Model-based Policy Optimization under Approximate Bayesian Inference

Chaoqi Wang [*1] **Yuxin Chen** [1] **Kevin Murphy** [2]

Abstract

Model-based reinforcement learning algorithms (MBRL) present an exceptional potential to enhance sample efficiency within the realm of online reinforcement learning (RL). Nevertheless, a substantial proportion of prevalent MBRL algorithms fail to adequately address the dichotomy of exploration and exploitation. Posterior sampling reinforcement learning (PSRL) emerges as an innovative strategy adept at balancing exploration and exploitation, albeit its theoretical assurances are contingent upon exact inference. In this paper, we show that adopting the same methodology as in exact PSRL can be suboptimal under approximate inference. Motivated by the analysis, we propose an improved factorization for the posterior distribution of polices by removing the conditional independence between the policy and data given the model. By adopting such a posterior factorization, we further propose a general algorithmic framework for PSRL under approximate inference and a practical instantiation of it. Empirically, our algorithm can surpass baseline methods by a significant margin on both dense rewards and sparse rewards tasks from the Deepmind control suite, OpenAI Gym and Metaworld benchmarks.

1. Introduction

Model-based reinforcement learning has proven instrumental in enhancing the sample efficiency of reinforcement learning. Despite this, prevalent MBRL algorithms (Kurutach et al., 2018; Chua et al., 2018; Janner et al., 2019; Eysenbach et al., 2022) often struggle to balance exploration and exploitation, leading to poor performance when exploration is pivotal. Existing algorithms typically employ one of three strategies to achieve this balance: 1) *optimism-based* (Auer et al., 2008; Pacchiano et al., 2021; Curi et al., 2020); 2) *posterior-sampling-based* (Strens, 2000; Osband et al., 2013; 2018; Fan & Ming, 2021); or 3) *information-directed sampling* (Russo & Roy, 2014).

As outlined by Osband & Roy (2017), posterior sampling reinforcement learning (PSRL) can match the statistical efficiency or regret bound of optimism-based algorithms while providing superior computational efficiency. Information-directed sampling methods can exhibit even greater statistical efficiency when dealing with intricate information structures (Russo & Roy, 2014), but they require estimators for the mutual information, which is challenging for high-dimensional random variables. Consequently, this paper focuses on PSRL in the episodic setting.

Within the PSRL framework, a posterior $p(\mathcal{M}|\mathcal{D}_{\mathcal{E}})$ of the Markov decision process (MDP) $\mathcal{M}$, based on the observations $\mathcal{D}_{\mathcal{E}}$ from a real-world environment $\mathcal{E}$, is maintained. At the onset of each episode, an MDP is sampled from the posterior and the optimal policy $\pi(\mathcal{M})$ for the chosen model $\mathcal{M}$ is computed. This procedure could equivalently be perceived as the sampling of this policy from a specialized posterior, termed "degenerate" in this context, which is formulated as $p(\pi|\mathcal{D}_{\mathcal{E}}) = \int \delta(\pi|\mathcal{M})p(\mathcal{M}|\mathcal{D}_{\mathcal{E}})d\mathcal{M}$. Here, $\delta(\pi|\mathcal{M}) = \delta(\pi - \pi(\mathcal{M}))$ represents a Dirac delta distribution of the optimal policy. This specific policy is then applied within the real environment to collect new data. Theoretically, such a direct approach has been proven to achieve a Bayesian regret of $\tilde{\mathcal{O}}(\sqrt{K})$ over K episodes, as corroborated by prior studies (Osband et al., 2013; Osband & Roy, 2014). Nonetheless, these theoretical guarantees hold true only under specific conditions of exact inference, namely, when one can access the exact posterior over models, denoted as $p(\mathcal{M}|\mathcal{D}_{\mathcal{E}})$, and when the computation of the optimal policy is achievable. In real-world scenarios, however, such precise conditions are unlikely to be met, and thus we may need approximations.

An often-employed heuristic approximation to PSRL, as seen in Fan & Ming (2021), is to substitute the posterior over models, $p(\mathcal{M}|\mathcal{D}_{\mathcal{E}})$, with an approximation $q(\mathcal{M}|\mathcal{D}_{\mathcal{E}})$, such as Bayesian linear regression (Box & Tiao, 2011; Gelman et al., 2013; Murphy, 2022) applied to repre-

[*]Work done in Google Brain (now Google DeepMind). [1]Department of Computer Science, University of Chicago [2]Google DeepMind. Correspondence to: Chaoqi Wang <chaoqi@uchicago.edu>.

Workshop on New Frontiers in Learning, Control, and Dynamical Systems at the International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA, 2023.

sentations learned via a neural network. In this way, the resulting policy is then sampled from $q^{\delta}(\pi|\mathcal{D}_{\mathcal{E}}) = \int \delta(\pi|\mathcal{M}) q(\mathcal{M}|\mathcal{D}_{\mathcal{E}}) d\mathcal{M}$, where we use $q(\mathcal{M}|\mathcal{D}_{\mathcal{E}})$ to approximate the true posterior $p(\mathcal{M}|\mathcal{D}_{\mathcal{E}})$.

While the heuristic $q^{\delta}(\pi|\mathcal{D}_{\mathcal{E}})$ may initially seem appropriate due to its resemblance to the true posterior $p(\pi|\mathcal{D}_{\mathcal{E}})$, our findings demonstrate that its performance could be substandard, as measured by the KL divergence between the heuristic and the true posterior, potentially leading to increased regret (Lu et al., 2021). To address this issue, we propose the substitution of the degenerate $\delta(\pi|\mathcal{M})$ with a non-degenerate distribution $q(\pi|\mathcal{M}, \mathcal{D}_{\mathcal{E}})$, which conditions on both the model and the empirical data $\mathcal{D}_{\mathcal{E}}$. In this way, the posterior of polices then become $q(\pi|\mathcal{D}_{\mathcal{E}}) = \int q(\pi|\mathcal{M}, \mathcal{D}_{\mathcal{E}}) q(\mathcal{M}|\mathcal{D}_{\mathcal{E}}) d\mathcal{M}$. This modification counterbalances any potential inadequacies in the posterior over models $q(\mathcal{M}|\mathcal{D}_{\mathcal{E}})$. Further tuning the dependency of π on $\mathcal{D}_{\mathcal{E}}$ and $\mathcal{M}$ enables a harmonious balance between data efficiency and inference error in $q(\mathcal{M}|\mathcal{D}_{\mathcal{E}})$. Crucially, this proposed modification comes with a performance guarantee: it is assured not to underperform compared to the conventional approach (i.e., $q^{\delta}(\pi|\mathcal{D}_{\mathcal{E}})$) that employs the degenerate distribution $\delta(\pi|\mathcal{M})$.

Capitalizing on these findings, we propose a novel framework for PSRL under approximate inference. To put this method into practice, we amalgamate deep ensembles (Lakshminarayanan et al., 2017) and Model-based Policy Optimization (MBPO) (Janner et al., 2019). We also put forward two distinct sampling strategies for policy selection, leveraging our posterior approximation. Empirical evidence demonstrates that our algorithm substantially outperforms existing baselines on both dense reward and sparse reward tasks (Brockman et al., 2016; Tunyasuvunakool et al., 2020; Yu et al., 2020). In addition to these performance evaluations, we perform numerous ablation studies to facilitate a deeper understanding of our algorithm's effectiveness.

In summary, the key contributions of this paper are:

1. We first conduct a study on how approximate inference impacts the performance in PSRL, demonstrating that maintaining the same methodology (i.e., $q^{\delta}(\pi|\mathcal{M})$) as in exact PSRL may be less effective when the true posterior is unavailable (Section 2).
2. Second, we design a novel framework for PSRL under approximate inference, building on the aforementioned observations. This framework integrates the use of deep ensembles and Model-based Policy Optimization (MBPO) along with two policy sampling strategies that exploit our posterior approximation. This forms our primary contribution (Section 3).
3. Finally, we conduct experiments on the DM control suite, OpenAI Gym, and Metaworld benchmarks to demonstrate the superiority of our approach. Our algorithm shows significant performance improvements over existing baselines on both dense and sparse reward tasks. We further elucidate the effectiveness through several ablation studies (Section 4).

2. Preliminary and Backgrounds

Notation. We consider the finite-horizon episodic Markov Decision Process (MDP) problem, of which we denote an instance as $\mathcal{M} := \{\mathcal{S}, \mathcal{A}, r_{\mathcal{M}}, p_{\mathcal{M}}, H, \rho\}$. For each instance $\mathcal{M}$, $\mathcal{S}$ and $\mathcal{A}$ denote the set of states and actions, respectively. $r_{\mathcal{M}} : \mathcal{S} \times \mathcal{A} \to [0, R_{\text{max}}]$ is the reward function, $p_{\mathcal{M}}$ is the transition distribution, H is the length of the episode, and ρ is the distribution of the initial state. We further define the value function of a policy π under MDP $\mathcal{M}$ at timestep i as

$$V_{\pi,i}^{\mathcal{M}}(\boldsymbol{s}) := \mathbb{E}_{\mathcal{M},\pi}\left[\sum_{t=i}^{H} r_{\mathcal{M}}(\boldsymbol{s}_t, \boldsymbol{a}_t) |\, \boldsymbol{s}_i = \boldsymbol{s}\right], \quad (1)$$

where $\boldsymbol{s}_{t+1} \sim p_{\mathcal{M}}(\boldsymbol{s}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ and $\boldsymbol{a}_t \sim \pi(\boldsymbol{a}|\boldsymbol{s}_t)$. We define $\pi^{\star}$ as the optimal policy for an MDP $\mathcal{M}$ if $V_{\pi^{\star},i}^{\mathcal{M}}(\boldsymbol{s}) = \max_{\pi} V_{\pi,i}^{\mathcal{M}}(\boldsymbol{s})$ for all $\boldsymbol{s} \in \mathcal{S}$ and $i \in [1, H]$. We define the cumulative reward obtained by policy π over H steps sampled from model $\mathcal{M}$ as follows:

$$R_{\mathcal{M}}(\pi) = \mathbb{E}_{\mathcal{M},\pi}\left[\sum_{t=1}^{H} r_{\mathcal{M}}(\boldsymbol{s}_t, \boldsymbol{a}_t)\right] \quad (2)$$
$$\text{where} \quad \boldsymbol{s}_{t+1} \sim p_{\mathcal{M}}(\boldsymbol{s}|\boldsymbol{s}_t, \boldsymbol{a}_t) \ \text{ and } \ \boldsymbol{a}_t \sim \pi(\boldsymbol{a}|\boldsymbol{s}_t),$$

where the initial state is sampled from $\boldsymbol{s}_1 \sim \rho(\boldsymbol{s})$.

PSRL. Posterior Sampling Reinforcement Learning or PSRL (Strens, 2000) is a well-known algorithmic framework for managing the trade-off between exploration and exploitation in online RL. The framework is generic and can be applied to various RL problems. At the core of PSRL lies the computation of the posterior distribution over MDPs, which includes the dynamics and reward models. The quality of this posterior distribution is crucial for the performance of PSRL. We denote the posterior of the model as $p(\mathcal{M}|\mathcal{D}_{\mathcal{E}})$, which is conditioned on the data $\mathcal{D}_{\mathcal{E}}$ collected from the environment. Therefore, the posterior distribution of policies can be expressed as follows:

$$p(\pi|\mathcal{D}_{\mathcal{E}}) = \int p(\pi|\mathcal{M})\, p(\mathcal{M}|\mathcal{D}_{\mathcal{E}}) d\mathcal{M}, \quad (6)$$

where $p(\pi|\mathcal{M}) = \delta(\pi|\mathcal{M})$, and $\delta(\pi|\mathcal{M})$ is a Dirac delta distribution[1], defined as $\delta(\pi(\mathcal{M})|\mathcal{M}) = 1$, and $\pi(\mathcal{M}) = \arg\max_{\pi} R_{\mathcal{M}}(\pi)$ is the optimal policy for solving the MDP $\mathcal{M}$. At the start of each episode, a Markov decision process

[1] Although an MDP, $\mathcal{M}$, can have multiple optimal policies, we'll simplify by assuming just one.

EXAMPLE 1. SUBOPTIMALITY OF $q^\delta(\pi|\mathcal{M})$.

Consider a toy setting, where the support set of MDPs is $\{\mathcal{M}_1, \mathcal{M}_2\}$, and the support set of policies is $\{\pi_1, \pi_2\}$. Suppose that the true posterior distribution of MDPs is $p(\mathcal{M}_1|\mathcal{D}_\mathcal{E}) = 1/3$, $p(\mathcal{M}_2|\mathcal{D}_\mathcal{E}) = 2/3$, and the optimal policy per MDP is $\delta(\pi_1|\mathcal{M}_1) = 1$ and $\delta(\pi_2|\mathcal{M}_2) = 1$. This we get the following exact distribution over policies: $p(\pi|\mathcal{D}_\mathcal{E})$ is

$$p(\pi|\mathcal{D}_\mathcal{E}) = \underbrace{\begin{bmatrix} \delta(\pi_1|\mathcal{M}_1)=1,\ \delta(\pi_1|\mathcal{M}_2)=0 \\ \delta(\pi_2|\mathcal{M}_1)=0,\ \delta(\pi_2|\mathcal{M}_2)=1 \end{bmatrix}}_{\delta(\pi|\mathcal{M})} \underbrace{\begin{bmatrix} p(\mathcal{M}_1|\mathcal{D}_\mathcal{E})=\frac{2}{3} \\ p(\mathcal{M}_2|\mathcal{D}_\mathcal{E})=\frac{1}{3} \end{bmatrix}}_{p(\mathcal{M}|\mathcal{D}_\mathcal{E})} = \begin{bmatrix} p(\pi_1|\mathcal{D}_\mathcal{E})=\frac{2}{3} \\ p(\pi_2|\mathcal{D}_\mathcal{E})=\frac{1}{3} \end{bmatrix} \tag{3}$$

Now suppose we use the approximate posterior distribution over models, $q(\mathcal{M}_1|\mathcal{D}_\mathcal{E}) = 0$ and $q(\mathcal{M}_2|\mathcal{D}_\mathcal{E}) = 1$. We can optimize $q(\pi|\mathcal{M})$ by minimizing $\mathbf{d}_{\text{KL}}\left(q(\pi|\mathcal{D}_\mathcal{E})|\, p(\pi|\mathcal{D}_\mathcal{E})\right)$. One solution could be

$$q(\pi|\mathcal{D}_\mathcal{E}) = \underbrace{\begin{bmatrix} q(\pi_1|\mathcal{M}_1)=\frac{1}{2},\ q(\pi_1|\mathcal{M}_2)=\frac{2}{3} \\ q(\pi_2|\mathcal{M}_1)=\frac{1}{2},\ q(\pi_2|\mathcal{M}_2)=\frac{1}{3} \end{bmatrix}}_{q(\pi|\mathcal{M})} \underbrace{\begin{bmatrix} q(\mathcal{M}_1|\mathcal{D}_\mathcal{E})=0 \\ q(\mathcal{M}_2|\mathcal{D}_\mathcal{E})=1 \end{bmatrix}}_{q(\mathcal{M}|\mathcal{D}_\mathcal{E})} = \begin{bmatrix} q(\pi_1|\mathcal{D}_\mathcal{E})=\frac{2}{3} \\ q(\pi_2|\mathcal{D}_\mathcal{E})=\frac{1}{3} \end{bmatrix} \tag{4}$$

We see that the optimal $q(\pi|\mathcal{M})$ requires modeling uncertainty in the policy even conditional on the model. By contrast, if we adopt $q^\delta(\pi|\mathcal{D}_\mathcal{E})$ as our approximation, we will have

$$\mathbf{d}_{\text{KL}}\left(q^\delta(\pi|\mathcal{D}_\mathcal{E})\middle|\, p(\pi|\mathcal{D}_\mathcal{E})\right) = \log 3 = \max_{q\in\Delta^1} \mathbf{d}_{\text{KL}}\left(q(\pi|\mathcal{D}_\mathcal{E})|\, p(\pi|\mathcal{D}_\mathcal{E})\right). \tag{5}$$

(MDP) or, equivalently, a policy, is drawn at random from the posterior distribution. This sampled MDP is then used to collect new data. Despite its simplicity, this algorithmic framework achieves a Bayesian regret of $\tilde{O}(\sqrt{K})$ (Osband et al., 2013), where K is the total number of episodes. However, it is important to note that the theoretical results only hold under the assumption of exact inference. In the remainder of this section, we will discuss the impact on performance when using approximate inference methods.

A typical instantiation of $q(\pi|\mathcal{D}_\mathcal{E})$ is to use a distribution that has the same functional form as the true posterior, which is commonly adopted in practice (Fan & Ming, 2021), i.e.,

$$q^\delta(\pi|\mathcal{D}_\mathcal{E}) := \int \delta(\pi|\mathcal{M})q(\mathcal{M}|\mathcal{D}_\mathcal{E})d\mathcal{M}. \tag{7}$$

However, this choice may not always be effective for approximating the posterior of policies, as demonstrated by the following proposition.

Proposition 1 *Under approximate inference (i.e., $q(\mathcal{M}|\mathcal{D}_\mathcal{E}) \neq p(\mathcal{M}|\mathcal{D}_\mathcal{E})$), the optimal $q(\pi|\mathcal{M})$ may not be a Dirac delta distribution, i.e., there exists other $q(\pi|\mathcal{D}_\mathcal{E})$ such that*

$$\mathbf{d}_{KL}\left(q(\pi|\mathcal{D}_\mathcal{E})|\, p(\pi|\mathcal{D}_\mathcal{E})\right) \leq \mathbf{d}_{KL}\left(q^\delta(\pi|\mathcal{D}_\mathcal{E})\middle|\, p(\pi|\mathcal{D}_\mathcal{E})\right).$$

As an illustration, we present an example in Example 1 that serves as a constructive proof of Proposition 1. This example demonstrates that $q^\delta(\pi|\mathcal{D}_\mathcal{E})$ can have arbitrarily poor performance in terms of the KL divergence. Given this observation, a reasonable and pressing question arises: what is a better alternative to $q^\delta(\pi|\mathcal{D}_\mathcal{E})$? The subsequent section provides an answer to this question by introducing a novel approach that outperforms $q^\delta(\pi|\mathcal{D}_\mathcal{E})$ in terms of the KL divergence.

3. Method

Prompted by the findings delineated in the preceding section, we initially propose a posterior decomposition that is assuredly superior to $q^\delta(\pi|\mathcal{D}_\mathcal{E})$. Subsequently, we present a pragmatic version of the algorithm based deep ensembles (Lakshminarayanan et al., 2017) and MBPO (Janner et al., 2019). Lastly, we propose a duo of sampling methodologies designed to foster efficient exploration.

3.1. An improved posterior decomposition

In section 2, we demonstrated that $q^\delta(\pi|\mathcal{D}_\mathcal{E})$ is not an advantageous choice, primarily due to its presumption that the policy π is predetermined once $\mathcal{M}$ is provided. This rationale prompts us to contemplate a more versatile posterior decomposition of $q(\pi|\mathcal{D}_\mathcal{E})$.

$$q(\pi|\mathcal{D}_\mathcal{E}, \lambda) = \int q(\pi|\mathcal{M}, \mathcal{D}_\mathcal{E}, \lambda)q(\mathcal{M}|\mathcal{D}_\mathcal{E})d\mathcal{M}. \tag{8}$$

From an intuitive standpoint, such a posterior decomposition no longer operates under the assumption that the model is fully capable of encapsulating all relevant attributes of the data. We exemplify these two posterior approximations in Figure 1. The additional parameter $\lambda \in [0, 1]$ provides a mechanism for calibrating the significance of fictitious data ($\mathcal{D}_\mathcal{M}$) originating from $\mathcal{M}$ and the actual data ($\mathcal{D}_\mathcal{E}$)

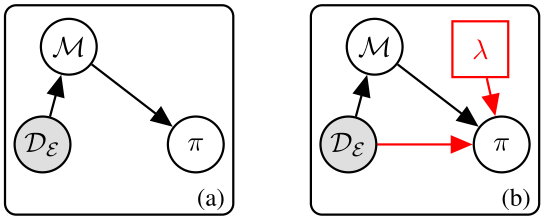

Figure 1: Graphical models for (a) the standard and (b) our posterior over policies π. Differences are shown in red.

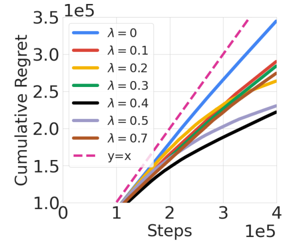

Figure 2: A comparison of cumulative regret for different λ.

sourced from the environment. In particular, we define

$$q(\pi|\mathcal{M}, \mathcal{D}_{\mathcal{E}}, \lambda = 0) = q(\pi|\mathcal{M}) = \delta(\pi|\mathcal{M}) \quad (9)$$

$$q(\pi|\mathcal{M}, \mathcal{D}_{\mathcal{E}}, \lambda = 1) = q(\pi|\mathcal{D}_{\mathcal{E}}) \quad (10)$$

Consequently,when λ is small, we trust our model more, whereas when λ is large we trust it less. In the extreme scenario where $\lambda = 0$, this framework degenerates to the posterior $q^{\delta}(\pi|\mathcal{D}_{\mathcal{E}})$. By fine-tuning λ, an optimal balance can be struck between minimizing the impact of approximate inference error and maximizing data efficiency. To encapsulate this in more formal terms, the subsequent proposition delineates the advantage of equation 8.

Proposition 2 *By adopting the posterior decomposition of equation 8, we have*

$$\min_{\lambda} \mathbf{d}_{KL}\left(q(\pi|\mathcal{D}_{\mathcal{E}}, \lambda) | \, p(\pi|\mathcal{D}_{\mathcal{E}})\right) \le \mathbf{d}_{KL}\left(q^{\delta}(\pi|\mathcal{D}_{\mathcal{E}}) \middle| \, p(\pi|\mathcal{D}_{\mathcal{E}})\right).$$

This proposition indicates that we have the capability to curtail the KL divergence, and thus potentially reduce the Bayesian regret, through a judicious selection of the λ value. As corroborative empirical evidence, Figure 2 depicts the cumulative regret with varying λ in the cartpole-swingup environment with sparse reward adopted from the DeepMind Control Suite (Tunyasuvunakool et al., 2020). We can observe that a λ value of approximately 0.4 or 0.5 is optimal, while $\lambda = 0$ is least favorable in this circumstance.

3.2. The proposed algorithm and its practical instantiation

Leveraging the aforementioned results, we present a simple yet general algorithmic framework for PSRL under approximate inference. This framework diverges from the standard PSRL framework solely in the decomposition of policy posterior, as delineated in Algorithm 2 in Appendix C. For practical implementation, we employ deep ensembles (Lakshminarayanan et al., 2017) to approximate the posterior distributions $q(\mathcal{M}|\mathcal{D}_{\mathcal{E}})$ and $q(\pi|\mathcal{M}, \mathcal{D}_{\mathcal{E}})$, as represented using Θ and Φ in Algorithm 1. This approach aligns with ME-TRPO (Kurutach et al., 2018), PETS (Chua et al., 2018), and MBPO (Janner et al., 2019), with the additional facet of modeling the uncertainty over policies, i.e., $q(\pi|\mathcal{M}, \mathcal{D}_{\mathcal{E}})$, as well as dynamics, i.e., $q(\mathcal{M}|\mathcal{D}_{\mathcal{E}})$.

Delving into greater detail, each constituent of the deep ensemble represents a conditional Gaussian distribution over outputs, marked by its mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$. For multi-dimensional predictions, we treat each dimension independently, predicting only the marginal mean and variance for the sake of simplicity. Each ensemble member is then trained independently by minimizing the negative log-likelihood, represented by the following equation:

$$-\log p_{\boldsymbol{\theta}}(y|\boldsymbol{x}) \propto \frac{\log \sigma^2_{\boldsymbol{\theta}}(\boldsymbol{x})}{2} + \frac{(y - \mu_{\boldsymbol{\theta}}(\boldsymbol{x}))^2}{2\sigma^2_{\boldsymbol{\theta}}(\boldsymbol{x})}. \quad (11)$$

We maintain N distinct dynamics models $\Theta = \{\hat{\boldsymbol{\theta}}_n\}_{n=1}^{N}$. For each of the dynamics model $\hat{\boldsymbol{\theta}}_n$, we compute M different policies (i.e., $\Phi = \{\hat{\phi}_{n,m}\}_{n,m=1}{}^{N,M}$) employing the soft actor-critic (SAC) (Haarnoja et al., 2018) method (please refer to the appendix for detailed information). The policy network $\pi_{n,m}$ with parameters $\hat{\phi}_{n,m}$ is updated based on synthetic data $\mathcal{D}^{n,m}_{\mathcal{M}}$, generated by dynamics model n and policy model m, in conjunction with environment data $\mathcal{D}_{\mathcal{E}}$ collected from real-world dynamics interaction with a sampled policy. The pseudo-code is in Algorithm1.

3.3. Sampling policies

Ensemble Sampling. Given the posterior distributions, it remains to specify the sampling approach for policies. The most rudimentary sampling strategy consists of uniform sampling at the beginning of each episode. For our algorithm that employs ensemble sampling, we shall use the term PS-MBPO.

$$\pi \sim \mathcal{U}(\{\pi_{1,1}, ..., \pi_{N,M}\}). \quad (12)$$

In the context of bandits (where $N = 1$, as there is no transition model), this elementary strategy has been demonstrated to achieve a regret of $\tilde{\mathcal{O}}(\sqrt{T} + T\sqrt{A/M})$ (Lu & Roy, 2017) for T steps in Gaussian linear bandits, with M denoting the size of the ensemble and A the number of arms. This regret bound analysis suggests that the regret can be mitigated by adding more ensemble members, although it remains

[2]By mixed dataset $\lambda\mathcal{D}_{\mathcal{E}} + (1-\lambda)\mathcal{D}^{n,m}_{\mathcal{M}}$, we mean that for each data point in the training batch, it is with probability of λ being sampled from the real data $\mathcal{D}_{\mathcal{E}}$ and probability of $1 - \lambda$ from the fictitious data $\mathcal{D}^{n,m}_{\mathcal{M}}$.

Algorithm 1 PSRL with approximate inference using **Ensemble Sampling (PS-MBPO)** or **Optimistic Ensemble Sampling (OPS-MBPO)**.

Require: Initialize an ensemble of dynamics models $\Theta = \{\hat{\boldsymbol{\theta}}_n\}_{n=1}^{N}$ i.i.d. $\sim q(\boldsymbol{\theta})$.
Require: Initialize an ensemble of policy networks $\Phi = \{\hat{\boldsymbol{\phi}}_{n,m}\}_{n,m=1}^{N,M}$ i.i.d. $\sim q(\boldsymbol{\phi})$.
Require: Initialize empty datasets $\mathcal{D}_{\mathcal{E}}$ and $\{\mathcal{D}_{\mathcal{M}}^{n,m}\}_{n,m=1}^{N,M}$. Real data vs. synthetic data ratio λ.
1: **for** K episodes **do**
    ▷ /* Dynamics training. (Line 2) */
2:     Train the ensemble models Θ on $\mathcal{D}_{\mathcal{E}}$ under the objective in equation 11.
    ▷ /* Policy sampling. (Line 3) */
3:     Sample a policy π from Φ **uniformly at random** (equation 12) or based on **the optimistic distribution** (equation 13).
4:     Sample state $\boldsymbol{s}_1$ from the initial state distribution $\rho(\boldsymbol{s})$
5:     **for** $h = 2 : H$ steps **do**
        ▷ /* Data collection. (Lines 6-11) */
6:         $\boldsymbol{s}_h$ = rollout(world dynamics $\mathcal{E}$, policy π, initial state $\boldsymbol{s}_{h-1}$, num. steps 1)
7:         Add $\boldsymbol{s}_h$ to $\mathcal{D}_{\mathcal{E}}$
8:         Sample state $\boldsymbol{s} \sim \mathcal{D}_{\mathcal{E}}$
9:         **for** each model n, policy m **do**
10:             $\mathcal{D}_{\mathcal{M}}^{n,m}$ = rollout(dynamics $\hat{\boldsymbol{\theta}}_n$, policy $\hat{\boldsymbol{\phi}}_{n,m}$, initial state $\boldsymbol{s}$, num. steps R)
11:             Created mixed dataset $D = \lambda \mathcal{D}_{\mathcal{E}} + (1-\lambda)\mathcal{D}_{\mathcal{M}}^{n,m}$[2]
            ▷ /* Policy optimization (Line 12) */
12:             $\hat{\boldsymbol{\phi}}_{n,m}$ = update-policy($\hat{\boldsymbol{\phi}}_{n,m}$, D, num. gradient steps G)
13:         **end for**
14:     **end for**
15:     **Update the optimistic policy distribution** (equation 13).
16: **end for**

uncertain how this theoretical result extrapolates to the RL setting—an intriguing avenue for future research. However, in our experiments, we observe that this aforementioned relationship also applies to RL.

Optimistic Ensemble Sampling. Unfortunately, ensemble sampling may instigate excessive exploration in certain unpromising regions, as it treats each member of the ensemble model equally. This could result in unnecessary or even wasteful explorations. To counter this, we propose an optimistic variant of ensemble sampling, which we term OPS-MBPO. Specifically, we monitor the performance of each ensemble member in terms of the accumulated episodic return. Alternatively, one could also employ the value function for each policy (Agarwal & Zhang, 2022). We then utilize this performance record to determine the probability of selecting each member, thereby gradually phasing out unpromising ensemble members.

More precisely, at the commencement of the k_{th} episode, we sample the policy from the subsequent Boltzmann distribution, as opposed to a uniform random selection.

$$p_k(\pi = \pi_i) := \frac{\exp\left(\sum_{l=1}^{k} R_{\mathcal{E}}(\pi_i, l)/\tau\right)}{\sum_{j=1}^{N\cdot M} \exp\left(\sum_{l=1}^{k} R_{\mathcal{E}}(\pi_j, l)/\tau\right)}, \quad (13)$$

where τ represents the temperature term used for regulating the level of optimism, while $R_{\mathcal{E}}(\pi_i, l)$ refers to the empirical cumulative reward obtained by policy π_i after the l_{th} episode. It is worth noting that when $\tau \to \infty$, the resulting sampling method becomes uniform, which we refer to as PS-MBPO.

4. Experiments

Our empirical investigation aims to: 1) verify the effectiveness of our proposed methodologies on standard benchmarks; 2) offer a profound understanding of the mechanisms instrumental to the improved performance; and 3) conduct supplementary ablation studies on remaining components. We commence by presenting the experimental setup.

4.1. Experimental Setup

We examine seven tasks sourced from OpenAI Gym (Brockman et al., 2016), DeepMind Control Suite (Tunyasuvunakool et al., 2020), and Metaworld (Yu et al., 2020). This set includes four dense reward tasks (namely, `ant`, `half-cheetah`, `walker2d`, and `hopper`), wherein the agent garners an immediate reward at each step. Additionally, we consider three sparse reward tasks (`ball-in-cup`, `cartpole-swingup`, and `window-open-v2`), where the agent is rewarded only upon successful completion of the pertinent task. It's important to note that efficient exploration of the environment is of paramount importance in the case of sparse reward

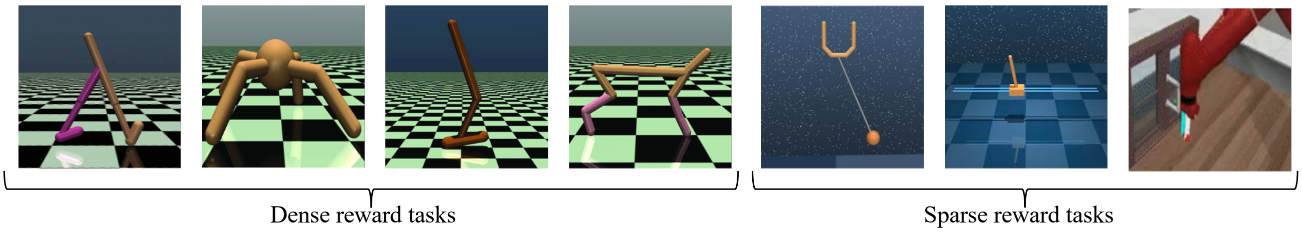

Figure 3: We consider seven tasks from three benchmarks: OpenAI Gym, DM Control and Metaworld. These seven tasks cover both dense reward and sparse reward tasks.

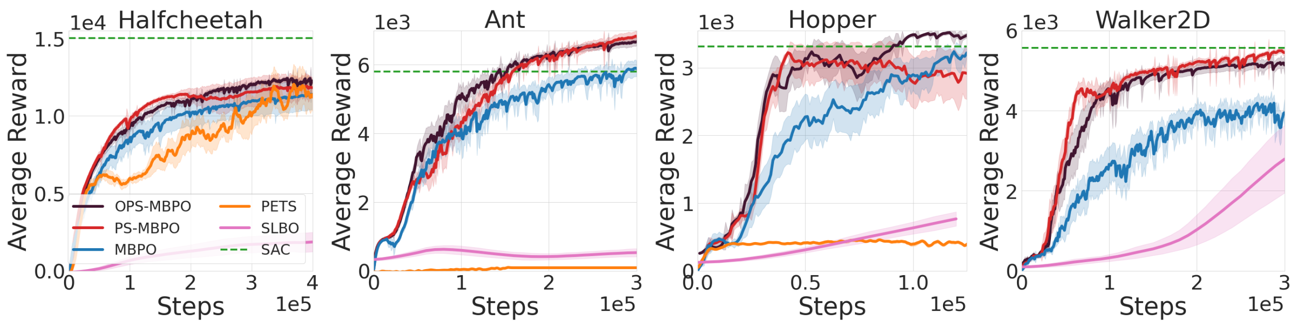

Figure 4: Comparisons on four tasks with dense rewards. The shaded region denotes the one-standard error. The dashed green curve corresponds to the asymptotic performance of SAC at 3M steps. PS-MBPO improves over MBPO across all of the four tasks, and the improvement is more significant on `Ant` and `Walker2D`. OPS-MBPO achieves similar sample efficiency with PS-MBPO.

tasks, more so than in dense reward tasks. For more detailed insights, we refer the reader to Figure 3, which offers visualizations of these tasks.

For the baseline methods, we consider a range of model-based methods including SLBO (Luo et al., 2019), PETS (Chua et al., 2018), and MBPO (Janner et al., 2019), as well as a model-free approach, SAC (Haarnoja et al., 2018). We juxtapose each method with respect to the average episodic reward, where each episode ends either upon reaching the $1,000$ timestep or the agent's arrival at the terminal state. To ensure the robustness of our findings, each experiment is repeated with 10 random seeds, and we report the mean and the standard error. For further details, please refer to Appendix A and C.

4.2. Comparison with existing methods

The results for the dense reward tasks are depicted in Figure 4. Initially, it is noteworthy that our (O)PS-MBPO method surpasses the baseline methods across all four tasks, including the MBPO method. We have verified that our implementation of MBPO either matches or exceeds the performance of the original implementation (please refer to the appendix for further details). Specifically, for the `hopper` task, our approach requires approximately 40K iterations to attain an average reward around $3,500$, in contrast to the MBPO method, which requires around 150K steps to achieve similar performance.

The results for the sparse reward tasks are then provided in Figure 5. Corresponding to the outcomes presented in Figure 4, we initially observe that both PS-MBPO and OPS-MBPO outperform the MBPO method across all three tasks. The magnitude of improvement is notably more pronounced on the `Cartpole-swingup` and `Window-open-v2` tasks. In contrast to the findings depicted in Figure 4, we additionally observe that OPS-MBPO significantly improves upon the performance of PS-MBPO on these three tasks. This observation underscores the potential benefits of integrating an optimistic sampling strategy within sparse reward tasks.

4.3. Ablation Studies

In this section, we conduct a series of ablation studies to better comprehend our proposed method. Additional experiments are available in the appendix, including an ablation study with forced exploration (Phan et al., 2019), random function prior network (Osband et al., 2018) and more.

Does the gain come from posterior sampling or ensemble? To ascertain the impact of posterior sampling, we draw a comparison between two performance conditions: one where a policy is sampled from the posterior at each episode, and another where we employ the

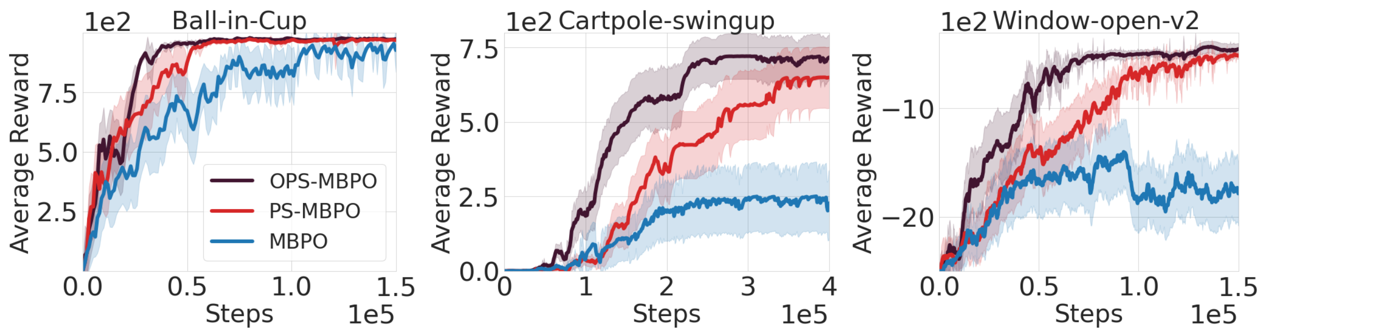

Figure 5: Comparisons on three tasks (`Ball-in-Cup`, `Cartpole-Swingup` and `Window-open-v2`) with sparse rewards. PS-MBPO improves over MBPO, and OPS-MBPO further improves over PS-MBPO in terms of sample efficiency.

average policy, computed by averaging the distribution over actions across all ensemble members. The results, as depicted in Figure 6, utilize $N = 5$ dynamics networks and $M \in \{1, 3, 5\}$ policy networks. A clear observation from the experiment is the significant performance degradation upon disabling the sampling procedure. Furthermore, it's noteworthy that performance improvements are observed as the number of ensemble members increases (i.e., as M increases) when posterior sampling is active. In contrast, when posterior sampling is deactivated, increasing M does not appear to result in performance enhancements. These findings strongly support the premise that the primary factor contributing to improved performance is the application of posterior sampling, rather than simply the use of a larger ensemble for both dynamics and policies.

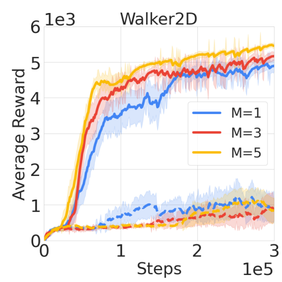

Figure 6: Ablation study on the performance of with (solid curves) and without (dashed curves) the sampling step.

Effect of N and M. Considering our employment of the deep ensemble approximation, it is natural to speculate whether enhanced performance could be realized through the application of a larger ensemble for both dynamics models and policies. In Figure 7, we present the average reward of the final 10 evaluations, each obtained with 10 distinct random seeds. These results utilize N dynamics models and M policy networks for each dynamics model, with the values of N and M varying within the set $\{1, 2, 3, 4, 5\}$. The results clearly indicate that increasing both N and M can yield performance improvements. Furthermore, both forms of uncertainty (pertaining to policies and dynamics) seem to play a significant role.

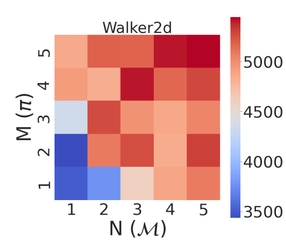

Figure 7: Average reward for varying number of dynamics model (N) and policies (M).

Visualization of the State Space. To gain a more nuanced understanding of the exploration behavior, we project the high-dimensional states of each trajectory gathered by PS-MBPO and MBPO into a two-dimensional space using Umap (McInnes et al., 2018). These visualizations are provided in Figure 8. During the initial phase, it is apparent that PS-MBPO (as represented in the top three figures on the left) demonstrates a more explorative approach than MBPO (top right three figures), which subsequently leads to superior final performance (refer to Figure 5 for details). Additionally, we illustrate two representative trajectories of PS-MBPO and MBPO for a qualitative comparison in Figure 8, corresponding to the same training iterations. Visually, it can be observed that PS-MBPO manages to manipulate the robot arm to cover more diverse regions, whereas MBPO is unable to do so, resulting in the two trajectories appearing markedly similar to each other. This observation is also reflected in the Umap visualization.

5. Related Works

Model-Based Reinforcement Learning (MBRL). The field of MBRL predominantly targets two core concerns: the manner of learning the dynamics model from the available data, and the subsequent usage of the acquired model. The most widespread technique for model learning involves the use of the L_2 loss for one-step transitions (Kurutach et al., 2018; Luo et al., 2019; Chua et al., 2018; Janner et al., 2019; Rajeswaran et al., 2020). This is equivalent to maximum likelihood estimation under a Gaussian assumption. Advancing beyond the scope of one-step training, Hafner et al. (2019); Asadi et al. (2019); Lutter et al. (2021) have demonstrated that multi-step training can further enhance prediction accuracy for long horizons. However, this comes with increased computational costs which scale quadratically with the number of prediction steps.

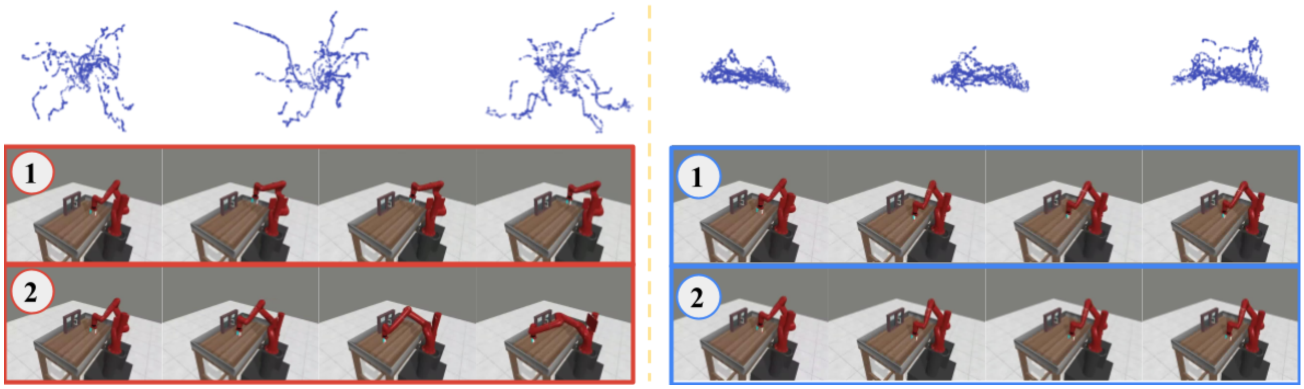

Figure 8: The visualization of the state spaces visited by PS-MBPO (top left) and MBPO (top right) on the `Window-open-v2` during the initial stages of training is provided. We also showcase two illustrative trajectories of PS-MBPO (bottom left) and MBPO (bottom right), which consist of four frames extracted from the corresponding videos.

Simultaneously, a separate line of research has emerged, focusing on the objective mismatch problem in MBRL (Ziebart, 2010; Farahmand et al., 2017; Luo et al., 2019; Lambert et al., 2020; Eysenbach et al., 2022). Here, the objective is to modify the model training goal to provide a performance guarantee for the policy when it operates in the real environment. This helps ensure that the policy is not merely optimized for the model, but remains effective when deployed in the real environment.

To utilize the learned model, several methodologies have been proposed. One common method is Model Predictive Control (MPC) (Camacho & Alba, 2013), a derivative-free optimization technique that has found wide acceptance in numerous preceding works (Nagabandi et al., 2018; Chua et al., 2018; Hafner et al., 2019; Fan & Ming, 2021; Lutter et al., 2021). However, while effective, MPC has some limitations. It is sensitive to the planning horizon and struggles to handle high-dimensional problems. As a mitigation strategy, Kurutach et al. (2018); Luo et al. (2019); Janner et al. (2019) propose training a policy on top of the model to amortize the planning cost. This approach leverages the model for generating synthetic data, which are then used to train a policy using model-free methods, thereby reducing the online sample complexity. Similarly, other research has proposed using the model to facilitate learning of value functions (Feinberg et al., 2018; Buckman et al., 2018), allowing for more efficient policy evaluation and improvement. However, few of them have investigated the tradeoff between exploration and exploitation in policy optimization.

Exploration and exploitation. Handling the exploration and exploitation tradeoff is the central problem in online learning. Typical methods can be categorized into the following three classes: 1) *optimism-based* (Auer et al., 2008; Pacchiano et al., 2021; Curi et al., 2020); 2) *posterior-sampling-based* (Strens, 2000; Osband et al., 2013; Osband & Roy, 2014; Osband et al., 2018; Fan & Ming, 2021); and 3) *information-directed sampling* (Russo & Roy, 2014; Nikolov et al., 2019) approaches. Optimism-based methods needs one to construct the confidence set that contains the target model/policy with high probability, which suffers from scalability issues (Osband & Roy, 2017); in addition this approach empirically performs worse than Thompson sampling (Chapelle & Li, 2011). Information-directed sampling can be better than optimism-based methods and Thompson sampling, as it directly minimizes the "regret per information bit" (Russo & Roy, 2014). However, it relies on estimating the mutual information between random variables, which is especially difficult for high-dimensional continuous random variables (McAllester & Stratos, 2020). Therefore, we focus on posterior sampling. However, different from prior works, we study the effect of approximate inference in an RL setting.

6. Summary and Future Works

We have introduced PS-MBPO and OPS-MBPO as innovative algorithms for efficient model-based reinforcement learning in intricate environments. We demonstrate that both PS-MBPO and OPS-MBPO significantly enhance the sample efficiency in online reinforcement learning and outperform various benchmark methods by a considerable margin, especially in sparse reward tasks. Our hope is that our analysis, beyond our specific approach, will inspire future research to propose advanced factorizations of the posterior over policies and models.

Moving forward, we intend to explore the possibility of automatically adapting the value of λ in an online manner. (See Section B.2 for preliminary results.) Furthermore, we aim to extend the results of Phan et al. (2019) and establish sublinear regret for PSRL under approximate inference. It would also be intriguing to examine epistemic neural networks (Osband et al., 2021) and transformers (Vaswani et al., 2017; Müller et al., 2022) as alternatives to deep ensembles for approximate posterior inference.

Acknowledgments

We thank Sergey Levine and Dale Schuurmans for their feedback on an early draft of this work, as well as the anonymous reviewers for their constructive feedback. Additionally, we express our gratitude to Michael Janner for providing the results of PETS, SLBO, and SAC on the Halfcheetah, Ant, Hopper, and Walker2D. Lastly, we acknowledge the Google TPU Research Cloud for providing the computing resources.

References

Agarwal, A. and Zhang, T. Model-based rl with optimistic posterior sampling: Structural conditions and sample complexity. *ArXiv preprint*, abs/2206.07659, 2022. URL https://arxiv.org/abs/2206.07659.

Agrawal, S. and Jia, R. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 1184–1194, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3621f1454cacf995530ea53652ddf8fb-Abstract.html.

Asadi, K., Misra, D., Kim, S., and Littman, M. L. Combating the compounding-error problem with a multi-step model. *ArXiv preprint*, abs/1905.13320, 2019. URL https://arxiv.org/abs/1905.13320.

Auer, P., Jaksch, T., and Ortner, R. Near-optimal regret bounds for reinforcement learning. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pp. 89–96. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper/2008/hash/e4a6222cdb5b34375400904f03d8e6a5-Abstract.html.

Box, G. E. and Tiao, G. C. *Bayesian inference in statistical analysis*. John Wiley & Sons, 2011.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016. URL https://arxiv.org/abs/1606.01540.

Buckman, J., Hafner, D., Tucker, G., Brevdo, E., and Lee, H. Sample-efficient reinforcement learning with stochastic ensemble value expansion. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8234–8244, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/f02208a057804ee16ac72ff4d3cec53b-Abstract.html.

Camacho, E. F. and Alba, C. B. *Model predictive control*. Springer science & business media, 2013.

Chapelle, O. and Li, L. An empirical evaluation of thompson sampling. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F. C. N., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pp. 2249–2257, 2011. URL https://proceedings.neurips.cc/paper/2011/hash/e53a0a2978c28872a4505bdb51db06dc-Abstract.html.

Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 4759–4770, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/3de568f8597b94bda53149c7d7f5958c-Abstract.html.

Curi, S., Berkenkamp, F., and Krause, A. Efficient model-based reinforcement learning through optimistic policy search and planning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/a36b598abb934e4528412e5a2127b931-Abstract.html.

Eysenbach, B., Khazatsky, A., Levine, S., and Salakhutdinov, R. R. Mismatched no more: Joint model-policy optimization for model-based rl. *Advances in Neural Information Processing Systems*, 35:23230–23243, 2022.

Fan, Y. and Ming, Y. 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, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3078–3087. PMLR, 2021. URL http://proceedings.mlr.press/v139/fan21b.html.

Farahmand, A. M., Barreto, A., and Nikovski, D. Value-aware loss function for model-based reinforcement learning. In Singh, A. and Zhu, X. J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1486–1494. PMLR, 2017. URL http://proceedings.mlr.press/v54/farahmand17a.html.

Feinberg, V., Wan, A., Stoica, I., Jordan, M. I., Gonzalez, J. E., and Levine, S. Model-based value estimation for efficient model-free reinforcement learning. *ArXiv preprint*, abs/1803.00101, 2018. URL https://arxiv.org/abs/1803.00101.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. *Bayesian data analysis*. CRC press, 2013.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1856–1865. PMLR, 2018. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

Hafner, D., Lillicrap, T. P., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019. URL http://proceedings.mlr.press/v97/hafner19a.html.

Hoffman, M., Shahriari, B., Aslanides, J., Barth-Maron, G., Behbahani, F., Norman, T., Abdolmaleki, A., Cassirer, A., Yang, F., Baumli, K., Henderson, S., Novikov, A., Colmenarejo, S. G., Cabi, S., Gulcehre, C., Paine, T. L., Cowie, A., Wang, Z., Piot, B., and de Freitas, N. Acme: A research framework for distributed reinforcement learning. *ArXiv preprint*, abs/2006.00979, 2020. URL https://arxiv.org/abs/2006.00979.

Janner, M., Fu, J., Zhang, M., and Levine, S. When to trust your model: Model-based policy optimization. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 12498–12509, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5faf461eff3099671ad63c6f3f094f7f-Abstract.html.

Kurutach, T., Clavera, I., Duan, Y., Tamar, A., and Abbeel, P. Model-ensemble trust-region policy optimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=SJJinbWRZ.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6402–6413, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html.

Lambert, N., Amos, B., Yadan, O., and Calandra, R. Objective mismatch in model-based reinforcement learning. *ArXiv preprint*, abs/2002.04523, 2020. URL https://arxiv.org/abs/2002.04523.

Lu, X. and Roy, B. V. Ensemble sampling. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 3258–3266, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

Lu, X., Van Roy, B., Dwaracherla, V., Ibrahimi, M., Osband, I., and Wen, Z. Reinforcement learning, bit by bit. *ArXiv preprint*, abs/2103.04047, 2021. URL https://arxiv.org/abs/2103.04047.

Luo, Y., Xu, H., Li, Y., Tian, Y., Darrell, T., and Ma, T. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=BJe1E2R5KX.

Lutter, M., Hasenclever, L., Byravan, A., Dulac-Arnold, G., Trochim, P., Heess, N., Merel, J., and Tassa, Y. Learning dynamics models for model predictive agents. *ArXiv preprint*, abs/2109.14311, 2021. URL https://arxiv.org/abs/2109.14311.

McAllester, D. and Stratos, K. Formal limitations on the measurement of mutual information. In Chiappa, S. and Calandra, R. (eds.), *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pp. 875–884. PMLR, 2020. URL http://proceedings.mlr.press/v108/mcallester20a.html.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *ArXiv preprint*, abs/1802.03426, 2018. URL https://arxiv.org/abs/1802.03426.

Müller, S., Hollmann, N., Pineda-Arango, S., Grabocka, J., and Hutter, F. Transformers can do bayesian inference. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=KSugKcbNf9.

Murphy, K. P. *Probabilistic machine learning: an introduction*. MIT press, 2022.

Nagabandi, A., Kahn, G., Fearing, R. S., and Levine, S. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7559–7566. IEEE, 2018.

Nikolov, N., Kirschner, J., Berkenkamp, F., and Krause, A. Information-directed exploration for deep reinforcement learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Byx83s09Km.

Osband, I. and Roy, B. V. Model-based reinforcement learning and the eluder dimension. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 1466–1474, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/1141938ba2c2b13f5505d7c424ebae5f-Abstract.html.

Osband, I. and Roy, B. V. 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, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2701–2710. PMLR, 2017. URL http://proceedings.mlr.press/v70/osband17a.html.

Osband, I., Russo, D., and Roy, B. V. (more) efficient reinforcement learning via posterior sampling. In Burges, C. J. C., Bottou, L., Ghahramani, Z., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pp. 3003–3011, 2013. URL https://proceedings.neurips.cc/paper/2013/hash/6a5889bb0190d0211a991f47bb19a777-Abstract.html.

Osband, I., Aslanides, J., and Cassirer, A. Randomized prior functions for deep reinforcement learning. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8626–8638, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/5a7b238ba0f6502e5d6be14424b20ded-Abstract.html.

Osband, I., Wen, Z., Asghari, M., Ibrahimi, M., Lu, X., and Van Roy, B. Epistemic neural networks. *ArXiv preprint*, abs/2107.08924, 2021. URL https://arxiv.org/abs/2107.08924.

Pacchiano, A., Ball, P. J., Parker-Holder, J., Choromanski, K., and Roberts, S. Towards tractable optimism in model-based reinforcement learning. In de Campos, C. P., Maathuis, M. H., and Quaeghebeur, E. (eds.), *Proceedings of the Thirty-Seventh Conference on Uncer-*

tainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021, volume 161 of *Proceedings of Machine Learning Research*, pp. 1413–1423. AUAI Press, 2021. URL https://proceedings.mlr.press/v161/pacchiano21a.html.

Phan, M., Abbasi-Yadkori, Y., and Domke, J. Thompson sampling and approximate inference. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8801–8811, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/f3507289cfdc8c9ae93f4098111a13f9-Abstract.html.

Pineda, L., Amos, B., Zhang, A., Lambert, N. O., and Calandra, R. Mbrl-lib: A modular library for model-based reinforcement learning. *Arxiv*, 2021.

Rajeswaran, A., Mordatch, I., and Kumar, V. A game theoretic framework for model based reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7953–7963. PMLR, 2020. URL http://proceedings.mlr.press/v119/rajeswaran20a.html.

Russo, D. and Roy, B. V. Learning to optimize via information-directed sampling. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 1583–1591, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/301ad0e3bd5cb1627a2044908a42fdc2-Abstract.html.

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

Strens, M. J. A. A bayesian framework for reinforcement learning. In Langley, P. (ed.), *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pp. 943–950. Morgan Kaufmann, 2000.

Tiapkin, D., Belomestny, D., Calandriello, D., Moulines, E., Munos, R., Naumov, A., Rowland, M., Valko, M., and MENARD, P. Optimistic posterior sampling for reinforcement learning with few samples and tight guarantees. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022a.

Tiapkin, D., Belomestny, D., Moulines, E., Naumov, A., Samsonov, S., Tang, Y., Valko, M., and Ménard, P. From dirichlet to rubin: Optimistic exploration in RL without bonuses. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 21380–21431. PMLR, 2022b. URL https://proceedings.mlr.press/v162/tiapkin22a.html.

Tunyasuvunakool, S., Muldal, A., Doron, Y., Liu, S., Bohez, S., Merel, J., Erez, T., imothy Lillicrap, Heess, N., and Tassa, Y. dm_control: Software and tasks for continuous control. *Software Impacts*, 6, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Ziebart, B. D. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

Contents

A. Extended Discussions

A.1. Dynamics model

We use deep ensemble for fitting the environment dynamics. For each network in the ensemble $f_{\boldsymbol{\theta}}$, it takes a whitened state and action pair as input, and predicts the residual of the next state as well as the reward, i.e.,

$$f_{\boldsymbol{\theta}}\left(\frac{\boldsymbol{s}_t-\boldsymbol{\mu}_s}{\boldsymbol{\sigma}_s},\frac{\boldsymbol{a}_t-\boldsymbol{\mu}_a}{\boldsymbol{\sigma}_a}\right)=\mathcal{N}\left(\begin{bmatrix}\Delta\boldsymbol{s}_t\\ r(\boldsymbol{s}_t,\boldsymbol{a}_t)\end{bmatrix},\begin{bmatrix}\mathrm{diag}(\boldsymbol{\sigma}^2_{\Delta\boldsymbol{s}_t}), & \mathbf{0}\\ \mathbf{0}, & \sigma^2_{r_t}\end{bmatrix}\right), \tag{14}$$

where $\boldsymbol{\mu}_s$, $\boldsymbol{\mu}_a$ are the empirical mean of the states and actions, $\boldsymbol{\sigma}_s$ and $\boldsymbol{\sigma}_a$ are the empirical standard deviation of them. Then, the predictions of next state and reward will be

$$\begin{bmatrix}\boldsymbol{s}_{t+1}\\ r_t\end{bmatrix}\sim\mathcal{N}\left(\begin{bmatrix}\boldsymbol{s}_t+\Delta\boldsymbol{s}_t\\ r(\boldsymbol{s}_t,\boldsymbol{a}_t)\end{bmatrix},\begin{bmatrix}\mathrm{diag}(\boldsymbol{\sigma}^2_{\Delta\boldsymbol{s}_t}), & \mathbf{0}\\ \mathbf{0}, & \sigma^2_{r_t}\end{bmatrix}\right). \tag{15}$$

Below is our implementation of each individual neural network in JAX (Bradbury et al., 2018).

```
class GaussianMLP(hk.Module):
  """MLP with Gaussian distribution outputs."""

  def __init__(
      self,
      output_size: int,
      hidden_sizes: Sequence[int],
      *,
      activation=jax.nn.swish,
      min_logvar: float = -10.0,
      max_logvar: float = 2.0,
      name: Optional[str] = None,
  ):
    super().__init__(name=name)
    self.output_size = output_size
    w_init = hk.initializers.VarianceScaling(1.0, 'fan_in', 'truncated_normal')
    self.mlp = hk.nets.MLP(
        hidden_sizes, w_init=w_init, activation=activation, activate_final=True)
    self.min_logvar = jnp.ones(output_size) * min_logvar
    self.max_logvar = jnp.ones(output_size) * max_logvar
    self.mean_and_logvar = hk.Linear(
        self.output_size * 2, w_init=w_init, name='mean_and_logvar')

  def __call__(self, x):
    h = self.mlp(x)
    mean, logvar = jnp.split(self.mean_and_logvar(h), 2, axis=-1)
    logvar = self.max_logvar - jax.nn.softplus(self.max_logvar - logvar)
    logvar = self.min_logvar + jax.nn.softplus(logvar - self.min_logvar)
    return mean, logvar
```

A.2. Soft actor-critic (SAC)

We use SAC for learning the policies. In a highlevel, SAC is a maximum entropy RL algorithm, which typically optimizes the following objective,

$$J(\pi)=\sum_{t=0}^{T}\mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim\rho_\pi}[r(\boldsymbol{s},\boldsymbol{a})+\alpha\mathbb{H}(\pi(\cdot|\boldsymbol{s}))]. \tag{16}$$

As a result, maximum entropy RL algorithm will favor those policies that not only optimize for the reward, but also has a large entropy. This can in turn improve the robustness of the optimized policy. As for SAC, it searches the policy by iteratively solving the policy evaluation and policy improvement steps.

$$\text{Policy Evaluation: } Q^{\pi_t}(\boldsymbol{s},\boldsymbol{a}) \leftarrow r(\boldsymbol{s},\boldsymbol{a}) + \gamma \mathbb{E}_{\boldsymbol{s}'\sim p(\cdot|\boldsymbol{s},\boldsymbol{a})}[V^{\pi_t}(\boldsymbol{s}')], \tag{17}$$

$$V^{\pi_t}(\boldsymbol{s}) = \mathbb{E}_{\boldsymbol{a}\sim\pi_t(\cdot|\boldsymbol{s})}[Q^{\pi_t}(\boldsymbol{s},\boldsymbol{a}) - \log \pi_t(\boldsymbol{a}|\boldsymbol{s})]; \tag{18}$$

$$\text{Policy Improvement: } \pi_{t+1} \leftarrow \arg\min_{\pi} \mathbf{d}_{\mathrm{KL}}\left(\pi(\cdot|\boldsymbol{s}_0) | \exp(Q^{\pi_t}(\boldsymbol{s}_0,\cdot))\right). \tag{19}$$

In the practical implementation of SAC, it uses a separate function approximator for the state value to stabilize the training. Specifically, there are three components in SAC, a parameterized state value function $V_{\boldsymbol{\psi}}(\boldsymbol{s})$, a soft Q-function $Q_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a})$, and a policy $\pi_{\boldsymbol{\phi}}(\boldsymbol{a}|\boldsymbol{s})$. The objectives for each component are

$$J_V(\boldsymbol{\psi}) = \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}}\left[\frac{1}{2}\left(V_{\boldsymbol{\psi}}(\boldsymbol{s}) - \mathbb{E}_{\boldsymbol{a}\sim\pi_{\boldsymbol{\phi}}(\cdot|\boldsymbol{s})}[Q^{\pi_\phi}(\boldsymbol{s},\boldsymbol{a}) - \log \pi_{\boldsymbol{\phi}}(\boldsymbol{a}|\boldsymbol{s})]\right)^2\right], \tag{20}$$

$$J_Q(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a}) - \hat{Q}(\boldsymbol{s},\boldsymbol{a})\right)^2\right], \tag{21}$$

$$J_\pi(\boldsymbol{\phi}) = \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}}\left[\mathbf{d}_{\mathrm{KL}}\left(\pi_{\boldsymbol{\phi}}(\cdot|\boldsymbol{s}) | \exp(Q_{\boldsymbol{\theta}}(\boldsymbol{s},\cdot))\right)\right], \tag{22}$$

where $Z_{\boldsymbol{\theta}}(\cdot)$ is a normalizing constant, and $\hat{Q}(\boldsymbol{s},\boldsymbol{a})$ is defined as

$$\hat{Q}(\boldsymbol{s},\boldsymbol{a}) \coloneqq r(\boldsymbol{s},\boldsymbol{a}) + \gamma \mathbb{E}_{\boldsymbol{s}'\sim p(\cdot|\boldsymbol{s},\boldsymbol{a})}\left[V_{\bar{\boldsymbol{\psi}}}(\boldsymbol{s}')\right]. \tag{23}$$

Additionally, $\bar{\boldsymbol{\psi}}$ is the exponentially moving average of the weights of the value network, and $J_\pi(\boldsymbol{\phi})$ can be optimized with reparameterization trick under Gaussian case, which can further reduces the variance of the gradient estimator and hence stabilizes the training. We adopt the SAC implementation from Acme (Hoffman et al., 2020).

A.3. Revisiting MBPO, PETS and ME-TRPO

Popular model-based reinforcement learning algorithms such as ME-TRPO (Kurutach et al., 2018), PETs (Chua et al., 2018) and MBPO (Janner et al., 2019) typically repeat the following three steps: 1) train a dynamics model (or an ensemble of models) $q(\mathcal{M}|\mathcal{D}_\mathcal{E})$; 2) train/extract a policy $\pi^\star$ from the learned model; 3) collect data from the environment with the policy. Consequently, their policy is (approximately) equivalent to the one obtained by solving

$$\pi^\star = \arg\max_{\pi\in\Pi} \mathbb{E}_\mathcal{M}[R_\mathcal{M}(\pi)] = \arg\max_{\pi\in\Pi} \int R_\mathcal{M}(\pi) q(\mathcal{M}|\mathcal{D}_\mathcal{E}) d\mathcal{M}, \tag{24}$$

where the posterior of the model $\mathcal{M}$ is approximated by an ensemble of neural networks, Π is the search space of policies, and the cumulative reward $R_\mathcal{M}(\pi)$ for an episode of length H of a policy π under dynamics model $\mathcal{M}$ is defined as

$$R_\mathcal{M}(\pi) = \mathbb{E}\left[\sum_{t=1}^{H} r_\mathcal{M}(\boldsymbol{s}_t,\boldsymbol{a}_t)\right] \quad \text{where} \quad \boldsymbol{s}_{t+1} \sim p_\mathcal{M}(\boldsymbol{s}|\boldsymbol{s}_t,\boldsymbol{a}_t) \text{ and } \boldsymbol{a}_t \sim \pi(\boldsymbol{a}|\boldsymbol{s}_t). \tag{25}$$

However, the above strategy only accounts for exploitation, so will lead to low data efficiency.

A.4. Posterior sampling reinforcement learning

The idea of PSRL is introduced by Strens (2000). The first regret bound $\tilde{\mathcal{O}}(HS\sqrt{AT})$ for PSRL is proved by Osband et al. (2013) for a tabular case with S, A, T, H denotes the number of state, number of actions, number of time steps, and the length of each episode, respectively. In Osband & Roy (2017), the bound was improved to $\tilde{\mathcal{O}}(H\sqrt{SAT})$. For the continuous case, Osband & Roy (2014) provides the first regret bound $\tilde{\mathcal{O}}(\sqrt{d_K d_E T})$ based on the eluder dimension d_E and Kolmogorov dimension d_K. More recently, Fan & Ming (2021) study the regret bound for PSRL under Gaussian process assumption, and obtain a regret bound of $\tilde{\mathcal{O}}(H^{3/2}d\sqrt{T})$. In addition to the bound on Bayesian regret, there is also a line of works studying the worst-case or frequentist regret bound for PSRL (Agrawal & Jia, 2017; Tiapkin et al., 2022b;a), and achieve a regret bound of order $\tilde{\mathcal{O}}(\sqrt{T})$. Nevertheless, all these regret bounds are derived under exact Thompson sampling.

A.5. Limitations of our approach

Our methodology hinges on the employment of an ensemble technique to approximate the posterior for both the dynamic models and the policies. Despite the ensemble method being lauded as one of the most efficacious methods for approximating the posterior distribution in practical scenarios, it may potentially necessitate extensive computational resources. Furthermore, while our algorithm might be assured to outperform traditional methodologies (i.e., q^{δ}), it remains ambiguous whether sublinear regret can be attained under approximate inference, and how the inference error is influenced by the ensemble size in reinforcement learning. These facets constitute the primary constraints of our research and provide intriguing prospects for future exploration.

B. Additional Results for (O)PS-MBPO

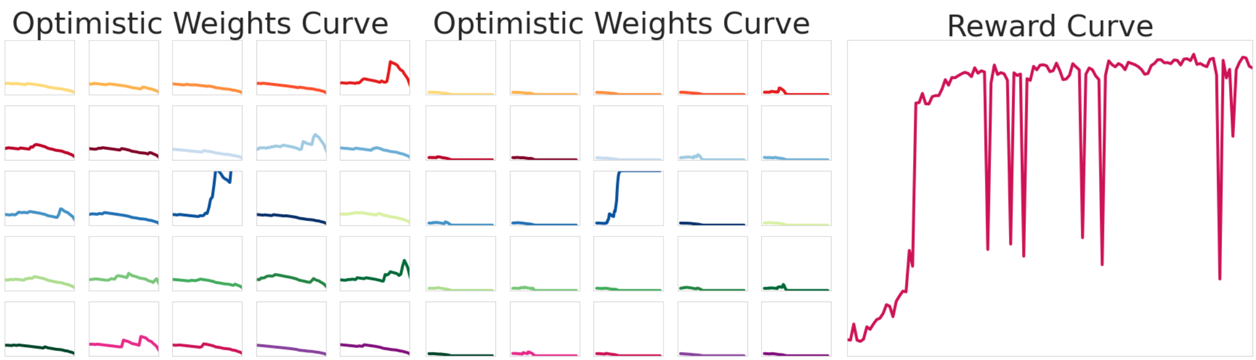

Figure 9: Visualization of the optimistic weights of the first 40K iterations (left) and during the entire training process (middle), and the reward curve (right) on `Hopper`. Each chart in the left and middle figures corresponds to the weights of each single policy.

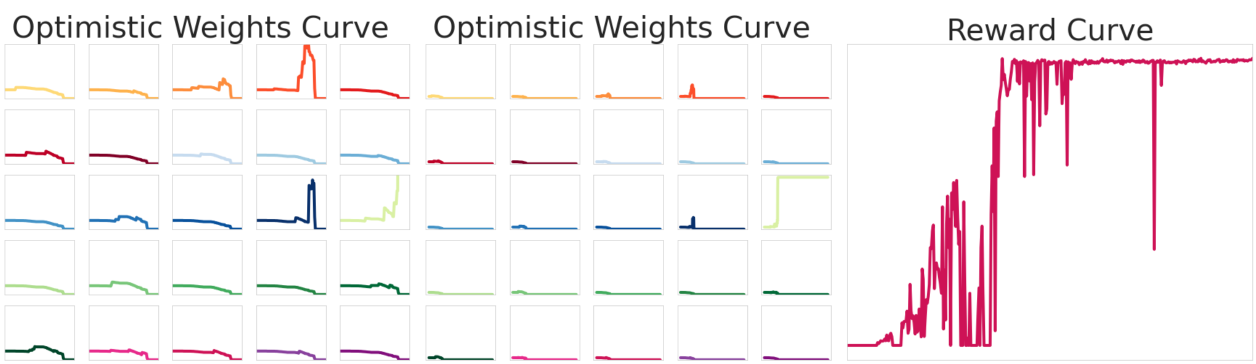

Figure 10: Visualization of the optimistic weights of the first 100K iterations (left) and during the entire training process (middle), and the reward curve (right) on `Cartpole-Swingup`. Each chart in the left and middle figures corresponds to the weights of each single policy.

B.1. Visualization of the optimistic weights

For OPS-MBPO, we will maintain the weights for each policy throughout the entire process of online learning. To investigate how those weights evolve, we plot the weights of each policy in Figure 9 and Figure 10, which covers one dense reward task and one sparse reward task. The leftmost figure corresponds to the weights in the initial phase, which will change more rapidly than the later phases. We observe that the weights of some polices will first go up and than go down, and finally it will converge to a single policy. More interestingly, the reward curve in the rightmost figure is also consistent with the pattern in the optimistic weights curve.

B.2. How does the temperature term τ affect the performance?

We further study how does the temperature term will affect the performance on both dense reward and sparse reward tasks. The results can be found in Figure 11 and Figure 12. We observe that the temperature term will affect the convergence speed of the reward on most of the tasks. For some of the tasks, such as `Ant`, `Hopper` and `Walker2d`, it will also affect the converged reward slightly. In general, we recommend the temperature term to be around five times of the best averaged

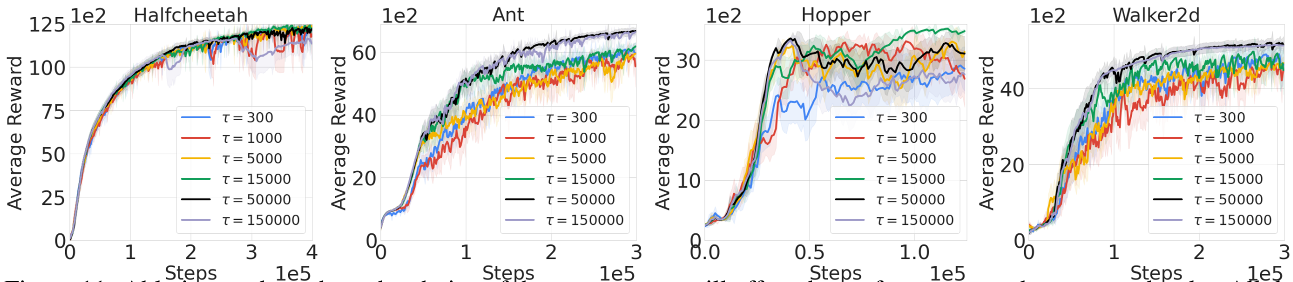

Figure 11: Ablation study on how the choice of the temperature will affect the performance on dense reward tasks. All the experimental setups are the same as those experiments in our main paper, except for the temperature term.

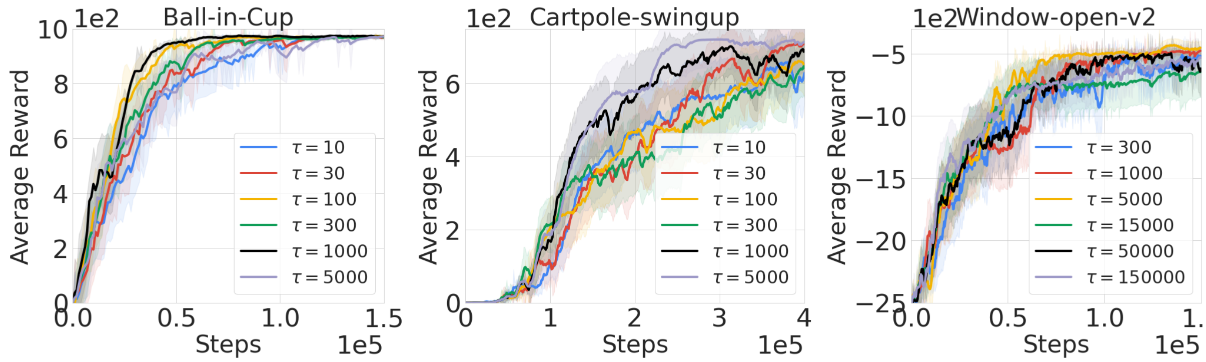

Figure 12: Ablation study on how the choice of the temperature will affect the performance on sparse reward tasks. All the experimental setups are the same as those experiments in our main paper, except for the temperature term.

episodic reward that can be achieved.

B.3. How does the schedule of λ affect the performance?

Since λ plays a role in balancing the effect of approximate inference error and data efficiency, we are interested in how different schedules of λ will affect the reward curve. We consider two schemes for adjusting λ, i.e., 1) a constant schedule; and 2) a linear schedule. For constant schedule, we fix the value of λ throughout training, whereas for linear schedule, we decrease the value of λ from 1 to 0 linearly. The rightmost figure in Figure 13 visualizes the difference between these two schedules. To make them comparable, we ensure that the areas under both curves are of the same size, so that the total amount of real-world data is the same. The comparison on three tasks are shown in the left three figures of Figure 13. We observe that the linear schedule has very little effect on the dense reward tasks, though slightly improves the the final reward in `Hopper`. For `Cartpole-swingup`, the performance of linear schedule improves faster than constant schedule, but achieves similar rewards in the end. Nevertheless, we believe that there might be more sophisticated schedules for λ that can achieve better performance than the constant schedule, e.g., adapting the value of λ based on the model's validation loss. For simplicity, we recommend to use a constant schedule in practice. For sparse reward tasks, the search range of λ can be $\{0, 0.1, 0.3, 0.5, 0.7\}$, and $\{0, 0.05, 0.15, 0.3\}$ for dense reward tasks.

In addition to the grid search, we believe it's also possible to adapt the value of λ online. We can cast the the problem of choosing the optimal value of λ as a bandit problem. The high-level idea of the algorithm is: 1) Initialize a set of possible values for λ, and treat each value of λ as an arm in bandit. 2) Apply any no-regret learning algorithms for solving it, e.g., explore-then-commit. However, we haven't test this algorithm yet, and it would be interesting as a future extension.

C. Additional Details about the Algorithm and Experiments

C.1. Algorithm Details

In Algorithm 1, we approximate the posterior of MDPs and policies, i.e., $q(\mathcal{M}|\mathcal{D}_\mathcal{E})$ and $q(\pi|\mathcal{M}, \mathcal{D}_\mathcal{E}, \lambda)$ using deep ensemble, which can be regarded as a finite particle approximation to the posterior. Specifically, $q(\mathcal{M}|\mathcal{D}_\mathcal{E})$ is approximated by $\{\mathcal{M}_{\hat{\boldsymbol{\theta}}_n}\}_{n=1}^N$ and $q(\pi|\mathcal{M}_{\hat{\boldsymbol{\theta}}_n}, \mathcal{D}_\mathcal{E}, \lambda)$ is approximated by $\{\pi_{\hat{\boldsymbol{\phi}}_{n,m}}\}_{m=1}^M$ for all $n \in [N]$, where both $\mathcal{M}_{\hat{\boldsymbol{\theta}}_n}$ and $\pi_{\hat{\boldsymbol{\phi}}_{n,m}}$ are

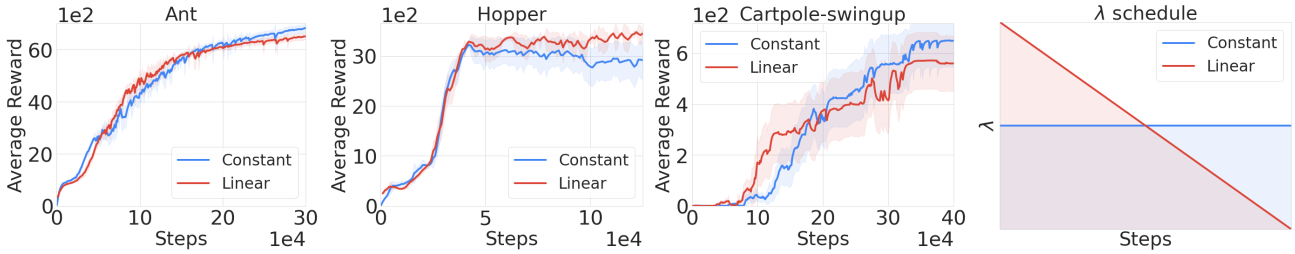

Figure 13: Ablation study on how the schedule of λ affect the performance with PS-MBPO. All the experimental setups are the same as the experiments in main paper.

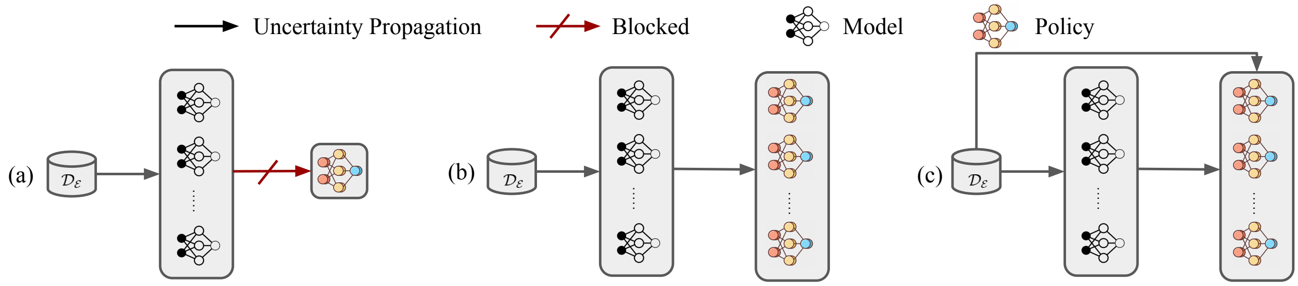

Figure 14: An illustration of the differences between (a) MBPO, (b) PS-MBPO with $\lambda = 0$ and (c) PS-MBPO with $\lambda \in (0, 1)$. MBPO adopt a point estimation to the policy, which is obtained by MAP inference. Thus, the uncertainty propagation from the dynamics model to the policy is blocked. For (b), it implicit assumes the dynamics model captures all the relevant properties of the data. For (c), we add a short cut from data directly to the policy, which is controlled by the value of λ. Hence, the policy can further utilize the information in the data that are not captured in the dynamics model.

implemented using a multi-layer perceptron (MLP) with parameters $\hat{\boldsymbol{\theta}}_n$ trained on $\mathcal{D}_\mathcal{E}$ and $\hat{\phi}_{n,m}$ trained on the mixed dataset $\lambda\mathcal{D}_\mathcal{E} + (1-\lambda)\mathcal{D}_\mathcal{M}^{n,m}$, respectively. By mixed dataset $\lambda\mathcal{D}_\mathcal{E} + (1-\lambda)\mathcal{D}_\mathcal{M}^{n,m}$, we mean that for each data point in the training batch, it is with probability of λ being sampled from the real data $\mathcal{D}_\mathcal{E}$ and probability of $1-\lambda$ from the fictitious data $\mathcal{D}_\mathcal{M}^{n,m}$.

C.2. Implementation Details

In this section, we provide the additional details about our algorithm and experiments. We provide a detailed description of our approach in Algorithm 1 and a visual illustration about its difference with MBPO in Figure 14. In terms of the hyperparameters, our choice of them are mostly the same as the ones adopted in MBPO (Janner et al., 2019) and Pineda et al. (2021) for `Ant`, `Halfcheetah`, `Hopper`, `Walker2D` and `Cartpole-swingup`, and Eysenbach et al. (2022) for `Window-open-v2`, which are sufficiently optimized by the authors for MBPO. Specifically, the hyperparameters of MBPO are directly adopted from `https://github.com/facebookresearch/mbrl-lib` for dense reward tasks. For sparse reward tasks, the hyperparameters are adopted from Eysenbach et al. (2022). The hyperparameters for our method on each task are reported in Table 1. We will also release our code for reproducing all the experiments. Next, we introduce the details about each tasks.

C.3. Task Details

Ant, Halfcheetah, Hopper and Walker2D. These tasks are taken from the official Github repository of MBPO (Janner et al., 2019), `https://github.com/jannerm/mbpo`.

Ball-in-Cup and Cartpole-swingup. These tasks are taken from the deepmind control suite (Tunyasuvunakool et al., 2020). More details can be found in the Github repository at `https://github.com/deepmind/dm_control/`.

Window-open-v2. This task is based on the original `Window-open-v2` in Metaworld benchmakr (Yu et al., 2020). The sparse reward is 0 only if the window is within 3 units of the open position, and -10 for all other positions.

Algorithm 2 PS-MBPO (abstract formulation)

Require: Prior distributions $q(\mathcal{M})$, $q(\pi)$ and tuning hyperparameter λ.
Require: Initialize an empty dataset $\mathcal{D}_\mathcal{E}$ for storing data collected from the environment.
for K episodes **do**
  Fit the posterior of the policy $q(\pi|\mathcal{D}_\mathcal{E}, \lambda)$ on data $\mathcal{D}_\mathcal{E}$ using equation 8.
  Sample a policy $\pi^k \sim q(\pi|\mathcal{D}_\mathcal{E}, \lambda)$ from the posterior distribution.
  for H steps **do**
    Run the policy π^k in the environment and add the collected data to $\mathcal{D}_\mathcal{E}$.
  end for
end for

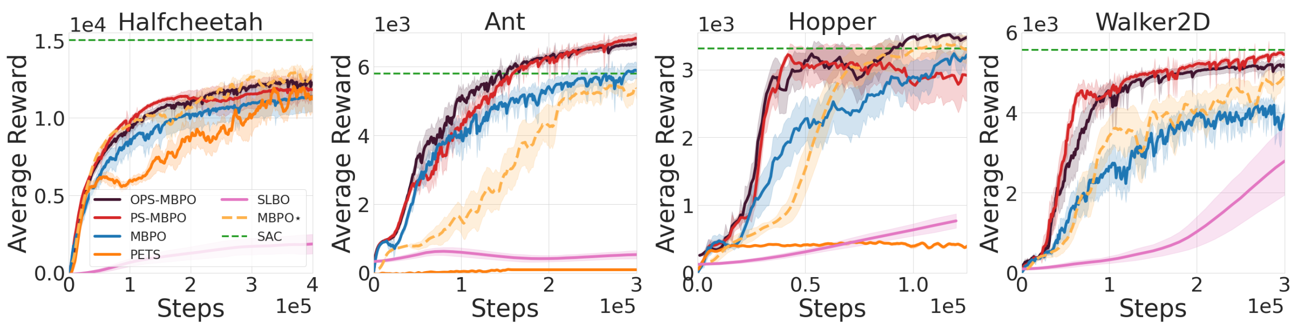

Figure 15: Comparisons on four tasks with dense rewards including `Halfcheetah`, `Ant`, `Hopper` and `Walker2D`. MBPO* is the curve from the original paper by Janner et al. (2019). To be noted, in the original implementation of MBPO, they use 7 networks for the ensemble of dynamics model, whereas our implementation only uses 5 networks. But still, our implementation mostly reproduces their results and sometimes is even better.

D. Forced Exploration

Forced exploration is proposed in Phan et al. (2019) to improve approximate Thompson sampling for bandit problems. Without properly dealing with the approximate inference error, there will be an extra term in the regret that is linear in T, regardless how small the error is. In their paper, they use the α-divergence for measuring the approximate inference error, defined as

$$D_\alpha(P, Q) = \frac{1 - \int p(x)^\alpha q(x)^{1-\alpha} dx}{\alpha(1-\alpha)}. \tag{26}$$

The α-divergence can capture many divergences, including forward KL divergence ($\alpha \to 1$), backward KL divergence ($\alpha \to 0$), Hellinger distance ($\alpha = 0.5$) and χ^2 divergence ($\alpha = 2$). Different inference methods will give error guarantee measured by α-divergence with different α.

We are interested in the error guarantee under the reverse KL-divergence, i.e., $\alpha = 0$, as the ensemble sampling (Lu & Roy, 2017) provides error guarantees under the reverse Kl-divergence. In Phan et al. (2019), they prove that forced exploration can make the posterior concentrate and hence restore the sub-linear regret bound, if the inference error is bounded by α-divergence with $\alpha \leq 0$. The reverse KL-divergence falls in this case. Specifically, the forced exploration is a simple method, that maintains a probability of random exploration. This probability decays as t, the online steps, grows.

Though the above results only hold for bandit setting and it's unclear for reinforcement learning, we are interested in testing its empirical performance in RL. In our experiments, we consider the following exploration rate

$$p_k(\text{random explore=True}) = \text{Bern}(\tau/k), \tag{27}$$

where k is the index for the episode, and τ is the hyperparameter for controlling the frequency of forced exploration. As k increases, the random exploration probability will decrease. In our experiments, we consider $\tau \in \{1, 5, 10\}$. All the other settings are the same as our experiments in the main paper. The results are presented in Figure 16. We observe that forced exploration is mostly not helpful in our experiments, except for the `Hopper` task. Moreover, increasing τ usually make the performance even worse. On the other hand, this may not be so surprising as the forced exploration is designed for

Table 1: Hyperparameters for each task. $x \to y$ over episodes $a \to b$ denotes a segment linear function, $f(k) = \lfloor \min(\max(x + (k-a)/(b-a) \cdot (y-x), x), y) \rfloor$. We use `Ball`, `Cart`, `Cheetah`, `Walker` and `Window` as abbreviations for `Ball-in-Cup`, `Cartpole-Swingup`, `Halfcheetah`, `Walker2D` and `Window-open-v2` so as to make the table fit in the page.

Hyper-parameter	`Ant`	`Ball`	`Cart`	`Cheetah`	`Hopper`	`Walker`	`Window`
Replay buffer capacity	10^6	10^6	10^6	10^6	10^6	10^6	10^6
Episode length	1000	1000	1000	1000	1000	1000	250
Number of episodes	300	150	400	300	125	300	600
Batch size for model	256	256	256	256	256	256	256
Model update frequency	250	250	250	250	250	250	250
Model Hidden dim.	200	200	200	200	200	200	200
Model #Hidden layers	4	4	4	4	4	4	4
Learning rate for model	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
Weight decay for model	10^{-4}	5×10^{-5}	5×10^{-5}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
#Model ensemble (N)	5	5	5	5	5	5	5
Validation ratio	20%	20%	20%	20%	20%	20%	20%
Rollout batch size	10^5	10^5	10^5	10^5	10^5	10^5	5×10^4
Model horizon	$1 \to 25$ over episodes $20 \to 100$	1	1	1	$1 \to 15$ over episodes $20 \to 100$	1	1
Batch size for policy	256	256	256	256	256	256	256
Learning rate for policy	3e-4	3e-4	3e-4	3e-4	3e-4	3e-4	3e-4
#Policy per model (M)	5	5	5	5	5	5	5
Discount (γ)	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Target entropy	-4	-0.05	-0.05	-3	-1	-3	-1
Policy update frequency	1	1	1	1	1	1	1
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam	Adam
λ	0.05	0.5	0.5	0.05	0.05	0.05	0.5
τ for OPS	50000	1000	5000	15000	15000	50000	5000

approximate Thompson sampling in the bandit setting, and the result may not necessary generalize to the RL setting. We leave the theoretical analysis as a future work.

E. Random Function Prior

The random function prior (RFP) is proposed in Osband et al. (2018) for improving the uncertainty estimation. The prior network are chosen for modelling the uncertainty that does not come from the observed data. The RFPs can also be viewed as a regularization in the output space. In contrast to weight space regularization, RFP makes it easier to incorporate different property (e.g., periodicity) of the function to be learned as a prior information. More importantly, when using deep ensemble, incorporating the RFP is fairly simple. It modifies the original training objective $\ell(f_{\boldsymbol{\theta}}, \mathcal{D})$ by adding an extra regularization term,

$$\ell^{RFP}(f_{\boldsymbol{\theta}}, \mathcal{D}) := \ell(f_{\boldsymbol{\theta}} + \beta f_{\boldsymbol{\theta}_0}, \mathcal{D}), \tag{28}$$

where β is a scaling term for adjusting the effect of the prior, $f_{\boldsymbol{\theta}_0}$ is the prior network which is held fixed during training. Hence, we also conduct experiments with RFPs in our experiments to investigate how does the RFPs will affect the learning of dynamics models.

We vary the value of β in $\{0.1, 0.3, 1\}$. The results are reported in Figure 17. Firstly, by properly choosing the value of β, RFPs slightly improve the performance on `Hopper`, and don't affect the performance a lot on `Walker2D`. However, for `Window-open-v2`, RFPs will hurt the performance a lot. We conjecture that this might because our choice of the prior function on `Window-open-v2` is not suitable for the task, i.e., the reward is sparse in `Window-open-v2`, but the RFPs

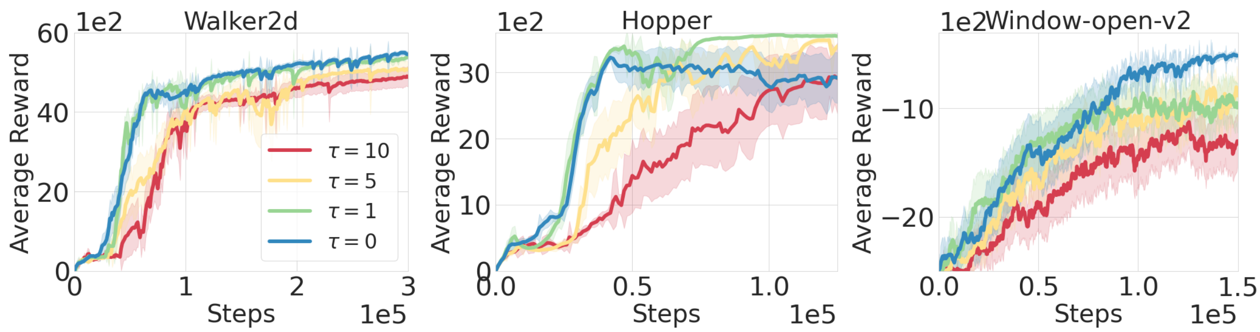

Figure 16: Experiments of forced exploration on `Walker2d`, `Hopper` and `Window-open-v2`. The shaded region denotes the one-standard error. $\tau = 0$ is the one without forced exploration.

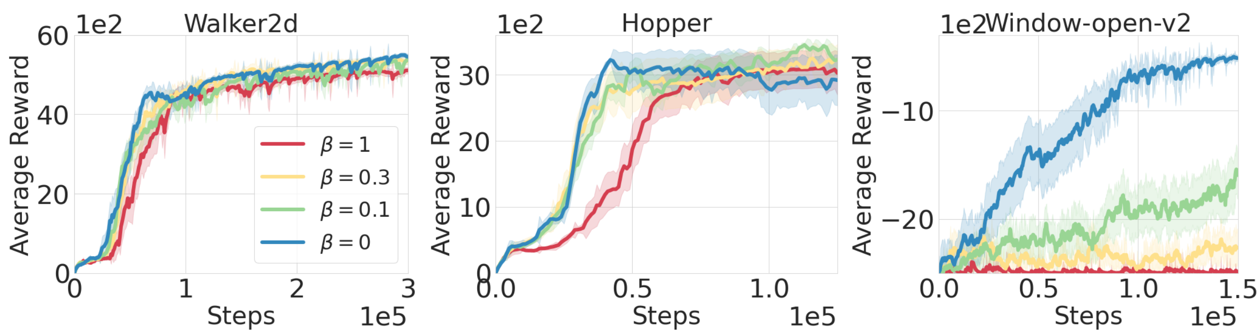

Figure 17: Experiments with random function prior networks (RFPs) on `Walker2d`, `Hopper` and `Window-open-v2`. The shaded region denotes the one-standard error. $\beta = 0$ corresponds to the one without using random function prior networks.

don't induce sparsity on the predictions.

Nevertheless, one interesting observation is that both forced exploration and RFPs seem to help on `Hopper`, and their overall pattern on three tasks is a bit consistent. Therefore, it would be interesting to figure out if there is a deep connection between the forced exploration and RFPs.

F. Theoretical Analysis of PSRL under Approximate Inference

In this section, we present the proof of Theorem 1. The proof of this theorem is inspired by the techniques in Russo & Van Roy (2016), with some additional modifications to extend the results from bandit setting to the reinforcement learning setting.

Regret. For a given MDP $\mathcal{M}$, the regret is defined as the difference between value function of the optimal policy in hindsight and that of the actual policy executed by the algorithm $\mathscr{A}$,

$$\text{Regret}(T, \mathscr{A}, \mathcal{M}) := \sum_{k=1}^{K} \underbrace{\int \rho(\boldsymbol{s}_1) \left(V^{\mathcal{M}}_{\pi^\star,1}(\boldsymbol{s}_1) - V^{\mathcal{M}}_{\pi^k,1}(\boldsymbol{s}_1) \right) d\boldsymbol{s}_1}_{:=\Delta_k}, \tag{29}$$

where $\pi^\star$ is the optimal policy for $\mathcal{M}$, and π^k is the policy employed by the algorithm for k_{th} episode. Correspondingly, the Bayesian regret is defined as the expectation of the above regret, i.e.,

$$\text{BayesianRegret}(T, \mathscr{A}, p(\mathcal{M})) := \mathbb{E}\left[\text{Regret}(T, \mathscr{A}, \mathcal{M})\right].$$

Here the expectation is taken over the prior distribution of dynamics models $p(\mathcal{M})$ and the randomness in the algorithm $\mathscr{A}$ and environment.

Bayesian regret under approximate inference. Let us denote the approximate and true posterior distribution of polices at k_{th} episode by

$$q_k(\pi) = q(\pi|\mathcal{D}^k_{\mathcal{E}}) \text{ and } p_k(\pi) = \int \delta(\pi|\mathcal{M}) p(\mathcal{M}|\mathcal{D}^k_{\mathcal{E}}) d\mathcal{M}.$$

where $\mathcal{D}^k_{\mathcal{E}}$ denotes all the data collected from the environment $\mathcal{E}$ till the k-th episode . Next we characterize how approximate posterior inference affects the Bayesian regret.

Theorem 1 *For K episodes, the Bayesian regret of posterior sampling reinforcement learning algorithm $\mathscr{A}$ with any approximate posterior distribution q_k at episode k is upper bounded by*

$$\sqrt{CK(HR_{\max})^2 \mathbb{H}(\pi^\star)} + 2HR_{\max} \sum_{k=1}^{K} \sqrt{\mathbb{E}\left[\mathbf{d}_{KL}\left(q_k(\pi) \| p_k(\pi) \right) \right]}, \tag{30}$$

where $\mathbb{H}(\pi^\star)$ is the entropy of the prior distribution of optimal polices, i.e., $p(\pi) = \int \delta(\pi|\mathcal{M}) p(\mathcal{M}) d\mathcal{M}$, C is a problem-dependent constant (see Appendix for details) and $\mathbf{d}_{KL}\left(\cdot \| \cdot \right)$ is the KL-divergence.

Our Theorem 1 is a general result with minimal assumptions. For a specific problem setup, it remains to instantiate the problem-dependent constant C and $\mathbb{H}(\pi^\star)$ for deriving the regret bound. Although a detailed investigation is out the scope of this work, we provide a concrete example for showing that the constant C can be bounded well.

Remark 1 *When the number of policies $|\Pi|$ is finite, and the value function $V^{M}_{\pi,1}$ is linear with its parameter lives in $\mathbb{R}^d$, then C can be upper bounded by d, i.e., $C \leq d$.*

To be noted, the first term of the regret is $\tilde{\mathcal{O}}(\sqrt{K})$, which is the standard result. The second term will be zero under exact posterior inference. However, when performing approximate inference (which is usually the case in practice), the second term could result in a *linear* regret (i.e., $\tilde{\mathcal{O}}(K)$) due to the approximation error (i.e., $\min_q \mathbf{d}_{\text{KL}}\left(q \| p \right) > 0$). Therefore, the second term will dominate the entire regret under approximate inference. To reduce it, we should choose an approximate posterior distribution $q(\pi|\mathcal{D}_{\mathcal{E}})$ as "close" to the true distribution $p(\pi|\mathcal{D}_{\mathcal{E}})$ as possible.

F.1. Proof of Theorem 1

Proof: Recall the definition of Bayesian regret,

$$\text{BayesianRegret}(T, \mathscr{A}, p(\mathcal{M})) \coloneqq \mathbb{E}\left[\text{Regret}(T, \mathscr{A}, M)\right] = \mathbb{E}\left[\sum_{k=1}^{K} \Delta_k\right]. \tag{31}$$

Let's denote history at the beginning of episode k as H_k. Then, we can rewrite the Bayesian regret as

$$\text{BayesianRegret}(T, \mathscr{A}, p(\mathcal{M})) = \sum_{k=1}^{K} \mathbb{E}_{H_k}\left[\mathbb{E}\left[\Delta_k | H_k\right]\right]. \tag{32}$$

By doing so, we can bound each term $\mathbb{E}[\Delta_k|H_k]$ separately. For convenience, we define $\mathbb{E}_k[\Delta_k] \coloneqq \mathbb{E}[\Delta_k|H_k]$. Then, by Lemma 1, we can further decompose it into,

$$\mathbb{E}[\Delta_k|H_k] = G_k + D_k, \tag{33}$$

where

$$G_k \coloneqq \int \sqrt{q_k(\pi)p_k(\pi)}\left(\mathbb{E}_k\left[V_{\pi,1}^{M}(\boldsymbol{s}_1)|\pi^\star = \pi\right] - \mathbb{E}_k\left[V_{\pi,1}^{M}(\boldsymbol{s}_1)\right]\right) d\pi \tag{34}$$

and

$$D_k \coloneqq \int \left(\sqrt{p_k(\pi)} - \sqrt{q_k(\pi)}\right)\left(\sqrt{p_k(\pi)}\mathbb{E}_k\left[V_{\pi,1}^{M}(\boldsymbol{s}_1)|\pi^\star = \pi\right] + \sqrt{q_k(\pi)}\mathbb{E}_k\left[V_{\pi,1}^{M}(\boldsymbol{s}_1)\right]\right) d\pi. \tag{35}$$

Then, it remains to bound $\sum_{k=1}^{K}\mathbb{E}[G_k]$ and $\sum_{k=1}^{K}\mathbb{E}[D_k]$. By Lemma 2, we can bound the sum of expectation of D_k by

$$\sum_{k=1}^{K}\mathbb{E}[D_k] \leq 2HR_{\max}\sum_{k=1}^{K}\sqrt{\mathbb{E}\left[\mathbf{d}_{\mathrm{KL}}\left(q_k \| p_k\right)\right]}. \tag{36}$$

By Lemma 3, the upper bound for the sum of the expectation of G_k is

$$\sum_{k=1}^{K}\mathbb{E}[G_k] \leq \sqrt{CK\left((HR_{\max})^2/2\right)\mathbb{H}\left(\pi^\star\right)}. \tag{37}$$

Hence, the term D_k captures the regret incurred by the approximate inference error, and G_k captures the standard regret for Thompson sampling, which is of order $\tilde{\mathcal{O}}(\sqrt{K})$. By combining them together, we finally arrive at the upper bound of the Bayesian regret

$$\begin{aligned}\text{BayesianRegret}(T, \mathscr{A}, p(\mathcal{M})) \leq & \sqrt{CK(HR_{\max})^2\mathbb{H}\left(\pi^\star\right)} \\ & + 2HR_{\max}\sum_{k=1}^{K}\sqrt{\mathbb{E}\left[\mathbf{d}_{\mathrm{KL}}\left(q_k(\pi) \| p_k(\pi)\right)\right]}.\end{aligned} \tag{38}$$

$\square$

F.2. Proof of Remark 1

Proof: Recall that the definition of C is

$$C = \max_{k\in\mathbb{Z}_+}\frac{\left[\sum_{\pi} g_k(\pi,\pi))\right]^2}{\sum_{\pi}\sum_{\pi'}[g_k(\pi,\pi')]^2} \tag{39}$$

where $g_k(\pi, \pi')$ is defined as

$$g_k(\pi,\pi') = \sqrt{q_k(\pi)p_k(\pi')}(\mathbb{E}_k[V_{\pi,1}^{M}|\pi^\star = \pi'] - \mathbb{E}_k[V_{\pi,1}^{M}]) \tag{40}$$

Since the number of polices $|\Pi|$ is finite, we then define the following matrix

$$G_k = \begin{bmatrix} g_k(\pi_1,\pi_1) & g_k(\pi_1,\pi_2) & \dots & g_k(\pi_1,\pi_{|\Pi|}) \\ g_k(\pi_2,\pi_1) & g_k(\pi_2,\pi_2) & \dots & g_k(\pi_2,\pi_{|\Pi|}) \\ \dots & \dots & \dots & \dots \\ g_k(\pi_{|\Pi|},\pi_1) & g_k(\pi_{|\Pi|},\pi_2) & \dots & g_k(\pi_{|\Pi|},\pi_{|\Pi|}) \end{bmatrix} \tag{41}$$

Then, we can rewrite C as

$$C = \max_{k\in\mathbb{Z}_+} \frac{\text{Trace}(G_k)^2}{\|G_k\|_F^2} \tag{42}$$

By the fact that, $\text{Trace}(A)^2 \le \text{rank}(A)\|A\|_F^2$, we will have

$$\frac{\text{Trace}(G_k)^2}{\|G_k\|_F^2} \le \text{rank}(G_k) \tag{43}$$

Since G_k is a $|\Pi|$-by- $|\Pi|$ matrix, we must have

$$\text{rank}(G_k) \le |\Pi| \tag{44}$$

Since we also assume that the value function is linear in its parameters which is in $\mathbb{R}^d$, then by the linearity of expectation, we can write each $g_k(\pi_i,\pi_j)$ as (for some $\mathbf{v}_i, \theta_j$)

$$g_k(\pi_i,\pi_j) = \sqrt{q_k(\pi_i)p_k(\pi_j)}(\mathbf{v}_i^\top\theta_j - \mathbf{v}_i^\top\theta) \tag{45}$$

Then, we can define $\mathbf{u}_i = \sqrt{q_k(\pi_i)}\mathbf{v}_i$ and $\mathbf{w}_j = \sqrt{p_k(\pi_j)}(\theta_j - \theta)$, which further gives us

$$G_k = \underbrace{\begin{bmatrix} \mathbf{u}_1^\top \\ \mathbf{u}_2^\top \\ \dots \\ \mathbf{u}_{|\Pi|}^\top \end{bmatrix}}_{:=\mathbf{U}} \underbrace{\begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_{|\Pi|} \end{bmatrix}}_{:=\mathbf{W}} = \mathbf{U}\mathbf{W} \tag{46}$$

Since the parameters θ is in $\mathbb{R}^d$, we must have the rank of both $\mathbf{U}, \mathbf{W}$ no greater than d. Therefore, we then have $\text{rank}(G_k) \le d$. By combining the above two, we can conclude that $\forall k \in \mathbb{Z}_+$

$$\text{rank}(G_k) \le \min\{|\Pi|, d\}, \tag{47}$$

which further implies that $C \le \min\{|\Pi|, d\}$ and concludes the proof. □

F.3. Technical Lemmas

Lemma 1 *For each time $k = 1, ..., K$, we have*

$$\mathbb{E}[\Delta_k|H_k] = \mathbb{E}\left[V^M_{\pi^\star,1}(\boldsymbol{s}_1) - V^M_{\pi^k,1}(\boldsymbol{s}_1)|H_k\right] := \mathbb{E}_k\left[V^M_{\pi^\star,1}(\boldsymbol{s}_1) - V^M_{\pi^k,1}(\boldsymbol{s}_1)\right] = G_k + D_k, \tag{48}$$

where

$$G_k := \int \sqrt{q_k(\pi)p_k(\pi)}\left(\mathbb{E}_k\left[V^M_{\pi,1}(\boldsymbol{s}_1)|\pi^\star = \pi\right] - \mathbb{E}_k\left[V^M_{\pi,1}(\boldsymbol{s}_1)\right]\right) d\pi \tag{49}$$

and

$$D_k := \int \left(\sqrt{p_k(\pi)} - \sqrt{q_k(\pi)}\right)\left(\sqrt{p_k(\pi)}\mathbb{E}_k\left[V^M_{\pi,1}(\boldsymbol{s}_1)|\pi^\star = \pi\right] + \sqrt{q_k(\pi)}\mathbb{E}_k\left[V^M_{\pi,1}(\boldsymbol{s}_1)\right]\right) d\pi. \tag{50}$$

Proof: Conditioning on the history H_k, we can write the conditional Bayesian regret as

$$\mathbb{E}_k\left[V^M_{\pi^\star,1}(\boldsymbol{s}_1) - V^M_{\pi^k,1}(\boldsymbol{s}_1)\right] \tag{51}$$

$$= \int p_k(\pi)\mathbb{E}_k\left[V^M_{\pi,1}(\boldsymbol{s}_1)|\pi^\star = \pi\right]d\pi - \int q_k(\pi)\mathbb{E}_k\left[V^M_{\pi,1}(\boldsymbol{s}_1)|\pi^k = \pi\right]d\pi \tag{52}$$

$$= \int p_k(\pi)\mathbb{E}_k\left[V^M_{\pi,1}(\boldsymbol{s}_1)|\pi^\star = \pi\right]d\pi - \int q_k(\pi)\mathbb{E}_k\left[V^M_{\pi,1}(\boldsymbol{s}_1)\right]d\pi \tag{53}$$

$$= G_k + D_k, \tag{54}$$

where the second equality holds because the value function is independent of the instantiation of the policy π^k when given the history H_k. □

Lemma 2 *For any $k = 1, ..., K$, we have*

$$\sum_{k=1}^{K}\mathbb{E}[D_k] \leq 2HR_{\max}\sum_{k=1}^{K}\sqrt{\mathbb{E}\left[\mathbf{d}_{KL}\left(q_k|\,p_k\right)\right]}. \tag{55}$$

Proof: Recall D_k,

$$D_k \coloneqq \int\left(\sqrt{p_k(\pi)} - \sqrt{q_k(\pi)}\right)\left(\sqrt{p_k(\pi)}\mathbb{E}_k\left[V^M_{\pi,1}(\boldsymbol{s}_1)|\pi^\star = \pi\right] + \sqrt{q_k(\pi)}\mathbb{E}_k\left[V^M_{\pi,1}(\boldsymbol{s}_1)\right]\right)d\pi \tag{56}$$

By using the Cauchy-Schwarz inequality, we have

$$D_k \leq \left(\sqrt{\int\left(\sqrt{p_k(\pi)} - \sqrt{q_k(\pi)}\right)^2 d\pi}\right) \cdot \left(\sqrt{\int p_k(\pi)\mathbb{E}\left[V^M_{\pi,1}(\boldsymbol{s}_1)|\pi^\star = \pi\right]^2 d\pi} + \sqrt{\int q_k(\pi)\mathbb{E}_k\left[V^M_{\pi,1}(\boldsymbol{s}_1)\right]^2 d\pi}\right). \tag{57}$$

By the definition of Hellinger distance $\mathbf{d}_{\mathrm{H}}\left(\cdot|\,\cdot\right)$ between two random variables, we have

$$D_k \leq \mathbf{d}_{\mathrm{H}}\left(q_k|\,p_k\right)\left(\sqrt{\int p_k(\pi)\mathbb{E}\left[V^M_{\pi,1}(\boldsymbol{s}_1)^2|\pi^\star = \pi\right]d\pi} + \sqrt{\int q_k(\pi)\mathbb{E}_k\left[V^M_{\pi,1}(\boldsymbol{s}_1)^2\right]d\pi}\right). \tag{58}$$

Since $[\mathbf{d}_{\mathrm{H}}\left(\cdot|\,\cdot\right)]^2 \leq \mathbf{d}_{\mathrm{KL}}\left(\cdot|\,\cdot\right)$ and $V^M_{\pi,1}$ is a bounded random variable with $HR_{\max}$ as its upper bound, we have

$$D_k \leq 2\mathbf{d}_{\mathrm{H}}\left(q_k|\,p_k\right)HR_{\max} \leq 2\sqrt{\mathbf{d}_{\mathrm{KL}}\left(q_k|\,p_k\right)}HR_{\max}. \tag{59}$$

Hence, we have

$$\sum_{k=1}^{K}\mathbb{E}[D_k] \leq 2HR_{\max}\sum_{k=1}^{K}\sqrt{\mathbb{E}\left[\mathbf{d}_{\mathrm{KL}}\left(q_k|\,p_k\right)\right]}. \tag{60}$$

□

Lemma 3 *For each $k = 1, ..., K$, we have*

$$\sum_{k=1}^{K}\mathbb{E}[G_k] \leq \sqrt{CK\left((HR_{\max})^2/2\right)\mathbb{H}\left(\pi^\star\right)}. \tag{61}$$

Proof: Recall the definition of G_k,

$$G_k \coloneqq \int\sqrt{q_k(\pi)p_k(\pi)}\left(\mathbb{E}_k\left[V^M_{\pi,1}(\boldsymbol{s}_1)|\pi^\star = \pi\right] - \mathbb{E}_k\left[V^M_{\pi,1}(\boldsymbol{s}_1)\right]\right)d\pi. \tag{62}$$

Since $V^M_{\pi,1}$ (here, we drop the dependency on $\boldsymbol{s}_1$ for clearness) is a bounded random variable, and more specifically, it's $((HR_{\max})/2)$-sub-Gaussian. Hence, by Lemma 4, the following holds,

$$\mathbb{E}_k\left[V^M_{\pi,1}|\pi^\star=\pi\right]-\mathbb{E}_k\left[V^M_{\pi,1}\right]\le\left(\frac{HR_{\max}}{2}\right)\sqrt{2\mathbf{d}_{\mathrm{KL}}\left(p_k(V^M_{\pi,1}|\pi^\star=\pi)\middle|\, p_k(V^M_{\pi,1})\right)}. \tag{63}$$

This gives us that

$$G_k\le\int\sqrt{q_k(\pi)p_k(\pi)}\left(\frac{HR_{\max}}{2}\right)\sqrt{2\mathbf{d}_{\mathrm{KL}}\left(p_k(V^M_{\pi,1}|\pi^\star=\pi)\middle|\, p_k(V^M_{\pi,1})\right)}d\pi. \tag{64}$$

Then, we can further rewrite the KL-divergence using the conditional mutual information $\mathbb{I}_k(\cdot;\cdot)$ (i.e., conditioning on the history H_k),

$$\iint q_k(\pi)p_k(\pi')\mathbf{d}_{\mathrm{KL}}\left(p_k(V^M_{\pi,1}|\pi^\star=\pi')\middle|\, p_k(V^M_{\pi,1})\right)d\pi d\pi'=\int q_k(\pi)\mathbb{I}_k(\pi^\star;V^M_{\pi,1})d\pi. \tag{65}$$

When conditioning on the history H_k, the optimal policy $\pi^\star$ and M is independent of the π^k, hence we have

$$\int q_k(\pi)\mathbb{I}_k(\pi^\star;V^M_{\pi,1})d\pi=\int q_k(\pi)\mathbb{I}_k(\pi^\star;V^M_{\pi^k,1}|\pi^k=\pi)d\pi=\mathbb{I}_k(\pi^\star;V^M_{\pi^k,1}|\pi^k). \tag{66}$$

By the fact that $\pi^\star$ is jointly independent of $V^M_{\pi^k,1}$ and π^k when conditioning on the history H_k, hence we have

$$\mathbb{I}_k(\pi^\star;V^M_{\pi^k,1}|\pi^k)=\mathbb{I}_k(\pi^\star;V^M_{\pi^k,1}|\pi^k)+\mathbb{I}_k(\pi^\star;\pi^k). \tag{67}$$

By the chain rule of mutual information, we finally get

$$\mathbb{I}_k(\pi^\star;V^M_{\pi^k,1}|\pi^k)+\mathbb{I}_k(\pi^\star;\pi^k)=\mathbb{I}_k(\pi^\star;(V^M_{\pi^k,1},\pi^k)). \tag{68}$$

Now, let's define the following function g_k and C,

$$g_k(\pi,\pi'):=\sqrt{q_k(\pi)p_k(\pi')}\left(\mathbb{E}_k\left[V^M_{\pi,1}|\pi^\star=\pi'\right]-\mathbb{E}_k\left[V^M_{\pi,1}\right]\right). \tag{69}$$

$$C:=\max_{k\in\mathbb{Z}_+}\frac{(\int g_k(\pi,\pi)d\pi)^2}{\iint[g_k(\pi,\pi')]^2d\pi d\pi'}. \tag{70}$$

Thus, we further have

$$\mathbb{I}_k(\pi^\star;(V^M_{\pi^k,1},\pi^k))\ge\frac{2}{(HR_{\max})^2}\iint[g_k(\pi,\pi')]^2d\pi d\pi'. \tag{71}$$

On the other hand, we can rewrite G_k as

$$G_k=\int g_k(\pi,\pi)d\pi. \tag{72}$$

By rearranging the terms, we get

$$\frac{G_k^2}{\mathbb{I}_k(\pi^\star;(\pi^k,V^M_{\pi^k,1}))}\le\frac{\left((HR_{\max})^2/2\right)(\int g_k(\pi,\pi)d\pi)^2}{\iint\left[g_k(\pi,\pi')\right]^2d\pi d\pi'}\le C\left((HR_{\max})^2/2\right). \tag{73}$$

Hence,

$$G_k\le\sqrt{C\left((HR_{\max})^2/2\right)\mathbb{I}_k(\pi^\star;(\pi^k,V^M_{\pi^k,1}))}. \tag{74}$$

Hence, we have

$$\sum_{k=1}^{K}\mathbb{E}[G_k] \le \sum_{k=1}^{K}\mathbb{E}\left[\sqrt{C\left((HR_{\max})^2/2\right)\mathbb{I}_k(\pi^\star;(\pi^k, V^M_{\pi^k,1}))}\right] \tag{75}$$

$$= \sqrt{C\left((HR_{\max})^2/2\right)}\sum_{k=1}^{K}\mathbb{E}\left[\sqrt{\mathbb{I}_k(\pi^\star;(\pi^k, V^M_{\pi^k,1}))}\right] \tag{76}$$

$$\le \sqrt{C\left((HR_{\max})^2/2\right)}\sqrt{K\mathbb{E}\left[\sum_{k=1}^{K}\mathbb{I}_k(\pi^\star;(\pi^k, V^M_{\pi^k,1}))\right]} \tag{77}$$

$$= \sqrt{CK\left((HR_{\max})^2/2\right)}\sqrt{\mathbb{E}\left[\sum_{k=1}^{K}\mathbb{I}_k(\pi^\star;(\pi^k, V^M_{\pi^k,1}))\right]} \tag{78}$$

$$\le \sqrt{CK\left((HR_{\max})^2/2\right)\mathbb{H}\left(\pi^\star\right)}, \tag{79}$$

□

Lemma 4 (Russo & Van Roy (2016)) *Suppose that there is a H_k-measurable random variable η, such that for each $\pi \in \Pi$, $V^M_{\pi,1}$ is a η-sub-Gaussian random variable when conditioned on H_k, then for every $\pi, \pi' \in \Pi$, the following holds with probability* 1*,*

$$\mathbb{E}_k[V^M_{\pi,1}|\pi^\star = \pi'] - \mathbb{E}_k[V^M_{\pi,1}] \le \eta\sqrt{2\mathbf{d}_{KL}\left(p_k(V^M_{\pi,1}|\pi^\star = \pi')\middle|\, p_k(V^M_{\pi,1})\right)}. \tag{80}$$

F.4. An Alternative Analysis of the Bayesian Regret Bound of PSRL under Approximate Inference

Theorem 2 *For K episodes, the Bayesian regret of posterior sampling reinforcement learning algorithm $\mathscr{A}$ with any approximate posterior distribution q_k at episode k is upper bounded by*

$$\mathcal{BR}(K, \text{TS}, p(\mathcal{M})) + 4HR_{\max}\sum_{k=1}^{K}\sqrt{\mathbf{d}_{KL}\left(q(\pi_{0:k-1})\middle|\, p_{\text{exact}}(\pi_{0:k-1})\right)} + 2HR_{\max}\sum_{k=1}^{K}\sqrt{\mathbb{E}\left[\mathbf{d}_{KL}\left(q_k(\pi)\middle|\, p_k(\pi)\right)\right]}. \tag{81}$$

where $\mathcal{BR}(K, \text{TS}, p(\mathcal{M}))$ denotes the Bayesian regret under exact Thompson sampling, $\mathbf{d}_{KL}(\cdot|\cdot)$ is the KL-divergence, and $q(\pi_{0:k-1})$ and $q_{\text{exact}}(\pi_{0:k-1})$ are the joint likelihood of the polices under the approximate posterior q and the exact posterior p, respectively.

Proof: We first define the following notations,

$$V^{\mathcal{M}}_{\pi,1} = \int \rho(\boldsymbol{s})V^{\mathcal{M}}_{\pi,1}(\boldsymbol{s})d\boldsymbol{s}. \tag{82}$$

Recall the definition of the Bayesian regret,

$$\mathbb{E}\left[\sum_{k=1}^{K}\Delta_k\right] = \mathbb{E}\left[\sum_{k=1}^{K}V^{\mathcal{M}}_{\pi^\star,1} - V^{\mathcal{M}}_{\pi^k,1}\right]. \tag{83}$$

By denoting $\pi^{\star,k}$ as a sample from the true posterior distribution $p_k(\pi)$, we can further rewrite the Bayesian regret as

$$\mathbb{E}\left[\sum_{k=1}^{K}\Delta_k\right] = \mathbb{E}\left[\sum_{k=1}^{K}V^{\mathcal{M}}_{\pi^\star,1} - V^{\mathcal{M}}_{\pi^{\star,k},1} + V^{\mathcal{M}}_{\pi^{\star,k},1} - V^{\mathcal{M}}_{\pi^k,1}\right] \tag{84}$$

$$= \mathbb{E}\left[\sum_{k=1}^{K}V^{\mathcal{M}}_{\pi^\star,1} - V^{\mathcal{M}}_{\pi^{\star,k},1}\right] + \mathbb{E}\left[\sum_{k=1}^{K}V^{\mathcal{M}}_{\pi^{\star,k},1} - V^{\mathcal{M}}_{\pi^k,1}\right] \tag{85}$$

We can further expand the first term of Equation 85 as

$$\mathbb{E}\left[\sum_{k=1}^{K} V^{\mathcal{M}}_{\pi^\star,1} - V^{\mathcal{M}}_{\pi^{\star,k},1}\right] = \mathbb{E}\left[\sum_{k=1}^{K} \mathbb{E}\left[V^{\mathcal{M}}_{\pi^\star,1} - V^{\mathcal{M}}_{\pi^{\star,k},1}\middle| H_{k-1} \sim q(H_{k-1})\right]\right], \tag{86}$$

where $q(H_{k-1})$ is the marginal likelihood of the history H_{k-1} under approximate inference. We can further rewrite the above equation as

$$\begin{aligned}
&\mathbb{E}\left[\sum_{k=1}^{K} \mathbb{E}\left[V^{\mathcal{M}}_{\pi^\star,1} - V^{\mathcal{M}}_{\pi^{\star,k},1}\middle| H_{k-1} \sim q(H_{k-1})\right]\right] && (87)\\
&= \sum_{k=1}^{K} \int \mathbb{E}\left[V^{\mathcal{M}}_{\pi^\star,1} - V^{\mathcal{M}}_{\pi^{\star,k},1}\middle| H_{k-1}\right] q(H_{k-1}) dH_{k-1} && (88)\\
&= \sum_{k=1}^{K} \int \mathbb{E}\left[V^{\mathcal{M}}_{\pi^\star,1} - V^{\mathcal{M}}_{\pi^{\star,k},1}\middle| H_{k-1}\right] p_{\text{exact}}(H_{k-1}) dH_{k-1} && (89)\\
&\quad + \int \mathbb{E}\left[V^{\mathcal{M}}_{\pi^\star,1} - V^{\mathcal{M}}_{\pi^{\star,k},1}\middle| H_{k-1}\right] \left(q(H_{k-1}) - p_{\text{exact}}(H_{k-1})\right) dH_{k-1} \\
&= \mathcal{BR}(K, \text{TS}, p(\mathcal{M})) + \sum_{k=1}^{K} \int \mathbb{E}\left[V^{\mathcal{M}}_{\pi^\star,1} - V^{\mathcal{M}}_{\pi^{\star,k},1}\middle| H_{k-1}\right] \left(q(H_{k-1}) - p_{\text{exact}}(H_{k-1})\right) dH_{k-1} && (90)\\
&\le \mathcal{BR}(K, \text{TS}, p(\mathcal{M})) + 2HR_{\max} \sum_{k=1}^{K} \int |q(H_{k-1}) - p_{\text{exact}}(H_{k-1})| dH_{k-1} && (91)\\
&= \mathcal{BR}(K, \text{TS}, p(\mathcal{M})) + 4HR_{\max} \sum_{k=1}^{K} \mathbf{d}_{\text{TV}}(q(H_{k-1}) | p_{\text{exact}}(H_{k-1})). && (92)
\end{aligned}$$

where BayesianRegret$(K, \text{TS}, p(\mathcal{M}))$ denotes the Bayesian regret under exact Thompson sampling, which has been well-studied in Agrawal & Jia (2017); Osband et al. (2013); Osband & Roy (2014; 2017); Fan & Ming (2021); Tiapkin et al. (2022a;b). $p_{\text{exact}}(H_{k-1})$ is the marginal likelihood of the history H_{k-1} under the exact posterior. By Pinsker's inequality, for any two distributions p and q, we have

$$\mathbf{d}_{\text{TV}}(q|p) \le \sqrt{\frac{1}{2}\mathbf{d}_{\text{KL}}\left(q \middle| p\right)}. \tag{93}$$

Therefore, we have

$$\begin{aligned}
\mathbf{d}_{\text{TV}}(q(H_k) | p_{\text{exact}}(H_k)) &\le \sqrt{\frac{1}{2}\mathbf{d}_{\text{KL}}\left(q(H_k) \middle| p_{\text{exact}}(H_k)\right)} && (94)\\
&\le \sqrt{\frac{1}{2}\mathbf{d}_{\text{KL}}\left(q(H_k, \pi_{0:k-1}) \middle| p_{\text{exact}}(H_k, \pi_{0:k-1})\right)}. && (95)
\end{aligned}$$

By the definition of the joint distributions,

$$\begin{aligned}
q(H_k, \pi_{0:k-1}) &= p(H_k | \pi_{0:k-1}) q(\pi_{0:k-1}) && (96)\\
p_{\text{exact}}(H_k, \pi_{0:k-1}) &= p(H_k | \pi_{0:k-1}) p_{\text{exact}}(\pi_{0:k-1}) && (97)
\end{aligned}$$

Thus, we can further simplify the Equation 95 as

$$\sqrt{\frac{1}{2}\mathbf{d}_{\text{KL}}\left(q(H_k, \pi_{0:k-1}) \middle| p_{\text{exact}}(H_k, \pi_{0:k-1})\right)} = \sqrt{\frac{1}{2}\mathbf{d}_{\text{KL}}\left(q(\pi_{0:k-1}) \middle| p_{\text{exact}}(\pi_{0:k-1})\right)} \tag{98}$$

Similarly, we can bound the second term in Equation 85 by

$$\mathbb{E}\left[\sum_{k=1}^{K} V^{\mathcal{M}}_{\pi^{\star,k},1} - V^{\mathcal{M}}_{\pi^{k},1}\right] = \mathbb{E}\left[\sum_{k=1}^{K} (V^{\mathcal{M}}_{\pi^{\star,k},1} - V^{\mathcal{M}}_{\pi^{k},1})\mathbb{1}(\pi^k = \pi^{\star,k})\right]$$
$$+ \mathbb{E}\left[\sum_{k=1}^{K} (V^{\mathcal{M}}_{\pi^{\star,k},1} - V^{\mathcal{M}}_{\pi^{k},1})\mathbb{1}(\pi^k \neq \pi^{\star,k})\right] \tag{99}$$
$$= \mathbb{E}\left[\sum_{k=1}^{K} (V^{\mathcal{M}}_{\pi^{\star,k},1} - V^{\mathcal{M}}_{\pi^{k},1})\mathbb{1}(\pi^k \neq \pi^{\star,k})\right] \tag{100}$$
$$\leq 2HR_{\max}\mathbb{E}\left[\sum_{k=1}^{K} \mathbb{1}(\pi^k \neq \pi^{\star,k})\right]. \tag{101}$$

By the maximal coupling, we have that the probability of $\pi^k \neq \pi^{\star,k}$ is $\mathrm{TV}(p_k, q_k)$. Thus, in together with Jensen's inequality, we can bound the above equation by

$$\mathbb{E}\left[\sum_{k=1}^{K} V^{\mathcal{M}}_{\pi^{\star,k},1} - V^{\mathcal{M}}_{\pi^{k},1}\right] \leq 2HR_{\max}\mathbb{E}\left[\sum_{k=1}^{K} \sqrt{\frac{1}{2}\mathbf{d}_{\mathrm{KL}}\left(q_k(\pi) | \, p_k(\pi)\right)}\right] \tag{102}$$
$$\leq 2HR_{\max}\sum_{k=1}^{K} \sqrt{\mathbb{E}\left[\mathbf{d}_{\mathrm{KL}}\left(q_k(\pi) | \, p_k(\pi)\right)\right]}. \tag{103}$$

By combining the above results, we can conclude that the regret bound, i.e., $\mathcal{BR}(K, \mathscr{A}, p(\mathcal{M}))$, is at most

$$\mathcal{BR}(K, \mathrm{TS}, p(\mathcal{M})) + 4HR_{\max}\sum_{k=1}^{K} \sqrt{\mathbf{d}_{\mathrm{KL}}\left(q(\pi_{0:k-1}) | \, p_{\text{exact}}(\pi_{0:k-1})\right)} + 2HR_{\max}\sum_{k=1}^{K} \sqrt{\mathbb{E}\left[\mathbf{d}_{\mathrm{KL}}\left(q_k(\pi) | \, p_k(\pi)\right)\right]}. \tag{104}$$

□

G. Additional Umap Visualizations

We provide the Umap visualization of the state embeddings of PS-MBPO and MBPO during training in Figure 18 and Figure 19. We observe that PS-MBPO will mostly cover a more broad range of the embedding space, and its pattern also evolves more rapidly than MBPO.

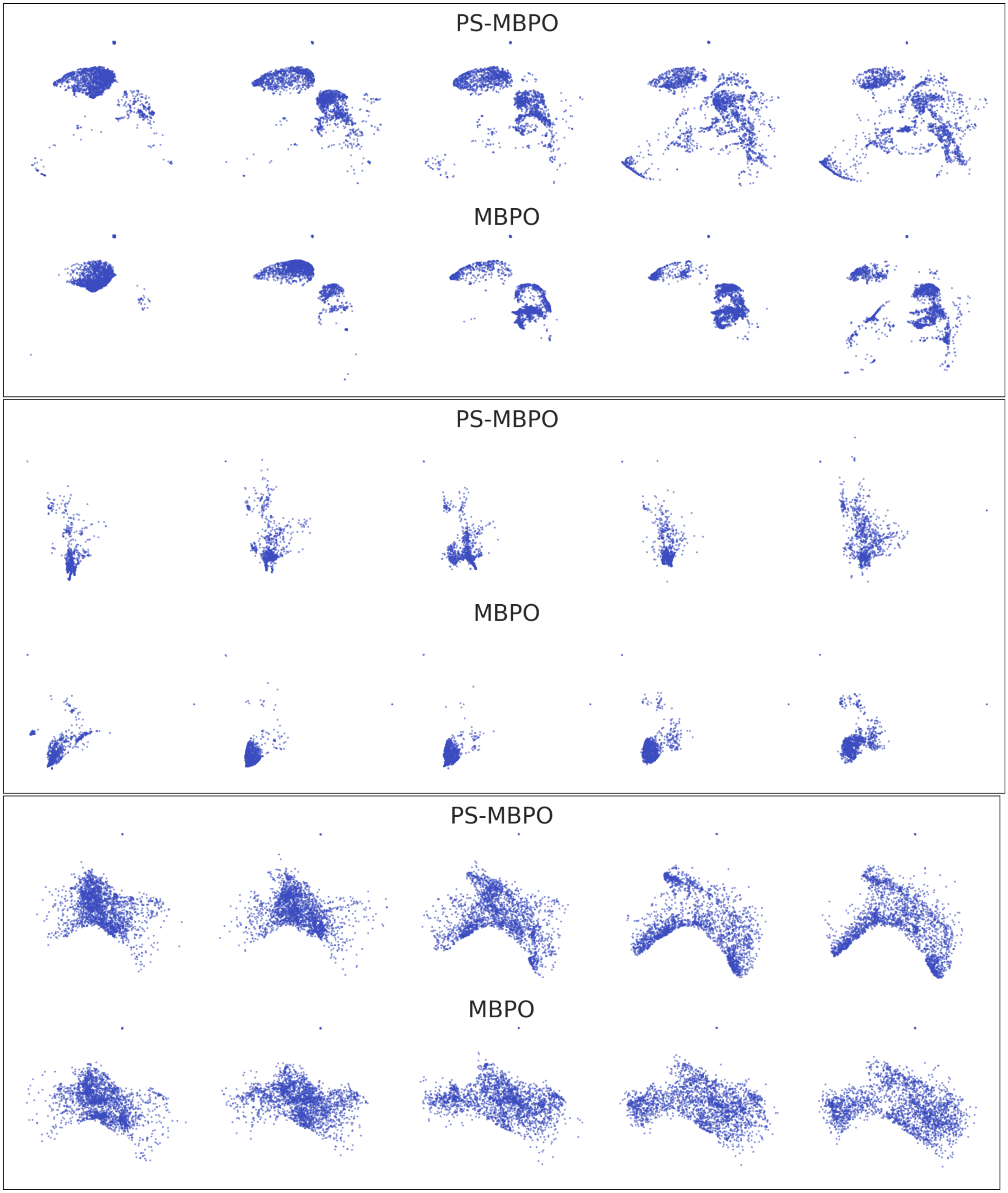

Figure 18: Visualization of the Umap embeddings of PS-MBPO and MBPO from consecutive training periods on `Hopper`, `Ant` and `Halfcheetah` (from top to bottom).

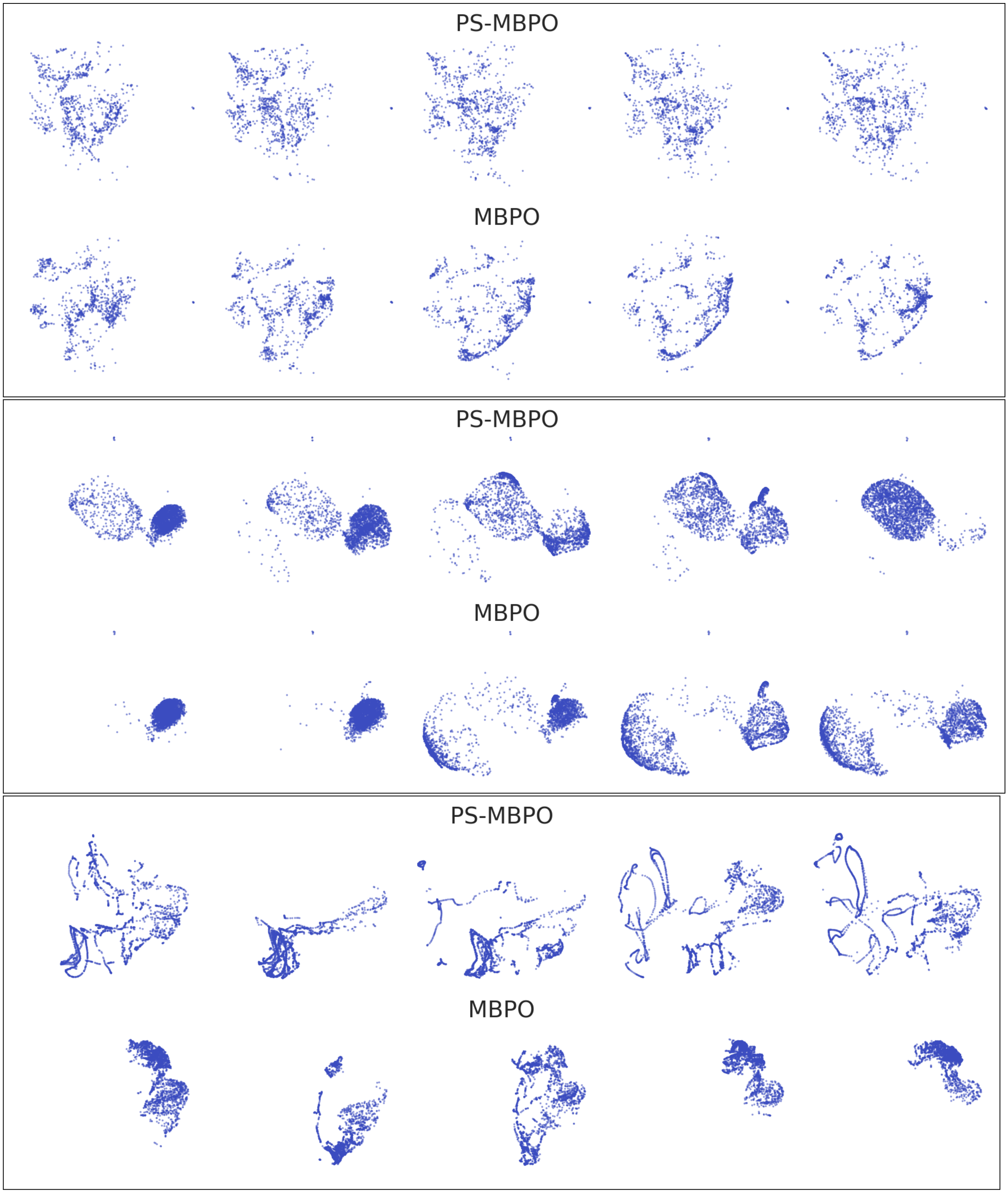

Figure 19: Visualization of the Umap embeddings of PS-MBPO and MBPO from consecutive training periods on `Walker2d`, `Ball-in-Cup` and `Cartpole-swingup` (from top to bottom).