Posterior Coreset Construction with Kernelized Stein Discrepancy for Model-Based Reinforcement Learning

Souradip Chakraborty[1], Amrit Singh Bedi[1], Pratap Tokekar[1], Alec Koppel[2], Brian Sadler[3], Furong Huang[1], Dinesh Manocha[1]

[1]University of Maryland, College Park, USA,
[2]JP Morgan AI Research, NY, USA,
[3]DEVCOM Army Research Laboratory, Adeplhi, USA
{schakra3,amritbd,tokekar, furongh, dmanocha}@umd.edu, alec.koppel@jpmchase.com, brian.m.sadler6.civ@army.mil

Abstract

Model-based approaches to reinforcement learning (MBRL) exhibit favorable performance in practice, but their theoretical guarantees in large spaces are mostly restricted to the setting when transition model is Gaussian or Lipschitz, and demands a posterior estimate whose representational complexity grows unbounded with time. In this work, we develop a novel MBRL method (i) which relaxes the assumptions on the target transition model to belong to a generic family of mixture models; (ii) is applicable to large-scale training by incorporating a compression step such that the posterior estimate consists of a Bayesian coreset of only statistically significant past state-action pairs; and (iii) exhibits a sublinear Bayesian regret. To achieve these results, we adopt an approach based upon Stein's method, which, under a smoothness condition on the constructed posterior and target, allows distributional distance to be evaluated in closed form as the kernelized Stein discrepancy (KSD). The aforementioned compression step is then computed in terms of greedily retaining only those samples which are more than a certain KSD away from the previous model estimate. Experimentally, we observe that this approach is competitive with several state-of-the-art RL methodologies, and can achieve up-to 50 percent reduction in wall clock time in some continuous control environments.

1 Introduction

Reinforcement learning, mathematically characterized by a Markov Decision Process (MDP) (Puterman 2014), has gained traction for addressing sequential decision-making problems with long-term incentives and uncertainty in state transitions (Sutton and Barto 1998). A persistent debate exists as to whether model-free (approximate dynamic programming (Sutton 1988) or policy search (Williams 1992)), or model-based (model-predictive control, MPC (Garcia, Prett, and Morari 1989; Kamthe and Deisenroth 2018)) methods, are superior in principle and practice (Wang et al. 2019). A major impediment to settling this debate is that performance certificates are presented in disparate ways, such as probably approximate correct (PAC) bounds (Strehl, Li, and Littman 2009; Dann, Lattimore, and Brunskill 2017), frequentist regret (Jin et al. 2018, 2020), Bayesian regret (Agrawal and Jia 2017; Xu and Tewari 2020; O'Donoghue 2021), and convergence in various distributional metrics (Borkar and Meyn 2002; Amortila et al. 2020; Kose and Ruszczynski 2021). In this work, we restrict focus to regret, as it imposes the fewest requirements on access to a generative model underlying state transitions.

In evaluating the landscape of frequentist regret bounds for RL methods, both model-based and model-free approaches have been extensively studied (Jin et al. 2018). Value-based methods in episodic settings have been shown to achieve regret bounds $\tilde{\mathcal{O}}(d^p H^q \sqrt{T})$ (Yang and Wang 2020) (with $p = 1$, $q = 2$), where H is the episode length, and d is the aggregate dimension of the state and action space. This result has been improved to $p = q = 3/2$ in (Jin et al. 2020), and further to $p = 3/2$ and $q = 1$ in (Zanette et al. 2020). Recently, model-based methods have gained traction for improving upon the best known regret model-free methods with $p = 1$ and $q = 1/3$ (Ayoub et al. 2020). A separate line of works seek to improve the dependence on T to be logarithmic through instance dependence (Jaksch, Ortner, and Auer 2010; Zhou, Gu, and Szepesvari 2021). These results typically impose that the underlying MDP has a transition model that is linearly factorizable, and exhibit regret depends on the input dimension d. This condition can be relaxed through introduction of a nonlinear feature map, whose appropriate selection is highly nontrivial and lead to large gaps between regret and practice (Nguyen et al. 2013), or meta-procedures (Lee et al. 2021). Aside from the feature selection issue, these approaches require evaluation of confidence sets which is computationally costly and lead to statistical inefficiencies when approximated (Osband and Van Roy 2017a).

Thus, we prioritize *Bayesian* approaches to RL (Ghavamzadeh et al. 2015; Kamthe and Deisenroth 2018) popular in robotics (Deisenroth, Fox, and Rasmussen 2013). While many heuristics exist, performance guarantees take the form of Bayesian regret (Osband, Russo, and Van Roy 2013), and predominantly build upon posterior (Thompson) sampling (Thompson 1933). In particular, beyond the tabular setting, (Osband and Van Roy 2014) establishes a $\tilde{O}(\sigma_R\sqrt{d_K(R)d_E(R)T}+\mathbb{E}[L^*]\sigma_p\sqrt{d_K(P)d_E(P)})$ Bayesian regret for posterior sampling RL (PSRL) combined with greedy action selections with respect to the estimated value. Here L^* is a global Lipschitz constant for the future value function, d_K and d_E are Kolmogorov and eluder dimensions, and R and P refers to function classes of rewards and transitions. The connection between H and L is left im-

plicit; however, (Fan and Ming 2021)[Sec. 3.2] shows that L can depend exponentially on H. Similar drawbacks manifest in the Lipschitz parameter of the Bayesian regret bound of (Chowdhury and Gopalan 2019), which extends the former result to continuous spaces through kernelized feature maps. However, recently an augmentation of PSRL is proposed which employs feature embedding with Gaussian (symmetric distribution) dynamics to alleviate this issue (Fan and Ming 2021), yielding the Bayesian regret of $\tilde{\mathcal{O}}(H^{\frac{3}{2}}d\sqrt{T})$ that is polynomial in H and has no dependence on Lipschitz constant L. These results are still restricted in the sense that it requires (i) the transition model target to be Gaussian, (ii) its representational complexity to grow unsustainably large with time. Therefore, in this work we as the following question:

Can we achieve a trade-off between the Bayesian regret and the posterior representational complexity (aka coreset size) without oracle access to a feature map at the outset of training, in possibly continuous state-action spaces?

We provide an affirmative answer by honing in on the total variation norm used to quantify the posterior estimation error that appears in the regret analysis of (Fan and Ming 2021), and identify that it can be sharpened by instead employing an integral probability metric (IPM). Specifically, by shifting to an IPM, and then imposing structural assumptions on the target, that is, restricting it to a class of smooth densities, we can employ *Stein's identity* (Stein et al. 1956; Kattumannil 2009) to evaluate the distributional distance in closed form using the kernelized Stein discrepancy (KSD) (Gorham and Mackey 2015; Liu, Lee, and Jordan 2016). This restriction is common in Markov Chain Monte Carlo (MCMC) (Andrieu et al. 2003), and imposes that the target, for instance, belongs to a family of mixture models.

This modification in the metric of convergence alone leads to improved regret because we no longer require the assumption that the posterior is Gaussian (Fan and Ming 2021). However, our goal is to translate the scalability of PSRL from tabular settings to continuous spaces which requires addressing the parameterization complexity of the posterior estimate, which grows linearly unbounded (Fan and Ming 2021) . With the power to evaluate KSD in closed form, then, we sequentially remove those state-action pairs that contribute least (decided by a compression budget ϵ) in KSD after each episode (which is completely novel in the RL setting) from the posterior representation according to (Hawkins, Koppel, and Zhang 2022). Therefore, the posterior estimate only retains statistically significant past samples from the trajectories, i.e., it is defined by a Bayesian coreset of the trajectory data (Campbell and Broderick 2018, 2019). The budget parameter ϵ then is calibrated in terms of a rate determining factor α to yield both sublinear Bayesian regret and sublinear representational complexity of the learned posterior – see Table 1. The resultant procedure we call Kernelized Stein Discrepancy-based Posterior Sampling for RL (`KSRL`). Our main contributions are, then, to:

- ▷ introduce Stein's method in MBRL for the first time, and use it to develop a novel transition model estimate based upon it, which operates in tandem with a KSD compression step to remove statistically insignificant past state-action pairs, which we abbreviate as `KSRL`;
- ▷ establish Bayesian regret bounds of the resultant procedure that is sublinear in the number of episodes experienced, without any prior access to a feature map, alleviating difficult feature selection drawbacks of prior art. Notably, these results relax Gaussian and Lipschitz assumptions of prior related results;
- ▷ mathematically establish a tunable trade-off between Bayesian regret and posterior's parameterization complexity (or dictionary size) via introducing parameter $\alpha \in (0, 1]$ for the first time in this work;
- ▷ experimentally demonstrate that `KSRL` achieves favorable tradeoffs between sample and representational complexity relative to several strong benchmarks.

2 Problem Formulation

We consider the problem of modelling an episodic finite-horizon Markov Decision Process (MDP) where the true unknown MDP is defined as $M^* := \{\mathcal{S}, \mathcal{A}, R^*, P^*, H, R_{\max}, \rho\}$, where $\mathcal{S} \subset \mathbb{R}^{d_s}$ and $\mathcal{A} \subset \mathbb{R}^{d_a}$ denote continuous state and action spaces, respectively. Here, P^* represents the true underlying generating process for the state action transitions and R^* is the true rewards distribution. After every episode of length H, the state will reset according to the initial state distribution ρ. At time step $i \in [1, H]$ within an episode, the agent observe $s_i \in \mathcal{S}$, selects $a_i \in \mathcal{A}$ according to a policy μ, receives a reward $r_i \sim R^*(s_i, a_i)$ and transitions to a new state $s_{i+1} \sim P^*(\cdot|s_i, a_i)$. We consider M^* itself as a random process, as is the often the case in Bayesian Reinforcement Learning, which helps us to distinguish between the true and fitted transition/reward model.

Next, we define policy μ as a mapping from state $s \in \mathcal{S}$ to action $a \in \mathcal{A}$ over an episode of length H. For a given MDP M, the value for time step i is the reward accumulation during the episode:

$$V^M_{\mu,i}(s) = \mathbb{E}[\Sigma^H_{j=i}[\bar{r}^M(s_j, \mu(s_j, j))|s_i = s], \qquad (1)$$

where actions are under policy $\mu(s_j, j)$ (j denotes the timestep within the episode) and $\bar{r}^M(s, a) = \mathbb{E}_{r\sim R^M(s,a)}[r]$. Without loss of generality, we assume the expected reward an agent receives at a single step is bounded $|\bar{r}^M(s, a)| \leq R_{\max}$, $\forall s \in \mathcal{S}, a \in \mathcal{A}$. This further implies that $|V(s)| \leq HR_{\max}$, $\forall s$. For a given MDP M, the optimal policy μ^M is defined as

$$\mu^M = \arg\max_{\mu} V^M_{\mu,i}(s), \qquad (2)$$

for all s and $i = 1, \ldots, H$. Next, we also define future value function $U^M_i(P)$ to be the expected value of the value function over all initializations and trajectories

$$U^M_i(P) = \mathbb{E}_{s'\sim P(s'|s,a),a=\mu^M(s,i)}[V^M_{\mu^M,i}(s')|s_i = s], \quad (3)$$

where P is the transition distribution under MDP M. According to these definitions, one would like to find the optimal policy (2) for the true model $M = M^*$.

Next, we review the PSRL algorithm, which is an adaption of Thompson sampling to RL (Osband and Van Roy 2014) (see Algorithm in Appendix (Chakraborty et al. 2022)). In PSRL, we start with a prior distribution over

Setting	Refs	Bayes Regret	Coreset Size
Tabular	PSRL (Osband, Benjamin, and Daniel 2013)	$\tilde{\mathcal{O}}(HS\sqrt{AT})$	$\Omega(T)$
Tabular	PSRL2 (Osband and Van Roy 2017b)	$\tilde{\mathcal{O}}(H\sqrt{SAT})$	$\Omega(T)$
Tabular	TSDE (Ouyang, Gagrani, and Jain 2017)	$\tilde{\mathcal{O}}(HS\sqrt{AT})$	$\Omega(T)$
Tabular	General PSRL (Agrawal and Jia 2017)	$\tilde{\mathcal{O}}(DS\sqrt{AT})$	$\Omega(T)$
Tabular	DS-PSRL (Theocharous et al. 2018)	$\tilde{\mathcal{O}}(CH\sqrt{C'T})$	$\Omega(T)$
Tabular	PSRL3 (Osband and Van Roy 2014)	$\tilde{\mathcal{O}}(\sqrt{d_k d_E T})$	$\Omega(T)$
Continuous/Gaussian	MPC-PSRL (Fan and Ming 2021)	$\tilde{\mathcal{O}}(H^{\frac{3}{2}}d\sqrt{T})$	$\Omega(T)$
Continuous/ Smooth	KSRL (**This work**)	$\tilde{\mathcal{O}}\left(dH^{1+(\alpha/2)}T^{1-(\alpha/2)}\right)$	$\Omega(\sqrt{T^{1+\alpha}})$

Table 1: A comparison of Bayes regret (cf. (5)) and Bayesian Coreset (the number of stored data points in dictionary $\mathcal{D}_k$ to represent posterior at k). We introduce KSD-based compression to model-based RL (KSRL), with tuning parameter α to obtain sublinear Bayesian regret *and* coreset size for any $\alpha \in (0, 1]$. For $\alpha = 1$, we recover the state of the art results of MPC-PSRL ($\tilde{\mathcal{O}}(dH^{3/2}\sqrt{T})$). But our results hold for general transitions, which are smooth and not restricted to Gaussian assumption.

MDP given by ϕ. Then at each episode, we take sample M^k from the posterior given by $\phi(\cdot|\mathcal{D}_k)$, where $\mathcal{D}_k := \{s_{1,1}, a_{1,1}, r_{1,1}, \cdots, s_{k-1,H}, a_{k-1,H}, r_{k-1,H}\}$ is a data set containing past trajectory data, i.e., state-action-reward triples, which we call a *dictionary*. That is, where $s_{k,i}, a_{k,i}$ and $r_{k,i}$ indicate the state, action, and reward at time step zi in episode k. Then, we evaluate the optimal policy $\mu^k := \mu^{M^k}$ via (2). Thereafter, information from the latest episode is appended to the dictionary as $\mathcal{D}_{k+1} = \mathcal{D}_k \cup \{s_{k,1}, a_{k,1}, r_{k,1}, \cdots, s_{k,H}, a_{k,H}, r_{k,H}\}$.

Bayes Regret and Limitations of PSRL: To formalize the notion of performance in the model-based RL setting, we define Bayes regret for episode k as (Osband and Van Roy 2014; Fan and Ming 2021)

$$\Delta_k = \int \rho(s_1)(V^{M^*}_{\mu^*,1}(s_1) - V^{M^*}_{\mu^k,1}(s_1))ds_1, \qquad (4)$$

where $\rho(s_1)$ is the initial state distribution, and μ^k is the optimal policy for M^k sampled from posterior at episode k. The total regret for all the episodes is thus given by

$$Regret_T := \Sigma_{k=1}^{\lceil \frac{T}{H} \rceil} \Delta_k, \quad \text{and} \qquad (5)$$

$$BayesRegret_T := \mathbb{E}[Regret_T \mid M^* \sim \phi]. \qquad (6)$$

The Bayes regret of PSRL (cf. Algorithm **??**) is established to be $\tilde{\mathcal{O}}(\sqrt{d_K(R)d_E(R)T} + \mathbb{E}[L^*]\sqrt{d_K(P)d_E(P)})$ where d_K and d_E are Kolmogorov and Eluder dimensions, R and P refer to function classes of rewards and transitions, and L^* is a global Lipschitz constant for the future value function. Although it is mentioned that system noise smooths the future value functions in (Osband and Van Roy 2014), an explicit connection between H and L is absent, which leads to an exponential dependence on horizon length H in the regret (Osband and Van Roy 2014, Corollary 2) for LQR. This dependence has been improved to a polynomial rate in H: $\tilde{\mathcal{O}}(H^{\frac{3}{2}}d\sqrt{T})$ in (Fan and Ming 2021), which is additionally linear in d and sublinear in T.

A crucial assumption in deriving the best known regret bound for PSRL with continuous state action space is of target distribution belonging to Gaussian/symmetric class, which is often violated. For instance, if we consider a variant of inverted pendulum with an articulated arm, the transition model has at least as many modes as there are minor-joints in the arm. Another major challenge is related to *posterior's parameterization complexity* $M(T) := |\mathcal{D}_k|$, which we subsequently call the dictionary size (step 10 in Algorithm **??**) which is used to parameterize the posterior distribution. We note that $M(T) = \Omega(T)$ for the PSRL (Osband and Van Roy 2014) and MPC-PSRL (Fan and Ming 2021) algorithms which are state of the art approaches.

To alleviate the Gaussian restriction, we consider an alternative metric of evaluating the distributional estimation error, namely, the kerneralized Stein discrepancy (KSD). Additionally, that KSD is easy to evaluate under appropriate conditions on the target distribution, i.e., the target distribution is smooth, one can compare its relative quality as a function of which data is included in the posterior. Doing so allows us to judiciously choose which points to retain during the learning process in order to ensure small Bayesian regret. To our knowledge, this work is the first to deal with the compression of posterior estimate in model-based RL settings along with provable guarantees. These aspects are derived in detail in the following section.

3 Proposed Approach

3.1 Posterior Coreset Construction via KSD

The core of our algorithmic development is based upon the computation of Stein kernels and KSD to evaluate the merit of a given transition model estimate, and determine which past samples to retain. Doing so is based upon the consideration of integral probability metrics (IPM) rather than total variation (TV) distance. This allows us to employ Stein's identity, which under a hypothesis that the score function of the target (which is gradient of log likelihood of target) is computable. This approach is well-known to yield methods to improve the sample complexity of Markov Chain Monte Carlo (MCMC) methods (Chen et al. 2019). That this turns out to be the case in model-based RL as well is a testament to its power (Stein et al. 1956; Ross 2011). This method to approximate a target density P consists of defining an IPM (Sriperumbudur et al. 2012) based on a set $\mathcal{G}$ consisting of test functions on

Algorithm 1: **K**ernelized **S**tein Discrepancy-based Posterior Sampling for **RL** (KSRL)

1: **Input** : Episode length H, Total timesteps T, Dictionary $\mathcal{D}$, prior distribution $\phi = \{\mathcal{P}, \mathcal{R}\}$ for true MDP M^*, planning horizon τ for MPC Controller, thinning budget $\{\epsilon_k\}_{k=1}^{K}$
2: **Initialization** : Initialize dictionary $\mathcal{D}_1$ at with random actions from the controller as $\mathcal{D}_1 := \{s_{1,1}, a_{1,1}, r_{1,1}, \cdots, s_{1,H}, a_{1,H}, r_{1,H}\}$, posterior $\phi_{\mathcal{D}_1} = \{\mathcal{P}_{\mathcal{D}_1}, \mathcal{R}_{\mathcal{D}_1}\}$
3: **for** Episodes $k = 1$ to K **do**
4: **Sample** a transition $P^k \sim \mathcal{P}_{\mathcal{D}_k}$ and reward model $r^k \sim \mathcal{R}_{\mathcal{D}_k}$ and initialize empty $\mathcal{C} = []$
5: **for** timesteps $i = 1$ to H **do**
6: **Evaluate** optimal action sequence $a^*_{k,i:k,i+\tau} = \arg\max_{a_{k,i:k,i+\tau}} \sum_{t=i}^{i+\tau} \mathbb{E}[r(s_{k,t}, a_{k,t})]$
7: **Execute** $a^*_{k,i}$ from the optimal sequence $a^*_{k,i:k,i+\tau}$
8: **Update** $\mathcal{C} \leftarrow \mathcal{C} \cup \{(s_{k,i}, a_{k,i}, s_{k,i+1}, r_{k,i})\}$
9: **end for**
10: **Update** dictionary $\widetilde{\mathcal{D}}_{k+1} \leftarrow \mathcal{D}_k \cup \mathcal{C}$
11: **Perform** thinning operation (cf. Algorithm 2)

$$(\phi_{\mathcal{D}_{k+1}}, \mathcal{D}_{k+1}) = \text{KSD-Thinning}(\phi_{\widetilde{\mathcal{D}}_{k+1}}, \widetilde{\mathcal{D}}_{k+1}, \epsilon_k)$$

12: **end for**

$\mathcal{X} \subset \mathbb{R}^{2d_s+d_a}$, and is defined as:

$$D_{\mathcal{G},P}(\{x_i\}_{i=1}^n) \;:=\; \sup_{g\in\mathcal{G}} \left|\tfrac{1}{n}\sum_{i=1}^n g(x_i) - \int_{\mathcal{X}} g \mathrm{d}P\right|, \tag{7}$$

where n denotes the number of samples. We can recover many well-known probability metrics, such as total variation distance and the Wasserstein distance, through different choices of $\mathcal{G}$. Although IPMs efficiently quantify the discrepancy between an estimate and target, (7) requires P to evaluate the integral, which may be unavailable.

To alleviate this issue, Stein's method restricts the class of test functions g to be those such that $\mathbb{E}_P[g(z)] = 0$. In this case, IPM (7) only depends on the Dirac-delta measure (δ) from the stream of samples, removing dependency on the exact integration in terms of P. Then, we restrict the class of densities to those that satisfy this identity, which are simply those for which we can evaluate the score function of the target distribution. Surprisingly, in practice, we need not evaluate the score function of the true posterior, but instead only the score function of estimated posterior, for this approach to operate (Liu, Lee, and Jordan 2016, proposition). Then, by supposing that the true density is smooth, the IPM can be evaluated in closed form through the kernelized Stein discrepancy (KSD) as a function of the *Stein kernel* (Liu, Lee, and Jordan 2016), to be defined next.

To be more precise, we define each particle h_i as a tuple of the form $h_i := (s_i, a_i, s'_i) \in \mathbb{R}^d$ (with $d = 2d_s + d_a$) and the state-action tuple $\hat{h}_i := (s_i, a_i) \in R^{d_s+d_a}$. We would like a transition model estimate over past samples, and some appropriately defined test function g. This test function turns out to be a Stein kernel κ_0, which is explicitly defined in terms of base kernel κ, e.g., a Gaussian, or inverse multi-quadratic associated with the RKHS that imposes smoothness properties on the estimated transition model (Liu, Lee, and Jordan 2016). The explicit form of the Stein kernel κ_0 is given as follows

$$\begin{aligned}\kappa_0(h_i,h_j) =& s_P(h_i)^T s_P(h_j)\kappa(h_i,h_i) + s_P(h_j)^T \nabla_{h_i}\kappa(h_i,h_j) \\ &+ s_P(h_i)^T \nabla_{h_j}\kappa(h_i,h_j) + \sum_{l=1}^{d} \frac{\partial^2 \kappa(h_i,h_j)}{\partial h_j(l) \partial h_j(l)} \;, \end{aligned} \tag{8}$$

where $h_j(l)$ is the l^{th} element of the $d-$dimensional vector and $s_P(h_i) := \nabla_{h_i} \log P(h_i)$ is the score function of true transition model P. Observe that this is only evaluated over *samples*, and hence the score function of the true transition model is unknown. The key technical upshot of employing Stein's method is that we can now evaluate the integral probability metric of posterior $\phi_{\mathcal{D}_k} := \phi(\cdot|\mathcal{D}_k)$ parameterized by dictionary $\mathcal{D}_k$ through the KSD, which is efficiently computable:

$$\text{KSD}(\phi_{\mathcal{D}_k}) := \sqrt{\frac{1}{|\mathcal{D}_k|^2} \sum_{h_i,h_j} \kappa_0(h_i,h_j)} \;. \tag{9}$$

Therefore, we no longer require access to the true unknown target transition model of the MDP in order to determine the quality of a given posterior estimate of unknown target P. This is a major merit of utilizing Stein's method in MBRL, and allows us to improve the regret of model-based RL methods based on posterior sampling.

This the previous point is distinct from the computational burden of storing dictionary $\mathcal{D}_k$ that parameterizes $\phi_{\mathcal{D}_k}$ and evaluating the optimal value function according to the current belief model (2). After this novel change (see Lemma 4.1 in Sec. 4), we can utilize the machinery of KSD to derive the regret rate for the proposed algorithm in this work (cf. Algorithm 1) in lieu of concentration inequalities, as in (Fan and Ming 2021). We shift to the computational storage requirements of the posterior in continuous space next.

KSD Thinning: We develop a principled way to avoid the requirement that the dictionary $\mathcal{D}_k$ retains all information from past episodes, and is instead parameterized by a coreset of statistically significant samples. More specifically, observe that in step 10 and 11 in PSRL (see Algorithm in Appendix (Chakraborty et al. 2022)), the dictionary at each episode k retains H additional points, i.e., $|\mathcal{D}_{k+1}| = |\mathcal{D}_k| + H$. Hence, as the number of episodes experienced becomes large, the posterior representational complexity grows linearly and unbounded with episode index k. On top of that, the posterior update in step 11 in PSRL (cf. Algorithm **??**) is also parameterized by data collected in $\mathcal{D}_{k+1}$. For instance, if the prior is assumed to be Gaussian, the posterior update of step 11 in PSRL (cf. Algorithm **??**) boils down to GP posterior parameter evaluations which is of complexity $\mathcal{O}(|\mathcal{D}_k|^3)$ for each k (Rasmussen 2004).

To deal with this bottleneck, we propose to sequentially remove those particles from $\mathcal{D}_{k+1}$ that contribute least in terms of KSD. This may be interpreted as projecting posterior estimates onto "subspaces" spanned by only statistically representative past state-action-state triples. This notion of

Algorithm 2: Posterior Coreset with **KSD Thinning** for **R**einforcement **L**earning (KSD-Thinning)

Input: $(q_{\mathcal{W}}, \mathcal{W}, \epsilon)$
Require: Target score function
Compute the reference KSD as $\alpha := \text{KSD}(q_{\mathcal{W}})$ via (9)
while $\text{KSD}(q_{\mathcal{W}})^2 < \alpha^2 + \epsilon$ **do**
  Compute the least influential point x_j as the minimial $h_i \in \widetilde{\mathcal{D}}_{k+1}$ (10)
  if $\text{KSD}(q_{\mathcal{W}\setminus\{x_j\}})^2 < \alpha^2 + \epsilon$ **then**
    Remove the least influential point, set $\mathcal{W} = \mathcal{W} \setminus \{x_j\}$
  else
    Break loop
  end if
end while
Output thinned dictionary $\mathcal{W}$ satisfying $\text{KSD}(q_{\mathcal{W}})^2 < \alpha^2 + \epsilon$

representing a nonparametric posterior using only most representative samples has been shown to exhibit theoretical and numerical advantages in probability density estimation (Campbell and Broderick 2018, 2019), Gaussian Processes (Koppel, Pradhan, and Rajawat 2021), and Monte Carlo methods (Elvira, Míguez, and Djurić 2016). Here we introduce it for the first time in model-based RL, which allows us to control the growth of the posterior complexity, which in turn permits us to obtain computationally efficient updates.

To be more specific, suppose we are at episode k with dictionary $\mathcal{D}_k$ associated with posterior $\phi_{\mathcal{D}_k}$, and we denote the dictionary after update as $\widetilde{\mathcal{D}}_{k+1} = \mathcal{D}_k + H$ and corresponding posterior as $\phi_{\widetilde{\mathcal{D}}_{k+1}}$. For a given dictionary $\mathcal{D}_k$, we can calculate the KSD of posterior $\phi_{\mathcal{D}_k}$ to target via (9). We note that (9) goes to zero as $k \to \infty$ due to the posterior consistency conditions (Gorham and Mackey 2015). At each episode k, after performing the dictionary update to obtain $\widetilde{\mathcal{D}}_{k+1}$ (step 10 in Algorithm 1), we propose to thin dictionary $\widetilde{\mathcal{D}}_{k+1}$ such that

$$\text{KSD}(\phi_{\mathcal{D}_{k+1}})^2 < \text{KSD}(\phi_{\widetilde{\mathcal{D}}_{k+1}})^2 + \epsilon_{k+1}, \tag{10}$$

where $\mathcal{D}_{k+1}$ is the dictionary following thinning and $\epsilon_k > 0$ is a scalar parameter we call the thinning budget proposed. This means the posterior defined by compressed dictionary $\phi_{\widetilde{\mathcal{D}}_{k+1}}$ is at most ϵ_{k+1} in KSD from its uncompressed counterpart. See (Hawkins, Koppel, and Zhang 2022) for related development of this compression routine in the context of MCMC. We will see in the regret analysis section (cf. Sec. 4) how ϵ_k permits us to trade off regret and dictionary size in practice. (10) may be succinctly stated as

$$(\phi_{\mathcal{D}_{k+1}}, \mathcal{D}_{k+1}) = \text{KSD-Thinning}(\phi_{\widetilde{\mathcal{D}}_{k+1}}, \widetilde{\mathcal{D}}_{k+1}, \epsilon_k). \tag{11}$$

We summarize the proposed algorithm in Algorithm 1 with compression subroutine in Algorithm 2, where `KSRL` is an abbreviation for Kernelized Stein Discrepancy Thinning for Model-Based Reinforcement Learning. Please refer to discussion Appendix (Chakraborty et al. 2022) for MPC-based action selection.

4 Bayesian Regret Analysis

In this section, we establish the regret of Algorithm 1. Our analysis builds upon (Osband and Van Roy 2014), but exhibits fundamental departures in the sense that we consider an alternative metric for quantifying the posterior estimation error using IPMs that exploit's salient structural properties of Stein's method, which additionally provides a basis for establishing tradeoffs between regret and posterior representational complexity which is novel in this work. Begin then by restating the model error at episode k from (4) as (Osband and Van Roy 2014)

$$\begin{aligned}\Delta_k &= \int \rho(s_1)(V_{\mu^*}^{M^*}(s_1) - V_{\tilde{\mu}^k}^{M^*}(s_1))ds_1 \\ &= \underbrace{\int \rho(s_1)(V_{\mu^*}^{M^*}(s_1) - V_{\mu^k}^{M^k}(s_1))ds_1}_{=:\Delta_k^I} \\ &\quad + \underbrace{\int (V_{\mu^k}^{M^k}(s_1) - V_{\mu^k}^{M^*}(s_1))ds_1}_{=:\Delta_k^{II}}, \end{aligned} \tag{12}$$

where we add and subtract the term $\int \rho(s_1)(V_{\mu^k}^{M^k}(s_1))ds_1$, μ^k represents the optimal policy (cf. (2)) under constructed model M^k from the thinned posterior we obtain via procedure proposed in Algorithm 1. Hence, the regret for episode k can be decomposed as $\Delta_k = \Delta_k^I + \Delta_k^{II}$ which implies

$$Regret_T = \sum_{k=1}^{\lceil \frac{T}{H} \rceil} \Delta_k^I + \sum_{k=1}^{\lceil \frac{T}{H} \rceil} \Delta_k^{II}, \tag{13}$$

where $\lceil \frac{T}{H} \rceil$ denotes the number of total episodes T is the total number of timesteps and H is the number of timesteps per episode. The equation in (13) matches with the regret decomposition in (Osband and Van Roy 2014) but there is a fundamental departure: specifically, (13) the sample M_k for episode k is sampled from the ***thinned*** posterior following the procedure proposed in Algorithm 1. Similar to prior works (Osband and Van Roy 2014; Fan and Ming 2021), since we are also performing posterior sampling $M^k \sim \phi(\cdot \mid \mathcal{D}_k)$ for each k, hence we note that $\mathbb{E}[\Delta_k^I] = 0$. Next, we take the expectation on both sides in (13) to write

$$\mathbb{E}[Regret_T] = \sum_{k=1}^{\lceil \frac{T}{H} \rceil} \mathbb{E}[\Delta_k^{II}], \tag{14}$$

which implies that the first term in (13) is null, which allows us to shift focus to analyzing the expected value of Δ_k^{II} for each k. To proceed with the analysis, we relate the estimation error of the future value function to the KSD in Lemma 4.1, which is a key novelty of this work that exploit's Stein's method. To keep the exposition simple, We first derive Lemma 4.1 as follows.

Lemma 4.1. *(Lipschitz in Kernel Stien Discrepancy) Recall the definition of the future value function in* (3)*. Under the assumption that posterior distributions are continuously differentiable (also called smooth) the future value function*

estimation error of P^k with respect to P^* is upper-bounded in terms of the KSD of P^k

$$U_i^k(P^k(\hat{h}_i)) - U_i^k(P^*(\hat{h}_i)) \leq HR_{max}KSD(P^k(h_i)), \quad (15)$$

for all i and k.

See Appendix (Chakraborty et al. 2022) for proof. Observe that Lemma 4.1 is unique to this work, and is a key point of departure from (Osband and Van Roy 2014; Fan and Ming 2021). An analogous inequality in (Osband and Van Roy 2014) mandates that the future value function is Lipschitz continuous without an explicit value of Lipschitz parameter. This implicit dependence is made explicit in (Fan and Ming 2021, Lemma 2), where the Lipschitz parameter is explicitly shown to depend on episode length H under the assumption that underlying model in Gaussian. Furthermore, an important point to note that both in (Osband and Van Roy 2014; Fan and Ming 2021), this upper bound on the future value function is provided in terms of total variation norm between the distributions P^k and P^*. In contrast, we take a different route based on Stein's method that alleviates the need for Gaussian assumption for the underlying transition model, and replaces the TV norm with KSD of $P^k(h_i)$.

Observe that the right hand side of (15) depends on the KSD of joint posterior $P^k(h_i)$ which we define for h_i. The Stein's method allows us to consider the KSD of joint because the score function of conditional distribution and joint distribution are the same and does not depend upon the normalizing constant. The salient point to note here is that we do not require access to P^* to evaluate the right hand side of (15). True to our knowledge, this is the first time that power of Stein's methods is being utilized in model-based RL. Next, we proceed towards deriving the regret for Algorithm 1. First, we need an additional result which upper bounds the KSD of current posterior KSD$(\phi_{\mathcal{D}_k})$ at episode k which we state next in Lemma 4.2.

Lemma 4.2. *(KSD Upper bound) Under the Assumptions of Lemma 4.1 and thinning budget $\epsilon_k = \frac{\log(k)}{f(k)^2}$, for the iterates of proposed Algorithm 1, it holds that*

$$\mathbb{E}\left[KSD(\Lambda_{\mathcal{D}_k})\right] = \mathcal{O}\left(\frac{\sqrt{k\log(k)}}{f(k)}\right), \quad (16)$$

where $f(k)$ lower-bounds the growth rate of the posterior's parameterization complexity (coreset size) as $|\mathcal{D}_k| \geq f(k)$.

See Appendix (Chakraborty et al. 2022) for proof. The inequality established in Lemma 4.2 relates the KSD to the number of episodes experienced by a model-based RL method, and may be interpreted as an adaption of related rates of posterior contraction in terms of KSD that have appeared in MCMC (Chen et al. 2019; **?**). In particular, with this expression, we can relate the expected value of the KSD of current thinned posterior $\phi_{\mathcal{D}_k}$ to the target density for each episode k. For the statistical consistency of the posterior estimate, we note that it is sufficient to show that $\mathbb{E}\left[\text{KSD}(\phi_{\mathcal{D}_k})\right] \to 0$ as $k \to \infty$, which imposes a lower bound on the dictionary size growth rate $f(k) > \sqrt{k\log(k)}$ from the statement of Lemma 4.1 required for convergence. This result communicates that it is possible to achieve statistical consistency without having a linear growth in the dictionary size that is a drawback of prior art (Osband and Van Roy 2014; Fan and Ming 2021). Next, we ready to combine Lemmas 4.1 - 4.2 to establish our main result.

Theorem 4.3 (Regret and Coreset Size Tradeoff KSRL)**.** *Under the thinning budget $\epsilon_k = \frac{\log(k)}{f(k)^2}$ and coreset size growth condition $f(k) = \sqrt{k^{\alpha+1}\log(k)}$ where $\alpha \in (0, 1]$, the total Bayes regret for our KSD based posterior thinning algorithm for model-based RL (cf. Algorithm 1) is given by*

$$\mathbb{E}[Regret_T] = \mathcal{O}\left(dT^{1-\frac{\alpha}{2}}H^{1+\frac{\alpha}{2}}\right) \quad (17)$$

and coreset size ($M(T)$) order is given by $M(T) = \tilde{\Omega}\left(\sqrt{T^{1+\alpha}}\right)$, where T denotes the total number of state-action pairs processed, and H is the length of each episode.

The proof is provided in Appendix (Chakraborty et al. 2022). To establish Theorem 4.3, we start with the upper bound on the term Δ_k^{II} via utilizing the result from the statement of Lemma 4.1. Then we upper bound the right hand side of (15) via the relationship between KSD and the number of past samples (Lemma 4.2).

Remark 1 (Dependence on Dimension): Prior results (Osband and Van Roy 2014; Fan and Ming 2021) exhibit a linear dependence on the input dimension d, which matches our dependence. However, these results require the posterior to belong to a symmetric class of distributions. We relax this assumption and only require our posterior to be smooth, a substantial relaxation.

Remark 2 (Tradeoff between regret and model complexity): An additional salient attribute of Theorem 4.3 is the introduction of compression budget ϵ_k which we specify in terms of tunable parameter α. This quantity determines the number of elements that comprise the thinned posterior during model-based RL training. We provide a more quantitative treatment in the Table 2. Here, in the table, we present the

α	$\mathbb{E}[\text{Regret}_T]$	$M(T)$
0	$\mathcal{O}(dHT)$	$\widetilde{\Omega}(\sqrt{T})$
0.5	$\mathcal{O}(dH^{\frac{7}{4}}T^{\frac{1}{4}})$	$\widetilde{\Omega}(T^{3/4})$
1	$\mathcal{O}(dH^{\frac{3}{2}}T^{\frac{1}{2}})$	$\widetilde{\Omega}(T)$

Table 2: Tradeoff for different values of α.

regret analysis for our Efficient Stein-based PSRL algorithm. Observe that we match the best known prior results of PSRL with $\alpha = 1$ as shown in (Fan and Ming 2021) in-terms of $O(dH^{\frac{3}{2}}T^{\frac{1}{2}})$, but with relaxed conditions on the posterior, allowing the approach to apply to a much broader class of problems. Moreover, for $\alpha < 1$, we obtain a superior tradeoff in model complexity and regret. Now, from a statistical consistency perspective a posterior is consistent for β if the posterior distribution on β concentrates in neighborhoods of the true value.

5 Experiments

In this section, we present a detailed experimental analysis of KSRL as compared to state of the art model-based and model-

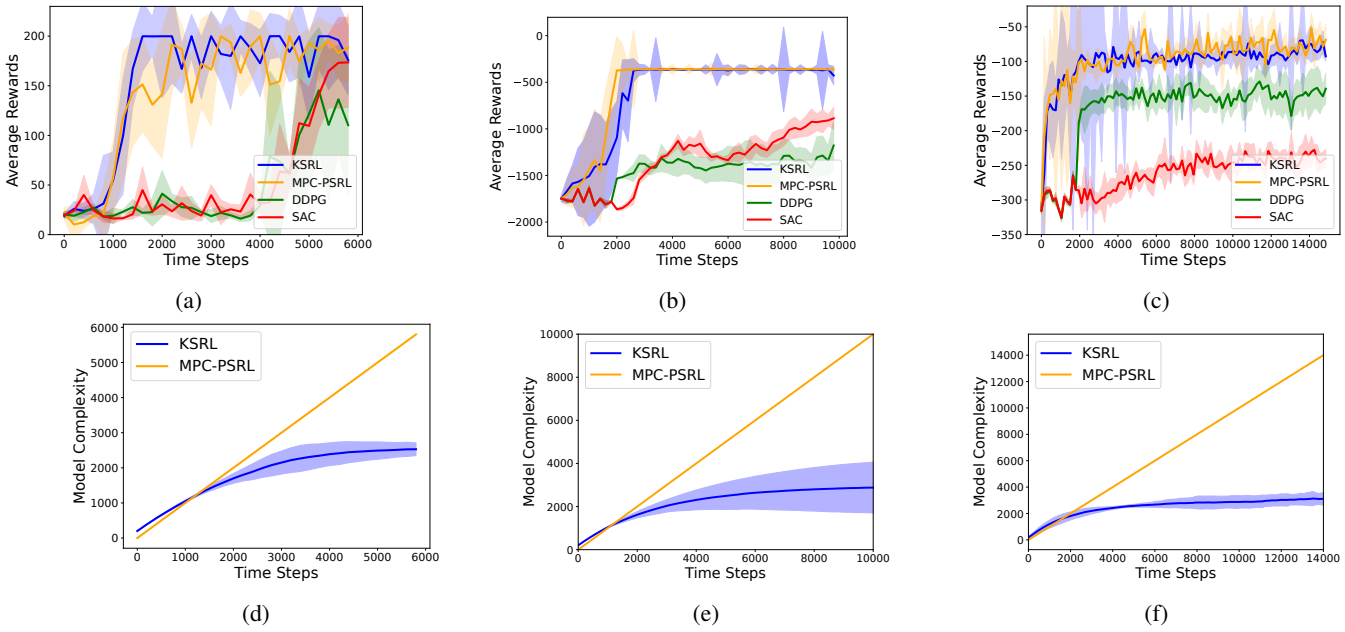

Figure 1: (a)-(c) compares the average cumulative reward return achieved by the proposed KSRL (shown in blue) algorithm with MPC-PSRL (Fan and Ming 2021), SAC (Haarnoja et al. 2018), and DDPG (Barth-Maron et al. 2018) for modified Cartpole, Pendulum, and Pusher without rewards. Figures with rewards are shown in the Appendix Experimental section (Chakraborty et al. 2022). (d)-(f) compares the model-complexity. We note that KSRL is able to achieve the maximum average reward at-par with the current SOTA MPC-PSRL with drastically reduced model complexity. Solid curves represent the average across five trials (seeds), shaded areas correspond to the standard deviation amongst the trials.

free RL methods on several continuous control tasks in terms of training rewards, model complexity and KSD convergence. First we discuss the different baseline algorithms to which we compare the proposed KSRL. Secondly, we details the experimental environments, and then we empirically validate and analyze the performance of KSRL in detail.

Baselines. For comparison to other model free approaches, we compare against MPC-PSRL method propose in (Fan and Ming 2021). There are other popular model based methods in literature such as MBPO (Janner et al. 2019) and PETS (Chua et al. 2018) but MPC-PSRL is already shown to outperform them in (Fan and Ming 2021, Fig. 1). Since the underlying environments are same, we just compare to MPC-PSRL and show improvements. For comparison to model-free approaches, we compare with Soft Actor-Critic (SAC) from (Haarnoja et al. 2018) and Deep Deterministic Policy Gradient (DDPG) (Barth-Maron et al. 2018).

Environment Details. We consider continuous control environments Stochastic Pendulum , Continuous Cartpole, Reacher and Pusher with and without rewards of modified OpenAI Gym (Brockman et al. 2016) & MuJoCo environments (Todorov, Erez, and Tassa 2012). These environments are of different complexity and provides a good range of performance comparison in practice. See Appendix (Chakraborty et al. 2022) for additional specific details of the environments and architecture.

Discussion. Fig. 1 compares the average reward return (top row) and model complexity (bottom row) for Cartpole, Pendulum, and Pusher, respectively. We note that KSRL performs equally good or even better as compared to the state-of-the-art MPC-PSRL algorithm with a significant reduction in model complexity (bottom row in Fig. 1) consistently across different environments. From the model complexity plots, we remark that KSRL is capable of automatically selecting the data points and control the dictionary growth across different environments which helps to achieve same performance in terms of average reward with fewer dictionary points to parameterize posterior distributions. This also helps in achieving faster compute time in practice to perform the same task as detailed in the Experimental section Appendix (Chakraborty et al. 2022). We also show improvements of our algorithm over MPC with fixed buffer size in Appendix(Chakraborty et al. 2022) with both random and sequential removal.

6 Conclusions

In this work, we develop a novel Bayesian regret analysis for model-based RL that can scale efficiently in continuous space under more general distributional settings and we achieve this objective by constructing a coreset of points which contributes the most to the posterior as quantified by KSD. Theoretical and experimental analysis bore out the practical utility of this methodology.

References

Agrawal, S.; and Jia, R. 2017. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. In

Advances in Neural Information Processing Systems, 1184–1194.

Amortila, P.; Precup, D.; Panangaden, P.; and Bellemare, M. G. 2020. A distributional analysis of sampling-based reinforcement learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, 4357–4366. PMLR.

Andrieu, C.; De Freitas, N.; Doucet, A.; and Jordan, M. I. 2003. An introduction to MCMC for machine learning. *Machine learning*, 50(1): 5–43.

Ayoub, A.; Jia, Z.; Szepesvari, C.; Wang, M.; and Yang, L. 2020. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, 463–474. PMLR.

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

Borkar, V. S.; and Meyn, S. P. 2002. Risk-sensitive optimal control for Markov decision processes with monotone cost. *Mathematics of Operations Research*, 27(1): 192–209.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. Openai gym. *arXiv preprint arXiv:1606.01540*.

Campbell, T.; and Broderick, T. 2018. Bayesian coreset construction via greedy iterative geodesic ascent. In *International Conference on Machine Learning*, 698–706. PMLR.

Campbell, T.; and Broderick, T. 2019. Automated scalable Bayesian inference via Hilbert coresets. *The Journal of Machine Learning Research*, 20(1): 551–588.

Chakraborty, S.; Bedi, A. S.; Koppel, A.; Sadler, B. M.; Huang, F.; Tokekar, P.; and Manocha, D. 2022. Posterior Coreset Construction with Kernelized Stein Discrepancy for Model-Based Reinforcement Learning. arXiv:2206.01162.

Chen, W. Y.; Barp, A.; Briol, F.-X.; Gorham, J.; Girolami, M.; Mackey, L.; and Oates, C. 2019. Stein point markov chain monte carlo. In *International Conference on Machine Learning*, 1011–1021. PMLR.

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

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

Dann, C.; Lattimore, T.; and Brunskill, E. 2017. Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 30.

Deisenroth, M. P.; Fox, D.; and Rasmussen, C. E. 2013. Gaussian processes for data-efficient learning in robotics and control. *IEEE transactions on pattern analysis and machine intelligence*, 37(2): 408–423.

Elvira, V.; Míguez, J.; and Djurić, P. M. 2016. Adapting the number of particles in sequential Monte Carlo methods through an online scheme for convergence assessment. *IEEE Transactions on Signal Processing*, 65(7): 1781–1794.

Fan, Y.; and Ming, Y. 2021. Model-based Reinforcement Learning for Continuous Control with Posterior Sampling. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 3078–3087. PMLR.

Garcia, C. E.; Prett, D. M.; and Morari, M. 1989. Model predictive control: Theory and practice—A survey. *Automatica*, 25(3): 335–348.

Ghavamzadeh, M.; Mannor, S.; Pineau, J.; Tamar, A.; et al. 2015. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6): 359–483.

Gorham, J.; and Mackey, L. 2015. Measuring sample quality with Stein's method. *Advances in Neural Information Processing Systems*, 28.

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

Hawkins, C.; Koppel, A.; and Zhang, Z. 2022. Online, informative mcmc thinning with kernelized stein discrepancy. *arXiv preprint arXiv:2201.07130*.

Jaksch, T.; Ortner, R.; and Auer, P. 2010. Near-optimal Regret Bounds for Reinforcement Learning. *Journal of Machine Learning Research*, 11(51): 1563–1600.

Janner, M.; Fu, J.; Zhang, M.; and Levine, S. 2019. When to trust your model: Model-based policy optimization. *Advances in Neural Information Processing Systems*, 32.

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

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

Kamthe, S.; and Deisenroth, M. 2018. Data-efficient reinforcement learning with probabilistic model predictive control. In *International conference on artificial intelligence and statistics*, 1701–1710. PMLR.

Kattumannil, S. K. 2009. On Stein's identity and its applications. *Statistics & Probability Letters*, 79(12): 1444–1449.

Koppel, A.; Pradhan, H.; and Rajawat, K. 2021. Consistent online gaussian process regression without the sample complexity bottleneck. *Statistics and Computing*, 31(6): 1–18.

Kose, U.; and Ruszczynski, A. 2021. Risk-averse learning by temporal difference methods with markov risk measures. *Journal of machine learning research*, 22.

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

Liu, Q.; Lee, J.; and Jordan, M. 2016. A kernelized Stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, 276–284. PMLR.

Nguyen, T.; Li, Z.; Silander, T.; and Leong, T. Y. 2013. Online feature selection for model-based reinforcement learning. In *International Conference on Machine Learning*, 498–506. PMLR.

O'Donoghue, B. 2021. Variational bayesian reinforcement learning with regret bounds. *Advances in Neural Information Processing Systems*, 34.

Osband, I.; Benjamin, V. R.; and Daniel, R. 2013. (More) Efficient Reinforcement Learning via Posterior Sampling. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, 3003–3011. USA: Curran Associates Inc.

Osband, I.; Russo, D.; and Van Roy, B. 2013. (More) efficient reinforcement learning via posterior sampling. *Advances in Neural Information Processing Systems*, 26.

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

Osband, I.; and Van Roy, B. 2017a. Why is posterior sampling better than optimism for reinforcement learning? In *International conference on machine learning*, 2701–2710. PMLR.

Osband, I.; and Van Roy, B. 2017b. 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*, 2701–2710. International Convention Centre, Sydney, Australia: PMLR.

Ouyang, Y.; Gagrani, M.; and Jain, R. 2017. Control of unknown linear systems with thompson sampling. In *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 1198–1205. IEEE.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Rasmussen, C. E. 2004. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, 63–71. Springer.

Ross, N. 2011. Fundamentals of Stein's method. *Probability Surveys*, 8: 210–293.

Sriperumbudur, B. K.; Fukumizu, K.; Gretton, A.; Schölkopf, B.; and Lanckriet, G. R. 2012. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6: 1550–1599.

Stein, C.; et al. 1956. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the Third Berkeley symposium on mathematical statistics and probability*, volume 1, 197–206.

Strehl, A. L.; Li, L.; and Littman, M. L. 2009. Reinforcement Learning in Finite MDPs: PAC Analysis. *Journal of Machine Learning Research*, 10(11).

Sutton, R. S. 1988. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1): 9–44.

Sutton, R. S.; and Barto, A. G. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Theocharous, G.; Wen, Z.; Abbasi Yadkori, Y.; and Vlassis, N. 2018. Scalar posterior sampling with applications. *Advances in Neural Information Processing Systems*, 31.

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

Todorov, E.; Erez, T.; and Tassa, Y. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, 5026–5033. IEEE.

Wang, T.; Bao, X.; Clavera, I.; Hoang, J.; Wen, Y.; Langlois, E.; Zhang, S.; Zhang, G.; Abbeel, P.; and Ba, J. 2019. Benchmarking model-based reinforcement learning. *arXiv preprint arXiv:1907.02057*.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3): 229–256.

Xu, Z.; and Tewari, A. 2020. Reinforcement learning in factored mdps: Oracle-efficient algorithms and tighter regret bounds for the non-episodic setting. *Advances in Neural Information Processing Systems*, 33: 18226–18236.

Yang, L.; and Wang, M. 2020. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, 10746–10756. PMLR.

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

Zhou, D.; Gu, Q.; and Szepesvari, C. 2021. Nearly minimax optimal reinforcement learning for linear mixture markov decision processes. In *Conference on Learning Theory*, 4532–4576. PMLR.