Model-based Reinforcement Learning for Continuous Control with Posterior Sampling

Ying Fan [1] **Yifei Ming** [1]

Abstract

Balancing exploration and exploitation is crucial in reinforcement learning (RL). In this paper, we study model-based posterior sampling for reinforcement learning (PSRL) in continuous state-action spaces theoretically and empirically. First, we show the first regret bound of PSRL in continuous spaces which is polynomial in the episode length to the best of our knowledge. With the assumption that reward and transition functions can be modeled by Bayesian linear regression, we develop a regret bound of $\tilde{O}(H^{3/2}d\sqrt{T})$, where H is the episode length, d is the dimension of the state-action space, and T indicates the total time steps. This result matches the best-known regret bound of non-PSRL methods in linear MDPs. Our bound can be extended to nonlinear cases as well with feature embedding: using linear kernels on the feature representation ϕ, the regret bound becomes $\tilde{O}(H^{3/2}d_\phi\sqrt{T})$, where d_ϕ is the dimension of the representation space. Moreover, we present MPC-PSRL, a model-based posterior sampling algorithm with model predictive control for action selection. To capture the uncertainty in models, we use Bayesian linear regression on the penultimate layer (the feature representation layer ϕ) of neural networks. Empirical results show that our algorithm achieves the state-of-the-art sample efficiency in benchmark continuous control tasks compared to prior model-based algorithms, and matches the asymptotic performance of model-free algorithms.

1. Introduction

In reinforcement learning (RL), an agent interacts with an unknown environment which is typically modeled as a Markov Decision Process (MDP). Efficient exploration in RL has been one of the main challenges: the agent is expected to balance between *exploring* unseen state-action pairs to gain more knowledge about the environment, and *exploiting* existing knowledge to optimize rewards in the presence of known data. Specifically, when the state-action spaces are *continuous*, function approximation is necessary to approximate the value function (in model-free settings) or the reward and transition functions (in model-based settings), which raises extra challenges for both computational and statistical efficiency compared to finite tabular cases[1].

Frequentist Regrets with Upper Confidence Bound Most existing works, which focus on using function approximations to achieve efficient exploration with performance guarantees, use algorithms based on Upper Confidence Bound (UCB) to develop frequentist regret bounds. In the model-based settings, the state-of-the-art frequentist bound is given by UC-MatrixRL (Yang & Wang, 2019), which achieves a regret bound of $\tilde{O}(H^2d\sqrt{T})$, where H is the episode length, d is the dimension of the state-action space, and T indicates the total time steps. In model-free settings, Jin et al. (2020) proposed LSVI-UCB and developed a bound of $\tilde{O}(H^{3/2}d^{3/2}\sqrt{T})$. This bound is further improved to $\tilde{O}(H^{3/2}d\sqrt{T})$ by (Zanette et al., 2020) and (Ayoub et al., 2020), which achieve the best-known frequentist bound among model-free and model-based algorithms. All above bounds are achieved in linear MDPs where both reward and transition functions are modeled as linear functions. Those results can be extended to non-linear cases using kernel functions, replacing d with d_ϕ where d_ϕ is the dimension of the feature space. However, there are two main drawbacks of UCB-based methods. First, UCB requires optimizing over a confidence set, which is likely to be computationally prohibitive. Second, the lack of statistical efficiency can emerge from the sub-optimal construction of the confidence set (Osband & Van Roy, 2017). Accordingly, this line of works mostly focuses on theoretical analysis rather than empirical applications.

[1]University of Wisconsin-Madison. Correspondence to: Ying Fan <yfan87@wisc.edu>, Yifei Ming <ming5@wisc.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

[1]Tabular RL has been extensively studied with a regret bound of $\tilde{O}(H\sqrt{SAT})$, where S and A denote the number of states and actions respectively. However, in continuous state-action spaces S and A can be infinite, hence the above results do not apply to continuous spaces.

Bayesian Regrets with Posterior Sampling Another line of works in Bayesian reinforcement learning treats MDP as a random variable with a prior distribution. This prior distribution of the MDP provides an initial uncertainty estimate of the environment, which generally contains distributions of transition dynamics and reward functions. The epistemic uncertainty (subjective uncertainty due to limited data) in reinforcement learning can be captured by posterior distributions given the data collected by the agent. Bayesian regrets naturally provide performance guarantees in this setting. Posterior sampling for reinforcement learning (PSRL), motivated by Thompson sampling in bandit problems (Thompson, 1933), serves as a provably efficient algorithm under Bayesian settings. In PSRL, the agent follows an optimal policy for a single MDP sampled from the posterior distribution for interaction in each episode, instead of optimizing over a confidence set, so PSRL is more computationally tractable than UCB-based methods.

Although PSRL with function approximation in continuous MDPs has been studied, to the best of our knowledge, there is no existing work that develops a regret bound which is *clearly dependent in episode length H with a polynomial order* while simultaneously sub-linear in T as bounds developed via UCB. Existing results either provide no clear dependency on H or suffer from an exponential order of H.

Limitations on the Order of H in PSRL with Function Approximation In model-based RL, (Osband & Van Roy, 2014) derive a regret bound of $\tilde{O}(\sigma_R\sqrt{d_K(R)d_E(R)T} + \mathbb{E}[L^*]\sigma_p\sqrt{d_K(P)d_E(P)})$ in their Corollary 1, where L^* is a global Lipschitz constant for the future value function (see Section 3.2), d_K and d_E are Kolmogorov and eluder dimensions, and R and P refers to function classes of rewards and transitions. However, L^* is actually dependent on H (see our remark in Section 3.2). Such dependency is not explored in their paper, and they didn't provide a clear dependency on H in their Corollary 1. Moreover, they give a very loose bound (exponential in H) in their Corollary 2 for LQR, which we will discuss in detail in Section 3.4. Chowdhury & Gopalan (2019) considers the regret bound for kernelized MDP which is sub-linear in T. However, they only mention that L^* basically measure the connectedness of the MDP without discussing the dependency of H in L^* in continuous state-action spaces, and they followed (Osband & Van Roy, 2014) in their Corollary 2 for LQR with an exponential order of H. In model-free settings, (Azizzadenesheli et al., 2018) develops a regret bound of $\tilde{O}(d_\phi\sqrt{T})$ using a linear function approximator in the Q-network, where d_ϕ is the dimension of the feature representation vector of the state-action space, but their bound is exponential in H as mentioned in their paper.

Motivated by the drawbacks of previously discussed UCB-based and PSRL works, we are interested in the following question: in continuous MDPs, *can PSRL achieve provably efficient exploration with polynomial orders of d and H in regret bounds, while still enjoy computational tractability of solving a single known MDP?*

Our Results In this paper, we study model-based PSRL in continuous state-action spaces. We assume that rewards and transitions can be modeled by Bayesian Linear regression (Rasmussen, 2003), and extend the assumption to non-linear settings using feature representation.

The key differences of our analysis compared to previous work in PSRL with function approximation are as follows: **First**, we show the order of H can be polynomial in PSRL with continuous state-action spaces: in Section 3.2, we use the property derived from any noise with a symmetric probability distribution, which includes many common noise assumptions, to derive a closed-form solution of the Lipschitz constant L^* mentioned in (Osband & Van Roy, 2014). As a result, in Section 3.4 we can develop a regret bound with polynomial dependency on H. **Second**, our analysis requires less assumptions (especially compared to Chowdhury & Gopalan (2019)): we omit their Lipschitz assumption (discussed in Section 3.2) and regularity assumption (discussed in Section 3.3). **Third**, our bound enjoys lower dimensionality (especially compared to Osband & Van Roy (2014)) as discussed in Section 3.5.

To the best of our knowledge, we are the first to show that the regret bound for PSRL in continuous state-action spaces can be polynomial in the episode length H and simultaneously sub-linear in T: For the linear case, we develop a Bayesian regret bound of $\tilde{O}(H^{3/2}d\sqrt{T})$. Using feature embedding, we derive a bound of $\tilde{O}(H^{3/2}d_\phi\sqrt{T})$. Our regret bound match the order of *best-known* regret bound of UCB-based methods (Zanette et al., 2020; Ayoub et al., 2020), which is also $\tilde{O}(H^{3/2}d\sqrt{T})$.[2] As far as is known, no previous works of UCRL have presented empirical results in continuous control. In contrast, PSRL only requires optimizing a **single** MDP, and thus enjoys more computationally tractability compared to UCB-based methods.

Moreover, we implement PSRL with function approximation as a computationally tractable method with performance guarantee: We use Bayesian linear regression (BLR) (Rasmussen, 2003) on the penultimate layer (for feature representation) of neural networks when fitting transition and reward models. We use model predictive control (MPC) (Camacho & Alba, 2013) as an approximate optimization solution of the sampled MDP, to optimize the policy under

[2] We can compare them together in the Bayesian framework as discussed in Osband & Van Roy (2017): A frequentist regret bound for a confidence set of MDPs implies an identical bound on the Bayesian regret for any prior distribution of MDPs with support on the same confidence set.

the sampled models in each episode as described in Section 4. Experiments show that our algorithm achieves more efficient exploration compared with previous model-based algorithms in control benchmark tasks (see Section 5).

2. Preliminaries

2.1. Problem Formulation

We model an episodic finite-horizon Markov Decision Process (MDP) M as $\{\mathcal{S}, \mathcal{A}, R^M, P^M, H, \sigma_r, \sigma_f, R_{\max}, \rho\}$, where $\mathcal{S} \subset \mathbb{R}^{d_s}$ and $\mathcal{A} \subset \mathbb{R}^{d_a}$ denote state and action spaces, respectively.

Each episode with length H has an initial state distribution ρ. At time step $i \in [1, H]$ within an episode, the agent observes $s_i \in \mathcal{S}$, selects $a_i \in \mathcal{A}$, receives a noised reward $r_i \sim R^M(s_i, a_i)$ and transitions to a noised new state $s_{i+1} \sim P^M(\cdot|s_i, a_i)$. More specifically, $r(s_i, a_i) = \bar{r}^M(s_i, a_i) + \epsilon_r$ and $s_{i+1} = f^M(s_i, a_i) + \epsilon_f$, where $\epsilon_r \sim \mathcal{N}(0, \sigma_r^2)$, $\epsilon_f \sim \mathcal{N}(0, \sigma_f^2 I_{d_s})$. Variances σ_r^2 and σ_f^2 are fixed to control the noise level. Without loss of generality, we assume the expected reward an agent receives at a single step is bounded $|\bar{r}^M(s, a)| \leq R_{\max}$, $\forall s \in \mathcal{S}, a \in \mathcal{A}$. Let $\mu \colon \mathcal{S} \to \mathcal{A}$ be a deterministic policy. Define the value function for state s at time step i with policy μ as $V^M_{\mu,i}(s) = \mathbb{E}[\Sigma_{j=i}^H [\bar{r}^M(s_j, a_j)|s_i = s]$, where $s_{j+1} \sim P^M(\cdot|s_j, a_j)$ and $a_j = \mu(s_j)$. With the bounded expected reward, we have that $|V(s)| \leq H R_{\max}$, $\forall s$.

We use M^* to indicate the real unknown MDP which includes R^* and P^*, and M^* itself is treated as a random variable. Thus, we can treat the real noiseless reward function $\bar{r}^*$ and transition function f^* as random processes as well. In the posterior sampling algorithm π^{PS}, M^k is a random sample from the posterior distribution of the real unknown MDP M^* in the kth episode, which includes the posterior samples of R^k and P^k , given history prior to the kth episode: $\mathcal{H}_k := \{s_{1,1}, a_{1,1}, r_{1,1}, \cdots, s_{k-1,H}, a_{k-1,H}, r_{k-1,H}\}$, where $s_{k,i}, a_{k,i}$ and $r_{k,i}$ indicate the state, action, and reward at time step i in episode k. We define the the optimal policy under M as $\mu^M(s_i) \in \arg\max_{\mu_2} V^M_{\mu_2,i}(s_i)$. In particular, μ^* indicates the optimal policy under M^* and μ^k represents the optimal policy under M^k. Define future value function: $U^M_i(P) = \mathbb{E}_{s' \sim P(s')}[V^M_{\mu^M,i+1}(s')]$. Let Δ_k denote the regret over the kth episode:

$$\Delta_k = \int \rho(s_1)(V^{M^*}_{\mu^*,1}(s_1) - V^{M^*}_{\mu^k,1}(s_1))ds_1 \tag{1}$$

Then we can express the regret of π^{ps} up to time step T as:

$$Regret(T, \pi^{ps}, M^*) := \Sigma_{k=1}^{\lceil \frac{T}{H} \rceil} \Delta_k, \tag{2}$$

Let $BayesRegret(T, \pi^{ps}, \phi)$ denote the Beyesian regret of π^{ps} as defined in Osband & Van Roy (2017), where ϕ is the prior distribution of M^*:

$$BayesRegret(T, \pi^{ps}, \phi) = \mathbb{E}[Regret(T, \pi^{ps}, M^* \sim \phi)]. \tag{3}$$

2.2. Gaussian Process Assumption

Generally, we consider modeling an unknown target function $g : \mathbb{R}^d \to \mathbb{R}$. We are given a set of noisy samples $y = [y_1 ...., y_T]^T$ at points $X = [x_1, ..., x_T]^T$, $X \subset D$, where D is compact and convex, $y_i = g(x_i) + \epsilon_i$ with $\epsilon_i \sim N(0, \sigma^2)$ i.i.d. Gaussian noise $\forall i \in \{1, \cdots, T\}$.

We model g as a sample from a Gaussian Process $GP(\mu(x), \mathcal{K}(x, x'))$, specified by the mean function $\mu(x) = \mathbb{E}[g(x)]$ and the covariance (kernel) function $\mathcal{K}(x, x') = \mathbb{E}[(g(x) - \mu(x)(g(x') - \mu(x')]$.

Let the prior distribution without any data as $GP(0, \mathcal{K}(x, x'))$. Then the posterior distribution over g given X and y is also a GP with mean $\mu_T(x)$, covariance $\mathcal{K}_T(x, x')$, and variance $\sigma^2_T(x)$ (Rasmussen, 2003):

$$\begin{aligned}
&\mu_T(x) = \mathcal{K}(x, X)(\mathcal{K}(X, X) + \sigma^2 I)^{-1} y, \\
&\mathcal{K}_T(x, x') = \mathcal{K}(x, x') \\
&\quad - \mathcal{K}(X, x)^T (\mathcal{K}(X, X) + \sigma^2 I)^{-1} \mathcal{K}(X, x), \\
&\sigma^2_T(x) = \mathcal{K}_T(x, x),
\end{aligned}$$

where $\mathcal{K}(X, x) = [\mathcal{K}(x_1, x), ..., \mathcal{K}(x_T, x)]^T$, $\mathcal{K}(X, X) = [\mathcal{K}(x_i, x_j)]_{1 \leq i \leq T, 1 \leq j \leq T}$.

We model our reward function $\bar{r}^M$ as a Gaussian Process with noise σ_r^2.

For transition models, we treat each dimension independently: each $f_i(s, a), i = 1, .., d_S$ is modeled independently as above, and with the same noise level σ_f^2 in each dimension. Thus it corresponds to our formulation in the RL setting. Since the posterior covariance matrix is only dependent on the input rather than the target value, the distribution of each $f_i(s, a)$ shares the same covariance matrice and only differs in the mean function.

3. Bayesian Regret Analysis

3.1. Regret Decomposition

In this section, we briefly describe the regret decomposition as presented in Osband & Van Roy (2014) to facilitate further analysis.

The regret in episode k can be rearranged as: $\Delta_k = \int \rho(s_1)(V^{M^*}_{\mu^*,}(s_1) - V^{M^k}_{\mu^k,1}(s_1)) + \tilde{\Delta}_k)ds_1$, where $\tilde{\Delta}_k = V^{M^k}_{\mu^k,1}(s_1) - V^{M^*}_{\mu^k,1}(s_1)$. In PSRL, $V^{M^*}_{\mu^*,} - V^{M^k}_{\mu^k,1}$ is zero in expectation, and thus we only need to bound $\tilde{\Delta}_k$ when deriving the Bayesian regret of PSRL.[3] For clarity, the value

[3] It suffices to derive bounds for any initial state s_1 as the

function $V_{\mu^k,1}^{M^k}$ is simplified to $V_{k,1}^k$ and $V_{\mu^k,1}^{M^*}$ to $V_{k,1}^*$. Let h_i indicate the state-action pair s_i, a_i, and h_i is the state-action pair that the agent encounters in the kth episode while using μ_k as policy in the real MDP M^* [4]. Consider the regret from concentration via the Bellman operator (details of the derivation can be found in (Osband & Van Roy, 2014)):

$$\mathbb{E}[\tilde{\Delta}_k|\mathcal{H}_k] = \mathbb{E}[\tilde{\Delta}_k(r) + \tilde{\Delta}_k(f)|\mathcal{H}_k],$$

where $\tilde{\Delta}_k(r) = \Sigma_{i=1}^H(\bar{r}^k(h_i) - \bar{r}^*(h_i))$, and $\tilde{\Delta}_k(f) = \Sigma_{i=1}^H(U_i^k(P^k(h_i)) - U_i^k(P^*(h_i)))$.

3.2. Dependency on H in the Lipschitz Constant of Future Value Functions

First, we present a lemma on the property derived from noises with a symmetric probability distribution.

Lemma 1 *(Property derived from noises with symmetric probability distribution) Consider two zero-mean noises $\boldsymbol{\epsilon_1}, \boldsymbol{\epsilon_2} \in \mathbb{R}^d$, and the noise in each dimension of $\boldsymbol{\epsilon_1}, \boldsymbol{\epsilon_2}$ is i.i.d. drawn from the same symmetric probability distribution. Let P_1, P_2 be the probability distribution of the random variables $\boldsymbol{\mu_1} + \boldsymbol{\epsilon_1}$ and $\boldsymbol{\mu_2} + \boldsymbol{\epsilon_2}$ respectively, where $\boldsymbol{\mu_1}, \boldsymbol{\mu_2} \in \mathbb{R}^d$.*

Then we have

$$||P_1 - P_2|| \leq C||\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2||_2,$$

where C is a constant that is only dependent on the variance of the noise.

The proof of Lemma 1 is in Appendix. Using this Lemma, we can develop a closed-form upper bound of the Lipschitz constant of the future value function.

Lemma 2 *(The dependency of H in the future value function) We have*

$$\begin{aligned} &U_i^k(P^k(h_i)) - U_i^k(P^*(h_i)) \\ &\leq CHR_{max}||f^k(h_i) - f^*(h_i)||_2, \end{aligned} \tag{4}$$

where C is the constant mentioned in Lemma 1.

Proof. For all i, we have

$$\begin{aligned} &U_i^k(P^k(h_i)) - U_i^k(P^*(h_i)) \\ &\leq \max_s |V_{k,i+1}^k(s)|||P^k(\cdot|h_i) - P^*(\cdot|h_i)|| \\ &\leq HR_{\max}||P^k(\cdot|h_i) - P^*(\cdot|h_i)|| \end{aligned} \tag{5}$$

Recall that $P^k(s'|h_i) = \mathcal{N}(f^k(h_i), \sigma_f^2 I)$ and $P^*(s'|h_i) = \mathcal{N}(f^*(h_i), \sigma_f^2 I)$. By Lemma 1 we have

$$||P^k(\cdot|h_i) - P^*(\cdot|h_i)|| \leq C||f^k(h_i) - f^*(h_i)||_2, \tag{6}$$

And the proof will be complete by combining (4) and (6). □

Remark Lemma 1 is crucial for developing a bound which is polynomially dependent on H in Section 3.4. Here $CHR_{\max}$ serves as the Lipschitz constant in the Lipschitz assumption of the future value function in Osband & Van Roy (2014) . In fact, the Lipschitz constant here is naturally dependent on H: when $i = 1$, the future value function includes the cumulative rewards within H steps. As the distance between two initial states propagates in H steps (and result in differences in the rewards), the resulting difference in the future value function is naturally dependent on H. However, Osband & Van Roy (2014) did not explore such dependency and directly assume the Lipschitz continuity of the future value function their Corollary 1 (presented in Section 1). In contrast, we present the Lipschitz continuity of the future value function as a result of the property of noises and provide its dependency on H.

3.3. Connecting Regret with Posterior Variances

In this section, we show the upper bound of $\Sigma_{k=1}^{[\frac{T}{H}]}\tilde{\Delta}_k(f)$ conditioned on any given history $\mathcal{H}_k$ with high probability.

Lemma 3 *(Upper bound by the sum of posterior variances) With probability at least $1-\delta$,*

$$\begin{aligned} &\Sigma_{k=1}^{[\frac{T}{H}]}[\tilde{\Delta}_k(f)|\mathcal{H}_k] \\ &\leq \Sigma_{k=1}^{[\frac{T}{H}]} 4CH^2R_{max}\sqrt{2d_s\sigma_k^2(h_{kmax})log\frac{4Td_s}{\delta}}. \end{aligned} \tag{7}$$

Proof. Given history $\mathcal{H}_k$, let $\bar{f}^k(h)$ indicate the posterior mean of $f^k(h)$ in episode k, and $\sigma_k^2(h)$ denotes the posterior variance of f^k in each dimension. Note that f^* and f^k share the same variance in each dimension given history $\mathcal{H}_k$, as described in Section 3. Consider all dimensions of the state space, We have that for N sub-Gaussian random variables: $X_1, ..., X_N$ with variance σ^2 (not required to be independent), and for any $t > 0$, $\mathbb{P}(\max_{1\leq i\leq N}|X_i| > t) \leq 2Ne^{-\frac{t^2}{2\sigma^2}}$. (Rigollet & Hütter, 2015). So with probability at least $1-\delta$, for any state-action pair h, $\max_{1\leq i\leq d_s}|f_i^k(h) - \bar{f}_i^k(h)| \leq \sqrt{2\sigma_k^2(h)\log\frac{2d_s}{\delta}}$. Also, we can derive an upper bound for the norm of the state difference $||f^k(h) - \bar{f}^k(h)||_2 \leq \sqrt{d_s}\max_{1\leq i\leq d_s}|f_i^k(h) - \bar{f}_i^k(h)|$, and so does $||f^*(h) - \bar{f}^k(h)||_2$ since f^* and f^k share the same posterior distribution. By the union bound, we have that with probability at least $1-2\delta$, $||f^k(h) - f^*(h)||_2 \leq 2\sqrt{2d_s\sigma_k^2(h)\log\frac{2d_s}{\delta}}$.

Then we look at the sum of the differences over horizon H, without requiring each variable in the sum to be independent:

regret bound will still hold through the integration of the initial distribution $\rho(s_1)$.

[4] where $a_i = \mu_k(s_i), s_{i+1} \sim P^*(\cdot|s_i, a_i)$.

$$\mathbb{P}(\Sigma_{i=1}^{H}||f^k(h_i)-f^*(h_i)||_2 > \Sigma_{i=1}^{H}2\sqrt{2d_s\sigma_k^2(h_i)\log\frac{2d_s}{\delta}})$$
$$\leq \mathbb{P}(\bigcup_{i=1}^{H}\{||f^k(h_i)-f^*(h_i)||_2 > 2\sqrt{2d_s\sigma_k^2(h_i)\log\frac{2d_s}{\delta}}\})$$
$$\leq \Sigma_{i=1}^{H}\mathbb{P}(||f^k(h_i)-f^*(h_i)||_2 > 2\sqrt{2d_s\sigma_k^2(h_i)\log\frac{2d_s}{\delta}}) \tag{8}$$

Thus, we have that with probability $1-\delta$,

$$\begin{aligned}&\Sigma_{i=1}^{H}||f^k(h_i)-f^*(h_i)||_2\\ &\leq \Sigma_{i=1}^{H}2\sqrt{2d_s\sigma_k^2(h_i)\log\frac{4Hd_s}{\delta}}\\ &\leq 2H\sqrt{2d_s\sigma_k^2(h_{\text{kmax}})\log\frac{4Hd_s}{\delta}},\end{aligned} \tag{9}$$

where we define the index: kmax $:= \arg\max_i \sigma_k(h_i), i = 1, ..., H$ in episode k. Here, since the posterior distribution is only updated every H steps, we have to use data points with the max variance in each episode to bound the result. Similarly, using the union bound for $[\frac{T}{H}]$ episodes, we have that with probability at least $1-\delta$,

$$\begin{aligned}&\Sigma_{k=1}^{[\frac{T}{H}]}[\tilde{\Delta}_k(f)|\mathcal{H}_k]\\ &\leq \Sigma_{k=1}^{[\frac{T}{H}]}\Sigma_{i=1}^{H}2CHR_{\text{max}}||f^k(h_i)-f^*(h_i)||_2\\ &\leq \Sigma_{k=1}^{[\frac{T}{H}]}4CH^2R_{\text{max}}\sqrt{2d_s\sigma_k^2(h_{\text{kmax}})\log\frac{4Td_s}{\delta}}.\end{aligned}$$

□

Remark Here we bound $\tilde{\Delta}_k(f)$ using *point-wise* concentration properties of rewards and transitions that applies to any state-action pair. Then we use the union bound on each state-action pair that the agent encounters in every episode. In contrast, Chowdhury & Gopalan (2019) use the uniform concentration on the reward and transition functions, which requires an extra assumption (their regularity assumption of the RKHS norm) compared to our analysis.

3.4. Regret with Linear Kernels

Theorem 1 *In the RL problem formulated in Section 2.1, under the assumption of Section 2.2 with linear kernels* [5]*, we have* $BayesRegret(T, \pi^{ps}, \phi) = \tilde{O}(H^{3/2}d\sqrt{T})$*, where* d *is the dimension of the state-action space,* H *is the episode length, and* T *is the time elapsed.*

[5] GP with linear kernel correspond to Bayesian linear regression $f(x) = w^T x$, where the prior distribution of the weight is $w \sim \mathcal{N}(0, \Sigma_p)$ (Rasmussen, 2003) .

Proof. In each episode k, let $\sigma_k^{'2}(h)$ denote the posterior variance given only a subset of data points $\{h_{1\text{max}}, ..., h_{(\text{k-1})\text{max}}\}$, where each element has the max variance in the corresponding episode. By Eq.(6) in Williams & Vivarelli (2000), we know that the posterior variance reduces as the number of data points grows. Hence $\forall h, \sigma_k^2(h) \leq \sigma_k^{'2}(h)$. By Theorem 5 in Srinivas et al. (2012) which provides a bound on the information gain, and Lemma 2 in Russo & Van Roy (2014) that bounds the sum of variances by the information gain, we have that $\Sigma_{k=1}^{[\frac{T}{H}]}\sigma_k^{'2}(h_{\text{kmax}}) = \mathcal{O}((d_s+d_a)\log[\frac{T}{H}])$ for linear kernels with bounded variances (See Appendix for details).

Thus with probability $1-\delta$, and let $\delta = \frac{1}{T}$,

$$\begin{aligned}&\Sigma_{k=1}^{[\frac{T}{H}]}[\tilde{\Delta}_k(f)|\mathcal{H}_k]\\ &\leq \Sigma_{k=1}^{[\frac{T}{H}]}4CH^2R_{\text{max}}\sqrt{2d_s\sigma_k^2(h_{\text{kmax}})\log\frac{4Td_s}{\delta}}\\ &\leq 8CH^2R_{\text{max}}\sqrt{\Sigma_{k=1}^{[\frac{T}{H}]}\sigma_k^{'2}(h_{\text{kmax}})}\sqrt{[\frac{T}{H}]}\sqrt{d_s\log(2Td_s)}\\ &\leq 8CH^{\frac{3}{2}}R_{\text{max}}\sqrt{T}\sqrt{d_s\log(2Td_s)}\sqrt{\mathcal{O}((d_s+d_a)\log(T))}\\ &= \tilde{\mathcal{O}}((d_s+d_a)H^{\frac{3}{2}}\sqrt{T})\end{aligned} \tag{10}$$

where $\tilde{\mathcal{O}}$ ignores logarithmic factors of T.

Therefore,

$$\begin{aligned}&\mathbb{E}[\Sigma_{k=1}^{[\frac{T}{H}]}\tilde{\Delta}_k(f)|\mathcal{H}_k]\\ &\leq (1-\frac{1}{T})\tilde{\mathcal{O}}((d_s+s_a)H^{\frac{3}{2}}T) + \frac{1}{T}2HR_{\text{max}}[\frac{T}{H}]\\ &= \tilde{\mathcal{O}}(H^{\frac{3}{2}}d\sqrt{T}),\end{aligned} \tag{11}$$

where $2HR_{\text{max}}$ is the upper bound on the difference of value functions, and $d = d_s + d_a$. Following similar derivation, $\mathbb{E}[\Sigma_{k=1}^{[\frac{T}{H}]}\tilde{\Delta}_k(r)|\mathcal{H}_k] = \tilde{\mathcal{O}}(\sqrt{dHT})$ (See Appendix for details). Finally, through the tower property we have $BayesRegret(T, \pi^{ps}, M^*) = \tilde{\mathcal{O}}(H^{\frac{3}{2}}d\sqrt{T})$. □

Remark Here we compare our result with Corollary 2 in (Osband & Van Roy, 2014). Note that we maintain the same assumptions of transitions and rewards as (Osband & Van Roy, 2014). However, in their Corollary 2 which describes the regret for LQR, they directly use the Lipschitz constant of the *underlying value function*, instead of the future value function. The Lipschitz constant of the underlying function in LQR is actually *exponential* in H; as a result, even if the reward is linear, their bound would still be exponential in H (See Appendix for details), while we present a regret bound polynomial in H. So their Corollary 2 is very loose and can be improved by our analysis.

3.5. Nonlinear Extension via Feature Representation

We can slightly modify the previous proof to derive the bound in settings that use feature representations. Consider the mapping: $(s, a) \to s$ in the transition model. We transform the state-action pair (s, a) to $\phi_f(s, a) \in \mathbb{R}^{d_\phi}$ as the input of the transition model, and transform the target s' to $\psi_f(s') \in \mathbb{R}^{d_\psi}$, then this transition model can be established with respect to this feature embedding. We further assume $d_\psi = O(d_\phi)$. Besides, we assume $d_{\phi'} = O(d_\phi)$ in the feature representation $\phi_r(s, a) \in \mathbb{R}^{d_{\phi'}}$, then the reward model can also be established with respect to the feature embedding. In this way, we can handle non-linear rewards and transitions with only linear kernels in GP. Following similar steps in previous analysis will lead to a Bayesian regret of $\tilde{O}(H^{3/2} d_\phi \sqrt{T})$.

Empirically, the linearity of rewards and transitions can be preserved by updating the feature representation. By updating representations of all state-action pairs in the history and the covariance correspondingly, the theoretical extension to nonlinear cases still holds in practice.

Remark The eluder dimension of neural networks in Osband & Van Roy (2014) can blow up to infinity, and the information gain used in Chowdhury & Gopalan (2019) yields exponential order of dimension d if nonlinear kernels are used, such as SE and Matérn kernels. But linear kernel can only model linear functions, thus the representation power is restricted if the polynomial order of d is desired in their result. We first derive results for linear kernels, and increase the representation power by extracting the penultimate layer of neural networks, and thus we can derive a bound linear in the dimension of the penultimate layer, which is generally much less than the exponential order of the input dimension of neural networks.

4. Algorithm Description

In this section, we elaborate our proposed algorithm, MPC-PSRL, as shown in Algorithm 1.

Algorithm 1 MPC-PSRL

Initialize data $\mathcal{D}$ with random actions for one episode
repeat
  Sample a transition model and a cost model at the beginning of each episode
  for $i = 1$ to H steps **do**
    Obtain action using MPC with planning horizon τ: $a_i \in \arg\max_{a_{i:i+\tau}} \sum_{t=i}^{i+\tau} \mathbb{E}[r(s_t, a_t)]$
    $\mathcal{D} = \mathcal{D} \cup \{(s_i, a_i, r_i, s_{i+1})\}$
  end for
  Train cost and dynamics representations ϕ_r and ϕ_f using data in $\mathcal{D}$
  Update $\phi_r(s, a)$, $\phi_f(s, a)$ for all (s, a) collected
  Perform posterior update of w_r and w_f in cost and dynamics models using updated representations $\phi_r(s, a)$, $\phi_f(s, a)$ for all (s, a) collected
until convergence

4.1. Predictive Model

When modeling the rewards and transitions, we use features extracted from the penultimate layer of fitted neural networks, and perform Bayesian linear regression on the feature vectors to update posterior distributions.

Feature representation: we first fit neural networks for transitions and rewards, using the same network architecture as Chua et al. (2018). Let x_i denote the state-action pair (h_i) and y_i denote the target value. Specifically, we use reward r_i as y_i to fit rewards, and we take the difference between two consecutive states $s_{i+1} - s_i$ as y_i to fit transitions. The penultimate layer of fitted neural networks is extracted as the feature representation, denoted as ϕ_f and ϕ_r for transitions and rewards, respectively. Note that in the transition feature embedding, we only use one neural network to extract features of state-action pairs from the penultimate layer to serve as ϕ, and leave the target states without further feature representation (the general setting is discussed in Section 3.5 where feature representations are used for both inputs and outputs), so the dimension of the target in the transition model d_ψ equals to d_s. Thus we have a modified regret bound of $\tilde{O}(H^{3/2}\sqrt{d d_\phi T})$. We do not find the necessity to further extract feature representations in the target space, as it might introduce additional computational overhead. Although higher dimensionality of the hidden layers might imply better representation, we find that only modifying the width of the penultimate layer to be the same order of $d = d_s + s_a$ suffices in our experiments for both reward and transition models. Note that how to optimize the dimension of the penultimate layer for more efficient feature representation deserves further exploration.

Bayesian update and posterior sampling: here we describe the Bayesian update of transition and reward models using extracted features. Recall that Gaussian process with linear kernels is equivalent to Bayesian linear regression. By extracting the penultimate layer as feature representation ϕ, the target value y and the representation $\phi(x)$ could be seen as linearly related: $y = w\top\phi(x) + \epsilon$, where ϵ is a zero-mean Gaussian noise with variance σ^2 (which is σ_f^2 for the transition model and σ_r^2 for the reward model as defined in Section 2.1). We choose the prior distribution of weights w as zero-mean Gaussian with covariance matrix Σ_p, then the posterior distribution of w is also multivariate Gaussian (Rasmussen (2003)):

$$p(w|\mathcal{D}) \sim \mathcal{N}\left(\sigma^{-2} A^{-1} \Phi Y, A^{-1}\right)$$

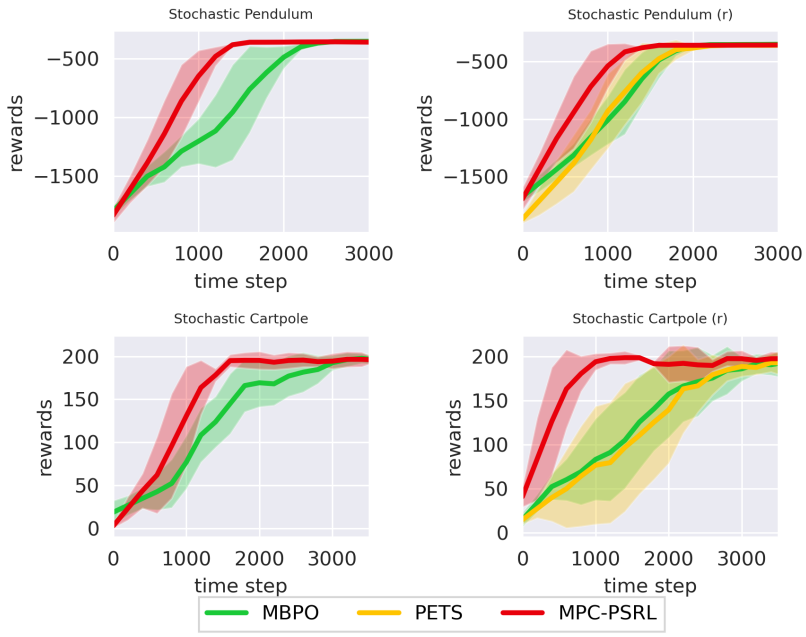

Figure 1. Training curves of MPC-PSRL (shown in red), and other model-based baseline algorithms in stochastic tasks. Solid curves are the mean of five trials, shaded areas correspond to the standard deviation among trials. (r) means with oracle rewards provided.

where $A = \sigma^{-2}\Phi\Phi^\top + \Sigma_p^{-1}$, $\Phi \in \mathcal{R}^{d_\phi \times N}$ is the concatenation of feature representations $\{\phi(x_i)\}_{i=1}^N$, and $Y \in \mathcal{R}^N$ is the concatenation of target values. At the beginning of each episode, we sample w from the posterior distribution to build the model, collect new data during the whole episode, and update the posterior distribution of w at the end of the episode using all the data collected. Here we present the complexity for posterior sampling: The matrix multiplication for covariance matrix A is $O(d_\phi^2 N)$; The inverse of A is $O(d_\phi^3)$.

Besides the posterior distribution of w, the feature representation ϕ is also updated in each episode with new data collected. We adopt a similar dual-update procedure as Riquelme et al. (2018): after representations for rewards and transitions are updated, feature vectors of all state-action pairs collected are re-computed. Then we apply Bayesian update on these feature vectors. See the description of Algorithm 1 for details.

4.2. Planning

During interaction with the environment, we use a MPC controller (Camacho & Alba (2013)) for planning. At each time step i, the controller takes state s_i and an action sequence $a_{i:i+\tau} = \{a_i, a_{i+1}, \cdots, a_{i+\tau}\}$ as the input, where τ is the planning horizon. We use transition and reward models to produce the first action a_i of the sequence of optimized actions $\arg\max_{a_{i:i+\tau}} \sum_{t=i}^{i+\tau} \mathbb{E}[r(s_t, a_t)]$, where the expected return of a series of actions can be approximated using the mean return of several particles propagated with noises of our sampled reward and transition models. To compute the optimal action sequence, we use CEM (Botev et al. (2013)), which samples actions from a distribution closer to previous action samples with high rewards.

5. Experiments

5.1. Baselines

We compare our method with the following state-of-the-art model-based and model-free algorithms on benchmark control tasks.

Model-free: Soft Actor-Critic (SAC) from Haarnoja et al. (2018) is an off-policy deep actor-critic algorithm that utilizes entropy maximization to guide exploration. Deep Deterministic Policy Gradient (DDPG) from Barth-Maron et al. (2018) is an off-policy algorithm that concurrently learns a Q-function and a policy, with a discount factor to guide exploration.

Model-based: Probabilistic Ensembles with Trajectory Sampling (PETS) from Chua et al. (2018) models the dynamics via an ensemble of probabilistic neural networks to capture epistemic uncertainty for exploration, and uses MPC for action selection, with a requirement to have access to oracle rewards for planning. Model-Based Policy

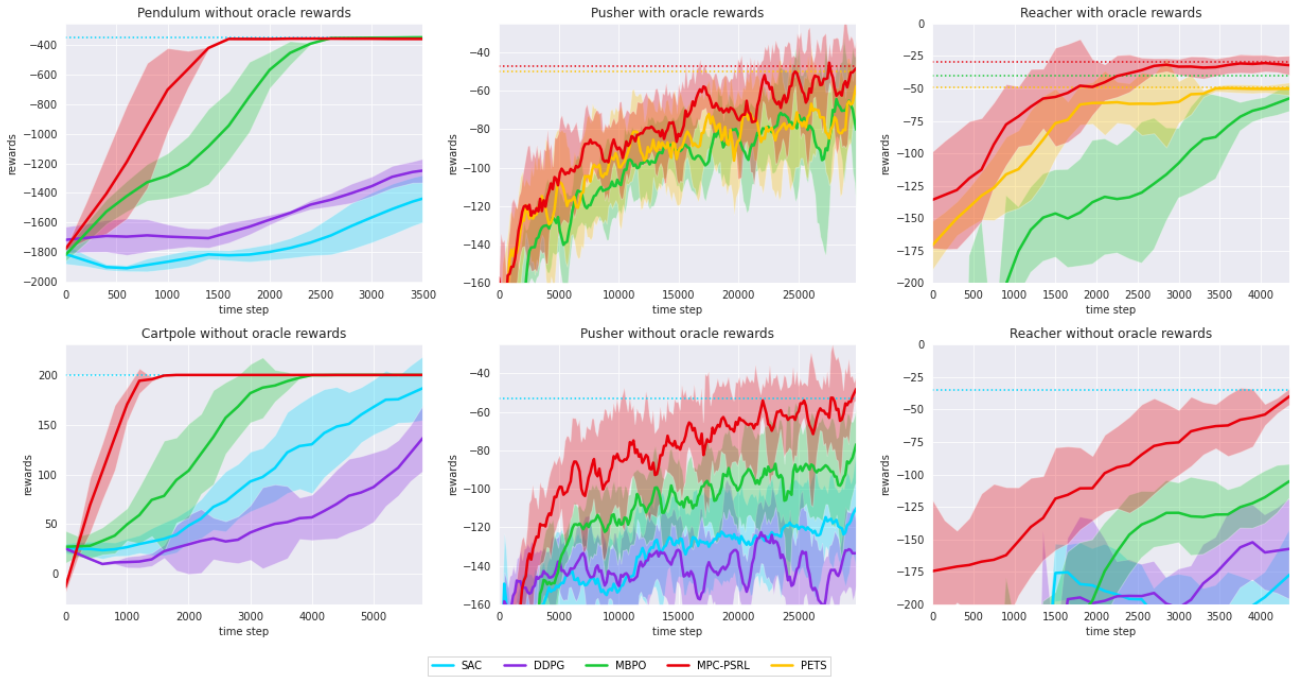

Figure 2. Training curves of MPC-PSRL (shown in red), and other baseline algorithms in different tasks. Solid curves are the mean of five trials, shaded areas correspond to the standard deviation among trials, and the dotted line shows the rewards at convergence.

Optimization (MBPO) from Janner et al. (2019) uses the same bootstrap ensemble techniques as PETS in modeling, but differs from PETS in policy optimization with a large amount of short model-generated rollouts, and can cope with environments with no oracle rewards provided. We do not compare with Gal et al. (2016), which adopts a single Bayesian neural network (BNN) with moment matching, as it is outperformed by PETS that uses an ensemble of BNNs with trajectory sampling. And we don't compare with GP-based trajectory optimization methods with real rewards provided (Deisenroth & Rasmussen (2011), Kamthe & Deisenroth (2018)), which are not only outperformed by PETS, but also computationally expensive and thus are limited to very small state-action spaces.

5.2. Environments and Results

We use environments with various complexity and dimensionality for evaluation:

Low-dimensional stochastic environments: continuous Cartpole ($d_s = 4$, $d_a = 1, H = 200$, with a continuous action space compared to the classic Cartpole, which makes it harder to learn) and Pendulum Swing Up ($d_s = 3$, $d_a = 1, H = 200$, a modified version of Pendulum where we limit the start state to make it harder for exploration). The transitions and rewards are originally deterministic in these environments, so we first modify the physics in Cartpole and Pendulum so that the transitions are stochastic with independent Gaussian noises ($\mathcal{N}(0, 0.01)$). We also use noises in the same form for stochastic rewards. The learning curves of model-based algorithms are shown in Figure 1, which shows our algorithm significantly outperforms model-based baselines in these stochastic environments.

Higher-dimensional environments: 7-DOF Reacher ($d_s = 17, d_a = 7, H = 150$) and 7-DOF pusher ($d_s = 20, d_a = 7, H = 150$) are two more challenging tasks as provided in (Chua et al., 2018), where we conduct experiments both with and without true rewards, to compare with all baseline algorithms mentioned.

The learning curves of all compared algorithms are shown in Figure 2, and the hyperparameters and other experimental settings in our experiments are provided in Appendix. Here we also included results from deterministic Cartpole and Pendulum without oracle rewards. When their trajectories are deterministic, optimization with oracle rewards in these two environments becomes very easy and there is no significant difference in the performances for all model-based algorithms we compare, so we omit those learning curves in Figure 2. However, when we add noise to the trajectory, these environments become harder to learn even when the rewards are provided, and we can observe the difference in the performance of different algorithms as in Figure 1.

When the oracle rewards are provided in Pusher and Reacher, our method outperforms PETS and MBPO: it converges

more quickly with similar performance at convergence in Pusher, while in Reacher, not only does it learn faster but also performs better at convergence. As we use the same planning method (MPC) as PETS, results indicate that our model better captures the uncertainty, which is beneficial to improving sample efficiency. When exploring in environments where both rewards and transition are unknown, our method significantly outperforms previous model-based and model-free methods which do no require oracle rewards. Meanwhile, it matches the performance of SAC at convergence. Convergence results are provided in Appendix.

5.3. Discussion

In our experiment, we have shown that our model-based algorithm outperforms Chua et al. (2018) and Janner et al. (2019) in given environments. Notice that Chua et al. (2018), Janner et al. (2019) have already greatly outperformed state-of-the-art model-free methods in sample efficiency as shown in their papers. Generally, model-based methods enjoy a significant advantage in sample efficiency over model-free methods. So we can safely expect that our algorithm can also outperform other model-free methods with exploration trick like Tang et al. (2017), Kamyar Azizzadenesheli (2018) and Bellemare et al. (2016).

From the experimental results, it can be verified that our algorithm better captures the model uncertainty, and makes better use of uncertainty using posterior sampling. In our methods, by sampling from a Bayesian linear regression on a fitted feature space, and optimizing under the same sampled MDP in the whole episode instead of re-sampling at every step, the performance of our algorithm is guaranteed from a Bayesian view as analyzed in Section 3. While PETS and MBPO use bootstrapped ensembles of models with a limited ensemble size to "simulate" a Bayesian model, in which the convergence of the uncertainty is not guaranteed and is highly dependent on the training of the neural network. However, in our method, there is a limitation of using MPC, which might fail in even higher-dimensional tasks as shown in Janner et al. (2019). Incorporating policy gradient techniques for action-selection might further improve the performance and we leave it for future work.

6. Conclusion

In our paper, we show that the regret for PSRL algorithm with function approximation can be polynomial in d, H with the assumption that true rewards and transitions (with or without feature embedding) can be modeled by GP with linear kernels. While matching the order of best-known bounds in UCB-based works, PSRL also enjoys computational tractability compared to UCB methods. Moreover, we propose MPC-PSRL in continuous environments, and experiments show that our algorithm exceeds existing model-based and model-free methods with more efficient exploration.

References

Ayoub, A., Jia, Z., Szepesvari, C., Wang, M., and Yang, L. Model-based reinforcement learning with value-targeted regression. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 463–474. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/ayoub20a.html.

Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. *arXiv preprint arXiv:1703.05449*, 2017.

Azizzadenesheli, K., Brunskill, E., and Anandkumar, A. Efficient exploration through bayesian deep q-networks. In *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–9. IEEE, 2018.

Barth-Maron, G., Hoffman, M. W., Budden, D., Dabney, W., Horgan, D., Tb, D., Muldal, A., Heess, N., and Lillicrap, T. Distributed distributional deterministic policy gradients. *arXiv preprint arXiv:1804.08617*, 2018.

Bellemare, M. G., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. *arXiv preprint arXiv:1606.01868*, 2016.

Botev, Z. I., Kroese, D. P., Rubinstein, R. Y., and L'Ecuyer, P. The cross-entropy method for optimization. In *Handbook of statistics*, volume 31, pp. 35–59. Elsevier, 2013.

Camacho, E. F. and Alba, C. B. *Model predictive control*. Springer Science & Business Media, 2013.

Chowdhury, S. R. and Gopalan, A. Online learning in kernelized markov decision processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3197–3205, 2019.

Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, pp. 4754–4765, 2018.

Deisenroth, M. and Rasmussen, C. E. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pp. 465–472, 2011.

Gal, Y., McAllister, R., and Rasmussen, C. E. Improving pilco with bayesian neural network dynamics models. In *Data-Efficient Machine Learning workshop, ICML*, volume 4, pp. 34, 2016.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

Janner, M., Fu, J., Zhang, M., and Levine, S. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, pp. 12498–12509, 2019.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pp. 4863–4873, 2018.

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

Kamthe, S. and Deisenroth, M. Data-efficient reinforcement learning with probabilistic model predictive control. In *International Conference on Artificial Intelligence and Statistics*, pp. 1701–1710. PMLR, 2018.

Kamyar Azizzadenesheli, Emma Brunskill, A. A. Efficient exploration through bayesian deep q-networks, 2018.

Osband, I. and Van Roy, B. Model-based reinforcement learning and the eluder dimension. In *Advances in Neural Information Processing Systems*, pp. 1466–1474, 2014.

Osband, I. and Van Roy, B. Why is posterior sampling better than optimism for reinforcement learning? In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, pp. 2701–2710, International Convention Centre, Sydney, Australia, 2017. PMLR.

Osband, I., Benjamin, V. R., and Daniel, R. (More) efficient reinforcement learning via posterior sampling. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pp. 3003–3011, USA, 2013. Curran Associates Inc.

Osband, I., Van Roy, B., Russo, D. J., and Wen, Z. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019.

Rasmussen, C. E. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pp. 63–71. Springer, 2003.

Rigollet, P. and Hütter, J.-C. High dimensional statistics. *Lecture notes for course 18S997*, 2015.

Riquelme, C., Tucker, G., and Snoek, J. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127*, 2018.

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

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

Tang, H., Houthooft, R., Foote, D., Stooke, A., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. # exploration: A study of count-based exploration for deep reinforcement learning. In *31st Conference on Neural Information Processing Systems (NIPS)*, volume 30, pp. 1–18, 2017.

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

Williams, C. K. and Vivarelli, F. Upper and lower bounds on the learning curve for gaussian processes. *Machine Learning*, 40(1):77–102, 2000.

Yang, L. F. and Wang, M. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. *arXiv preprint arXiv:1905.10389*, 2019.

Zanette, A., Lazaric, A., Kochenderfer, M., and Brunskill, E. Learning near optimal policies with low inherent bellman error. *arXiv preprint arXiv:2003.00153*, 2020.