A_{ij}^{\mathrm{TS}} \mid S_i) - \mathbb{E}[\log\pi^m(A^{\mathrm{TS}} \mid S)]\right| \\
&\quad\le 2\sup_{m\in\mathcal{M}}\left|\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{A^{\mathrm{TS}}\sim\pi^{\mathrm{TS}}(\cdot|S_i)}[\log\pi^m(A^{\mathrm{TS}} \mid S_i)] - \mathbb{E}[\log\pi^m(A^{\mathrm{TS}} \mid S)]\right| \\
&\qquad + 2\sup_{m\in\mathcal{M}}\left|\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{N_a}\sum_{j=1}^{N_a}\log\pi^m(A_{ij}^{\mathrm{TS}} \mid S_i) - \mathbb{E}_{A^{\mathrm{TS}}\sim\pi^{\mathrm{TS}}(\cdot|S_i)}[\log\pi^m(A^{\mathrm{TS}} \mid S_i)]\right)\right|
\end{aligned} \tag{20}$$

We proceed by separately bounding the two terms in the inequality (20). From Lemma 11, the second term is bounded by

$$4\mathbb{E}[\mathfrak{R}_N(\mathcal{G}_1)] + 2M\sqrt{\frac{2t}{N}}$$

with probability at least $1-2e^{-t}$. To bound the first term

$$Z_{N,N_a} := 2\sup_{m\in\mathcal{M}}\left|\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{N_a}\sum_{j=1}^{N_a}\log\pi^m(A_{ij}^{\mathrm{TS}} \mid S_i) - \mathbb{E}[\log\pi^m(A^{\mathrm{TS}} \mid S_i)]\right)\right|,$$

consider the Doob martingale

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

with $M_0 = \mathbb{E}[Z_{N,N_a}]$, which is 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 $k\ge 1$. Let $\bar{S}_k$ be an independent copy of S_k that is independent of all $S_i, A_{ij}^{\mathrm{TS}}$ for $i\ne k$, and let $\bar{A}_{kj}^{\mathrm{TS}} \sim \pi^{\mathrm{TS}}(\cdot \mid \bar{S}_k)$ independent of everything other $\bar{S}_k$. We can write

$$\begin{aligned}
\frac{1}{2}|D_k| = \mathbb{E}&\left[\sup_{m\in\mathcal{M}}\left|\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{N_a}\sum_{j=1}^{N_a}\log\pi^m(A_{ij}^{\mathrm{TS}} \mid S_i) - \mathbb{E}[\log\pi^m(A^{\mathrm{TS}} \mid S_i)]\right)\right| \middle| S_1,\ldots,S_k\right] \\
&- \mathbb{E}\Bigg[\sup_{m\in\mathcal{M}}\Bigg|\frac{1}{N}\sum_{i\ne k}\left(\frac{1}{N_a}\sum_{j=1}^{N_a}\log\pi^m(A_{ij}^{\mathrm{TS}} \mid S_i) - \mathbb{E}[\log\pi^m(A^{\mathrm{TS}} \mid S_i)]\right) \\
&\qquad + \frac{1}{N}\left(\frac{1}{N_a}\sum_{j=1}^{N_a}\log\pi^m(\bar{A}_{kj}^{\mathrm{TS}} \mid \bar{S}_k) - \mathbb{E}[\log\pi^m(A^{\mathrm{TS}} \mid \bar{S}_k)]\right)\Bigg| \Bigg| S_1,\ldots,S_k\Bigg].
\end{aligned}$$

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

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

where $\bar{S}_k$ is an independent copy of S_k, and similarly $\bar{A}_{kj}^{\mathrm{TS}} \overset{\mathrm{iid}}{\sim} \pi^{\mathrm{TS}}(\cdot \mid \bar{S}_k)$.

Next, we use a standard symmetrization result to bound the preceding display; see, for example, Chapter 2.3, van der Vaart and Wellner [56] for a comprehensive treatment.

Lemma 12. *If $X_i \overset{\text{iid}}{\sim} P$, we have*

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

Applying Lemma 12 to the bound on $|D_k|$. we conclude $|D_k| \le \frac{8}{N}\sup_{s\in\mathcal{S}}\mathbb{E}_{A_j^{\text{TS}}\overset{\text{iid}}{\sim}\pi^{\text{TS}}(\cdot|s)}[\mathfrak{R}_{N_a}(\mathcal{G}_2(s))]$. Then, Azuma-Hoeffding bound (Corollary 2.1, Wainwright [58]) yields

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

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 $(S_i, A_{ij}^{\text{TS}})$ are not i.i.d., a standard argument still applies, which we outline for completeness. Denoting by $(\bar{S}_i, \bar{A}_{ij}^{\text{TS}})$ independent copies of $(S_i, A_{ij}^{\text{TS}})$, note that

$$\begin{aligned}
\mathbb{E}[Z_{N,N_a}] &= 2\mathbb{E}\left[\sup_{m\in\mathcal{M}}\left|\frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_a}\log\pi^m(A_{ij}^{\text{TS}} \mid S_i) - \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_a}\log\pi^m(\bar{A}_{ij}^{\text{TS}} \mid \bar{S}_i)\right]\right|\right] \\
&\le 2\mathbb{E}\left[\sup_{m\in\mathcal{M}}\left|\frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_a}\log\pi^m(A_{ij}^{\text{TS}} \mid S_i) - \log\pi^m(\bar{A}_{ij}^{\text{TS}} \mid \bar{S}_i)\right|\right] \\
&= 2\mathbb{E}\left[\sup_{m\in\mathcal{M}}\left|\frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_a}\sum_{j=1}^{N_a}\epsilon_{ij}(\log\pi^m(A_{ij}^{\text{TS}} \mid S_i) - \log\pi^m(\bar{A}_{ij}^{\text{TS}} \mid \bar{S}_i))\right|\right] \\
&\le 8\mathbb{E}[\mathfrak{R}_{NN_a}(\mathcal{G}_3)].
\end{aligned}$$

Collecting these bounds, we conclude that with probability $1 - 3e^{-t}$, the right hand side of the inequality (20) is bounded by

$$4\mathbb{E}[\mathfrak{R}_N(\mathcal{G}_1)] + 2M\sqrt{\frac{2t}{n}} + \sqrt{\frac{32t}{N}}\sup_{s\in\mathcal{S}}\mathbb{E}_{A_j^{\text{TS}}\overset{\text{iid}}{\sim}\pi^{\text{TS}}(\cdot|s)}[\mathfrak{R}_{N_a}(\mathcal{G}_2(s))] + 8\mathbb{E}[\mathfrak{R}_{NN_a}(\mathcal{G}_3)].$$

F Experiment Details

F.1 Hyperparameters

We use hyperparameters from Riquelme et al. [45] 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. [45].

The Bayesian linear regression models used on the last linear layer for NeuralLinear-TS use the normal inverse gamma prior NIG($\mu_a = \mathbf{0}, \alpha_a = 3, \beta_a = 3, \Lambda_a = 0.25I_d$). Linear-TS uses a 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: $\hat{\pi}^{TS}(a|S_i) = \frac{1}{N_a}\sum_{j=1}^{N_a} \mathbb{1}(A_{ij} = a)$ 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.

Data preprocessing 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].

F.2 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 $\varepsilon \sim \mathcal{N}(0, \sigma_a^2)$.

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

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

$$\bar{\Lambda} = X_a^T X_a + \Lambda_a, \;\; \bar{\boldsymbol{\mu}}_a = \bar{\Lambda}_a^{-1}(\Lambda_a\boldsymbol{\mu}_a + X_a^T\boldsymbol{y}_a)$$

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

F.3 Benchmark Problem Datasets

Mushroom UCI Dataset: This real dataset contains 8,124 examples with 22 categorical valued features containing descriptive features about the mushroom and labels indicating if the mushroom is poisonous or not. With equal probability, a poisonous mushroom may be unsafe and hurt the consumer or it may be safe and harmless. At each time step, the policy must choose whether to eat the new mushroom or abstain. The policy receives a small positive reward (+5) for eating a safe mushroom, a large negative reward (-35) for eating an unsafe mushroom, and zero reward for abstaining. We one-hot encode all categorical features, which results in 117-dimesional contexts.

Pharamcological Dosage Optimization Warfarin is common anticoagulant (blood thinner) that is prescribed to patients with atrial fibrillation to prevent strokes [65]. The optimal dosage varies from person to person and prescribing the incorrect dosage can have severe consequences. The Pharmacogenetics and Pharmacogenomics Knowledge Base (PharmGKB) includes a dataset with a 17-dimensional feature set containing numeric features including age, weight, and height, along with one-hot encoded categorical features indicating demographics and the presence of genetic markers. The dataset also includes the optimal dosage for each patient refined by physicians over time. We use this supervised dataset as a contextual bandit benchmark using the dosage as the action and defining the reward function to be the distance between the selected dosage and the optimal dosage. We discretize the action space into 20 (or 50) equally spaced dosage levels.

Wheel Bandit Problem The wheel bandit problem is a synthetic problem specifically designed to require exploration [45]. 2-dimensional contexts are sampled from inside the unit circle with uniform random probability. There are 5 actions where one 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 other 4 actions are non-optimal with a mean reward of 1. If $||\boldsymbol{s}||_2 > \delta$, then 1 of the 4 remaining actions is optimal—and determined by the sign of the two dimensions of $\boldsymbol{s}$ —with a mean reward of 50. The remaining 3 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 a sampling a context on the perimeter ($||\boldsymbol{s}||_2 \geq \delta$) where one action yields a large reward is $1 - (0.95)^2 = 0.0975$.

Real World Video Upload Transcoding Optimization We demonstrate performance of the imitation learning algorithm on a real world video upload transcoding application. At each time step, the policy receives a request to upload a video along with contextual features and the policy is tasked

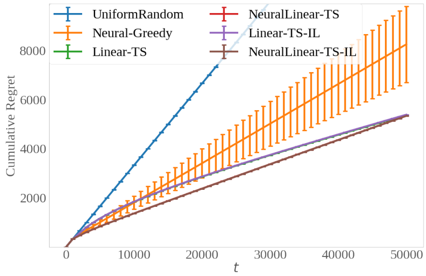


Figure 2: Cumulative regret on the Warfarin problem with 50 actions

with deciding how the video should be transcoded (e.g. what quality level to use) when uploading the video to the service. 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. Uploading a large file is more likely to to fail than uploading a small file; uploading a larger file takes more time, which increases the likelihood that the network connection will drop or that person uploading the video will grow frustrated and cancel.

The contextual information accompanying each video includes dense and sparse features 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 $i \geq j$. 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. [34]. The method evaluates a CB algorithm by performing rejection sampling on a stream of logged observation tuples of the form $(\mathbf{x}_t, a_t, r_t)$ collected under a uniform random policy. Specifically, the observation tuple t is rejected if the logged action does not match the action selected by the CB algorithm being evaluated. For this demonstration we leverage a real video upload transcoding dataset containing 8M 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.

F.4 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.

G Time and Space Complexity

G.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

$$\frac{1}{\tilde{\sigma}} L^T \boldsymbol{z} = \boldsymbol{\zeta} \tag{21}$$

and $\boldsymbol{\zeta} \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$. Since L^T is upper triangular, Eqn. (21) can be solved using a backward substitution in quadratic time: $O(d^2)$.[2]

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.

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

G.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 $m = 1...M$: $O\big(d_a h_1 + \sum_{m=1}^{M-1} h_m h_{m+1} + \sum_{m=1}^{M} kh_m\big)$.

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.

IL 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. [45] find that Thompson sampling using disjoint, exact linear bayesian regressions are a strong baseline in many applications. Furthermore, Riquelme et al. [45] observe that it is important to model the noise levels independently for each action.

G.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.[3] However, assuming independence among parameters has been observed to result in worse performance in some settings [45]. Low-rank approximations of the covariance matrix allow for sampling parameters in $O((n+1)\rho)$, 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 [66, 36]. 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.

[2] 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 $L_\Sigma \boldsymbol{\zeta}$.

[3] 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	$O(C_{\text{LAST-LAYER}}) + O(kh_M)$	