Deep Bandits Show-Off: Simple and Efficient Exploration with Deep Networks

Rong Zhu
Institute of Science and Technology for Brain-inspired Intelligence, Fudan University
rongzhu@fudan.edu.cn

Mattia Rigotti*
IBM Research AI
mr2666@columbia.edu

Abstract

Designing efficient exploration is central to Reinforcement Learning due to the fundamental problem posed by the exploration-exploitation dilemma. Bayesian exploration strategies like Thompson Sampling resolve this trade-off in a principled way by modeling and updating the distribution of the parameters of the action-value function, the outcome model of the environment. However, this technique becomes infeasible for complex environments due to the computational intractability of maintaining probability distributions over parameters of outcome models of corresponding complexity. Moreover, the approximation techniques introduced to mitigate this issue typically result in poor exploration-exploitation trade-offs, as observed in the case of deep neural network models with approximate posterior methods that have been shown to underperform in the deep bandit scenario.

In this paper we introduce *Sample Average Uncertainty (SAU)*, a simple and efficient uncertainty measure for contextual bandits. While Bayesian approaches like Thompson Sampling estimate outcomes uncertainty indirectly by first quantifying the variability over the parameters of the outcome model, SAU is a frequentist approach that directly estimates the uncertainty of the outcomes based on the value predictions. Importantly, we show theoretically that the uncertainty measure estimated by SAU asymptotically matches the uncertainty provided by Thompson Sampling, as well as its regret bounds. Because of its simplicity SAU can be seamlessly applied to deep contextual bandits as a very scalable drop-in replacement for epsilon-greedy exploration. We confirm empirically our theory by showing that SAU-based exploration outperforms current state-of-the-art deep Bayesian bandit methods on several real-world datasets at modest computation cost, and make the code to reproduce our results available at https://github.com/ibm/sau-explore.

1 Introduction

The *exploration-exploitation dilemma* is a fundamental problem in models of decision making under uncertainty in various areas of statistics, economics, machine learning, game theory, adaptive control and management. Given a set of actions associated with unknown probabilistic rewards, an agent has to decide whether to exploit familiar actions to maximizing immediate reward or to explore poorly understood or unknown actions for potentially finding ways to improve future rewards.

*Corresponding author

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

Quantifying the uncertainty associated with the value of each action is a key component of conventional algorithms for addressing the exploration-exploitation dilemma. In particular, it is central to the two most successful exploration strategies commonly adopted in bandit settings: *Upper Confidence Bound* (UCB) and *Thompson Sampling*. The UCB algorithm [1–11] follows the principle of *optimism in the face of uncertainty*, which promotes exploration by maintaining confidence sets for action-value estimates and then choosing actions optimistically within these confidence sets. Thompson Sampling (TS), introduced by [12] and successfully applied in a wide range of settings [13–16], is based on the principle of *sampling in the face of uncertainty*, meaning that it samples actions from the posterior distribution over action-values given past rewards.

In modern reinforcement learning (RL), the flexible generalization capabilities of neural networks brought about by Deep RL have proven successful in tackling complex environments by learning mappings from high-dimensional observations directly to value estimates [17]. However, obtaining uncertainty measures over complex value functions like neural network models becomes challenging because of the intractability of estimating and updating posteriors over their parameters, limiting the applicability of Bayesian exploration strategies like UCB and TS. Recently, several proposals to address this challenge have been put forth that rely on approximations of the posterior over value functions. Unfortunately, these methods tend to underperform empirically compared to much simpler heuristics. For instance, [18] showed that in contextual bandit tasks the main approximate Bayesian posterior methods for deep neural networks are consistently beaten by simple baselines such as combining neural network value functions with a basic exploration strategy like epsilon-greedy, or using simple action-values like linear regression where the exact posterior can be computed.

In this paper we propose a novel uncertainty measure which departs from the Bayesian approach of estimating the uncertainty over the parameters of the value prediction model. Our uncertainty measure, which we call *Sample Average Uncertainty (SAU)* is a frequentist quantity that only depends on the value prediction of each action. In particular, unlike UCB and TS, exploration based on SAU does not require the costly computation of a posterior distribution over models in order to estimate uncertainty of their predictions. In fact, instead of first estimating the uncertainty over the parameters of the value function to then use it to quantify the uncertainty over outcomes, SAU directly estimates uncertainty over outcomes by measuring the variance of sample averages. This result is then plugged into the current estimate of the outcome model.

With our new measure of uncertainty of the expected action-values, we build two SAU-based exploration strategies: one based on the principle of *"optimism in the face of SAU"* that we name SAU-UCB, and a second one based on *"sampling in the face of SAU"* that we name SAU-Sampling.

We investigate the use of these new exploration strategies to tackle contextual bandit problems, and show that SAU is closely related to the mean-squared error in contextual bandits. This allows us to show analytically that in the case of Bernoulli multi-armed bandits the SAU measure converges to the uncertainty of the action-value estimates that are obtained by TS, despite SAU being much simpler to compute and not needing to rely on maintaining the posterior distribution. In addition, we derive an upper bound on the expected regret incurred by our SAU algorithms in multi-armed bandits that shows that they achieve the optimal logarithmic regret.

Finally, we empirically study the deployment of SAU-UCB and SAU-Sampling in the deep bandit setting and use them as exploration strategy for deep neural network value function models. Concretely, we follow the study of [18] and show that SAU consistently outranks the deep Bayesian bandit algorithms that they analyzed on the benchmarks that they proposed.

2 Problem Formulation: Contextual Bandits

The contextual bandit problem is a paradigmatic model for the study of the exploration-exploitation trade-off and is formulated as follows. At each time step n we observe a context $\mathbf{x}_n$, select an action a_n from a set $\mathbb{K} = \{1, \dots, K\}$, after which we receive a reward r_n. The *value of an action* a (in context $\mathbf{x}_n \in \mathbb{R}^p$) is defined as the expected reward given that a is selected:

$$\mathbb{E}[r_n | a_n = a] = \mu(\mathbf{x}_n, \boldsymbol{\theta}_a), \tag{1}$$

where in general the action-values $\mu(\cdot)$ depend on unknown parameters $\boldsymbol{\theta}_a \in \mathbb{R}^p$.

Our goal is to design a sequential decision-making policy π that over time learns the action parameters $\boldsymbol{\theta}_a$ which maximize the expected reward. This goal is readily quantified in terms of minimizing

expected regret, where we say that at step n we incur expected regret

$$\max_{a' \in \mathbb{K}} \{\mu(\mathbf{x}_n, \boldsymbol{\theta}_{a'})\} - \mu(\mathbf{x}_n, \boldsymbol{\theta}_{a_n}), \tag{2}$$

i.e. the difference between the reward received by playing the optimal action and the one following the chosen action a_n. One way to design a sequential decision-making policy π that minimizes expected regret is to quantify the uncertainty around the current estimate of the unknown parameters $\boldsymbol{\theta}_a$. TS for instance does this by sequentially updating the posterior of $\boldsymbol{\theta}_a$ after each action and reward. This paper presents a novel and simpler alternative method to estimate uncertainty.

3 Exploration based on Sample Average Uncertainty

3.1 Sample Average Uncertainty (SAU)

In this section, we begin with introducing our novel measure of uncertainty SAU. Let $\mathbb{T}_a$ denote the set of time steps when action a was chosen so far, and let n_a be the size of this set. Based on the n_a rewards $\{r_n\}_{n \in \mathbb{T}_a}$ obtained with action a, the sample mean reward given action a is:

$$\bar{r}_a = n_a^{-1} \sum\nolimits_{n \in \mathbb{T}_a} r_n.$$

At this point we reiterate that exploitation and exploration are customarily traded off against each other with a Bayesian approach that estimates the uncertainty of the action-values on the basis of a posterior distribution over their parameters given past rewards. Instead, we propose a *frequentist approach* that directly measures the uncertainty of the sample average rewards that was just computed. Direct calculation using eq. (1) then gives us that the variance of the sample mean reward is

$$\text{Var}(\bar{r}_a) = \bar{\sigma}_a^2 / n_a, \quad \text{where} \quad \bar{\sigma}_a^2 = n_a^{-1} \sum\nolimits_{n \in \mathbb{T}_a} \sigma_{n,a}^2 \quad \text{with} \quad \sigma_{n,a}^2 = \mathbb{E}\left[(r_n - \mu(x_n, \boldsymbol{\theta}_a))^2\right].$$

Assuming that there is a sequence of estimators $\{\hat{\boldsymbol{\theta}}_{n,a}\}_{n \in \mathbb{T}_a}$ of $\boldsymbol{\theta}_a$, we can replace $\boldsymbol{\theta}_a$ with $\hat{\boldsymbol{\theta}}_{n,a}$ at each $n \in \mathbb{T}_a$ to approximate $\bar{\sigma}_a^2$ with a convenient statistics τ_a^2 defined as

$$\tau_a^2 = n_a^{-1} \sum\nolimits_{n \in \mathbb{T}_a} \left(r_n - \mu(x_n, \hat{\boldsymbol{\theta}}_{n,a})\right)^2. \tag{3}$$

With this we get an approximate sample mean variance of

$$\widehat{\text{Var}}(\bar{r}_a) = \tau_a^2 / n_a. \tag{4}$$

The central proposal of this paper is to use $\widehat{\text{Var}}(\bar{r}_a)$ as a measure of the uncertainty of the decision sequence. We call this quantity *Sample Average Uncertainty* (SAU), since it measures directly the uncertainty of sample mean rewards $\bar{r}_a$. In practice, τ_a^2 can be updated incrementally as follows:

1. Compute the *prediction residual*: $$e_n = r_n - \mu(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_{n,a_n}); \tag{5}$$
2. Update *Sample Average Uncertainty (SAU)*: $$\tau_{a_n}^2 \leftarrow \tau_{a_n}^2 + n_{a_n}^{-1}\left[e_n^2 - \tau_{a_n}^2\right]. \tag{6}$$

Let us take a moment to contrast the uncertainty measure given by SAU and existing exploration algorithms like TS, which as we said would estimate the uncertainty of the action-value function $\mu(\cdot)$ by maintaining and updating a distribution over its parameters $\boldsymbol{\theta}_a$. SAU instead directly quantifies the uncertainty associated with each action by measuring the uncertainty of the sample average rewards. The clear advantage of SAU is that it is simple and efficient to compute: all it requires are the prediction residuals $r_n - \mu(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_{n,a_n})$ without any need to model or access the uncertainty of $\mu(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_{n,a})$. Because of the simplicity of its implementation, SAU can be naturally adapted to arbitrary action-value functions. In particular, it can be used to implement an exploration strategy for action-value function parameterized as deep neural networks or other model classes for which TS would be infeasible because of the intractability of computing a probability distribution over models.

Note that in updating τ_a^2 we use the residuals obtained at each step rather than re-evaluating them using later estimates. This is a design choice motivated by the goal of minimizing the computation cost and implementation efficiency of SAU. Moreover, this choice can be justified from the viewpoint of the statistical efficiency, since, as the number of training samples increases, the impact of initial residuals

will decrease, so that the benefit of re-evaluating them incurs diminishing returns. Proposition 3 formalizes this argument by showing that indeed τ_a^2 as computed in eq. (6) is concentrated around its expectation. In addition, perhaps as importantly, the aim of SAU is to provide a quantity to support exploration. The effect of potentially inaccurate residuals in the initial steps may actually be beneficial due to the introduction of additional noise driving initial exploration. This might be in part at the root of the good empirical results.

3.2 SAU-based Exploration in Bandit Problems

We now use the SAU measure to implement exploration strategies for (contextual) bandit problems.

SAU-UCB. UCB is a common way to perform exploration. Central to UCB is the specification of an "exploration bonus" which is typically chosen to be proportional to the measure of uncertainty. Accordingly, we propose to use the SAU measure τ_a^2 as exploration bonus. Specifically, given value predictions $\hat{\mu}_{n,a} = \mu(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_{n,a})$ for each a at step n, we modify the values as

$$\widetilde{\mu}_{n,a} = \hat{\mu}_{n,a} + \sqrt{n_a^{-1}\tau_a^2 \log n}, \tag{7}$$

then choose the action by $a_n = \arg\max_a(\{\widetilde{\mu}_{n,a}\}_{a\in\mathbb{K}})$. We call this implementation of UCB using SAU as exploration bonus: SAU-UCB.

SAU-Sampling. "Sampling in the face of uncertainty" is an alternative exploration principle that we propose to implement with SAU in addition to UCB. This is inspired by TS which samples the success probability estimate $\hat{\mu}_a$ from its posterior distribution. Analogously, we propose to sample values from a parametric Gaussian distribution with a mean given by the value prediction and a variance given by $\bar{\sigma}_a^2$. This results in sampling values $\widetilde{\mu}_{n,a}$ at each time n as:

$$\widetilde{\mu}_{n,a} \sim \mathcal{N}\left(\hat{\mu}_{n,a}, \tau_a^2/n_a\right), \tag{8}$$

then choosing the action by $a_n = \arg\max_a(\{\widetilde{\mu}_{n,a}\}_{a\in\mathbb{K}})$. We call this use of SAU inspired by TS, SAU-Sampling.

SAU-UCB and SAU-Sampling are summarized in Algorithm 1.

Algorithm 1 SAU-UCB and SAU-Sampling for bandit problems

Initialize: $\hat{\boldsymbol{\theta}}_a$, $S_a^2 = 1$ and $n_a = 0$ for $a \in \mathbb{K}$.
for $n = 1, 2, \ldots$ **do**
  Observe context $\mathbf{x}_n$;
  for $a = 1, \ldots, K$ **do**
    Calculate the prediction $\hat{\mu}_{n,a} = \mu(\mathbf{x}_n; \hat{\boldsymbol{\theta}}_a)$ and $\tau_a^2 = S_a^2/n_a$;
    Draw a sample

$$\widetilde{\mu}_{n,a} = \hat{\mu}_{n,a} + \sqrt{\tau_a^2 n_a^{-1} \log n} \text{ (SAU-UCB)} \quad \text{or} \quad \widetilde{\mu}_{n,a} \sim \mathcal{N}\left(\hat{\mu}_{n,a}, n_a^{-1}\tau_a^2\right) \text{ (SAU-Sampling);}$$

  end for
  Compute $a_n = \arg\max_a(\{\widetilde{\mu}_{n,a}\}_{a\in\mathbb{K}})$ if $n > K$, otherwise $a_n = n$;
  Select action a_n, observe reward r_n;
  Update $\hat{\boldsymbol{\theta}}_{a_n}$ and increment $n_{a_n} \leftarrow n_{a_n} + 1$;
  Update $S_{a_n}^2 \leftarrow S_{a_n}^2 + e_n^2$ using prediction error calculated as $e_n = r_n - \hat{\mu}_{n,a_n}$;
end for

3.3 Novelty and comparison with related approaches

Using the variance estimation in MAB is not novel. For example [19] makes use of Bernstein's inequality to refine confidence intervals by additionally considering the uncertainty from estimating variance of reward noise. Our approach is fundamentally different from it with two aspects. First, Algorithm 1 is to propose a novel measure to approximate the uncertainty of the estimate of the mean reward that would afford such a flexible implementation and can therefore directly extended

and scaled up to complicated value models like deep neural networks. Second, our SAU quantity τ^2 is the per-step squared prediction error, i.e., the average cumulative squared prediction error, as opposed to an estimate of the variance of the different arms. In fact, τ^2 does not rely on the traditional variance estimation analyzed by[19], but is instead simply computed directly from the prediction. This difference makes SAU even easier to implement and adapt to settings like deep networks.

The exploration bonus in Algorithm 1 is not a function of the observed context, though it is updated from historical observations of the context. The algorithm could indeed be extended to provide a quantification of reward uncertainty that is a function of the current context by, for instance, fitting the SAU quantity as a function of context. Clearly, this will come at the cost of substantially increasing the complexity of the algorithm. Therefore to avoid this additional complexity, we instead focus the paper on the development of the SAU quantity as a simple estimate of uncertainty to efficiently drive exploration. However, exploring this possibility is a potentially exciting direction for future work.

4 SAU in Multi-Armed Bandits

4.1 SAU Approximates Mean-squared Error and TS in Multi-armed Bandits

Before considering the contextual bandits scenario, we analyze the measure of uncertainty provided by SAU in multi-armed bandits, and compare it to the uncertainty computed by TS. This will help motivate SAU and elucidate its functioning.

We assume a *multi-armed Bernoulli bandit*, i.e. at each step n each action $a \in \mathbb{K}$ results in a reward sampled from $r_n \sim \text{Bernoulli}(\mu_a)$ with fixed (unknown) means $\mu_a \in [0, 1]$. Assume that action a has been taken n_a times so far, and let $\hat{\mu}_a$ denote the sample averages of the rewards for each action. The *prediction residual* eq. (5) is $e_n = r_n - \hat{\mu}_{a_n}$ and is the central quantity to compute SAU.

TS in the case of Bernoulli bandits is typically applied by assuming that the prior follows a Beta distribution, i.e. the values are sampled from $\text{Beta}(\alpha_a, \beta_a)$ with parameters α_a and β_a for $a \in \mathbb{K}$. Uncertainty around the estimated mean values are then quantified by its variance denoted by $\hat{V}_a$ (see Appendix A.1). We then have the following proposition relating SAU and TS in Bernoulli bandits:

Proposition 1 *For Beta Bernoulli bandits the expectation of the average* prediction residual e_n^2/n_{a_n} *is an approximate unbiased estimator of the expectation of the* posterior variance $\hat{V}_a$ *in TS. Concretely:*

$$\mathbb{E}[\hat{V}_{a_n}] = \mathbb{E}[e_n^2/n_{a_n}] + O\left(n_{a_n}^{-2}\right).$$

Proof Proof of Proposition 1 is provided in Appendix A.1. ■

Proposition 1 says that SAU asymptotically approximates TS for Bernoulli bandits, despite not needing to assume a prior and update a posterior distribution over parameters. In Appendix A.3 we support this empirically by showing that in multi-armed bandits SAU rivals TS.

The following proposition further characterizes the prediction residual:

Proposition 2 *For Bernoulli bandits the expectation of the prediction residual used in SAU satisfies*

$$\mathbb{E}[e_n^2/n_{a_n}] = \mathbb{E}[(r_n - \hat{\mu}_{a_n})^2/n_{a_n}] = \mathbb{E}\left[(\hat{\mu}_{a_n} - \mu_{a_n})^2\right] + O\left(n_{a_n}^{-2}\right).$$

Proof Proof of Proposition 2 is provided in Appendix A.2. ■

Proposition 2 says that the prediction residual $e_n = r_n - \hat{\mu}_{a_n}$ is an approximately unbiased estimator of the mean squared error $\mathbb{E}\left[(\hat{\mu}_{a_n} - \mu_{a_n})^2\right]$. This means that for Bernoulli bandits, SAU closely approximates the uncertainty of the action-value estimates.

Armed with this characterization of the prediction residual $r_n - \hat{\mu}_{a_n}$ in Proposition 2, we now quantify the performance of the estimator τ_a^2 in eq. (3) in terms of its concentration around its expectation:

Proposition 3 *For* $\delta \in \left[2\exp\left(-\sigma_a^2 n_a/(32c)\right), 1\right)$*, where* σ_a^2 *is the variance of* r_j *for* $j \in \mathbb{T}_a$ *and* c *a constant, we have*

$$\Pr\left\{\left|\tau_a^2 - \mathbb{E}\left[\tau_a^2\right]\right| \geq \sigma_a \sqrt{8c/(n_a \log(\delta/2))}\right\} \leq \delta,$$

Proof Proof of Proposition 3 is provided in Appendix A.4. ■

Proposition 3 says that τ_a^2 is concentrated around its expectation, and thus remains stable as it is being updated. In Appendix A.6 we also show that $\mathbb{E}\left[\tau_a^2\right] \to \sigma_a^2$ as $n_a \to \infty$, and in Appendix A.7 we derive an upper bound on the expected regrets of SAU-UCB and SAU-Sampling in multi-armed bandits proving that the optimal logarithmic regrets are achievable uniformly over time, which says that the theoretical performance of SAU rivals TS in multi-armed bandits.

4.2 SAU in Linear Contextual Bandits: Theoretical analysis

We now show that the results in Proposition 2 also hold for another important bandit model beside Bernoulli bandits, i.e. *linear contextual bandits* defined by the following outcome model:

$$r_n = \mathbf{x}_n^\top \boldsymbol{\theta}_a + \epsilon_{n,a}, \;\; n = 1, 2, \ldots, \tag{9}$$

where $\mathbf{x}_n, \boldsymbol{\theta}_a \in \mathbb{R}^p$, and $\epsilon_{n,a}$ are iid random variables with variance σ_a^2. Assume action a was selected n_a times. We obtain the least-squares estimator $\hat{\boldsymbol{\theta}}_{n,a_n} = (\sum_{j \in \mathbb{T}_{n,a_n}} \mathbf{x}_j^\top \mathbf{x}_j)^{-1} (\sum_{j \in \mathbb{T}_{n,a_n}} \mathbf{x}_j^\top r_j)$. Accordingly, the prediction and the prediction residual at step n are, respectively,

$$\hat{\mu}_{n,a_n} = \mathbf{x}_n^\top \hat{\boldsymbol{\theta}}_{n,a_n} \quad \text{and} \quad e_n^2 = (r_n - \mathbf{x}_n^\top \hat{\boldsymbol{\theta}}_{n,a_n})^2. \tag{10}$$

Denote $h_n = \mathbf{x}_n^\top (\sum_{j \in \mathbb{T}_{n,a_n}} \mathbf{x}_j^\top \mathbf{x}_j)^{-1} \mathbf{x}_n$. The mean squared error of $\mathbf{x}_n^\top \hat{\boldsymbol{\theta}}_{n,a_n}$ is $\text{MSE}_n = \mathbb{E}[(\mathbf{x}_n^\top \hat{\boldsymbol{\theta}}_{n,a_n} - \mathbf{x}_n^\top \boldsymbol{\theta}_{a_n})^2]$. With direct calculation we see that $\text{MSE}_n = h_n \sigma_{a_n}^2$ and that $\mathbb{E}\left[e_n^2 / n_{a_n}\right] = (1 - h_n)\sigma_{a_n}^2 / n_{a_n}$. Therefore, we have the following proposition:

Proposition 4 *For linear contextual bandits* (9) *we have that*

$$\mathbb{E}[e_n^2 / n_{a_n}] = (h_n n_{a_n})^{-1} (1 - h_n)\, \text{MSE}_n.$$

Furthermore, assuming that there exist constants c_1 and c_2 so that $c_1 / n_{a_n} \le h_n \le c_2 / n_{a_n}$, then

$$c_2^{-1}(1 - c_2 / n_{a_n})\, \text{MSE}_n \le \mathbb{E}\left[e_n^2 / n_{a_n}\right] \le c_1^{-1}(1 - c_1 / n_{a_n})\, \text{MSE}_n.$$

Proposition 4 provides a lower and an upper bound for $\mathbb{E}\left[e_n^2 / n_{a_n}\right]$ in terms of MSE_n, meaning that on average SAU is a conservative measure of the uncertainty around $\mathbf{x}_n^\top \hat{\boldsymbol{\theta}}_{n,a_n}$. Noting that $0 \le h_j \le 1$ and $\sum_{j \in \mathbb{T}_{n,a_n}} h_j = p$, the assumption that $c_1 / n_{a_n} \le h_n \le c_2 / n_{a_n}$ requires that h_n does not dominate or is dominated by other terms h_j, with $j \in \mathbb{T}_{n,a_n}$, meaning that contexts should be "homogeneous" to a certain extent. To examine the robustness to violations of this assumption, in the simulation in Appendix B we empirically test the performance under a heavy-tailed t-distribution with $df = 2$. The results show that SAU works robustly even under such type of context inhomogeneity.

4.3 SAU in Linear Contextual Bandits: Empirical evaluation on synthetic data

In this section, we present simulation results quantifying the performance of our SAU-based exploration algorithms in linear contextual bandits. We evaluate SAU on synthetically generated datasets to address two questions: (1) How does SAU's performance compare against Thompson Sampling?, and (2) How robust is SAU in various parameter regimes?

We consider three scenarios for K (the number of actions) and p (the context dimensionality): (a) $K = 5, p = 5$, (b) $K = 20, p = 5$, and (b) $K = 5, p = 40$. The horizon is $N = 20000$ steps. For each action a, parameters $\boldsymbol{\theta}_a$ are drawn from a uniform distribution in $[-1, 1]$, then normalized so that $\|\boldsymbol{\theta}_a\| = 1$. Next, at each step n context $\mathbf{x}_n$ is sampled from a Gaussian distribution $\mathcal{N}(\mathbf{0}_p, \mathbf{I}_p)$. Finally, we set the noise variance to be $\sigma^2 = 0.5^2$ so that the signal-to-noise ratio equals 4.

We compare our SAU-based exploration algorithms, SAU-UCB and SAU-Sampling to Thompson Sampling ("TS" in Fig. 1). For TS on linear model, we follow [18] and use Bayesian linear regression for exact posterior inference. We also consider the *PrecisionDiag* approximation for the posterior covariance matrix of $\boldsymbol{\theta}_a$ with the same priors as in [18] ("TSdiag" in Fig. 1).

Fig. 1a) shows regret as a function of step for $(K, p) = (5, 5)$. From the figure we have two observations: SAU-Sampling is comparable to TS, and SAU-UCB achieves better regret than TS. In

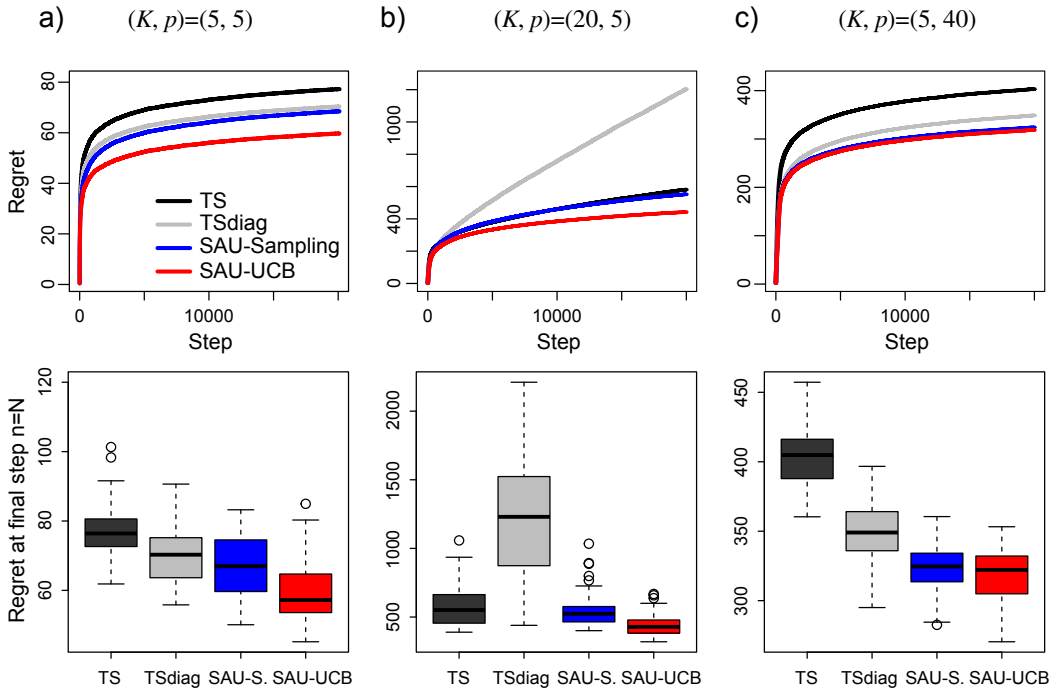


Figure 1: Performance on contextual linear bandits with various (K, p) parameters showing that our models (SAU-Sampling and SAU-UCB) consistently achieve lower regret than TS using Bayesian linear regression with exact posterior inference (TS) and TS with *PrecisionDiag* approximation (TSdiag). The upper panels report regret as a function of step n, where results are averaged over 100 runs. The lower panels show the distributions of the regret at the final step.

terms of cumulative regret SAU significantly outperforms TS and TSdiag. Figures 1b) and c) show the effects of larger K and p, respectively. The observations from Fig. 1a) still hold in these cases, implying that SAU's performance is robust to an increase in action space and context dimension.

We also consider four other cases: (1) the elements of $\boldsymbol{\theta}_a$ are sampled from $\mathcal{N}(0, 1)$ then are normalized; (2) the model errors are correlated with $AR(1)$ covariance structure with correlation $\rho = 0.5$; (3) the elements in $\mathbf{x}_i$ are correlated with $AR(1)$ covariance structure with correlation $\rho = 0.5$; and (4) the elements of $\mathbf{x}_i$ are sampled from a heavy-tailed t-distribution with $df = 2$ and are truncated at 5. These results are shown in Appendix B and are consistent with the results in Fig. 1 confirming SAU's robustness to various contextual linear bandit problems.

5 Deep Contextual Bandits

5.1 Deep Bayesian Bandit Algorithms

Deep contextual bandits refers to tackling contextual bandits by parameterizing the action-value function as a deep neural network $\mu(\mathbf{x}, \boldsymbol{\theta})$, thereby leveraging models that have been very successful in the large-scale supervised learning [20] and RL [17]. Notice that in the deep setting we denote all parameters with $\boldsymbol{\theta} = \{\boldsymbol{\theta}_a\}_{a \in \mathbb{K}}$, as common in the neural network literature. In particular, $\boldsymbol{\theta}$ includes the parameters that are shared across actions, as well as those of the last layer of the network which are specific to each action a. Algorithm 2 breaks down a generic deep contextual bandit algorithm in terms of an API exposing its basic subroutines: PREDICT (which outputs the set of action-values $\{\mu_{n,a}\}_{a \in \mathbb{K}}$ given the observation $\mathbf{x}_n$), ACTION (which selects an action given all the action-values), and UPDATE (which updates model parameters at the and of the step).

In this scheme Thompson Sampling (TS) is implemented as in Algorithm 3, which underlines where TS promotes exploration by sampling from a distribution over model parameters $P_n(\boldsymbol{\theta})$. In principle this provides an elegant Bayesian approach to tackle the exploration-exploitation dilemma embodied

Algorithm 2 Generic Deep Contextual Bandit algorithm

for $n = 1, 2, \ldots$ **do**
  Observe context $\mathbf{x}_n$;
  Compute values $\{\mu_{n,a}\}_{a\in\mathbb{K}} = \textsc{Predict}(\mathbf{x}_n)$;
  Choose $a_n = \textsc{Action}(\{\mu_{n,a}\}_{a\in\mathbb{K}})$, observe reward r_n;
  $\textsc{update}\,(r_n, a_n, \mathbf{x}_n)$;
end for

by contextual bandits. Unfortunately, representing and updating a posterior distribution over model parameters $P_n(\boldsymbol{\theta})$ exactly becomes intractable for complex models such as deep neural networks.

Algorithm 3 Thompson Sampling for Deep Contextual Bandits

function $\textsc{Predict}(\mathbf{x}_n)$
  Exploration: Sample model parameters from posterior distribution: $\hat{\boldsymbol{\theta}}_n \sim P_n(\boldsymbol{\theta})$;
  Return predicted values $\{\hat{\mu}_{n,a}\}_{a\in\mathbb{K}} = \mu(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_n)$, where
function $\textsc{Action}(\{\hat{\mu}_{n,a}\}_{a\in\mathbb{K}})$
  Return $a_n = \arg\max_a(\{\tilde{\mu}_{n,a}\}_{a\in\mathbb{K}})$;
function $\textsc{Update}(r_n, a_n, \mathbf{x}_n)$
  Use triplet $(r_n, a_n, \mathbf{x}_n)$ to update posterior distribution: $P_{n+1}(\boldsymbol{\theta}) \leftarrow P_n(\boldsymbol{\theta})$;

To obviate this problem, several techniques that heuristically approximate posterior sampling have emerged, such as randomly perturbing network parameters [21–23], or bootstrapped sampling [24]. Within the scheme of Algorithm 2 the role of random perturbation and bootstrapped sampling are to heuristically emulate the model sampling procedure promoting exploration in the PREDICT subroutine (see TS Algorithm 3). However, systematic empirical comparisons recently demonstrated that simple strategies such as epsilon-greedy [17, 25] and Bayesian linear regression [26] remain very competitive compared to these approximate posterior sampling methods in deep contextual bandit. In particular, [18] showed that linear models where the posterior can be computed exactly, and epsilon-greedy action selection overwhelmingly outrank deep methods with approximate posterior sampling in a suite of contextual bandit benchmarks based on real-world data.

5.2 SAU for Deep Contextual Bandits

We now re-examine the deep contextual bandits benchmarks in [18] and show that SAU can be seamlessly combined with deep neural networks, resulting in an exploration strategy whose performance is competitive with the best deep contextual bandit algorithms identified by [18].

Algorithm 4 shows the deep contextual bandit implementation of SAU. Notice that the PREDICT subroutine is remarkably simple, consisting merely in the forward step of the deep neural network value prediction model. In contrast to our extremely simple procedure, TS-based methods require at this step to (approximately) sample from the model posterior to implement exploration. In SAU exploration is instead taken care of by the ACTION subroutine, which takes the values as inputs and either explores through sampling from a distribution around the predicted values (SAU-Sampling) or through an exploration bonus added to them (SAU-UCB). SAU then selects the action corresponding to the maximum of these perturbed values. The UPDATE for SAU is also quite simple, and consists in updating the neural network parameters to minimize the reward prediction error loss l_n following action selection using SGD via backprop, or possibly its mini-batch version (which would then be carried out on a batch of $(r_n, a_n, \mathbf{x}_n)$ triplets previously stored in a memory buffer). UPDATE then updates the count and the SAU measure τ_{a_n} for the selected action a_n.

We notice that the simplicity of SAU for deep contextual bandits is akin to the simplicity of epsilon-greedy, for which exploration is also implemented in the ACTION subroutine (see Algorithms 5 in Appendix E). In fact, comparing the two algorithms it is clear that SAU can be used as a *drop-in replacement for epsilon-greedy exploration*, making it widely applicable.

Algorithm 4 SAU for Deep Contextual Bandits (SAU-Neural-Sampling and UCB)

function PREDICT($\mathbf{x}_n$)
  Return predicted values $\{\hat{\mu}_{n,a}\}_{a\in\mathbb{K}} = \mu(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_n)$;
function ACTION($\{\hat{\mu}_{n,a}\}_{a\in\mathbb{K}}$)
  Exploration: Compute $\widetilde{\mu}_{n,a} \sim \mathcal{N}\left(\hat{\mu}_{n,a}, \tau_a^2/n_a\right)$ (SAU-Sampling)
    or $\widetilde{\mu}_{n,a} = \hat{\mu}_{n,a} + \sqrt{\tau_a \log n / n_a}$ (SAU-UCB);
  Return $a_n = \arg\max_a(\{\tilde{\mu}_{n,a}\}_{a\in\mathbb{K}})$;
function UPDATE($r_n, a_n, \mathbf{x}_n$)
  Compute prediction error $e_n = r_n - \hat{\mu}_{n,a_n}$ and loss $l_n = \frac{1}{2}(r_n - \hat{\mu}_{n,a_n})^2$
  Update model parameters to $\hat{\boldsymbol{\theta}}_{n+1}$ using SGD with gradients $\frac{\partial l_n}{\partial \hat{\boldsymbol{\theta}}}$ (or mini-batch version);
  Update exploration parameters: $n_{a_n} \leftarrow n_{a_n} + 1, \quad S_{a_n}^2 \leftarrow S_{a_n}^2 + e_n^2 \quad \tau_{a_n}^2 = S_{a_n}^2/n_{a_n}$;

5.3 Empirical Evaluation of SAU on Deep Contextual Bandit Problems

Benchmarks and baseline algorithms. Our empirical evaluation of SAU's performance in the deep contextual bandit setting is based on the experiments by [18], who benchmarked the main TS-based approximate posterior sampling methods over a series of contextual bandit problems. We test SAU on the same contextual bandit problems against 4 competing algorithms consisting in the 4 best ranking algorithms identified by [18], which are: *LinearPosterior* (a closed-form Bayesian linear regression algorithm for exact posterior inference under the assumption of a linear contextual bandit [27]), *LinearGreedy* (epsilon-greedy exploration under the assumption of a linear contextual bandit), *NeuralLinear* (Bayesian linear regression on top of the last layer of a neural network trained with SGD [28]) and *NeuralGreedy* (a neural network with epsilon-greedy exploration trained with SGD). We neglected a direct comparison with NeuralUCB [29], since its scaling in memory and computational requirements make it quickly impractical for even moderately sized applications of practical interest. Moreover, its reported performance is substantially worse than SAU-UCB.

Implementations of SAU. We implemented and tested 4 versions of SAU on the benchmarks in [18]. In the Tables below we refer to them a follows: *Linear-SAU-S* and *Linear-SAU-UCB* refer to a linear regression model using SAU-Sampling and SAU-UCB as exploration strategies, respectively. *Neural-SAU-S* and *Neural-SAU-UCB* refer to a neural network model trained with SGD using SAU-Sampling and SAU-UCB, respectively.

Empirical evaluation on the Wheel Bandit. The *Wheel Bandit Problem* is a synthetic bandit designed by [18] to study the performance of bandit algorithms as a function of the need for exploration in the environment by varying a parameter $\delta \in [0, 1]$ that smoothly changes the importance of exploration. In particular, the difficulty of the problem increases with δ, since the problem is designed so that for δ close to 1 most contexts have the same optimal action, while only for a fraction $1 - \delta^2$ of contexts the optimal action is a different more rewarding action (see [18] for more details). In Appendix C, Table 2 quantifies the performance of SAU-Sampling and SAU-UCB in terms of cumulative regret in comparison to the 4 competing algorithms, and normalized to the performance of the *Uniform* baseline, which selects actions uniformly at random. There we can see that *Neural-SAU-S* is consistently the best algorithm with lower cumulative regret for a wide rage of the parameter δ. Only for very high values of δ ($\delta = 0.99$) the baseline algorithm *NeuralLiner* starts to overtake it, but even in this case, another variant of SAU, *SAU-Linear-S* still maintains the lead in performance.

Empirical evaluation on real-world Deep Contextual Bandit problems. Table 1 quantifies the performance of SAU-Sampling and SAU-UCB in comparison to the 4 competing baseline algorithms, and normalized to the performance of the *Uniform* baseline. These results show that a SAU algorithm is the best algorithm in each of the 7 benchmarks in terms of minimizing cumulative regret over all samples. *Neural-SAU-S* or *Neural-SAU-UCB* are the best combination 6 out of 7 times, and linear regression with SAU-UCB is the best on the bandit built from the Adult dataset. The next best algorithm in terms of minimizing cumulative regret is *NeuralLinear* [18], which incurs cumulative regret that on average is 32% higher than *Neural-SAU-S* and 34% higher than *Neural-SAU-UCB*.

As already mentioned, thanks to their implementation efficiency SAU-based algorithms are much less computation intensive than TS-based algorithms. This is reflected in the remarkably shorter

Table 1: Cumulative regret incurred on the contextual bandits in [18] by the 4 best algorithms that they identified (described in Appendix D) compared against our SAU-based algorithms. Results are relative to the cumulative regret of the Uniform algorithm. We report the mean and standard error of the mean over 50 trials. The best mean cumulative regret value for each task is marked in bold.

	Mushroom	**Statlog**	**Covertype**	**Financial**	**Jester**	**Adult**	**Census**
LinearPosterior	3.02 ± 0.15	10.29 ± 0.19	36.88 ± 0.07	10.77 ± 0.15	64.01 ± 0.40	75.87 ± 0.06	46.70 ± 0.08
LinearGreedy	4.64 ± 0.35	109.26 ± 0.95	53.50 ± 2.00	15.48 ± 1.02	62.71 ± 0.45	86.70 ± 0.24	65.35 ± 1.57
NeuralLinear	2.66 ± 0.08	1.26 ± 0.03	29.18 ± 0.07	10.16 ± 0.18	69.48 ± 0.43	78.28 ± 0.07	41.00 ± 0.09
NeuralGreedy	26.66 ± 0.32	40.17 ± 0.46	88.85 ± 0.12	85.85 ± 1.37	93.15 ± 0.77	98.96 ± 0.03	85.82 ± 0.26
Linear-SAU-S	4.58 ± 0.39	10.44 ± 0.26	36.29 ± 0.12	8.71 ± 0.52	62.59 ± 0.43	74.70 ± 0.13	39.65 ± 0.11
Linear-SAU-UCB	3.09 ± 0.15	10.02 ± 0.22	36.77 ± 0.17	6.51 ± 0.37	64.43 ± 0.40	**74.62 ± 0.07**	39.98 ± 0.09
Neural-SAU-S	**2.20 ± 0.05**	0.62 ± 0.01	**27.46 ± 0.06**	5.60 ± 0.11	**61.02 ± 0.57**	78.02 ± 0.07	38.76 ± 0.08
Neural-SAU-UCB	2.32 ± 0.06	**0.60 ± 0.01**	27.90 ± 0.06	**5.26 ± 0.15**	62.27 ± 0.61	78.09 ± 0.06	**38.54 ± 0.09**
Uniform	100.00 ± 0.24	100.00 ± 0.03	100.00 ± 0.02	100.00 ± 1.47	100.00 ± 0.96	100.00 ± 0.02	100.00 ± 0.04

execution time: on average *Neural-SAU-S* and *Neural-SAU-UCB* run more than 10 time faster than *NeuralLinear* [18] (see Appendix Table 5 for details), also making them extremely scalable.

6 Conclusion and Discussion

Existing methods to estimate uncertainty tend to be impractical for complex value function models like deep neural networks, either because exact posterior estimation become unfeasible, or due to how approximate algorithms coupled with deep learning training amplify estimation errors.

In this paper, we have introduced Sample Average Uncertainty (SAU), a simple and efficient uncertainty measure for contextual bandit problems which sidesteps the mentioned problems plaguing Bayesian posterior methods. SAU only depends on the value prediction, in contrast to methods based on Thompson Sampling that instead require an estimate of the variability of the model parameters. As a result, SAU is immune to the negative effects that neural network parameterizations and optimization have on the quality of uncertainty estimation, resulting in reliable and robust exploration as demonstrated by our empirical studies. SAU's implementation simplicity also makes it suitable as a drop-in replacement for epsilon-greedy action selection, resulting in a scalable exploration strategy that can be effortlessly deployed in large-scale and online contextual bandit scenarios.

We also have provided theoretical justifications for SAU-based exploration by connecting SAU with posterior variance and mean-squared error estimation. However, the reasons why SAU is in practice consistently better than TS-based exploration in deep bandits is still not settled theoretically. We hypothesize that this might be due to two main reasons: (1) TS-based methods implement exploration by estimating the uncertainty of the internal model parameters, which might introduce estimation errors, while SAU directly estimates uncertainty at the model output; (2) in addition, the approximation error from the approximate posterior implementations of TS-based models might result in inefficient uncertainty measures of the internal model parameters. Because of the importance of contextual bandit algorithms for practical applications like for instance recommendation and ad servicing systems, we believe that it will be important to further theoretically refine these hypotheses to help mitigate the possible negative societal impacts that could result from deploying inefficient, miscalibrated or biased exploration algorithms.

Another limitation of our work is that it developed SAU-based exploration in the specific and restricted case of bandits. Despite being an application of interest, we are excited and looking forward to further development that could extend methods based on SAU to more general sequential decision scenarios in RL beyond the bandit setting.

Acknowledgements

This work was partially supported by National Natural Science Foundation of China (No.11871459) and by Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01).

References

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

[2] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite time analysis of the multi-armed bandit problem. *Machine Learning*, 47:235–256, 2002.

[3] M. Woodroofe. A one-armed bandit problem with a concomitant variable. *Journal of the American Statistical Association*, 74:799–806, 1979.

[4] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2003.

[5] J. Langford and T. Zhang. The epoch-greedy algorithm for multiarmed bandits with side information. In *Advances in Neural Information Processing Systems*, volume 20, pages 817–824, 2008.

[6] Varsha Dani, Thomas Hayes, and Sham Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 355–366, 2008.

[7] Patt Rusmevichientong and John Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

[8] Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesväri. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, volume 24, pages 2312–2320, 2011.

[9] Yasin Abbasi-Yadkori. *Online learning for linearly parametrized control problems*. PhD thesis, University of Alberta, Edmonton, AB, Canada., 2012.

[10] V. Perchet and P. Rigollet. The multi-armed bandit problem with covariates. *The Annals of Statistics*, 41(2):693–721., 2013.

[11] A. Slivkins. Contextual bandits with similarity information. *Journal of Machine Learning Research*, 15(1):2533–2568, 2014.

[12] William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25:285–294, 1933.

[13] M. Strens. A bayesian framework for reinforcement learning. In *Proceedings of the 17th international conference on Machine learning*, pages 943–950, 2000.

[14] Olivier Chapelle and Lihong Li. An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

[15] Shipra Agrawal and Navin Goyal. Further optimal regret bounds for Thompson sampling. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, pages 99–107, 2013.

[16] D. Russo and B. Van Roy. Learning to optimize via information-directed sampling. In *Advances in Neural Information Processing Systems*, volume 27, pages 1583–1591, 2014.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

[18] Carlos Riquelme, George Tucker, and Jasper Snoek. Deep Bayesian bandits showdown: An empirical comparison of Bayesian deep networks for Thompson sampling. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[19] J.-Y. Audibert, R. Munosc, and C. Szepesvári. Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410:1876–1902, 2009.

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[21] Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. *arXiv preprint arXiv:1706.01905*, 2017.

[22] M. Fortunato, M.G. Azar, B. Piot, J. Menick, I Osband, A. Graves, R. Mnih, V. Munos, D. Hassabis, O. Pietquin, C. Blundell, and S. Legg. Noisy networks for exploration. In *International Conference on Representation Learning*, 2018.

[23] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International conference on machine learning*, 2016.

[24] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy. Deep exploration via bootstrapped DQN. In *Advances in Neural Information Processing Systems*, 2016.

[25] T. Schaul, J. Quan, I. Antonoglou, and D. Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016.

[26] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214. JMLR Workshop and Conference Proceedings, 2011.

[27] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[28] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. Scalable bayesian optimization using deep neural networks. In *International conference on machine learning*, pages 2171–2180. PMLR, 2015.

[29] D. Zhou, L. Li, and Q. Gu. Neural contextual bandits with ucb-based exploration. In *Proceedings of the 37th International Conference on Machine Learning*, 2009.

[30] R. Adamczak. A note on the hanson-wright inequality for random vectors with dependencies. *Electronic Communications in Probability*, 20(72):1–13, 2015.

[31] Branislav Kveton, Csaba Szepesväri, Sharan Vaswani, Zheng Wen, Mohammad Ghavamzadeh, and Tor Lattimore. Garbage in, reward out: Bootstrapping exploration in multi-armed bandits. In *Proceedings of the 36rd International Conference on Machine Learning*, 2018.

[32] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[33] Jeff Schlimmer. Mushroom records drawn from the Audubon society field guide to North American mushrooms. *GH Lincoff (Pres), New York*, 1981.

[34] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR, 2015.

[35] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *information retrieval*, 4(2):133–151, 2001.

[36] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Appendix A Multi-armed bandits

A.1 Thompson Sampling for Beta-Bernoulli bandits

Thompson Sampling (TS) in the case of Bernoulli bandits typically assumes that the true values μ_a are sampled from a Beta distribution with parameters α_a and β_a. Accordingly, TS samples the values from Beta(α_a, β_a). This distribution has mean $\hat{\mu}_a = \alpha_a/(\alpha_a + \beta_a)$ and variance $\hat{V}_a = (\alpha_a+\beta_a)^{-2}(\alpha_a+\beta_a+1)^{-1}\alpha_a\beta_a$. The mean $\hat{\mu}_a$ represents exploitation, while the posterior variance $\hat{V}_a$ promotes exploration.

After an action a is selected at step n, the posterior over μ_a is updated by updating the corresponding parameters as $(\alpha_a, \beta_a) \leftarrow (\alpha_a, \beta_a) + (r_n, 1 - r_n)$.

Direct calculation gives that

$$\mathbb{E}[\hat{V}_a] = \mathbb{E}[\hat{\mu}_a(1 - \hat{\mu}_a)/(n_a + 1)] = \mu_a(1 - \mu_a)/n_a + O\left(n_a^{-2}\right).$$

Comparing this last expression with eq. (11) proves Proposition 1.

A.2 Proof of Proposition 2

We first note that:

$$\begin{aligned}\mathbb{E}[e_n^2/n_{a_n}] &= \mathbb{E}\left[(r_n - \hat{\mu}_{a_n})^2/n_{a_n}\right] \\ &= \mathbb{E}\left[(r_n - \mu_{a_n} + \mu_{a_n} - \hat{\mu}_{a_n})^2\right]/n_a = \mu_{a_n}(1 - \mu_{a_n})/n_{a_n} + O\left(n_{a_n}^{-2}\right), \end{aligned} \tag{11}$$

where the last step follows by noticing that $\mathbb{E}\left[(r_n - \mu_{a_n})^2\right] = \mu_{a_n}(1 - \mu_{a_n})$.

Recalling that $\hat{\mu}_a$ are sample averages of Bernoulli variables with means μ_a, we note that

$$\mathbb{E}\left[(\hat{\mu}_a - \mu_a)^2\right] = \mu_a(1 - \mu_a)/n_a.$$

Inserting this result for $a = a_n$ in the previous expression results in

$$\mathbb{E}[e_n^2]/n_{a_n} = \mathbb{E}\left[(r_n - \hat{\mu}_{a_n})^2\right]/n_{a_n} = \mathbb{E}\left[(\hat{\mu}_{a_n} - \mu_{a_n})^2\right] + O\left(n_{a_n}^{-2}\right).$$

proving Proposition 2.

A.3 Empirical Studies on Multi-armed bandits

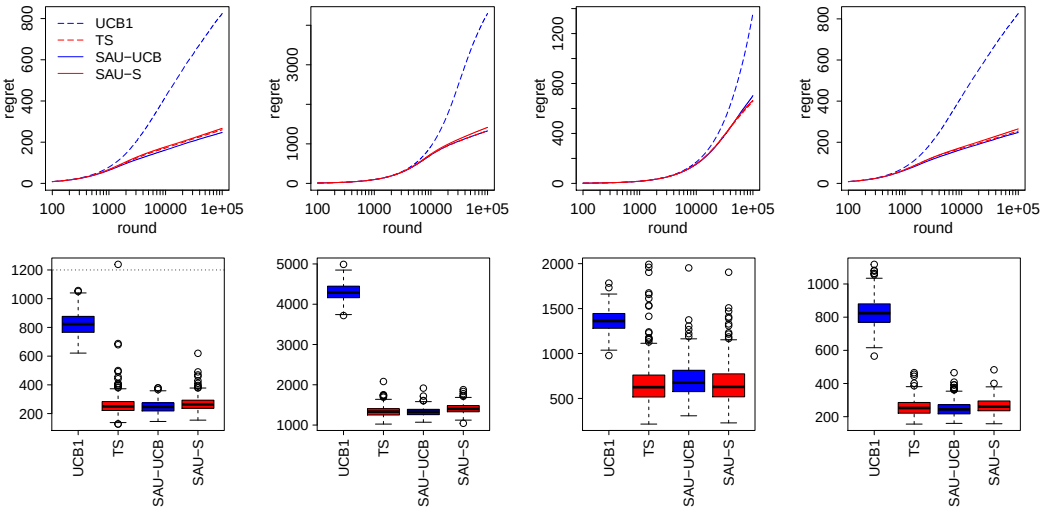
We present a simple synthetic example that simulates Bernoulli rewards and investigates its performance. The reward distribution of action $a \in \{1, \ldots, K\}$, $P_a = \text{Bernoulli}(\mu_a)$, is parameterized by the expected reward $\mu_a \in [0, 1]$. In this simulation, the optimal action has a reward probability of μ_1 and the $K - 1$ other actions have a probability of $\mu_1 - \epsilon$. We consider $\mu_1 = 0.5$, $\epsilon \in \{0.1, 0.02\}$ and $K \in \{10, 50\}$. The horizon is $n = 10^5$ rounds. Our SAU-based exploration approaches, SAU-UCB and SAU-Sampling, are compared to UCB1, and Thompson Sampling (TS).

Fig. 2a shows the regret as a function of round in the case $\epsilon = 0.1$ and $K = 10$. From the figure, we have two observations: (1) SAU-Sampling works similarly as TS; (2) SAU-UCB achieves a much smaller regret than UCB1, even work more or less than TS. This empirical performance shows the potential of our SAU measure.

We continue to empirically investigate the effects of K and ϵ (Fig. 2c), and show the performance in non-Bernoulli payoff (Fig. 2d). We consider a larger number of actions $K = 50$ (Fig. 2b) and a smaller $\epsilon = 0.02$ (Fig. 2c). From these figures, the performance of $K = 50$ or $\epsilon = 0.02$ is consistent with the case $(K, \epsilon) = (10, 0.1)$. We also run a non-Bernoulli bandits, uniform bandit, in which a random variable u is from uniform distribution $[0, 1]$, then the binary reward $r = 1$ if $u \leq \mu_a$, 0 otherwise. We plot the result in Fig. 2d, where the results works consistently as Bernoulli bandits. These results show that SAU works robustly to various multi-armed bandit problems.

A.4 Proofs of bound Proposition 3

Let $\mathbf{r}_{n_a}$ be the sub-vector of $(r_1, r_2, \ldots)^\top$ corresponding to the indices in $\mathbb{T}_a$, and $\mathbf{1}_j = (1, \cdots, 1, 0, \cdots, 0)^\top$ a vector whose first j components are 1 and all the others are zero. Denote



(a) $(K, \epsilon) = (10, 0.1)$. (b) $(K, \epsilon) = (50, 0.1)$. (c) $(K, \epsilon) = (10, 0.02)$. (d) Non-Bernoulli bandit.

Figure 2: Performance on 4 multi-armed bandit problems: three Bernoulli bandits (a,b,c) and one non-Bernoulli (uniform) bandit (d). Upper panels report regret performance as a function of play n averaged over 500 runs. Lower panels report regret distribution at the last play. For Bernoulli bandits rewards are sampled from Bernoulli(μ_a), best action has a reward probability of μ and the $K-1$ other actions have a probability of $\mu - \epsilon$. For the uniform bandit the rewards are sampled from $\mathbb{1}\{U[0,1] < \mu_a\}$, where $U[0,1]$ is a random variable from uniform distribution in $[0,1]$, $K = 10$ and $\epsilon = 0.1$.

$\mathbf{Q}_{n_a} = (\mathbf{1}_1, 2^{-1}\mathbf{1}_2, \cdots, n_a^{-1}\mathbf{1}_{n_a})^\top$. Eqn. (3) then becomes

$$\tau_a^2 = n_a^{-1} \sum\nolimits_{j \in \mathbb{T}_a} [r_j - j^{-1}\mathbf{1}_j^\top \mathbf{r}_{n_a}]^2 = n_a^{-1} \mathbf{r}_{n_a}^\top (\mathbf{I}_{n_a} - \mathbf{Q}_{n_a})^\top (\mathbf{I}_{n_a} - \mathbf{Q}_{n_a}) \mathbf{r}_{n_a}.$$

Noting that $(\mathbf{I}_{n_a} - \mathbf{Q}_{n_a})\mathbf{1}_{n_a} = (0, 0, \cdots, 0)^\top$, we can write

$$\begin{aligned}
\tau_a^2 =& \frac{1}{n_a} \mathbf{r}_{n_a}^\top (\mathbf{I}_{n_a} - \mathbf{Q}_{n_a})^\top (\mathbf{I}_{n_a} - \mathbf{Q}_{n_a}) \mathbf{r}_{n_a} \\
=& \frac{1}{n_a} (\mathbf{r}_{n_a} - \mu_a \mathbf{1}_{n_a})^\top (\mathbf{I}_{n_a} - \mathbf{Q}_{n_a})^\top (\mathbf{I}_{n_a} - \mathbf{Q}_{n_a}) (\mathbf{r}_{n_a} - \mu_a \mathbf{1}_{n_a}) \\
&+ \frac{2\mu_a}{n_a} (\mathbf{r}_{n_a} - \mu_a \mathbf{1}_{n_a})^\top (\mathbf{I}_{n_a} - \mathbf{Q}_{n_a})^\top (\mathbf{I}_{n_a} - \mathbf{Q}_{n_a}) \mathbf{1}_{n_a} - \frac{\mu_a^2}{n_a} \mathbf{1}_{n_a}^\top (\mathbf{I}_{n_a} - \mathbf{Q}_{n_a})^\top (\mathbf{I}_{n_a} - \mathbf{Q}_{n_a}) \mathbf{1}_{n_a} \\
=& \frac{1}{n_a} (\mathbf{r}_{n_a} - \mu_a \mathbf{1}_{n_a})^\top (\mathbf{I}_{n_a} - \mathbf{Q}_{n_a})^\top (\mathbf{I}_{n_a} - \mathbf{Q}_{n_a}) (\mathbf{r}_{n_a} - \mu_a \mathbf{1}_{n_a}) \\
=:& \tau_{a;e}^2. \qquad (12)
\end{aligned}$$

Denote $\mathbf{P}_{n_a} = (\mathbf{I}_{n_a} - \mathbf{Q}_{n_a})^\top (\mathbf{I}_{n_a} - \mathbf{Q}_{n_a})$. By applying Theorem 2.5 in [30](Lemma 2 below), we have that for $t \le \sigma_a^2 n_a / 4$,

$$\begin{aligned}
\Pr\left\{ n_a | \tau_{a;e}^2 - \mathbb{E}[\tau_{a;e}^2] | \ge t \right\} \le& 2 \exp\left(-\frac{1}{8c} \min\left\{ \frac{t^2}{\|\mathbf{P}_{n_a}\|_F^2 \sigma_a^2}, \frac{t}{\|\mathbf{P}_{n_a}\|} \right\} \right) \\
\le& 2 \exp\left(-\frac{1}{8c} \frac{t^2}{\|\mathbf{P}_{n_a}\|_F^2 \sigma_a^2} \right) \\
\le& 2 \exp\left(-\frac{t^2}{8c n_a \sigma_a^2} \right), \qquad (13)
\end{aligned}$$

where c is some constant, the 2nd step is from that $\frac{t^2}{\|\mathbf{P}_{n_a}\|_F^2 \sigma_a^2} < \frac{t}{\|\mathbf{P}_{n_a}\|}$ when $t \le \sigma_a^2 n_a / 4$, due to $\|\mathbf{P}_{n_a}\|_F^2 / \|\mathbf{P}_{n_a}\| \ge n_a / 4$ for $n_a \ge 2$, and the last step is from $\|\mathbf{P}_{n_a}\|_F^2 < n_a$. Denoting

$\delta = 2\exp\left(-\frac{t^2}{8cn_a\sigma_a^2}\right)$, we have that for $\delta \in \left[2\exp\left(-\frac{n_a\sigma_a^2}{32c}\right), 1\right)$,

$$\Pr\left\{|\tau_{a;e}^2 - \mathbb{E}[\tau_{a;e}^2]| \geq \sqrt{\frac{8c\sigma_a^2\log(\delta/2)^{-1}}{n_a}}\right\} \leq \delta.$$

A.5 Two lemmas

Motivated by the analysis on the regret bounds for Thompson Sampling in [15], [31] proved the following lemma which supplies an upper bound of the expected n-round regret under sampling distribution in general.

Lemma 1 *Let the history of action a after n_a plays be a vector H_{a,n_a} of length n_a. without loss of generality we assume that action 1 is optimal. Let $Y \sim dist(H_{a,n_a})$, where $dist(H_{a,n_a})$ is a sampling distribution depending on H_{a,n_a}. Define for $\eta \in \mathbb{R}$*

$$Q_{a,n_a}(\eta) = \Pr(Y \geq \eta | H_{a,n_a}).$$

For $\eta \in \mathbb{R}$, the expected n-round regret can be bounded from above as

$$R(n) \leq \sum_{a:\mu_a<\mu_*} \Delta_a(R_a^{(1)} + R_a^{(2)}),$$

where

$$R_a^{(1)} = \sum_{n_1=0}^{n-1} \mathbb{E}\left[\min\left\{\frac{1}{Q_{1,n_1}(\eta)} - 1, n\right\}\right] \text{ and } R_a^{(2)} = \sum_{n_a=0}^{n-1} \Pr[Q_{a,n_a}(\eta) > 1/n] + 1.$$

The following result is provide by [30].

Lemma 2 *Let X be a mean zero random vector in $\mathbb{R}^n$. If X has the convex concentration property with constant K, then for any $n \times n$ matrix A and every $t > 0$,*

$$Pr\left(|X^\top AX - E(X^\top AX)| \geq t\right) \leq 2\exp\left(-\frac{1}{CK^2}\min\left\{\frac{t^2}{\|A\|_F^2\|Cov(X)\|}, \frac{t}{\|A\|}\right\}\right)$$

for some universal constant C.

A.6 Derivation of $\mathrm{E}(\tau_a^2)$

By simple calculation,

$$\begin{aligned}\text{trace}(\mathbf{P}_{n_a}) &= \text{trace}\left(\mathbf{I}_n - 2\mathbf{Q}_{n_a} + \mathbf{Q}_{n_a}^\top\mathbf{Q}_{n_a}\right)\\ &= n_a - (1 + 1/2 + \cdots + 1/n_a).\end{aligned}$$

Noting $\mathrm{E}(\tau_a^2) = n_a^{-1}\sigma_a^2\text{trace}(\mathbf{P}_{n_a})$ and $\log(n_a+1) < 1 + 1/2 + \cdots + 1/n_a \leq 1 + \log(n_a)$, we have that $\sigma_a^2\left[1 - \frac{\log(n_a)}{n_a}\right] \leq \mathrm{E}(\tau_a^2) < \sigma_a^2\left[1 - \frac{1+\log(1+n_a)}{n_a}\right]$. It follows that as $n_a \to \infty$,

$$\mathrm{E}(\tau_a^2) - \sigma_a^2 \to 0.$$

A.7 Expected Regret Analysis

In this section, we derive an upper bound on the expected regret of the algorithms in Algorithm 1 showing that the optimal logarithmic regrets are achievable uniformly over time. Define $\Delta_a = \mu_* - \mu_a$, where, we recall that μ_a is the expected reward for action a and $\mu_* = \max\{\mu_1, \cdots, \mu_K\}$.

Consider the algorithm SAU-Sampling. Thanks to the fact that Y_a in Algorithm 1 is sampled from a Gaussian, we can derive an upper bound on the expected n-round regret of Algorithm 1 by applying the Gaussian tail bound.

Theorem 1 *If SAU-Sampling is run on a K-armed bandit problem ($K \geq 2$) having arbitrary reward distribution $P_1, \cdots, P_K$ with support in $[0,1]$, then its expected regret after any number $n \geq \max\{\frac{24 \log n}{\min_a \Delta_a^2}, 4\}$ of plays is at most*

$$\sum_{a:\mu_a<\mu_*} \Delta_a \left(\frac{96 \log n}{\Delta_a^2} + 6 \right).$$

A proof of Theorem 1 is provided in Section A.8 below.

In SAU-UCB, $\tau_{a_n}^2$ is updated at every n, but to simplify the analysis we replace $\tau_{a_n}^2$ with a constant τ_a^{*2}. This is justified thanks to Proposition 3 which says that $\tau_{a_n}^2$ is concentrated around its expectation, which tends to $\mu_a^2 + \sigma_a^2$ for growing n_a. With this simplification, we can directly reuse the regret analysis of UCB1, and obtain the following Theorem.

Theorem 2 *If SAU-UCB with fixed $\tau_{a_n}^2 = \tau_a^{*2}$ is run on a K-armed bandit problem ($K \geq 2$) with arbitrary reward distributions $P_1, \ldots, P_K$ with support in $[0,1]$, then for any $\kappa \geq 2/\min\{\tau_a^*\}_{a=1}^K$, its expected regret after an arbitrary number n of plays is at most*

$$\sum_{a:\mu_a<\mu_*} \Delta_a \left(\frac{4 \log n}{\Delta_a^2} + 1 + \frac{\pi^2}{3} \right).$$

Generalization of this regret analysis in the case where the constant $\tau_{a_n}^2$ assumption is invalid is left for future work.

A.8 Proof of Theorem 1

Proof We prove Theorem 1 by applying the Gaussian tail bound and Lemma 1, which is attached in the Appendix. In what follows, without loss of generality we assume that action 1 refers to the optimal action, i.e., $\mu_1 = \mu_*$. Fix n, for simplifying notations, we drop the subindex of n in quantities if without confusion. Let $r_{a,1}, r_{a,2}, \cdots, r_{a,n_a}$ be the random variables referring to the rewards yielded by action a of successive n_a plays. The notation "a, j" means that the number of plays of action a is j. Y_a is random variable drawn from the normal distribution $\mathcal{N}(\hat{\mu}_{a,n_a}, \tau_a^2/n_a)$, where

$$\hat{\mu}_{a,n_a} = \frac{1}{n_a} \sum_{j=1}^{n_a} r_{a,j} \quad \text{and } \tau_a^2 = \frac{1}{n_a} \sum_{j=1}^{n_a} (r_{a,j} - \hat{\mu}_{a,j-1})^2.$$

Rewrite

$$Y_a = \hat{\mu}_{a,n_a} + \delta_{a,n_a}, \quad \text{where } \delta_{a,n_a} \sim \mathcal{N}(0, \tau_a^2/n_a). \tag{14}$$

We bound δ_{a,n_a} in probability. By the Gaussian tail bound (Fact 2), for $\alpha > 0$,

$$\Pr\left\{\delta_{a,n_a} \geq \alpha | \tau_a^2\right\} \leq \exp\left\{-\frac{\alpha^2 n_a}{2\tau_a^2}\right\} \leq \exp\left\{-n_a \alpha^2/2\right\}, \tag{15}$$

where the second inequality is from that $\tau_a^2 \leq 1$. Eqn. (15) follows that

$$\Pr\left\{\delta_{a,n_a} \geq \alpha\right\} \leq \exp\left\{-n_a \alpha^2/2\right\}. \tag{16}$$

Denote $c_{1,n} = \sqrt{n_1^{-1} \log n}$ and $c_{a,n} = \sqrt{n_a^{-1} \log n}$. Let

$$A_1 = \{\hat{\mu}_{1,n_1} > \mu_1 - c_{1,n}\}, \text{ and } A_a = \{\hat{\mu}_{a,n_a} < \mu_a + c_{a,n}\} \text{ for } a = 2, \cdots, K.$$

Denote $\bar{A}_1$ and $\bar{A}_a$ be the complements of A_1 and A_a respectively. $\Pr\left\{\bar{A}_1\right\}$ and $\Pr\left\{\bar{A}_a\right\}$ are bounded from the Azuma-Hoeffding inequality (Fact 1):

$$\Pr\left\{\bar{A}_1\right\} = \Pr\left\{\hat{\mu}_{1,n_1} \leq \mu_1 - c_{1,n}\right\} \leq \exp(-2 \log n) = n^{-2}; \tag{17}$$

$$\Pr\left\{\bar{A}_a\right\} = \Pr\left\{\hat{\mu}_{a,n_a} \geq \mu_a + c_{a,n}\right\} \leq \exp(-2 \log n) = n^{-2}. \tag{18}$$

Define for $\eta \in \mathbb{R}$

$$Q_{a,n_a}(\eta) = \Pr(Y_a \geq \eta).$$

Lemma 1, which is proved by [31], shows that

$$R(n) \leq \sum_{a:\mu_a<\mu_*} \Delta_a(R_a^{(1)} + R_a^{(2)}),$$

where

$$R_a^{(1)} = \sum_{n_1=0}^{n-1} \mathbb{E}\left[\min\left\{\frac{1}{Q_{1,n_1}(\eta)} - 1, n\right\}\right] \text{ and } R_a^{(2)} = \sum_{n_a=0}^{n-1} \Pr[Q_{a,n_a}(\eta) > 1/n] + 1.$$

We set

$$\eta_a = \mu_a + \frac{\Delta_a}{2} = \mu_1 - \frac{\Delta_a}{2}. \tag{19}$$

Our proof is to bound the terms $R_a^{(1)}$ and $R_a^{(2)}$ by applying Lemma 1. Now in the first step we derive the upper bound on $R_a^{(1)}$. Denote $\bar{n}_a = \frac{24\log n}{\Delta_a^2}$. Noting $Q_{1,0}(\eta_a) = 1$,

$$\begin{aligned}
R_a^{(1)} &= \sum_{n_1=1}^{n-1} \mathbb{E}\left[\min\left\{\frac{1}{Q_{1,n_1}(\eta_a)} - 1, n\right\}\right] \\
&= \sum_{n_1=1}^{\lceil \bar{n}_a \rceil - 1} \mathbb{E}\left[\min\left\{\frac{1}{Q_{1,n_1}(\eta_a)} - 1, n\right\}\right] + \sum_{n_1=\lceil \bar{n}_a \rceil}^{n-1} \mathbb{E}\left[\min\left\{\frac{1}{Q_{1,n_1}(\eta_a)} - 1, n\right\}\right] \\
&=: R_a^{(1)_L} + R_a^{(1)_U},
\end{aligned} \tag{20}$$

where $\lceil x \rceil$ is the smallest integer not less than x.

The law of total probability implies that, when $n_1 \geq \bar{n}_a$,

$$\begin{aligned}
Q_{1,n_1}(\eta_a) =& \Pr\{Y_1 > \eta_a\} = 1 - \Pr\{Y_1 < \eta_a\} \\
=& 1 - \Pr(A_1)\Pr\{Y_1 < \eta_a | A_1\} - \Pr(\bar{A}_1)\Pr\left\{Y_1 < \eta_a | \bar{A}_1\right\} \\
>& 1 - \Pr\{\hat{\mu}_{1,n_1} + \delta_{1,n_1} < \mu_1 - \Delta_a/2 | A_1\} - \Pr(\bar{A}_1) \\
\geq& 1 - \Pr\{\delta_{1,n_1} \leq -\Delta_a/2 + c_{1,n}\} - \Pr(\bar{A}_1) \\
>& 1 - \Pr\left\{\delta_{1,n_1} \leq -\sqrt{2}c_{1,n}\right\} - \Pr(\bar{A}_1) \\
\geq& 1 - n^{-1} - n^{-2},
\end{aligned} \tag{21}$$

where the 1st inequality is from the facts that $\Pr(A_1) < 1$ and $\Pr\left\{Y_1 < \eta_a | \bar{A}_1\right\} < 1$, the 2nd inequality is from the definition of A_1, the 3rd inequality is from $\Delta_a/2 - c_{1,n} \geq \sqrt{2}c_{1,n}$ as $n_1 \geq \bar{n}_a$, and the last inequality is from Eqns. (16) and (17).

Eqn. (21) implies that for $n \geq 3$,

$$\begin{aligned}
R_a^{(1)_U} &= \sum_{n_1=\lceil \bar{n}_a \rceil}^{n-1} \mathbb{E}\left[\min\left\{\frac{1}{Q_{1,n_1}(\eta_a)} - 1, n\right\}\right] = \sum_{n_1=\lceil \bar{n}_a \rceil}^{n-1} \min\left\{\frac{1}{Q_{1,n_1}(\eta_a)} - 1, n\right\} \\
&< \sum_{n_1=\lceil \bar{n}_a \rceil}^{n-1} \frac{1}{1 - n^{-1} - n^{-2}} - 1 < 2 \sum_{n_1=\lceil \bar{n}_a \rceil}^{n-1} (n^{-1} + n^{-2}) < 4,
\end{aligned} \tag{22}$$

where the 1st inequality is from that $\frac{1}{Q_{1,n_1}(\eta_a)} - 1 < \frac{1}{1-n^{-1}-n^{-2}} - 1 < n$, the 2nd inequality is from that $\frac{1}{1-n^{-1}-n^{-2}} - 1 < 2(n^{-1} + n^{-2})$ when $n \geq 3$.

Next we investigate the term $R_a^{(1)_L}$ in Eqn. (20). For it, we now provide another lower bound of the $Q_{1,n_1}(\eta_a)$. Let

$$A_1^L = \{\hat{\mu}_{1,n_1} \geq \mu_1\}.$$

Denote $\bar{A}_1^L$ be the complements of A_1^L. We have that

$$\Pr\left\{A_1^L\right\} = 1/2; \;\; \Pr\left\{\bar{A}_1^L\right\} = 1/2 \tag{23}$$

Similarly as Eqn. (21), we have that

$$\begin{aligned}
Q_{1,n_1}(\eta_a) =& \text{Pr}\left\{Y_{1,n_1} > \eta_a\right\} \\
=& 1 - \text{Pr}(A_1^L)\text{Pr}\left\{Y_{1,n_1} < \eta_a | A_1^L\right\} - \text{Pr}(\bar{A}_1^L)\text{Pr}\left\{Y_{1,n_1} < \eta_a | \bar{A}_1^L\right\} \\
>& 1 - 1/2\text{Pr}\left\{\hat{\mu}_{1,n_1} + c_{1,n}\delta_{1,n_1} < \mu_1 - \Delta_a/2 | A_1^L\right\} - \text{Pr}(\bar{A}_1^L) \\
\geq& 1/2 - 1/2\text{Pr}\left\{c_{1,n}\delta_{1,n_1} \leq -\Delta_a/2\right\} \\
\geq& \frac{1}{2}\left[1 - \exp\left(-\frac{n_1\Delta_a^2}{8\log n}\right)\right],
\end{aligned} \tag{24}$$

where the 1st inequality is from the facts that $\text{Pr}\left\{Y_{1,n_1} < \eta_a | \bar{A}_1^L\right\} < 1$, and the 2nd inequality is from the definition of A_1^L and Eqn. (23), and the last inequality is from Eqn. (16).

We have that (1) $\log(1 - \frac{2}{n+1}) \geq \frac{-3}{n+1}$ when $n \geq 4$; and (2) $-\frac{\Delta_a^2}{8\log n} \leq \frac{-3}{n+1}$ when $\frac{n}{\log n} \geq \frac{24}{\min_a \Delta_a^2}$. The two inequalities imply that when $n \geq \max\{\frac{24\log n}{\min_a \Delta_a^2}, 4\}$,

$$1 - \exp\left(-\frac{\Delta_a^2}{8\log n}\right) \geq \frac{2}{n+1}.$$

Thus, Eqn. (24) follows that

$$\begin{aligned}
R_a^{(1)_L} =& \sum_{n_1=1}^{\lceil \bar{n}_a \rceil - 1} \mathbb{E}\left[\min\left\{\frac{1}{Q_{1,n_1}(\eta_a)} - 1, n\right\}\right] \\
\leq& \sum_{n_1=1}^{\lceil \bar{n}_a \rceil - 1}\left[\frac{2}{1 - \exp\left(-\frac{n_1\Delta_a^2}{8\log n}\right)} - 1\right] \\
<& \sum_{n_1=1}^{\lceil \bar{n}_a \rceil - 1}\left[4\exp\left(-\frac{n_1\Delta_a^2}{8\log n}\right) - 1\right] \\
<& 3\bar{n}_a.
\end{aligned} \tag{25}$$

Therefore, inserting Eqns. (22) & (25) into Eqn. (20),

$$R_a^{(1)} \leq 3\bar{n}_a + 4. \tag{26}$$

In the next step we derive the upper bound on $R_a^{(2)}$. When $n_a \geq \bar{n}_a$,

$$\Delta_a/2 - c_{a,n} \geq \sqrt{2}c_{a,n}. \tag{27}$$

From the definition of event A_a, the law of total probability implies that

$$\begin{aligned}
\text{Pr}\left\{Q_{a,n_a}(\eta_a) > 1/n\right\} =& \text{Pr}(A_a)\text{Pr}\left\{Q_{a,n_a}(\eta_a) > 1/n | A_a\right\} + \text{Pr}(\bar{A}_a)\text{Pr}\left\{Q_{a,n_a}(\eta_a) > 1/n | \bar{A}_a\right\} \\
\leq& \text{Pr}\left\{Q_{a,n_a}(\eta_a) > 1/n | A_a\right\} + \text{Pr}(\bar{A}_a).
\end{aligned} \tag{28}$$

When $n_a \geq \bar{n}_a$, given event A_a,

$$\begin{aligned}
Q_{a,n_a}(\eta_a) =& \text{Pr}\left\{Y_a > \eta_a | A_a\right\} = \text{Pr}\left\{\hat{\mu}_a + \delta_{a,n_a} > \Delta_a/2 + \mu_a | A_a\right\} \\
\leq& \text{Pr}\left\{\delta_{a,n_a} \geq \Delta_a/2 - c_{a,n}\right\} \\
\leq& \text{Pr}\left\{\delta_{a,n_a} \geq \sqrt{2}c_{a,n}\right\} \\
\leq& \exp\left\{-\log n\right\} = n^{-1}.
\end{aligned} \tag{29}$$

where the 1st inequality is from the definition of event A_a, the 2nd inequality is from Eqn. (27), the 3rd inequality is from Eqn. (16).

Eqn. (29) follow that when $n_a \geq \bar{n}_a$,

$$\text{Pr}\left\{Q_{a,n_a}(\eta_a) > 1/n | A_a\right\} = 0. \tag{30}$$

Inserting Eqn. (30) into Eqn. (28),

$$\Pr[Q_{a,n_a}(\eta_a) > 1/n] \leq \Pr(\bar{A}_a) \leq n^{-1}, \tag{31}$$

where the last step is from Eqn. (18).

We have that

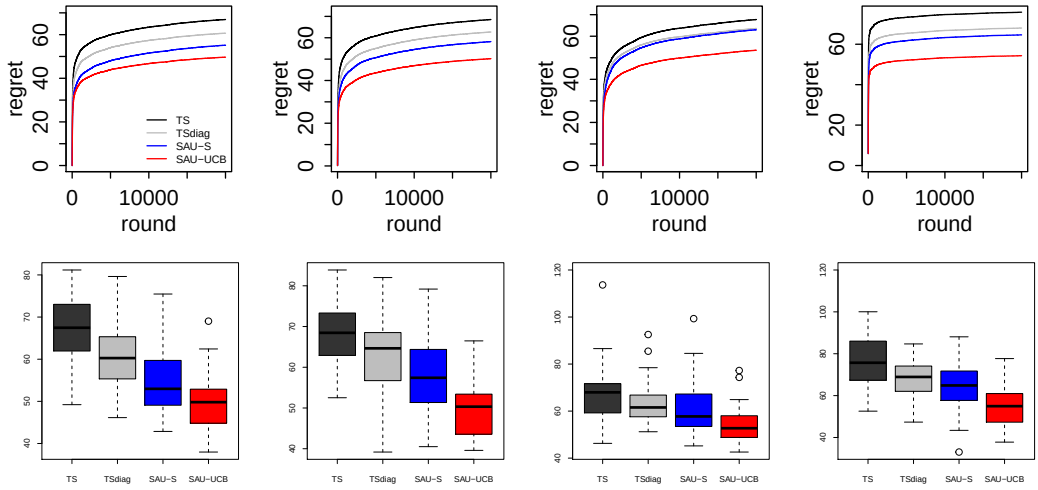
$$\begin{aligned} R_a^{(2)} &= \sum_{n_a=0}^{n-1} \Pr[Q_{a,n_a}(\eta_a) > 1/n] + 1 \\ &\leq \bar{n}_a + \sum_{n_a=\lceil \bar{n}_a \rceil}^{n-1} \Pr[Q_{a,n_a}(\eta_a) > 1/n] + 1 \\ &< \bar{n}_a + 2, \end{aligned} \tag{32}$$

where the 1st inequality is from direct calculation, the 2nd inequality is from Eqn. (31). Based on the bounds of $R_a^{(1)}$ in Eqn. (20) and $R_a^{(2)}$ in Eqn. (32), we have the statement

$$R(n) < \sum_{a=2}^{K} \Delta_a \left(\frac{96 \log n}{\Delta_a^2} + 6 \right).$$

■

Appendix B Performance of linear contextual bandits in other settings



(a) β is from Gaussian. (b) errors are correlated. (c) contexts are correlated. (d) $x_{ij} \sim t(2)$.

Figure 3: Performance of other cases: the elements of β are from $\mathcal{N}(0, 1)$ then are normalized; the model errors are correlated; and the contexts are correlated; and the elements of $\mathbf{x}_i$ are from a heavy-tailed truncated t-distribution with $df = 2$.

Appendix C Wheel Bandit Problem

The *Wheel Bandit Problem* is a synthetic bandit designed by [18] where the effect of need for exploration can be systematically studied by varying a parameter $\delta \in [0, 1]$. The difficulty of the problem increases with δ, since the problem is designed so that for δ close to 1 most contexts have the same optimal action, while only for a fraction $1 - \delta^2$ of contexts the optimal action is a different more rewarding action (see [18] for more details).

Table 2: Cumulative regret incurred on the Wheel Bandit Problem by the 4 best algorithms identified by [18] compared to our SAU-based algorithms (in bold) with increasing values of δ. We report the mean and standard error of the mean over 50 trials. The best cumulative regret value for each task is marked in bold.

δ	0.50	0.70	0.90	0.95	0.99
Linear	60.62 ± 3.55	83.56 ± 5.21	84.11 ± 4.72	86.71 ± 4.08	93.61 ± 2.74
LinearGreedy	88.95 ± 0.19	124.38 ± 0.46	122.18 ± 1.05	119.85 ± 1.55	99.59 ± 2.61
NeuralGreedy	35.00 ± 0.90	56.65 ± 1.50	72.04 ± 1.56	83.97 ± 2.53	91.18 ± 1.91
NeuralLinear	18.71 ± 2.05	26.63 ± 1.21	45.47 ± 2.91	65.44 ± 3.23	**86.03 ± 3.03**
SAU-Linear-S	34.74 ± 1.63	42.69 ± 2.27	54.88 ± 3.01	71.37 ± 3.57	**87.38 ± 2.85**
SAU-Linear-UCB	42.23 ± 1.55	60.08 ± 2.12	78.51 ± 4.93	92.42 ± 4.67	96.79 ± 2.50
SAU-Neural-S	**12.49 ± 4.11**	**13.72 ± 0.85**	**36.54 ± 3.03**	**63.30 ± 3.53**	106.89 ± 14.17
SAU-Neural-UCB	26.29 ± 2.32	51.92 ± 2.03	73.87 ± 4.58	75.95 ± 2.24	89.99 ± 2.65
Uniform	100.00 ± 28.43	100.00 ± 0.46	100.00 ± 0.86	100.00 ± 1.41	100.00 ± 2.75

Table 3: Same results as the Table 2 above, but not normalized with respect to the uniform exploration strategy

δ	0.50	0.70	0.90	0.95	0.99
Linear	49742.3 ± 2911.4	33109.1 ± 2063.5	12816.6 ± 719.3	6954.2 ± 327.0	1864.8 ± 54.6
LinearGreedy	72982.5 ± 158.6	49287.7 ± 183.8	18617.9 ± 160.2	9611.9 ± 124.3	1983.9 ± 51.9
NeuralGreedy	28722.7 ± 739.4	22448.0 ± 592.6	10977.4 ± 237.2	6733.9 ± 203.1	1816.5 ± 38.0
NeuralLinear	15350.4 ± 1685.2	10550.6 ± 479.5	6929.1 ± 443.4	5248.1 ± 259.2	**1713.9 ± 60.4**
SAU-Linear-S	28506.8 ± 1336.8	16914.3 ± 899.3	8362.1 ± 459.0	5723.6 ± 286.2	**1740.8 ± 56.9**
SAU-Linear-UCB	34648.4 ± 1275.8	23806.7 ± 839.6	11962.9 ± 751.8	7411.6 ± 374.3	1928.2 ± 49.8
SAU-Neural-S	**10247.5 ± 3369.4**	**5437.5 ± 335.5**	**5568.3 ± 461.1**	**5076.8 ± 283.1**	2129.3 ± 282.2
SAU-Neural-UCB	21572.5 ± 1903.9	20571.7 ± 803.7	11256.4 ± 697.7	6091.3 ± 179.7	1792.8 ± 52.9
Uniform	82053.3 ± 23324.5	39625.3 ± 182.9	15238.1 ± 130.8	8019.8 ± 113.2	1992.1 ± 54.7

Appendix D Real-world Datasets

Here below we list all the dataset used to reproduce the deep bandit experiments in [18]. Most datasets are from he UCI Machine Learning Repository [32] and were manually inspected to verify that they do not contain personally identifiable information or offensive content.

Mushroom. The Mushroom Dataset [33] has $N = 8124$ samples with $d = 117$ features (one-hots from 22 categorical attributes), divided in 2 classes ('poisonous' and 'edible'). As in [34], the bandit problem has $k = 2$ actions: 'eat' mushroom or 'pass'. If sample is in class 'edible' action 'eat' gets reward $r = +5$. If sample is in class 'poisonous' action 'eat' gets reward $r = +5$ with probability 1/2 and reward $r = -35$ otherwise. Action 'pass' gets reward of 0 in all cases. We use a number of steps of $T = 50000$.

Statlog. The Shuttle Statlog Dataset [32] has $N = 43500$ samples with $d = 9$ features, divided in 7 classes. In the bandit problem the agent has to select the right class ($k = 7$ actions), in which case it gets reward $r = 1$, and $r = 0$ otherwise. The number of steps is $T = 43500$. Note: one action is the optimal one in 80% of cases, meaning that a successful algorithm has to avoid committing to this action and explore instead.

Covertype. The Covertype Dataset [32] has $N = 581012$ samples with $d = 54$, divided in 7 classes. In the bandit problem the agent has to select the right class ($k = 7$ actions), in which case it gets reward $r = 1$, and $r = 0$ otherwise. The number of steps is $T = 50000$.

Financial. The Financial Dataset was created as in [18] by pulling stock prices of 21 publicly traded companies in NYSE and Nasdaq between 2004 and 2018. The dataset has $N = 3713$ samples with $d = 21$ features. The arms are synthetically created to be a linear combination of the features, representing $k = 8$ different portfolios. The number of steps is $T = 3713$. Note: this is a very the small horizon, meaning that algorithms may over-explore at the beginning with no time to amortize the initial regret.

Jester. The Jester Dataset [35] is preprocessed following [18] to have $N = 19181$ samples with $d = 32$ features associated to 8 continuous outputs. The agent selects one of the $k = 8$ outputs and obtains the reward corresponding to selected output. The number of steps is $T = 19181$.

Adult. The training partition of Adult Dataset [32] has $N = 30162$ samples (after dropping rows with NA values). As in [18] we turn the dataset into a contextual bandit by considering the $k = 14$ occupations as feasible actions to be selected based on the remaining $d = 92$ features (after encoding categorical variables into one-hots). The agent gets a reward of $r = 1$ for making the right prediction, and $r = 0$ otherwise. The number of steps is $T = 30162$.

Census. The US Census (1990) Dataset [32] has $N = 2458285$ samples with $d = 387$ features (after encoding categorical features to one-hots). The goal in the bandit tasks is to predict the occupation feature among $k = 9$ classes. The agent obtains reward $r = 1$ for making the right prediction, and $r = 0$ otherwise. The number of steps is $T = 25000$.

Appendix E Additional algorithms and results

Algorithm 5 ϵ-greedy exploration for Deep Contextual Bandits (NeuralGreedy)

function PREDICT($\mathbf{x}_n$)
    Return predicted values $\{\hat{\mu}_{n,a}\}_{a\in\mathbb{K}} = \mu(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_n)$;
function ACTION($\{\hat{\mu}_{n,a}\}_{a\in\mathbb{K}}$)
    Compute $a^* = \arg\max_a(\{\tilde{\mu}_{n,a}\}_{a\in\mathbb{K}})$;
    Return $a_n = \begin{cases} a^*, \text{with probability } 1-\epsilon \\ a \in \mathbb{K} \text{ uniformly at random, otherwise;} \end{cases}$ (**Exploration**)
function UPDATE($r_n, a_n, \mathbf{x}_n$)
    Compute loss $l_n = (r_n - \hat{\mu}_{n,a_n})^2$;
    Update model parameters $\hat{\boldsymbol{\theta}}_{n+1}$ using SGD with gradients $\frac{\partial l_n}{\partial \hat{\boldsymbol{\theta}}_n}$ (or mini-batch version);
    Update exploration parameter: decrease ϵ according to annealing schedule;

Table 4: Cumulative regret incurred on the contextual bandits in [18] by the 4 best algorithms that they identified and described in Appendix D compared to our SAU-based algorithms (in bold). This table shows the same results as Table 1 without normalizing to the Uniform algorithm. We report the mean and standard error of the mean over 50 trials. The best cumulative regret value for each task is marked in bold.

	Mushroom	**Statlog**	**Covertype**	**Financial**	**Jester**	**Adult**	**Census**
LinearPosterior	7373.7 ± 376.1	3837.4 ± 70.0	15807.6 ± 28.2	509.4 ± 7.3	61674.5 ± 383.8	31861.4 ± 26.7	10380.0 ± 17.6
LinearGreedy	11332.3 ± 862.1	40725.9 ± 354.8	22933.0 ± 858.6	731.9 ± 48.4	60427.6 ± 433.7	36409.3 ± 99.9	14525.9 ± 348.7
NeuralLinear	6503.8 ± 200.8	469.7 ± 9.6	12509.9 ± 28.1	480.7 ± 8.6	66948.0 ± 411.1	32874.2 ± 30.3	9113.7 ± 19.9
NeuralGreedy	65153.3 ± 771.9	14974.8 ± 173.3	38085.6 ± 53.5	4060.0 ± 64.7	89748.9 ± 740.6	41559.7 ± 11.3	19075.3 ± 57.9
Linear-SAU-S	11188.6 ± 954.0	3892.8 ± 97.2	15554.1 ± 49.5	411.9 ± 24.8	60304.6 ± 413.2	31370.9 ± 55.6	8812.4 ± 25.5
Linear-SAU-UCB	7563.5 ± 375.3	3736.8 ± 80.5	15761.4 ± 74.7	307.7 ± 17.7	62078.3 ± 381.8	**31337.0 ± 29.2**	8885.6 ± 20.7
Neural-SAU-S	**5381.1 ± 131.0**	232.6 ± 5.2	**11771.5 ± 26.9**	264.7 ± 5.3	**58795.1 ± 548.8**	32765.8 ± 30.7	8614.5 ± 18.2
Neural-SAU-UCB	5665.7 ± 143.5	**225.0 ± 5.5**	11958.2 ± 24.5	**248.8 ± 7.1**	59997.0 ± 586.6	32793.4 ± 27.3	**8565.7 ± 19.6**
Uniform	244431.4 ± 583.4	37275.8 ± 9.9	42864.5 ± 9.6	4729.0 ± 69.4	96353.3 ± 923.2	41994.5 ± 7.1	22226.4 ± 8.0

Table 5: Running times for the simulations in Tables 1 and 4 run on Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz. Results are averaged over 3 trials and the corresponding standard errors are also shown. Simulation time of SAU algorithms is consistently close the corresponding epsilon-greedy algorithm, confirming its computational efficiency.

	Mushroom	**Statlog**	**Covertype**	**Financial**	**Jester**	**Adult**	**Census**
LinearPosterior	1203.2 s ± 4.9	384.0 s ± 1.9	900.5 s ± 37.5	10.8 s ± 0.2	115.0 s ± 3.1	1554.1 s ± 8.7	7273.3 s ± 31.3
LinearGreedy	542.6 s ± 6.7	334.0 s ± 7.9	406.4 s ± 0.8	3.0 s ± 0.0	56.4 s ± 1.0	359.5 s ± 3.8	141.9 s ± 1.1
NeuralGreedy	112.5 s ± 4.0	70.7 s ± 1.2	85.8 s ± 1.4	6.0 s ± 0.1	28.5 s ± 0.1	87.3 s ± 2.5	78.4 s ± 0.4
NeuralLinear	790.6 s ± 23.0	1032.4 s ± 44.8	1282.5 s ± 13.8	80.0 s ± 0.9	452.5 s ± 13.8	1997.1 s ± 6.8	784.2 s ± 5.7
Linear-SAU-S	463.6 s ± 1.1	332.3 s ± 9.0	479.7 s ± 70.3	3.2 s ± 0.0	55.6 s ± 0.5	335.4 s ± 24.9	150.6 s ± 3.5
Linear-SAU-UCB	460.2 s ± 2.2	338.8 s ± 8.0	398.1 s ± 1.5	3.2 s ± 0.0	56.6 s ± 1.3	308.3 s ± 0.9	144.7 s ± 0.6
Neural-SAU-S	116.4 s ± 3.9	77.5 s ± 2.2	99.2 s ± 3.4	6.3 s ± 0.6	29.7 s ± 0.3	86.0 s ± 3.0	81.8 s ± 0.4
Neural-SAU-UCB	115.6 s ± 4.2	81.0 s ± 2.0	97.5 s ± 1.4	6.1 s ± 0.4	29.3 s ± 0.1	85.6 s ± 3.3	82.1 s ± 0.1
Uniform	31.9 s ± 0.7	13.5 s ± 0.0	20.8 s ± 0.1	0.7 s ± 0.0	3.4 s ± 0.0	14.5 s ± 0.0	34.4 s ± 0.2

Hyperparameters for deep contextual bandits

Here we list the settings of the main hyperparameter of our simulations. For more details on how to reproduce our results, please refer to our released code.

- All neural network models are an MLP with 2 hidden layers of size 100 with ReLU activations.

- Training is generally done with Adam SGD [36] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and learning rate 0.003.
- The default model updating consisted in training for $t_s = 10$ mini-batches of size 64 every $t_f = 20$ steps.
- The hyperparameters for the competing Bayesian bandit algorithms are based on those mentioned in [18].
- For *Linear-SAU-Sampling* and *Linear-SAU-UCB* the parameter `lambda_prior` (the initial variance of the inverse-covariance matrix) is set to 20.0 for all datasets, apart from Financial (which is rather small) for which it is set to 0.25.

Appendix F Paper Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We support our claims with both theoretical proofs and empirical validations.
   (b) Did you describe the limitations of your work? [Yes] Particularly in the discussion section of our paper we mention some theoretical limitations of our work, as well as its limitations due to the restricted application domain.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] In the discussion we mentioned the potential negative societal impacts that deploying inaccurate exploration algorithms might have due to their use for instance in recommendation and ad servicing systems.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] All our theoretical results and their proofs clearly state the assumptions under which they are valid.
   (b) Did you include complete proofs of all theoretical results? [Yes] We provided complete proofs of all our theoretical results in the indicated Appendix sections.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We included pseudocode of our algorithms in the paper, and provide the code along with instructions to reproduce our experiments as a zipped repo.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We specified how the data was used in the Appendix, as well as in the released code.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We reported standard errors around the average performance scores.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In the Appendix we mentioned what type of machine the main experiments were run on and the time it took to run them.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Citations are in the Appendix.
   (b) Did you mention the license of the assets? [Yes] Licenses are mentioned in the released code.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We included our code released as an anonymized github repository.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We mentioned citations and licenses in the Appendix and in the released code.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] In the Appendix we mention that the dataset were inspected to check that they do not contain personally identifiable information or offensive content.