Dual Conservative Policy Update for Efficient Model-Based Reinforcement Learning

Shenao Zhang
Georgia Institute of Technology
shenao@gatech.edu

Abstract

Model-based reinforcement learning (MBRL) algorithms by acquiring predictive models are data-efficient. Different from greedy model exploitation algorithms, provably-efficient MBRL algorithms based on optimism or posterior sampling are ensured to achieve the optimal performance asymptotically when with additional model complexity measures. However, such complexity measure bound is strong for many model function classes, which poses challenges to achieve optimality in practice. Due to the aggressive policy update and over-exploration, the convergence can be a very slow process and the policy may even end up suboptimal. Thus, in addition to the asymptotic guarantee, ensuring iterative policy improvement is the key to achieving high performance with finite timesteps. To this end, we propose *Dual Conservative Policy Update* (DCPU) for MBRL that involves a *locally greedy update* procedure and a *conservative exploration update* procedure. By greedily exploiting the local model and maximizing the expected value within the trust region, DCPU agents explore efficiently. We theoretically provide the iterative policy improvement bound of DCPU and show the monotonic improvement property under the Lipschitz assumption and with a proper constraint threshold. Besides, we prove the asymptotic optimality of DCPU with $\widetilde{\mathcal{O}}(\sqrt{T})$ Bayes regret bound. Empirical results demonstrate the superiority of DCPU on several Mujoco tasks.

1 Introduction

Model-Based Reinforcement Learning (MBRL) algorithms, which involve acquiring a predictive model by interacting with the environment and learning to make the optimal decision using the model, have shown great success [41, 24]. Benefiting from the learned model, MBRL is appealing due to its significantly reduced sample complexity compared to model-free RL methods. However, greedy model exploitation algorithms that assume the model sufficiently accurately resembles the real environment lack theoretical guarantees for asymptotic optimality and may lead to suboptimal policies that get stuck in local maxima even in simple environments [8].

As such, several provably-efficient MBRL algorithms have been proposed. One provable approach is based on the principle of *optimism in the face of uncertainty* (OFU) [42, 35, 8]. OFU achieves the asymptotic optimality by ensuring that the optimistically biased value is close to the real value in the long run, which requires the model families to have restricted complexity measure, *e.g.*, eluder dimension [35, 33] or witness rank[45]. However, in practice when the model complexity bound is strong, the global optimality is hard to guarantee and the effectiveness of these algorithms will be crippled, resulting from over-exploration and the slow convergence process. Even worse, the policy may end up suboptimal when the epistemic uncertainty remains large, especially in complex high-dimensional environments. In fact, it is shown recently [10] that nonlinear models do not have polynomially-bounded eluder dimension and an exploration step can only eliminate an exponentially small portion of the model hypothesis, making it hard to find a global optimum even in the simplified

Preprint. Under review.

bandit or deep RL settings. Achieving high performance with finite executed timesteps thus motivates to also seek for policy improvement guarantees *during* training. An alternative provable algorithm is Posterior Sampling RL (PSRL) [43, 32, 33]. Based on Thompson sampling [47], PSRL agents explore by selecting the best action within an action-value set constructed by sampling from the model posterior. Unfortunately, inefficient over-exploration and policy degradation still exists.

In this work, we propose *Dual Conservative Policy Update* (DCPU), a provable model-based RL algorithm. With a dual policy update procedure, which we call *locally greedy update* and *conservative exploration update*, DCPU makes full use of the optimality information in the model and efficiently explores the environment while avoiding over-exploration and harmful aggressive policy updates. Specifically, when the epistemic model uncertainty is low in the regions that the agent has explored, greedy exploitation of the *locally* accurate model offers reliable policy gradient. Then the conservative exploration update follows as a constrained *expected* value maximization procedure. And agents conservatively update the policies by executing exploring actions that could be plausibly optimal under multiple sampled models within the trust region.

Theoretically, we give the policy iterative improvement bound of DCPU and show that monotonic improvement over iterations can be achieved when with a proper trust region threshold and under Lipschitz assumption. Additionally, we provide the near-optimal $\widetilde{\mathcal{O}}(\sqrt{T})$ Bayes expected regret bound of DCPU that is sublinear in the elapsed iterations T. By enjoying these two properties simultaneously, ***i.e.***, iterative policy improvement and asymptotic optimality, DCPU achieves arbitrarily close to the optimality and offers considerable performance in practice. We conduct experiments on several Mujoco tasks, including the high-dimensional locomotion task. Empirical results reveal the superiority of DCPU and validate the benefits of the dual policy update procedure.

2 Problem Formulation

2.1 Model-based Reinforcement Learning

We consider the problem of learning to optimize a random finite-horizon γ-discounted Markov Decision Process (MDP) over repeated episodes of interaction. Denote the state space and action space as $\mathcal{S}$ and $\mathcal{A}$, respectively. In an episode $h = 1, \ldots, H$ where H is the episode length, the agent takes action $a_h \in \mathcal{A}$ at state $s_h \in \mathcal{S}$ and receives reward $r(s_h, a_h)$ that is bounded in absolute value R. The state s_h then transits to the next state $s_{h+1} \sim f^*(\cdot|s_h, a_h)$, where f^* is the real transition function that governs the dynamical system. For deterministic dynamics, f^* is a dirac measure. And for probabilistic dynamics, f^* can be represented by some parametric probability distribution.

In model-based RL, the true dynamical model f^* is unknown and needs to be learned using the collected data. Through episodic (or iterative) interaction, the transition data is collected at each iteration. The measurements up to iteration t then form $\mathcal{D}_t = \{\{(s_{n,i}, a_{n,i}), s_{n+1,i}\}_{n=0}^{N-1}\}_{i=1}^{t}$. Consider the model function class $\mathcal{F} = \{f : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}\}$. By treating the unknown model as a random variable, the posterior distribution of the dynamics model is estimated as $\phi(f|\mathcal{D}_t)$. The frequentist model of the mean and uncertainty can also be estimated. Specifically, the confidence set (or model hypothesis set) $\mathcal{F}_t \subset \mathcal{F}$ is introduced to represent the range of dynamics that is statistically plausible [35, 33, 8]. To ensure that the real model f^* is contained in $\mathcal{F}_t$ with high probability, one way is to construct the confidence set as $\mathcal{F}_t := \{f \in \mathcal{F} | \|f - \hat{f}_t^{LS}\|_{2,E_t} \leq \sqrt{\beta_t}\}$. Here, β_t is an appropriately chosen confidence parameter, the least squares estimate $\hat{f}_t^{LS} \in \operatorname{argmin}_f \sum_{i=1}^{t-1} \|f - f^*\|_2^2$ and the cumulative empirical 2-norm is defined as $\|g\|_{2,E_t}^2 := \sum_{i=1}^{t-1} \|g(x_i)\|_2^2$. Alternatively, recent works [8, 18] show that learning a calibrated statistical model [22] also suffices to provide accurate uncertainty measure and thus containing the real model f^* in $\mathcal{F}_t$ with high probability.

2.2 Cumulative Regret for Asymptotic Optimality

The objective of RL is to learn a policy that maximizes the cumulative episode reward under the *true* dynamics f^*. Define value function V_π^f to be the episodic return when running policy π on model f:

$$V_\pi^f = \mathop{\mathbb{E}}_{\substack{a_h \sim \pi(\cdot|s_h) \\ s_{h+1} \sim f(\cdot|s_h, a_h)}} \left[\sum_{h=0}^{H} \gamma^h r(s_h, a_h) \mid s_0 \right]. \tag{1}$$

Then the optimal policy is $\pi^* = \operatorname{argmax}_\pi V_\pi^{f^*}$, *i.e.*, optimally behave in the real environment.

To evaluate the performance of an RL algorithm, a common criterion is the cumulative regret, defined as the cumulative performance discrepancy between policy π_t at each iteration t and the optimal policy π^* under the real system f^* over the run of the algorithm. The (cumulative) regret up to iteration T is defined as:

$$\text{Regret}(T, \pi, f^*) := \sum_{t=1}^{T} \mathfrak{R}_t, \tag{2}$$

where $\mathfrak{R}_t$ denotes the regret over iteration t, defined as the performance discrepancy under the state visitation measure $\rho(s)$:

$$\mathfrak{R}_t := \int_{s \in \mathcal{S}} \rho(s)(V_{\pi^*}^{f^*}(s) - V_{\pi_t}^{f^*}(s)). \tag{3}$$

Notably, the above regret is not deterministic since it depends on the unknown model f^*, the learning policy π and through the collected data on the sampled transitions. For algorithms such as posterior sampling reinforcement learning [33, 34], Bayesian expected regret is widely adopted:

$$\text{BayesRegret}(T, \pi, \phi) := \mathbb{E}\left[\text{Regret}(T, \pi, f^*) \mid f^* \sim \phi\right], \tag{4}$$

where f^* is distributed according to ϕ.

One way to prove the asymptotic optimality is to show that the (expected) regret is sublinear in T, so that π_t converges to π^* with sufficient iterations. To obtain the regret bound, the *width of confidence set* $\omega_t(s, a)$ is introduced to represent the maximum deviation between any two members in $\mathcal{F}_t$:

$$\omega_t(s, a) = \sup_{\underline{f}, \overline{f} \sim \mathcal{F}_t} \|\overline{f}(\cdot|s, a) - \underline{f}(\cdot|s, a)\|_2. \tag{5}$$

3 Analysis of Provably-Efficient MBRL Algorithms

In this section, we analyze the central ideas and limitations of greedy model exploitation algorithms as well as two theoretically justified frameworks: optimistic algorithms and posterior sampling algorithms. The pseudocode of the algorithms for this section can be found in Appendix F.

3.1 Greedy Model Exploitation Reinforcement Learning

Before introducing the provably-efficient algorithms, we first analyze the greedy model exploitation algorithms that are broadly adopted by MBRL. In this framework, the agent takes actions assuming that the learned model sufficiently accurately resembles the real environment.

Algorithms that lie in this category can be further divided into two groups: model-based planning and model-augmented policy optimization. Compared to model-free algorithms [39, 12], model-augmented policy optimization facilitates learning by leveraging the acquired model. For instance, Dyna agents [46, 15, 13] optimize policies using model-free learners with model-generated data. And the learned model can also be exploited for back-propagation through paths [9, 7] or value expansion [11, 4]. On the other hand, model-based planning, or model predictive control (MPC) [29, 30], directly generates optimal action sequences under the model in a receding horizon fashion.

However, greedily exploiting the model without effective exploration mechanisms will lead to suboptimal performance. The resulting policy can suffer from premature convergence, leaving the potentially high-reward region unexplored. Unlike supervised learning where the data is already provided, the transition data that the MBRL model fits is generated by the agent taking actions and interacting with the real environment. And such dual effect [3, 20], *i.e.*, the current action influences both the next state and the model uncertainty, is not considered by the greedy model exploitation algorithms. In this regard, several heuristic yet provable algorithms are proposed.

3.2 Optimistic Reinforcement Learning

The first way that enables provably-efficient exploration is to adopt the principle of *optimism in the face of uncertainty* (OFU) [42, 35, 8]. With OFU, the agent assigns to its policy an optimistically biased estimate of virtual value by *jointly* optimizing over the policies and models inside the confidence set $\mathcal{F}_t$. At iteration t, the OFU policy π_t is defined as:

$$\pi_t = \operatorname*{argmax}_{\pi} \max_{f_t \in \mathcal{F}_t} V_{\pi}^{f_t}. \tag{6}$$

Most analysis of optimistic RL algorithms can be abstracted as showing two properties: the virtual value V_{π}^{f} is sufficiently high, and it is close to the real value $V_{\pi}^{f^*}$ in the long run. To ensure that V_{π}^{f} is close to $V_{\pi}^{f^*}$ asymptotically, additional assumptions and complexity measure is required, *e.g.*, Lipschitz value function assumption and eluder dimension d_E [35, 33, 49]. Intuitively, the eluder dimension d_E captures how effectively the model learned from the observed data can extrapolate to future data. And it is induced to bound the sum of confidence set width $\sum_{t=1}^{T} \omega_t$, which counts for the dual effect and appears in the cumulative regret bound (c.f. Sec. 5.2 and Appendix Lemma 3). By doing so, previous works eventually prove the $\widetilde{\mathcal{O}}(\sqrt{d_E T})$ regret bound [35, 33, 34].

Nevertheless, for many practical models, the bound of complexity measure is strong. Even the simplest nonlinear models, *e.g.*, one-layer neural networks, do *not* have polynomially-bounded eluder dimension (at least exponential in dimension, see Appendix C) [10]. And the additional complexity caused by over-exploration is hidden in the eluder dimension, which further leads to inefficiency in practice. Specifically, with nonlinear models adopted, the cumulative model error causes large epistemic uncertainty when the imagined trajectories transit into unexplored regions, which further results in an unrealistically large optimistic return that drives agents for uninformative exploration. Such exploration steps are not only suboptimal, but also eliminate only a small portion of the model hypothesis [10]. Thus, convergence to the optimal policy is a slow process and may end up suboptimal in practical high-dimensional complex tasks. Recent works on MBRL also show the harm of OFU and shelve optimistic algorithms both theoretically and empirically [10, 26].

3.3 Posterior Sampling Reinforcement Learning

Another provable exploration mechanism is posterior sampling for reinforcement learning (PSRL) [43, 32, 33], based on Thompson Sampling (TS) [47, 38]. The algorithm begins with a prior distribution of the true model f^*. At each iteration t, a model f_t is sampled from the posterior $\phi(f|\mathcal{D}_t)$, and π_t is updated to be optimal under f_t.

$$f_t \sim \phi(\cdot|\mathcal{D}_t),\ \pi_t = \operatorname*{argmax}_{\pi} V_{\pi}^{f_t}. \tag{7}$$

The insight is to keep away from actions that are unlikely to be optimal in the real environment. And agents explore by taking different actions with different sampled models, which is also proven to achieve the asymptotic optimality [32, 33]. Unfortunately, the same model complexity measure in OFU is also required for PSRL. And executing actions that are only optimal under a single sampled model can cause learning policy degradation between successive iterations and the exploration is inefficient in practice [38, 37].

Specifically, the posterior sampling exploration leads to policy π_t that is completely determined by the sampled (imperfect) model f_t. Such aggressive update results in policies that deviate from the optimal π^* with suboptimality degree depending on the epistemic model uncertainty. And executing π_t is not intended to offer performance improvement for follow-up policy learning, but only to narrow down the model hypothesis set $\mathcal{F}_t$. However, when the model is with large generalization error, which is quantitatively formulated in the model complexity measure, elimination of the hypothesis will be slow and thus causes inefficient exploration steps and suboptimal performance.

4 Dual Conservative Policy Update

As analyzed in Section 3, the theoretically justified RL algorithms by taking dual effect into account, asymptotically achieve the global optimality. However, guaranteeing such optimality is hard in practice due to the inefficient exploration when the model complexity measure bound is strong, *e.g.*, for nonlinear models in complex high-dimensional environments. To avoid aggressive policy updates and over-exploration, policy improvement guarantee *during* training is the key to achieving high performance with finite executed timesteps, in addition to the hard global optimality.

In this regard, we propose *Dual Conservative Policy Update* (DCPU) that efficiently explores the environment. The policy is updated following two successive procedures, which we call *locally greedy update* and *conservative exploration update*. The pseudocode of DCPU is in Algorithm 1.

4.1 Locally Greedy Update

Although the asymptotic optimality of the OFU and PSRL policy is widely studied, additional care must be taken if we also expect the learning policy update can indeed bring improvements during training. Since obtaining the *globally* accurate model is difficult, the policy is first *locally* updated within a trust region to leverage the reliable optimality information after model estimation.

Specifically, when the transition data following the *past* policy π_{t-1} is well fitted at the supervised model learning procedure (Line 9 in Algorithm 1), the model uncertainty set centered around the trajectories of π_{t-1} is narrowed down, *i.e.*, the mean model $\tilde{f}_{t-1} = \mathbb{E}_{\mathcal{F}_{t-1}}[f_{t-1}]$ is *locally* accurate. Thus, as long as the policy update is constrained within a trust region characterized by π_{t-1}, the agent can *greedily* exploit the model and optimize the intermediate policy q_t to be optimal under $\tilde{f}_{t-1}$:

$$q_t = \operatorname*{argmax}_q V_q^{\tilde{f}_t}, \ s.t. \mathop{\mathbb{E}}_{s\sim\tilde{\rho}^{\pi_{t-1}}} \left[D_{\text{TV}}\left(q_t(\cdot|s)||\pi_{t-1}(\cdot|s)\right)\right] \le \eta_1, \tag{8}$$

where D_{TV} stands for the total variation and η_1 is a hyperparameter. The state visitation measure $\tilde{\rho}^{\pi_{t-1}}$ by running π_{t-1} on $\tilde{f}_t$ is $\tilde{\rho}^{\pi_{t-1}} = (1-\gamma)\sum_h \gamma^h P(s_h = s; \pi_{t-1}, \tilde{f}_{t-1})$.

4.2 Conservative Exploration Update

Algorithm 1 Dual Conservative Policy Update

Input: Model-based policy optimization algorithm MBPO.

```
1: for iteration $t = 1, ..., T$ do
2:     $q_t \leftarrow$ MBPO($\pi_{t-1}, \tilde{f}_{t-1}$, Eq.(8))
3:     Sample $N$ models $\{f_{t-1,n}\}_{n=1}^N$ from $\mathcal{F}_{t-1}$
4:     $\pi_t \leftarrow$ MBPO($q_t, \{f_{t-1,n}\}_{n=1}^N$, Eq.(9))
5:     for episode step $h = 1, ..., H$ do
6:         Execute $\pi_t$ in real environment
7:         Update $\mathcal{D}_t = \mathcal{D}_t \cup (s_h, a_h, r_h, s_{h+1})$
8:     end for
9:     Update model hypothesis $\mathcal{F}_t$
10: end for
11: return policy $\pi_T$
```

The intermediate policy q_t brings non-negative value improvement over the reactive policy π_{t-1}, benefiting from the locally greedy update. However, there is *no* guarantee that the asymptotic q_T is optimal. In fact, the region where the model uncertainty remains large will not be effectively explored if only locally greedy policy update and dithering exploration mechanism (*e.g.*, epsilon-greedy) is taken.

For this reason, we propose *conservative exploration update* for efficient exploration. Define $\rho_t^\pi = (1-\gamma)\sum_h \gamma^h \mathbb{E}_{f_t\sim\mathcal{F}_t}[P(s_h = s; \pi, f_t)]$ as the expected state visitation measure under policy π on $\mathcal{F}_t$. This gives rise to the update of reactive policy π_t:

$$\pi_t = \operatorname*{argmax}_\pi \mathop{\mathbb{E}}_{f_{t-1}\sim\mathcal{F}_{t-1}} \left[V_\pi^{f_{t-1}}\right], \ s.t. \mathop{\mathbb{E}}_{s\sim\rho_{t-1}^{q_t}} \left[D_{\text{TV}}\left(\pi_t(\cdot|s)||q_t(\cdot|s)\right)\right] \le \eta_2. \tag{9}$$

The intuition behind it is to execute actions that could plausibly be optimal under multiple sampled models while maintaining the improvement brought by the locally greedy update. We note two ingredients that make the conservative exploration update desirable. First, the policy π_t is optimized to maximize the *expected* value over the models, which provides the potential to perform well in the real environment. Compared to the aggressive PSRL update in Eq. (7), the update rule in Eq. (9) is conservative. When the model suffers from large bias, the PSRL policy can be far from the true optimal policy, while the proposed algorithm avoids such pitfalls, which lead to significant policy degradation and over-exploration during learning. We will show in Sec. 5 that DCPU has bounded policy improvement while still enjoying the asymptotically optimal policy guarantee inheriting from the posterior sampling algorithms. Second, the total variation constraint plays a role in the trade-off between exploration and exploitation, which is formalized in Sec. 5.1.

In Algorithm 1, the model-based policy optimization algorithm MBPO that obtains q_t and π_t can be either model-augmented optimization *e.g.*, performing policy gradient in a Dyna-style [46], back-propagation through model paths [7], or model-based planning, *e.g.*, MPC with policy network [50]. Pseudocode with different optimization choices can be found in Appendix D.

5 Analysis

In this section, we prove that the proposed algorithm not only achieves the optimal policy asymptotically but also enjoys monotonic policy improvement under certain conditions.

5.1 Policy Iterative Improvement

To begin with, we first make the assumption of the Lipschitz continuous value function, which is widely adopted by previous MBRL works [26, 5, 10].

Assumption 1. *At iteration t, the value function $V_\pi^{f_t}$ is Lipschitz continuous in the sense that $|V_\pi^{f_t}(s_1) - V_\pi^{f_t}(s_2)| \leq L^t \|s_1 - s_2\|_2$.*

We note that the above assumption is reasonable since many RL settings can be satisfied [10], *e.g.*, nonlinear models with stochastic Lipschitz policies and Lipschitz reward models.

Define the *expected width of confidence set* when running π as ω_t^π, *i.e.*, $\omega_t^\pi = \mathbb{E}_{s\sim\rho_t^\pi, a\sim\pi}\left[\omega_t(s,a)\right]$.

Let q_t^* represent the policy that satisfies the same η_1 TV constraint as q_t but is optimal under the *real* model, *i.e.*, $q_t^* = \operatorname{argmax}_q V_q^{f^*}$, $s.t. \, \mathbb{E}_{s\sim\tilde{\rho}^{\pi_{t-1}}}\left[D_{\text{TV}}\left(q_t^*(\cdot|s)||\pi_{t-1}(\cdot|s)\right)\right] \leq \eta_1$. Then under Assumption 1, we have the following policy iterative improvement bound.

Theorem 1. *Denote $\epsilon = \omega_{t-1}^{\pi_{t-1}}$. Then the policy improvement between successive iterations, i.e., the improvement of π_t over π_{t-1}, is bounded by*

$$V_{\pi_t}^{f^*} - V_{\pi_{t-1}}^{f^*} \geq \Delta_t(\eta_1) + \Delta_t(\eta_2) - L^{t-1}\omega_{t-1}^{\pi_t} + O(\epsilon + \eta_1), \tag{10}$$

where $\Delta_t(\eta_1) = V_{q_t^*}^{f^*} - V_{\pi_{t-1}}^{f^*}$ and $\Delta_t(\eta_2) = \mathbb{E}_{f_{t-1}}\left[V_{\pi_t}^{f_{t-1}} - V_{q_t}^{f_{t-1}}\right]$ are non-negative.

Notably, ϵ is a small number that is determined by the model supervision error. Ideally, with a rich model function class and a proper supervised learning setting, the expected local uncertainty ϵ approaches zero. Together with the small η_1, the $O(\epsilon + \eta_1)$ term is induced in Eq. (10). And both $\Delta_t(\eta_1)$ and $\Delta_t(\eta_2)$ are non-negative policy value improvements since q_t^* is locally (around π_{t-1}) optimal on the *real* model, and π_t maximizes the *expected* value within the η_2-trust-region.

Corollary 1. *For any η_1, there exists η_2 at iteration t that ensures at least $\Delta_t(\eta_1)$ policy improvement over iteration $t-1$, i.e.,*

$$V_{\pi_t}^{f^*} - V_{\pi_{t-1}}^{f^*} \geq \Delta_t(\eta_1). \tag{11}$$

The necessary condition for η_2 at iteration t to guarantee the $\Delta_t(\eta_1)$ monotonic improvement is

$$\eta_{2,t} \geq \frac{(1-\gamma)L^{t-1}}{2R}\omega_{t-1}^{\pi_t}. \tag{12}$$

The intuition for policy improvement is that if η_2 approaches zero, at least $\Delta_t(\eta_1)$ iterative improvement is brought by the local update. Thus, for a proper small η_2, policy π_t that is better than q_t can be easily found within the trust region. In Corollary 1, $\omega_{t-1}^{\pi_t}$ captures the model uncertainty under the *upcoming* policy π_t. Choosing η_2 as the RHS of Eq. (12) that adapts with the model epistemic uncertainty makes intuitive sense. If the model uncertainty is large when running policy π_t, which indicates that transiting into the corresponding under-explored region may offer valuable policy optimization signals, then $\eta_{2,t}$ is large and can thus encourage exploration. On the other hand, a tight constraint is induced for model exploitation if the model bias is small. Therefore, the *uncertainty-dependent* η_2-trust-region can help trade-off between exploration and exploitation.

5.2 Asymptotic Optimality Guarantee

We then prove that the DCPU algorithm achieves the optimal performance asymptotically.

As discussed earlier, additional model complexity measure is required. In this work, we adopt eluder dimension with exactly the same definition as in [33]. Denote the state-action space as $\mathcal{X} = \mathcal{S} \times \mathcal{A}$.

Definition 1. *(($\mathcal{F}, \varepsilon$)-dependence). If $x \in \mathcal{X}$ is ($\mathcal{F}, \varepsilon$)-dependent on $x_1, ..., x_n \subseteq \mathcal{X}$, then*

$$\forall f_1, f_2 \in \mathcal{F}, \sum_{i=1}^{n} \|f_1(x_i) - f_2(x_i)\|_2^2 \leq \varepsilon^2 \Rightarrow \|f_1(x) - f_2(x)\|_2 \leq \varepsilon.$$

$x \in \mathcal{X}$ is ($\mathcal{F}, \varepsilon$)-independent of $x_1, ..., x_n$ iff it does not satisfy the definition for dependence.

Definition 2. *(Eluder dimension. Osband and Van Roy, 2014). The eluder dimension $dim_E(\mathcal{F}, \varepsilon)$ is the length of the longest possible sequence of elements in $\mathcal{X}$ such that for some $\varepsilon' \geq \varepsilon$, every element is ($\mathcal{F}, \varepsilon'$)-independent of its predecessors.*

Proposition 1. *If for all $f \in \mathcal{F}$, $\|f\|_2 \leq C$, then the Bayes expected regret is bounded by:*

$$\text{BayesRegret}(T, \pi, \phi) \leq 3L\frac{T}{T-1}\left(HCd_E + 4\sqrt{Td_E\beta_T}\right) + 3C + 3, \tag{13}$$

where $d_E = dim_E\left(\mathcal{F}, T^{-1}\right)$ and $L = \mathbb{E}[L^t] = \mathbb{E}[L^*]$.

What remains is to bound the confidence parameter β_T since it is the only term that still depends on T. Using results from previous works [35, 33], we show that the regret is indeed sublinear in T.

Theorem 2. *Let $N(\mathcal{F}, \alpha, \|\cdot\|_2)$ be the α-covering number of $\mathcal{F}$ and denote $n = \log(4N(\mathcal{F}, \frac{1}{T^2}, \|\cdot\|_2)T)$. If the model is with bounded mean $\|f\|_2 \leq C$ and the true dynamics is with additive σ-sub-Gaussian noise, then*

$$\text{BayesRegret}(T, \pi, \phi) \leq 3L\frac{T}{T-1}\tilde{D}(\mathcal{F}) + 3C + 3 \tag{14}$$

where $\tilde{D}(\mathcal{F}) = \left(HCd_E + 8\sigma\sqrt{2nd_ET} + 8\sqrt{d_E\left(4C + \sqrt{2\sigma^2\log(16T^3)}\right)}\right)$.

Remark 1. *By ignoring terms that are logarithmic in T, the regret bound is $\widetilde{\mathcal{O}}(L\sigma\sqrt{nd_ET})$. By replacing n with the Kolmogorov dimension d_K of the function class $\mathcal{F}$ [33], the $\widetilde{\mathcal{O}}(\sqrt{d_Kd_ET})$ near-optimal regret bound is obtained.*

The d_E term captures how effectively the unobserved transitions can be inferred from the observed samples. And the d_K term captures the sensitivity of $\mathcal{F}$ to statistical overfitting (c.f. Appendix C for definition). Theorem 2 ensures that DCPU eventually achieves arbitrarily close to the optimality.

5.3 Limitations and Practical Algorithm

We have shown that DCPU policies are asymptotically optimal, and choosing a proper η_2 brings iterative policy improvement. However, in Corollary 1, the *necessary* condition of η_2 for monotonic improvement does not immediately suggest the choice of η_2 by itself. Although the uncertainty-dependent η_2 trades off between exploration and exploitation, such design needs additional hyperparameter tuning and, even worse, is still not *sufficient* for strict iterative improvement. A possible modification is to adopt more sophisticated model exploitation approaches in the local update procedure for larger $\Delta_t(\eta_1)$ and looser restrictions on η_2, which we'll discuss in more detail at Sec. 8. In practice, we can simply set η_2 as a hyperparameter. Empirical results and ablation studies on the η_2-trust-region also show that a proper fixed η_2 suffices to offer superior performance.

Besides, the total variation constraint is replaced by the commonly used KL divergence [39, 2] by noticing that $D_{\text{TV}}\left(P||Q\right) \leq \sqrt{\frac{1}{2}D_{\text{KL}}\left(P||Q\right)}$. And we follow previous works [8, 18] to use neural network ensembles for model estimation and use calibrations [22] for accurate uncertainty measure.

6 Additional Related Work

In addition to the mentioned greedy model exploitation algorithms, some MBRL works also concern policy improvement. SLBO [26] provides a trust-region policy optimization framework based on OFU. However, the conditions for monotonic improvement cannot be satisfied by most parameterized models [26, 10], which leads to a greedy algorithm in practice. The close prior works also contain DPI [44] and GPS [23, 28]. The locally greedy update procedure of DCPU shares similarities with DPI and GPS, as the intermediate policies are optimized with the locally accurate model. Sun et al. [44] show that local updates can bring improvement. However, the successive *supervised* reactive policy update procedure in DPI weakens the theorem and poses additional challenges for effective exploration, resulting in greedy model exploitation and suboptimal policies. In fact, greedy algorithms are provably optimal only in very limited cases, *e.g.*, linear-quadratic regulator (LQR) setting [27].

The well-known OFU principle has shown to achieve an optimal $\widetilde{\mathcal{O}}(\sqrt{T})$ regret rate when applied to online LQR [1], tabular MDPs [14] and linear MDPs [16]. Among them, HUCRL [8] is a deep MBRL algorithm that is recently proposed to deal with the joint optimization intractability. PSRL as another provable MBRL algorithm, is shown to match the statistical efficiency of any standard OFU-RL algorithm in Bayesian expectation up to constant factors [34]. Notably, Russo and Van Roy

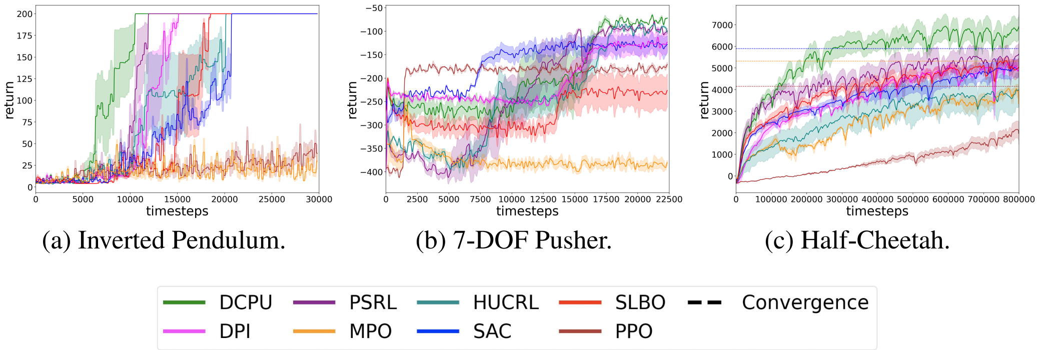
[35, 36] unify many previous bounds and show that a broad class of online decision problems satisfy $\widetilde{\mathcal{O}}(\sqrt{nd_E T})$ regret, where d_E is the eluder dimension and n is the log-covering number measuring the sensitivity to statistical overfitting. And in the RL setting, expected regret $\widetilde{\mathcal{O}}(\sqrt{d_K d_E T})$ is given in [33], which is exactly the same as ours. The bound of eluder dimension is given in [33] for linear functions, and is analyzed in [10] for nonlinear functions. Other theoretical MBRL works that also study families of dynamical models with restricted complexity measure include witness rank [45], linear dimensionality [52] and sequential Rademacher complexity [10]. Some works hide such model complexity by making strong assumptions, *e.g.*, the Lipschitz continuity of model uncertainty [8].

7 Experiments

7.1 Comparisons to Prior RL Algorithms

In this section, we conduct experiments on several Mujoco benchmark control tasks [48], including inverted pendulum swing-up, pusher goal-reaching, and half-cheetah locomotion. From low-dimensional to high-dimensional tasks, the state-action dimension and episode length are increasing.

We compare the performance of the DCPU algorithm to other state-of-the-art model-based and model-free RL algorithms. All the experiments are repeated with 5 random seeds. Nonlinear neural network models are adopted in the evaluated MBRL algorithms. For DCPU, we use Dyna-style `MBPO` algorithm as default (Algorithm 2 in Appendix). And η_1 and η_2 are set to be 0.1. Implementation details including hyperparameter choices are provided in Appendix E.



(a) Inverted Pendulum.

(b) 7-DOF Pusher.

(c) Half-Cheetah.

Figure 1: Training curves for Mujoco tasks. In the half-cheetah task where the curves of model-free algorithms have not converged, the dashed lines represent the asymptotic performance at convergence.

In Figure 1, the model-based algorithms that we evaluate include greedy MBRL methods and provably-efficient algorithms. The most closed related works are the provable OFU and PSRL algorithms. For OFU algorithms, we compare with HUCRL [8], which uses the same calibrated ensemble models like ours. Since OFU and PSRL suffer from over-exploration and aggressive policy updates especially in high-dimensional environments, the exploration is inefficient in practice and leads to suboptimal convergence in the half-cheetah locomotion task. On the contrary, DCPU achieves higher performance within the executed timesteps benefiting from the learning policy improvement.

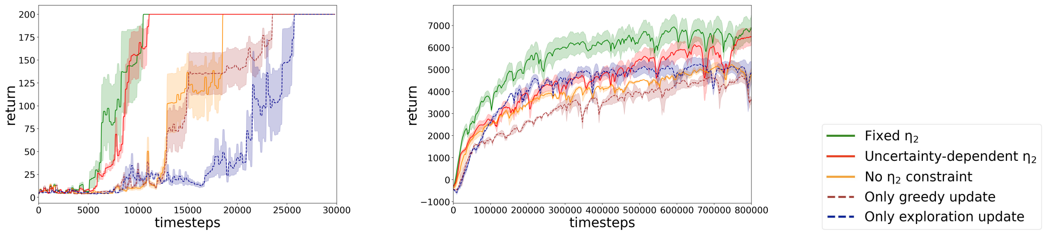
For greedy model exploitation MBRL algorithms, we examine DPI [44] and SLBO [26], both of which are proved to satisfy monotonic improvement under certain conditions. In the low-dimensional inverted pendulum environment, agents learn efficiently and converge faster than model-free algorithms by leveraging the acquired dynamical model. However, due to the lack of effective exploration mechanisms, the greedy model exploitation results in premature convergence and worse asymptotic performance compared to DCPU in the more complex pusher and half-cheetah tasks. As DCPU and DPI share similarities in the local update procedure, the results highlight the importance of the conservative exploration update of DCPU, which we'll discuss in more detail in the ablation study.

Another important RL research area is model-free RL, which offers appreciable returns with more data and timesteps. In addition to the strong baseline SAC [12], we also compare with trust-region model-free algorithms: PPO [40] and MPO [2]. As a model-based RL algorithm, DCPU achieves higher asymptotic performance while requiring significantly fewer samples, *e.g.*, 5, 7 and 10 times fewer samples than SAC, MPO, and PPO respectively on the half-cheetah task.

7.2 Ablation Study

We conduct ablation studies to further validate the benefits of the two update procedures of DCPU and investigate how different designs of the η_2-trust-region affect the algorithm. From Fig. 2, one can observe that the policies obtained *only* with the locally greedy update or the conservative exploration update lag behind DCPU in terms of efficiency and asymptotic performance. In inverted pendulum, the low-dimensional dynamics is fitted with high accuracy. And greedily exploiting the model thus achieves considerable performance. On the other hand, the conservative update procedure plays a key role in exploration and better asymptotic performance in the more complex half-cheetah task.

In Sec. 7.1, the constraint threshold η_2 is set to be a fixed hyperparameter. Here, we also evaluate the uncertainty-dependent $\eta_{2,t} = \alpha\omega_{t-1}^{\pi_t}$ suggested by Corollary 1. By choosing a proper coefficient α, this design achieves similar performance with the fixed η_2, but requires additional hyperparameter tuning. Besides, removing the η_2 constraint results in suboptimal behaviors in the high-dimensional half-cheetah task since the model is biased and the policy update involves less optimality information when model uncertainty ω_{t-1} is large in the complex environments. Thus, the unconstrained optimization leads to unclear policy improvement compared to the update within the η_2-trust-region.



(a) Inverted Pendulum.

(b) Half-Cheetah.

Figure 2: Ablation studies on the benefits of the two update procedures and different designs of η_2.

8 Conclusion & Discussions

In this work, we present Dual Conservative Policy Update, a provable model-based reinforcement learning algorithm. By optimizing the policy with *locally greedy update* and *conservative exploration update* iteratively, DCPU explores while avoiding inefficient over-exploration and aggressive policy update. We theoretically bound the policy iterative improvement and the cumulative Bayes expected regret, which offers the properties of *monotonic policy improvement* and *asymptotic optimality*, respectively. Practical DCPU algorithm by achieving high performance within the executed timesteps, shows superiority on several Mujoco tasks over a wide range of state-of-the-art RL algorithms.

Considering the simplicity and generalizability of DCPU, we expect our idea that achieving provable exploration with improvement guarantees can give rise to a variety of algorithm designs. Specifically, in the greedy update procedure, the constrained policy optimization is a *very naive* way to exploit the local model. To obtain higher iteration improvement, more sophisticated model-based policy learning methods can be adopted, *e.g.*, reality-aware MBPO [53], approximating value functions for better policies beyond local solutions [25], and probabilistic action ensembles [31]. Besides, the model supervised learning can also be more efficient by, *e.g.*, incorporating the long-term predictions [17].

Our work also opens some new problems. In Corollary 1, the uncertainty-dependent η_2 does not suffice to provide strict iterative improvement. A key reason is the naive model exploitation and limited policy improvement in the locally greedy update procedure. To obtain larger $\Delta_t(\eta_1)$, more sophisticated model exploitation such as the ones discussed above can be adopted. In this way, less restrictions will be imposed on η_2 since $\Delta_t(\eta_1) + \Delta_t(\eta_2) - L^{t-1}\omega_{t-1}^{\pi_t} \geq 0$ is easier to guarantee. Although obtaining the theoretical bound may be more challenging with such modifications, practical improvements can be obtained and we would like to explore the inspired algorithms as future work.

Broader Impact. For real-world applications, *e.g.*, robotics, interactions with the system imply energy or economic costs. With practical efficiency, DCPU reduces the training investment and is aligned with the principle of responsible AI. However, as an RL algorithm, DCPU is unavoidable to introduce safety concerns, *e.g.*, self-driving cars make mistakes during RL training. Although DCPU does not explicitly address them, it may be used in conjunction with safety controllers to minimize negative impacts, while drawing on its powerful MBRL roots to enable efficient learning.

References

[1] Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret bounds for the adaptive control of linear quadratic systems. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 1–26. JMLR Workshop and Conference Proceedings, 2011.

[2] Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

[3] Yaakov Bar-Shalom and Edison Tse. Dual effect, certainty equivalence, and separation in stochastic control. *IEEE Transactions on Automatic Control*, 19(5):494–500, 1974.

[4] Jacob Buckman, Danijar Hafner, George Tucker, Eugene Brevdo, and Honglak Lee. Sample-efficient reinforcement learning with stochastic ensemble value expansion. *arXiv preprint arXiv:1807.01675*, 2018.

[5] Sayak Ray Chowdhury and Aditya Gopalan. Online learning in kernelized markov decision processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3197–3205. PMLR, 2019.

[6] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *arXiv preprint arXiv:1805.12114*, 2018.

[7] Ignasi Clavera, Violet Fu, and Pieter Abbeel. Model-augmented actor-critic: Backpropagating through paths. *arXiv preprint arXiv:2005.08068*, 2020.

[8] Sebastian Curi, Felix Berkenkamp, and Andreas Krause. Efficient model-based reinforcement learning through optimistic policy search and planning. *arXiv preprint arXiv:2006.08684*, 2020.

[9] Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472. Citeseer, 2011.

[10] Kefan Dong, Jiaqi Yang, and Tengyu Ma. Provable model-based nonlinear bandit and reinforcement learning: Shelve optimism, embrace virtual curvature. *arXiv preprint arXiv:2102.04168*, 2021.

[11] Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I Jordan, Joseph E Gonzalez, and Sergey Levine. Model-based value estimation for efficient model-free reinforcement learning. *arXiv preprint arXiv:1803.00101*, 2018.

[12] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018.

[13] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[14] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(4), 2010.

[15] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *arXiv preprint arXiv:1906.08253*, 2019.

[16] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.

[17] Nan Rosemary Ke, Amanpreet Singh, Ahmed Touati, Anirudh Goyal, Yoshua Bengio, Devi Parikh, and Dhruv Batra. Modeling the long term future in model-based reinforcement learning. In *International Conference on Learning Representations*, 2018.

[18] Rahul Kidambi, Jonathan Chang, and Wen Sun. Optimism is all you need: Model-based imitation learning from observation alone. *arXiv preprint arXiv:2102.10769*, 2021.

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

[20] Edgar D Klenske and Philipp Hennig. Dual control for approximate bayesian reinforcement learning. *The Journal of Machine Learning Research*, 17(1):4354–4383, 2016.

[21] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014. Citeseer, 2000.

[22] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, pages 2796–2804. PMLR, 2018.

[23] Sergey Levine and Pieter Abbeel. Learning neural network policies with guided policy search under unknown dynamics. In *NIPS*, volume 27, pages 1071–1079. Citeseer, 2014.

[24] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[25] Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch. Plan online, learn offline: Efficient learning and exploration via model-based control. *arXiv preprint arXiv:1811.01848*, 2018.

[26] Yuping Luo, Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. *arXiv preprint arXiv:1807.03858*, 2018.

[27] Horia Mania, Stephen Tu, and Benjamin Recht. Certainty equivalence is efficient for linear quadratic control. *arXiv preprint arXiv:1902.07826*, 2019.

[28] William Montgomery and Sergey Levine. Guided policy search as approximate mirror descent. *arXiv preprint arXiv:1607.04614*, 2016.

[29] Manfred Morari and Jay H Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4-5):667–682, 1999.

[30] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7559–7566. IEEE, 2018.

[31] Masashi Okada and Tadahiro Taniguchi. Variational inference mpc for bayesian model-based reinforcement learning. In *Conference on Robot Learning*, pages 258–272. PMLR, 2020.

[32] Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. *arXiv preprint arXiv:1306.0940*, 2013.

[33] Ian Osband and Benjamin Van Roy. Model-based reinforcement learning and the eluder dimension. *arXiv preprint arXiv:1406.1853*, 2014.

[34] Ian Osband and Benjamin Van Roy. Why is posterior sampling better than optimism for reinforcement learning? In *International Conference on Machine Learning*, pages 2701–2710. PMLR, 2017.

[35] Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *NIPS*, pages 2256–2264. Citeseer, 2013.

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

[37] Daniel Russo and Benjamin Van Roy. Satisficing in time-sensitive bandit learning. *arXiv preprint arXiv:1803.02855*, 2018.

[38] Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A tutorial on thompson sampling. *arXiv preprint arXiv:1707.02038*, 2017.

[39] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[40] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[41] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[42] Alexander L Strehl and Michael L Littman. A theoretical analysis of model-based interval estimation. In *Proceedings of the 22nd international conference on Machine learning*, pages 856–863, 2005.

[43] Malcolm Strens. A bayesian framework for reinforcement learning. In *ICML*, volume 2000, pages 943–950, 2000.

[44] Wen Sun, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. Dual policy iteration. *arXiv preprint arXiv:1805.10755*, 2018.

[45] Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*, pages 2898–2933. PMLR, 2019.

[46] Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pages 216–224. Elsevier, 1990.

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

[48] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[49] Ruosong Wang, Russ R Salakhutdinov, and Lin Yang. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *Advances in Neural Information Processing Systems*, 33, 2020.

[50] Tingwu Wang and Jimmy Ba. Exploring model-based planning with policy networks. *arXiv preprint arXiv:1906.08649*, 2019.

[51] Tian Xu, Ziniu Li, and Yang Yu. On value discrepancy of imitation learning. *arXiv preprint arXiv:1911.07027*, 2019.

[52] Lin Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pages 10746–10756. PMLR, 2020.

[53] Guangxiang Zhu, Minghao Zhang, Honglak Lee, and Chongjie Zhang. Bridging imagination and reality for model-based deep reinforcement learning. *arXiv preprint arXiv:2010.12142*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Sec. 5.3 and Sec. 8.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Sec. 8.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Assumption 1.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix A.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Our code can be found in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A] Our code can be found in the supplemental material.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Proofs

A.1 Proof of Theorem 1

Proof. By decomposing the value discrepancy between iterations, we have:

$$V_{\pi_t}^{f^*} - V_{\pi_{t-1}}^{f^*} = V_{\pi_t}^{f^*} - V_{q_t}^{f^*} + V_{q_t}^{f^*} - V_{\pi_{t-1}}^{f^*} \tag{15}$$

The $V_{q_t}^{f^*} - V_{\pi_{t-1}}^{f^*}$ term can be bounded using Lemma 2:

$$V_{q_t}^{f^*} - V_{\pi_{t-1}}^{f^*} \geq \Delta_t(\eta_1) + O(\epsilon + \eta_1), \tag{16}$$

where $\Delta_t(\eta_1) = V_{q_t^*}^{f^*} - V_{\pi_{t-1}}^{f^*}$ is a non-negative value improvement.

The insight behind the above inequality is that resulting from the constraint between π_{t-1} and q_t, the trajectories of executing policy π_{t-1} and q_t should be similar. Thus, as long as the model is *locally* accurate around the trajectory of π_{t-1}, the dynamics when following policy q_t is also well modeled. And obtaining a locally accurate model is easier because at iteration t, we have already collected the transition data following the *past* policy π_{t-1}. Thus, the $O(\epsilon + \eta_1)$ term is induced to represent the small model estimation error $\epsilon = \omega_{t-1}^{\pi_{t-1}}$ and the small divergence η_1 between π_{t-1} and q_t.

With a similar intuition, we expect the value discrepancy between $V_{q_t}^{f_{t-1}}$ and $V_{q_t}^{f^*}$ to be small. Specifically, denote $\omega_m = \max_s \omega_t(s)$, then with the small ϵ and small η_1, we get

$$\begin{aligned}
\mathop{\mathbb{E}}_{f_{t-1}}\left[\left|V_{q_t}^{f_{t-1}} - V_{q_t}^{f^*}\right|\right] \leq & L^{t-1}\omega_{t-1}^{q_t} \\
= & L^{t-1}\omega_{t-1}^{\pi_{t-1}} + L^{t-1}\left|\omega_{t-1}^{q_t} - \omega_{t-1}^{\pi_{t-1}}\right| \\
\leq & L^{t-1}\epsilon + 2L^{t-1}\omega_m\eta_1 \\
= & O(\epsilon + \eta_1)
\end{aligned} \tag{17}$$

Next, the $V_{\pi_t}^{f^*} - V_{q_t}^{f^*}$ term in Eq. (15) is bounded by

$$\begin{aligned}
V_{\pi_t}^{f^*} - V_{q_t}^{f^*} &= \mathop{\mathbb{E}}_{f_{t-1}}\left[V_{\pi_t}^{f^*} - V_{q_t}^{f^*}\right] \\
&= \mathop{\mathbb{E}}_{f_{t-1}}\left[V_{\pi_t}^{f^*} - V_{\pi_t}^{f_{t-1}} + V_{\pi_t}^{f_{t-1}} - V_{q_t}^{f_{t-1}} + V_{q_t}^{f_{t-1}} - V_{q_t}^{f^*}\right] \\
&= \mathop{\mathbb{E}}_{f_{t-1}}\left[V_{\pi_t}^{f^*} - V_{\pi_t}^{f_{t-1}} + V_{\pi_t}^{f_{t-1}} - V_{q_t}^{f_{t-1}}\right] + O(\epsilon + \eta_1) \\
&= \Delta_t(\eta_2) + \mathop{\mathbb{E}}_{f_{t-1}}\left[V_{\pi_t}^{f^*} - V_{\pi_t}^{f_{t-1}}\right] + O(\epsilon + \eta_1) \\
&\geq \Delta_t(\eta_2) - L^{t-1}\omega_{t-1}^{\pi_t} + O(\epsilon + \eta_1),
\end{aligned} \tag{18}$$

where $\Delta_t(\eta_2) = \mathbb{E}_{f_{t-1}}\left[V_{\pi_t}^{f_{t-1}} - V_{q_t}^{f_{t-1}}\right]$ is the expected value improvement within the trust region determined by η_2 and is also non-negative according to the definition of π_t. The last inequality holds following Lemma 1.

Finally, by combining Eq. (15), (16), and (18), we obtain the policy iterative improvement bound

$$V_{\pi_t}^{f^*} - V_{\pi_{t-1}}^{f^*} \geq \Delta_t(\eta_1) + \Delta_t(\eta_2) - L^{t-1}\omega_{t-1}^{\pi_t} + O(\epsilon + \eta_1). \tag{19}$$

□

A.2 Proof of Corollary 1

Proof. By omitting the $O(\epsilon + \eta_1)$ term in Theorem 1, we have

$$V_{\pi_t}^{f^*} - V_{\pi_{t-1}}^{f^*} \geq \Delta_t(\eta_1) + \Delta_t(\eta_2) - L^{t-1}\omega_{t-1}^{\pi_t}, \tag{20}$$

where $\Delta_t(\eta_1) = V_{q_t^*}^{f^*} - V_{\pi_{t-1}}^{f^*}$ and $\Delta_t(\eta_2) = \mathbb{E}_{f_{t-1}}\left[V_{\pi_t}^{f_{t-1}} - V_{q_t}^{f_{t-1}}\right]$.

We first show that we can always find a η_2 such that the value improvement at successive iterations is at least $\Delta_t(\eta_1)$.

Consider the conservative exploration update step from q_t to π_t. Consider the case when η_2 is chosen to approach zero, which is reasonable if the model is believed to be globally accurate and the greedy policy q_t is already near-optimal. Such constraint results in the same policy $\pi_t = q_t$ and zero improvement $\Delta_t(\eta_2)$. And enlarging η_2 provides the potential to offer positive improvement $\Delta_t(\eta_2)$ as the constraint gets looser. So the η_2 that guarantees at least $\Delta_t(\eta_1)$ improvement always exists.

Now we derive the necessary conditions for such monotonic improvement. Using the results from Lemma 5, we have

$$\Delta_t(\eta_2) = \mathop{\mathbb{E}}_{f_{t-1}}\left[\left|V_{\pi_t}^{f_{t-1}} - V_{q_t}^{f_{t-1}}\right|\right] \le \frac{2R}{1-\gamma} \mathop{\mathbb{E}}_{s\sim\rho_{t-1}^{q_t}}\left[D_{\mathrm{TV}}\left(\pi_t(\cdot|s)||q_t(\cdot|s)\right)\right] \le \frac{2R}{1-\gamma}\eta_{2,t}. \tag{21}$$

To ensure the conservative exploration update step at iteration t, *i.e.*, updating policy from q_t to π_t, gives performance improvement, we need

$$\Delta_t(\eta_2) \ge L^{t-1}\omega_{t-1}^{\pi_t}, \tag{22}$$

Since we also have $\Delta_t(\eta_2) \le \frac{2R}{1-\gamma}\eta_{2,t}$, the necessary condition to guarantee at least $\Delta_t(\eta_1)$ value improvement from π_{t-1} to π_t is

$$\eta_{2,t} \ge \frac{(1-\gamma)L^{t-1}}{2R}\omega_{t-1}^{\pi_t}, \tag{23}$$

□

A.3 Proof of Proposition 1

Proof. Define π_{f_t} as the best-performing policy under a sampled model f_t, *i.e.*, $\pi_{f_t} = \max_\pi V_\pi^{f_t}$. And the difference between π_{f_t} and the expected value maximization policy π_t in DCPU should be noticed: when taking expectation over f_t, π_{f_t} changes with different f_t. On the contrary, π_t is a single policy that aims to maximize the expected policy value (within certain trust region).

Whenever $f^* \in \mathcal{F}_t$, the value expectation over the model posterior satisfies

$$\mathop{\mathbb{E}}_{f_t}\left[V_{\pi^*}^{f^*} - V_{\pi_t}^{f_t} \mid \mathcal{D}_t\right] \le 0, \tag{24}$$

The reason is that the true dynamics f^* and the sampled model f_t are identically distributed when conditioned upon $\mathcal{D}_t$, which is also known as posterior sampling lemma [33]. And for every sampled f_t, we have $V_{\pi^*}^{f_t} \le V_{\pi_{f_t}}^{f_t}$ according to the definition of π_{f_t}.

We adopt the notation $\mathfrak{D}_{\mathcal{F}_t}(\pi)$ and $\tilde{\mathfrak{D}}_{\mathcal{F}_t}(\pi)$ to represent the expected value discrepancy when running policy π on the estimated model and on the real model f^*, mean model $\tilde{f}_t$, respectively, *i.e.*,

$$\mathfrak{D}_{\mathcal{F}_t}(\pi) = \mathop{\mathbb{E}}_{f_t\sim\mathcal{F}_t}[V_\pi^{f_t} - V_\pi^{f^*}],\ \tilde{\mathfrak{D}}_{\mathcal{F}_t}(\pi) = \mathop{\mathbb{E}}_{f_t\sim\mathcal{F}_t}[V_\pi^{f_t} - V_\pi^{\tilde{f}_t}], \tag{25}$$

where $\tilde{f}_t$ is the mean model estimation, *i.e.*, the expectation of the estimated model parameters with respect to the posterior distribution at iteration t.

Then the expectation of regret $\mathfrak{R}_t$ over iteration t can be decomposed by

$$\begin{aligned}
\mathbb{E}\left[\mathfrak{R}_t\right] &= \mathbb{E}\left[V_{\pi^*}^{f^*} - V_{\pi_t}^{f^*}\right] \\
&= \mathbb{E}\left[V_{\pi^*}^{f^*} - V_{\pi_{f_t}}^{f_t} + V_{\pi_{f_t}}^{f_t} - V_{\pi_t}^{f^*}\right] \\
&\le \mathbb{E}\left[V_{\pi_{f_t}}^{f_t} - V_{\pi_t}^{f^*}\right] \\
&= \mathbb{E}\left[V_{\pi_{f_t}}^{f_t} - V_{\pi_t}^{f_t} + V_{\pi_t}^{f_t} - V_{\pi_t}^{f^*}\right] \\
&= \mathbb{E}\left[V_{\pi_{f_t}}^{f_t} - V_{\pi_t}^{f_t}\right] + \mathfrak{D}_{\mathcal{F}_t}(\pi_t)
\end{aligned} \tag{26}$$

The terms in the expectation can be further decomposed as follows:

$$\begin{aligned}\mathop{\mathbb{E}}_{f_t}\left[V^{f_t}_{\pi_{f_t}} - V^{f_t}_{\pi_t}\right] &= \mathop{\mathbb{E}}_{f_t}\left[V^{f_t}_{\pi_{f_t}} - V^{\tilde{f}_t}_{\pi_{f_t}} + V^{\tilde{f}_t}_{\pi_{f_t}} - V^{f_t}_{\pi_t}\right] \\ &= \tilde{\mathfrak{D}}_{\mathcal{F}_t}(\pi_{f_t}) + \mathop{\mathbb{E}}_{f_t}\left[V^{\tilde{f}_t}_{\pi_{f_t}} - V^{f_t}_{\pi_t}\right].\end{aligned} \tag{27}$$

Further, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{f_t}\left[V^{\tilde{f}_t}_{\pi_{f_t}} - V^{f_t}_{\pi_t}\right] &\leq \mathop{\mathbb{E}}_{f_t}\left[V^{\tilde{f}_t}_{q_t} - V^{f_t}_{\pi_t}\right] \\ &\leq \mathop{\mathbb{E}}_{f_t}\left[V^{\tilde{f}_t}_{q_t} - V^{f_t}_{q_t}\right] \\ &= \tilde{\mathfrak{D}}_{\mathcal{F}_t}(q_t),\end{aligned} \tag{28}$$

where the first inequality holds since q_t is the intermediate policy that maximizes the value under model $\tilde{f}_t$. And the second inequality holds since π_t maximizes the expected value, which indicates $\mathbb{E}\left[V^{f_t}_{\pi_t}\right] \geq \mathbb{E}\left[V^{f_t}_{q_t}\right]$.

Combining Eq. (26), (27) and (28), we get:

$$\begin{aligned}\mathbb{E}\left[\mathfrak{R}_t\right] &\leq \mathbb{E}\left[V^{f_t}_{\pi_{f_t}} - V^{f_t}_{\pi_t} + V^{f_t}_{\pi_t} - V^{f^*}_{\pi_t}\right] \\ &\leq \mathfrak{D}_{\mathcal{F}_t}(\pi_t) + \tilde{\mathfrak{D}}_{\mathcal{F}_t}(\pi_{f_t}) + \tilde{\mathfrak{D}}_{\mathcal{F}_t}(q_t)\end{aligned} \tag{29}$$

Let $A = \{f^*, f_k \in \mathcal{F}_k \ \forall k\}$. Recall that the model is bounded by $\|f\|_2 \leq C$. Then we have:

$$\begin{aligned}\text{BayesRegret}(T, \pi, \phi) =& \mathbb{E}\left[\text{Regret}(T, \pi, f^*)\right] \\ =& \mathbb{E}\left[\sum_{t=1}^{T} \mathfrak{R}_t\right] \\ \leq& \sum_{t=1}^{T}\Big(\mathfrak{D}_{\mathcal{F}_t}(\pi_t) + \tilde{\mathfrak{D}}_{\mathcal{F}_t}(\pi_{f_t}) + \tilde{\mathfrak{D}}_{\mathcal{F}_t}(q_t)\Big) \\ \leq& \sum_{t=1}^{T}\sum_{h=1}^{H} 3\{\mathbb{E}\left[L^t \mid A\right]\omega_t + 4\delta C\},\end{aligned} \tag{30}$$

where δ is the variable that determines the probability of containing the real model and characterizes the control parameter $\beta_t(\delta, \alpha)$ defined in Eq. (43). And the last inequality can be proven by using very similar techniques as in Lemma 1. The slight differences lie in that the real model in Lemma 1 is replaced by the mean model and the $4\delta C$ term is induced when the real model is not contained in the model hypothesis.

For $L = \mathbb{E}[L^t] = \mathbb{E}[L^*]$. And we have

$$\mathbb{E}\left[L^t \mid A\right] \leq \frac{\mathbb{E}[L^t]}{P(A)} \leq \frac{L}{1 - 4\delta} \tag{31}$$

Then we can apply the off-the-shelf sum of set width bound in Lemma 3 to give the expected regret bound. By setting $\delta = \frac{1}{4T}$ and denote $\omega_t = \sup_{\underline{f}, \overline{f} \sim \mathcal{F}_t} \|\overline{f}(s, \cdot) - \underline{f}(s, \cdot)\|_2$,

$$\begin{aligned}\text{BayesRegret}(T, \pi, \phi) \leq& 3L\frac{T}{T-1}\sum_{t=1}^{T}\sum_{h=1}^{H}\omega_t + 3C \\ \leq& 3L\frac{T}{T-1}\left(HCd_E + 4\sqrt{Td\beta_T(\frac{1}{4T}, \alpha)}\right) + 3C + 3,\end{aligned} \tag{32}$$

where d_E is shorthand for $\dim_E(\mathcal{F}, T^{-1})$. □

A.4 Proof of Theorem 2

Proof. From Proposition 1, the expected regret is bounded by:

$$\text{BayesRegret}(T, \pi, \phi) \leq 3L \frac{T}{T-1} \left(HCd_E + 4\sqrt{Td_E \beta_T(\frac{1}{4T}, \alpha)} \right) + 3C + 3 \tag{33}$$

From Lemma 4, by setting $\alpha = \frac{1}{T^2}$ and plugging into Eq. (43), we have the following confidence parameter that can guarantee the real dynamics is contained in the confidence set with high probability:

$$\beta_T(\frac{1}{4T}, \frac{1}{T^2}) = 8\sigma^2 \log(4nT) + \frac{2}{T} \left(8C + \sqrt{8\sigma^2 \log(16T^3)} \right), \tag{34}$$

where $n = \log(4N(\mathcal{F}, \frac{1}{T^2}, \|\cdot\|_2)T)$.

Plugging Eq. (34) into Eq. (33) and apply triangle inequality, the bound is obtained:

$$\text{BayesRegret}(T, \pi, \phi) \leq 3L \frac{T}{T-1} \tilde{D}(\mathcal{F}) + 3C + 3 \tag{35}$$

where $\tilde{D}(\mathcal{F}) = \left(HCd_E + 8\sigma\sqrt{2nd_E T} + 8\sqrt{d_E \left(4C + \sqrt{2\sigma^2 \log(16T^3)} \right)} \right)$. □

B Useful Lemmas

Lemma 1. *If the real model is in the model hypothesis, i.e., $f^* \in \bigcap_{t=1}^{T} \mathcal{F}_t$, then the expected value discrepancy when executing policy π under the estimated model $f_t \sim \mathcal{F}_t$ and under the real model f^* is bounded by*

$$\mathop{\mathbb{E}}_{f_t \sim \mathcal{F}_t} \left[\left| V_\pi^{f_t} - V_\pi^{f^*} \right| \right] \leq L^t \omega_t^\pi. \tag{36}$$

Proof. Following Osband et al. [32, 33], we can rewrite the value function by introducing the Bellman operator. Specifically, the Bellman operator $\mathcal{T}_\pi^f$ is defined as:

$$\mathcal{T}_\pi^f V(s) := r^\pi(s) + \int_{s' \sim \mathcal{S}} f^\pi(s'|s) V(s') \tag{37}$$

where $r^\pi(s) := \int_{a \sim \mathcal{A}} \pi(a|s) r(s, a)$ and $f^\pi(s'|s) := \int_{a \sim \mathcal{A}} \pi(a|s) f(s'|s, a)$.

Below, we rewrite the value function V_π^f as $V_{\pi,1}^f$ to denote the value at the first step. Then the value at the h-th step $V_{\pi,h}^f$ satisfies

$$V_{\pi,h}^f = \mathcal{T}_\pi^f V_{\pi,h+1}^f \tag{38}$$

For any policy π, we have

$$\begin{aligned}
\left(V_{\pi,1}^{f_t} - V_{\pi,1}^{f^*} \right)(s) &= \left(\mathcal{T}_\pi^{f_t} V_{\pi,2}^{f_t} - \mathcal{T}_\pi^{f^*} V_{\pi,2}^{f^*} \right)(s) \\
&= \left(\mathcal{T}_\pi^{f_t} - \mathcal{T}_\pi^{f^*} \right) V_{\pi,2}^{f_t}(s) + \int_{s' \sim \mathcal{S}} f^{*,\pi}(s'|s) \left(V_{\pi,2}^{f^*} - V_{\pi,2}^{f_t} \right)(s') \\
&= \sum_{h=1}^{H} \left(\mathcal{T}_\pi^{f_t} - \mathcal{T}_\pi^{f^*} \right) V_{\pi,h+1}^{f_t}(s_h) \\
&\quad + \sum_{h=1}^{H} \int_{s' \sim \mathcal{S}} f_\pi^*(s'|s_h) \left(V_{\pi,h+1}^{f^*} - V_{\pi,h+1}^{f_t} \right)(s') - \left(V_{\pi,h+1}^{f^*} - V_{\pi,h+1}^{f_t} \right)(s_h).
\end{aligned} \tag{39}$$

This expresses the regret in terms of two factors. The first factor is the one-step Bellman error under the model f_t. And the second factor captures the randomness in the transitions of f^*. We can observe that the *expected* value of $\int_{s'\sim\mathcal{S}} f^{*,\pi}(s'|s)\left(V^{f^*}_{\pi,h+1} - V^{f_t}_{\pi,h+1}\right)(s')$ is exactly $\left(V^{f^*}_{\pi,h+1} - V^{f_t}_{\pi,h+1}\right)(s_h)$. Thus,

$$
\begin{aligned}
\mathop{\mathbb{E}}_{f_t\sim\mathcal{F}_t}\left[\left|V^{f_t}_{\pi,1} - V^{f^*}_{\pi,1}\right|\right] =& \mathop{\mathbb{E}}_{f_t}\left[\sum_{h=1}^{H}\left|\mathcal{T}^{f_t}_{\pi} - \mathcal{T}^{f^*}_{\pi}\right| V^{f_t}_{\pi,h+1}\right] \\
=& \mathop{\mathbb{E}}_{f_t}\left[\sum_{h=1}^{H} r^{\pi}(s_h) - r^{\pi}(s_h) + \int_{s'} \left|f^{\pi}_t(s'|s_h) - f^{*,\pi}(s'|s_h)\right| V^{f_t}_{\pi,h+1}\right] \\
\le& \mathop{\mathbb{E}}_{f_t}\left[\sum_{h=1}^{H} L^t \|f^{\pi}_t(\cdot|s_h) - f^{*,\pi}(\cdot|s_h)\|_2\right] \\
\le& L^t\omega^{\pi}_t.
\end{aligned}
\tag{40}
$$

The first inequality holds due to the Lipschitz assumption of the value function and the last inequality holds by the definition of ω^{π}_t. □

Lemma 2. *(Sun et al., 2018, Theorem 3.1). If the real model is in the model hypothesis, i.e.,* $f^* \in \bigcap_{t=1}^{T}\mathcal{F}_t$, *then the value improvement from policy* π_{t-1} *to* q_t *under the real model is bounded by:*

$$
V^{f^*}_{q_t} - V^{f^*}_{\pi_{t-1}} \ge \Delta_t(\eta_1) + O(\epsilon + \eta_1), \tag{41}
$$

where $\epsilon = \omega^{\pi_{t-1}}_{t-1}$ *is a small number when the model is locally accurate,* $\Delta_t(\eta_1) = V^{f^*}_{q^*_t} - V^{f^*}_{\pi_{t-1}}$ *is a non-negative policy improvement within the* η_1 *trust region, and* q^*_t *is the optimal policy under the real model that satisfies the same* η_1 *constraint as* q_t*, i.e.,* $q^*_t = \operatorname{argmax}_q V^{f^*}_q$, *s.t.* $\mathbb{E}_{s\sim\tilde{\rho}^{\pi_{t-1}}}\left[D_{TV}\left(q^*_t(\cdot|s)||\pi_{t-1}(\cdot|s)\right)\right] \le \eta_1$.

Proof. The proof is pretty similar to the proof in [44]. Since Sun et al. consider the cost minimizing objective, the inequality is reversed since our objective is to maximize cumulative reward. And the maximum modeling error in [44] is replaced by the expected uncertainty ϵ in Eq. (41). □

Lemma 3. *(Osband and Van Roy, 2014, Proposition 6). If* $\{\beta_t|t \in \mathbb{N}\}$ *is nondecreasing with* $\mathcal{F}_t = \mathcal{F}_t(\beta_t)$ *and* $\|f\|_2 \le C$ *for all* $f \in \mathcal{F}$*, then:*

$$
\sum_{t=1}^{T}\sum_{h=1}^{H}\omega_t(s,a) \le 1 + HC\dim_E(\mathcal{F}, T^{-1}) + 4\sqrt{\dim_E(\mathcal{F}, T^{-1})\beta_T T}, \tag{42}
$$

where $\omega_t(s,a) = \sup_{\underline{f},\overline{f}\sim\mathcal{F}_t}\|\overline{f}(s,a) - \underline{f}(s,a)\|_2$.

Lemma 4. *(Osband and Van Roy, 2014, Proposition 5). If the control parameter* $\beta_t(\delta,\alpha)$ *is set to*

$$
\beta_t(\delta,\alpha) = 8\sigma^2\log(N(\mathcal{F},\alpha,\|\cdot\|_2)/\delta) + 2\alpha t\left(8C + \sqrt{8\sigma^2\log(4t^2/\delta)}\right), \tag{43}
$$

then for all $\delta > 0$, $\alpha > 0$ *and* $t \in \mathbb{N}$*, the confidence set* $\mathcal{F}_t = \mathcal{F}_t(\beta_t(\delta,\alpha))$ *satisfies:*

$$
P\left(f^* \in \bigcap_t \mathcal{F}_t\right) \ge 1 - 2\delta. \tag{44}
$$

Lemma 5. *If the reward is bounded in absolute value* R*, then the policy value discrepancy is bounded by:*

$$
\mathop{\mathbb{E}}_{f_t}\left[\left|V^{f_t}_{\pi_1} - V^{f_t}_{\pi_2}\right|\right] \le \frac{2R}{1-\gamma}\mathop{\mathbb{E}}_{s\sim\rho^{\pi_2}_t}\left[D_{TV}\left(\pi_1(\cdot|s)||\pi_2(\cdot|s)\right)\right], \tag{45}
$$

where D_{TV} *is total variation.*

Proof. The following proof is based on the state visitation measure and the total variation. We note that similar techniques are widely used in many previous works [39, 2, 51] and some steps below are borrowed from them.

With slight notion abuse, let ρ^{π_1} denote the state visitation measure under any function f following π_1. By rewriting the value function we have:

$$\begin{aligned} &\left|V^f_{\pi_1} - V^f_{\pi_2}\right| \\ =&\left|\sum_{s,a}\left[\pi_1(a|s)\rho^{\pi_1}(s) - \pi_2(a|s)\rho^{\pi_2}(s)\right] r(s,a)\right| \\ \leq&\sum_{s,a}\left|\pi_1(a|s)\rho^{\pi_1}(s) - \pi_2(a|s)\rho^{\pi_2}(s)\right| r(s,a) \\ \leq& R\sum_{s,a}\left|\pi_1(a|s)\rho^{\pi_1}(s) - \pi_2(a|s)\rho^{\pi_2}(s)\right|. \end{aligned} \tag{46}$$

We rewrite the state transition measure as:

$$\begin{aligned} &\sum_{s,a}\left|\pi_1(a|s)\rho^{\pi_1}(s) - \pi_2(a|s)\rho^{\pi_2}(s)\right| \\ &= \sum_{s,a}\left|\left[\pi_1(a|s) - \pi_2(a|s)\right]\rho^{\pi_1}(s) + \left[\rho^{\pi_1}(s) - \rho^{\pi_2}(s)\right]\pi_2(a|s)\right| \\ &\leq \sum_{s,a}\left|\pi_1(a|s) - \pi_2(a|s)\right|\rho^{\pi_1}(s) + \left|\rho^{\pi_1}(s) - \rho^{\pi_2}(s)\right|\pi_2(a|s) \\ &= 2\mathop{\mathbb{E}}_{s\sim\rho^{\pi_1}}\left[D_{\text{TV}}\left(\pi_1(\cdot|s)||\pi_2(\cdot|s)\right)\right] + \sum_s\left|\rho^{\pi_1}(s) - \rho^{\pi_2}(s)\right|\sum_a \pi_2(a|s), \\ &= 2\mathop{\mathbb{E}}_{s\sim\rho^{\pi_1}}\left[D_{\text{TV}}\left(\pi_1(\cdot|s)||\pi_2(\cdot|s)\right)\right] + 2D_{\text{TV}}\left(\rho^{\pi_1}||\rho^{\pi_2}\right) \end{aligned} \tag{47}$$

where $D_{\text{TV}}(p,q)$ stands for total variation between two distributions p and q, defined as $D_{\text{TV}}(p,q) = \frac{1}{2}\|p-q\|_1$.

We then bound the second term $D_{\text{TV}}\left(\rho^{\pi_1}||\rho^{\pi_2}\right)$. According to the definition of ρ^{π}, we rewrite it as:

$$\rho^{\pi} = (I - \gamma P_{\pi})^{-1}\delta_0, \tag{48}$$

where $P_{\pi}(s'|s) = \sum_a f(s'|s,a)\pi(a|s)$ and f (or P in previous works) is the state transition probability and δ_0 is is a Dirac delta function of the initial state.

Then we have:

$$\begin{aligned} \rho^{\pi_1} - \rho^{\pi_2} &= \left[(I-\gamma P_{\pi_1})^{-1} - (I-\gamma P_{\pi_2})^{-1}\right]\delta_0 \\ &= (I-\gamma P_{\pi_1})^{-1}\left[(I-\gamma P_{\pi_2}) - (I-\gamma P_{\pi_1})\right](I-\gamma P_{\pi_2})^{-1}\delta_0 \\ &= \gamma(P_{\pi_1} - P_{\pi_2})(I-\gamma P_{\pi_1})^{-1}(I-\gamma P_{\pi_2})^{-1}\delta_0 \\ &= \gamma(P_{\pi_1} - P_{\pi_2})(I-\gamma P_{\pi_2})^{-1}\rho^{\pi_1} \end{aligned} \tag{49}$$

Thus, the total variation can be written as:

$$
\begin{aligned}
D_{\mathrm{TV}}\left(\rho^{\pi_1} || \rho^{\pi_2}\right) &= \frac{\gamma}{2} \|(P_{\pi_1} - P_{\pi_2})(I - \gamma P_{\pi_2})^{-1} \rho^{\pi_1}\|_1 \\
&\leq \frac{\gamma}{2} \|(I - \gamma P_{\pi_2})^{-1}\|_1 \|(P_{\pi_1} - P_{\pi_2}) \rho^{\pi_1}\|_1 \\
&\leq \frac{\gamma}{2(1-\gamma)} \|(P_{\pi_1} - P_{\pi_2}) \rho^{\pi_1}\|_1 \\
&= \frac{\gamma}{2(1-\gamma)} \sum_{s'} \left| \sum_{s} \left(P_{\pi_1}(s'|s) - P_{\pi_2}(s'|s)\right) \rho^{\pi_1} \right| \\
&\leq \frac{\gamma}{2(1-\gamma)} \sum_{s,s'} \left| P_{\pi_1}(s'|s) - P_{\pi_2}(s'|s) \right| \rho^{\pi_1} \qquad (50) \\
&= \frac{\gamma}{2(1-\gamma)} \sum_{s,s'} \left| \sum_{a} f(s'|s,a) \left(\pi_1(a|s) - \pi_2(a|s)\right) \right| \rho^{\pi_1} \\
&\leq \frac{\gamma}{2(1-\gamma)} \sum_{s} \rho^{\pi_1} \sum_{a} |\pi_1(a|s) - \pi_2(a|s)| \\
&= \frac{\gamma}{1-\gamma} \mathop{\mathbb{E}}_{s \sim \rho^{\pi_1}} \left[D_{\mathrm{TV}}\left(\pi_1(\cdot|s) || \pi_2(\cdot|s)\right)\right]
\end{aligned}
$$

Plugging Eq. (50) into Eq. (47), we obtain:

$$
\begin{aligned}
&\sum_{s,a} |\pi_1(a|s) \rho^{\pi_1}(s) - \pi_2(a|s) \rho^{\pi_2}(s)| \\
&\leq 2 \mathop{\mathbb{E}}_{s \sim \rho^{\pi_1}} \left[D_{\mathrm{TV}}\left(\pi_1(\cdot|s) || \pi_2(\cdot|s)\right)\right] + 2 D_{\mathrm{TV}}\left(\rho^{\pi_1} || \rho^{\pi_2}\right) \\
&\leq \left(2 + \frac{2\gamma}{1-\gamma}\right) \mathop{\mathbb{E}}_{s \sim \rho^{\pi_1}} \left[D_{\mathrm{TV}}\left(\pi_1(\cdot|s) || \pi_2(\cdot|s)\right)\right] \qquad (51) \\
&= \frac{2}{1-\gamma} \mathop{\mathbb{E}}_{s \sim \rho^{\pi_1}} \left[D_{\mathrm{TV}}\left(\pi_1(\cdot|s) || \pi_2(\cdot|s)\right)\right]
\end{aligned}
$$

Finally, by plugging Eq. (51) into Eq. (46), we have:

$$
\begin{aligned}
\left|V_{\pi_1}^{f} - V_{\pi_2}^{f}\right| &\leq R \sum_{s,a} |\pi_1(a|s) \rho^{\pi_1}(s) - \pi_2(a|s) \rho^{\pi_2}(s)| \\
&\leq \frac{2R}{1-\gamma} \mathop{\mathbb{E}}_{s \sim \rho^{\pi_2}} \left[D_{\mathrm{TV}}\left(\pi_1(\cdot|s) || \pi_2(\cdot|s)\right)\right]
\end{aligned} \qquad (52)
$$

It follows immediately that the expectation of value discrepancy is also bounded by

$$
\mathop{\mathbb{E}}_{f_t} \left[\left|V_{\pi_1}^{f_t} - V_{\pi_2}^{f_t}\right|\right] \leq \frac{2R}{1-\gamma} \mathop{\mathbb{E}}_{s \sim \rho_t^{\pi_2}} \left[D_{\mathrm{TV}}\left(\pi_1(\cdot|s) || \pi_2(\cdot|s)\right)\right]. \qquad (53)
$$

□

C Additional Complexity in Regret Bound

In Theorem 2, the *eluder dimension* d_E appears in the Bayes expected regret bound to capture how effectively the observed samples can extrapolate to unobserved transitions.

For some specific function classes, Osband et al. [33] provide the corresponding eluder dimension bound, *e.g.*, for (generalized) linear function classes. However, for non-linear models, Dong et al. [10] show that the ε-eluder dimension of one-layer neural networks is *at least* exponential in model dimension. We refer to Section 4.1 in [33] and Section 5 in [10] for more details.

Theorem 3. *(Dong et al., 2021, Theorem 5.2). The ε-eluder dimension of one-layer neural networks is at least $\Omega(\varepsilon^{(d-1)})$, where the state-action space is $\mathcal{X} = \{x \in \mathbb{R}^d\}$.*

Besides, to clarify the asymptotics, the covering number $N(\mathcal{F}, \alpha, \|\cdot\|_2)$ in Theorem 2 can be replaced by the *Kolmogorov dimension* d_K for $\mathcal{F}$ following [33].

Definition 3. *(Osband and Van Roy, 2014, Definition 1). The Kolmogorov dimension d_K of the model function class $\mathcal{F}$ is defined as:*

$$d_K(\mathcal{F}) := \limsup_{\alpha \downarrow 0} \frac{\log(N(\mathcal{F}, \alpha, \|\cdot\|_2))}{\log(1/\alpha)}. \tag{54}$$

D Algorithm Instantiations

The model-based policy optimization algorithm `MBPO` in Algorithm 1 can be instantiated as one of the following algorithms, Dyna style policy optimization in Algorithm 2, model-based back-propagation in Algorithm 3 and model predictive control policy optimization in Algorithm 4. As default, the `MBPO` is instantiated to be the Dyna-style policy optimization in our experiments. We note that the instantiations is not restricted to the listed algorithms, and many other `MBPO` algorithms that augment policy learning with a predictive model can also be leveraged, *e.g.*, model-based value expansion [11, 4].

Dyna Policy Optimization involves simulating data using the learned model and optimizing the policy with any model-free RL method, *e.g.*, REINFORCE or actor-critic [21]. And the state-action value can be estimated by learning a value function or rolling out the model. In Algorithm 1, the input objective function is constrained optimization of the policy, *i.e.*, Eq. (8) and Eq. (9). Thus, similar to the model-free trust-region algorithms [39, 40, 2], Lagrangian equations are introduced. And several gradient steps give the updated policy, *i.e.*, q_t and π_t in the dual update procedures. In the pseudocode below, N is the horizon.

Algorithm 2 Dyna Model-Based Policy Optimization

```
Input: Policy $\pi$, model set $\{f\}$, objective function.
Initialize a simulation data buffer $\hat{\mathcal{D}}$
Sample a batch of initial state from the initial distribution $d(s_0)$
▷ Data simulation
for each initial state sample $s_0$ do
  for model $f$ in model set $\{f\}$ do
    for timestep $h = 1, ..., N$ do
      Sample action $\hat{a}_h \sim \pi(\cdot|\hat{s}_h)$
      Sample simulation state $\hat{s}_{h+1} \sim f(\hat{s}_h, \hat{a}_h)$
      Append simulation data to buffer $\hat{\mathcal{D}} = \hat{\mathcal{D}} \cup (\hat{s}_h, \hat{a}_h, \hat{r}_h, \hat{s}_{h+1})$
    end for
  end for
end for
▷ Policy optimization with any model-free algorithm ModelFree
Objective optimization of policy on the simulated data $\pi \leftarrow$ ModelFree($\hat{\mathcal{D}}, \pi$)
```

Back-Propagation Through Time is an alternative way for model-based policy optimization. Specifically, the policy parameters are updated by directly computing the derivatives of the performance w.r.t. the parameters [9, 8, 7]. When the optimization of objective function is constrained, the accumulating J step in Algorithm 3 Line 9 can be $J \leftarrow J + \gamma^n \hat{r}(\hat{s}_h, \hat{a}_h) - \lambda D_{\mathrm{KL}}$, where λ is the Lagrangian multiplier and D_{KL} is the corresponding KL constraint.

Model Predictive Control Policy Optimization is a model-based *planning* algorithm. Different from the above model-augmented policy optimization methods, MPC policy optimization directly generates optimal action sequences under the model and then distills the policy. Specifically, the pseudocode in Algorithm 4 begins with initial actions generated by the policy. Then with a shooting method, *e.g.*, cross-entropy method (CEM), the actions are refined and the policy that generates these optimal actions are distilled. Below, the algorithm to obtain the refined actions `EliteActions` can be CEM with action noise added to the action or policy parameter, *i.e.*, POPLIN-A and POPLIN-P in [50]. And the policy can be updated by `UpdatePolicy` using behavior cloning.

Algorithm 3 Model-Based Back-Propagation Policy Optimization

Input: Policy π, model set $\{f\}$, objective function.

```
Initialize a simulation data buffer $\hat{\mathcal{D}}$
Start from initial state $s_0$
Reset $J \leftarrow 0$
▷ Data simulation
for model $f$ in model set $\{f\}$ do
    for timestep $h = 1, ..., N$ do
        Sample action $\hat{a}_h \sim \pi(\cdot|\hat{s}_h)$
        Sample simulation state $\hat{s}_{h+1} \sim f(\hat{s}_h, \hat{a}_h)$
        Accumulate reward and constraint to $J$
    end for
end for
▷ Policy optimization
Compute policy gradient with back-propagation through time
Objective optimization of policy $\pi \leftarrow$ PolicyGradient
```

Algorithm 4 Model Predictive Control Policy Optimization

Input: Policy π, model set $\{f\}$, objective function, algorithm to update actions `EliteActions`, algorithm to update policy `UpdatePolicy`.

```
Start from initial state $s_0$
Reset $J \leftarrow 0$
▷ Model-based planning
for model $f$ in model set $\{f\}$ do
    for timestep $h = 1, ..., N$ do
        Sample action $\hat{a}_h \sim \pi(\cdot|\hat{s}_h)$
        Sample simulation state $\hat{s}_{h+1} \sim f(\hat{s}_h, \hat{a}_h)$
        Accumulate reward and constraint to $J$
    end for
end for
$\mathbf{a} \leftarrow$ EliteActions$(J, \hat{a}_{1:N})$
▷ Policy distillation
$\pi \leftarrow$ UpdatePolicy$(\mathbf{a})$
```

E Experimental Settings

In experiments, we use a 5-layer neural network with supervised learning to fit the dynamical model. And we use deterministic ensembles [6] to capture the model epistemic uncertainty. Specifically, different ensembles are learned with independent transition data to construct the 1-step ahead confidence interval at every timestep. Each ensemble is separately trained using Adam [19]. And the number of ensemble heads can be set to 3, 4 or, 5, each of which is shown to be able to provide considerable performance in our experiments.

Since neural networks are not calibrated in general, *i.e.*, the model uncertainty set is not guaranteed to contain the real dynamics, we follow [8] to re-calibrate [22] the model. When using the Dyna model-based policy optimization, the number of gradient steps for each optimization procedure in an iteration is set to 20. And we empirically find that the KL divergence (or total variance) constraint makes the algorithm more efficient when computing the $\arg\max$ in the optimization step, since optimizing from π_{t-1} at iteration t needs fewer policy gradient steps if the policy update is constrained within a certain trust region.

For the three experimental tasks, the task settings follow standard Mujoco settings. The task-specific and task-common settings and parameters are listed below in Table 1.

Table 1: Experimental parameters.

	Inverted Pendulum	Pusher	Half-Cheetah
episode length H	200	150	1000
dimension of state	4	23	18
dimension of action	1	7	6
action penalty	0.001	0.1	0.1
ensemble number	3-5		
hidden nodes	(200, 200, 200, 200, 200)		
activation function	Swish		
optimizer	Adam		
learning rate	10^{-3}		

F Algorithmic Comparisons between MBRL Algorithms

We provide the algorithmic comparisons of four MBRL frameworks, including the naive greedy model exploitation algorithms, OFU, PSRL, and the proposed DCPU algorithm.

And the differences mainly lie in the model selection and policy update procedures. The high-level psuedocode is given in Algorithm 5, 6, 7 and 8. Among them, the greedy model exploitation algorithm is a pretty naive instantiation. And other instantiations of greedy model exploitation also include the ones that augment Algorithm 5 by *e.g.*, receding horizon model planning or adopting more complex models such as probabilistic dynamics models [6].

Algorithm 5 Naive Greedy Model Exploitation

```
for iteration $t = 1, ..., T$ do
    Compute mean model $\tilde{f}_t = \mathbb{E}[f_t]$
    Compute $\pi_t = \text{argmax}_\pi V_\pi^{\tilde{f}_t}$
    for timestep $h = 1, ..., H$ do
        Execute $\pi_t$ in real environment
        Update $\mathcal{D}_t = \mathcal{D}_t \cup (s_h, a_h, r_h, s_{h+1})$
    end for
end for
return policy $\pi_T$
```

Algorithm 6 OFU Algorithm

```
for iteration $t = 1, ..., T$ do
    Construct confidence set $\mathcal{F}_t$
    Compute $\pi_t = \text{argmax}_{\pi, f \sim \mathcal{F}_t} V_\pi^{f_t}$
    for timestep $h = 1, ..., H$ do
        Execute $\pi_t$ in real environment
        Update $\mathcal{D}_t = \mathcal{D}_t \cup (s_h, a_h, r_h, s_{h+1})$
    end for
end for
return policy $\pi_T$
```

Algorithm 7 PSRL Algorithm

```
for iteration $t = 1, ..., T$ do
    Sample $f_t \sim \phi(\cdot \mid \mathcal{D}_t)$
    Compute $\pi_t = \text{argmax}_\pi V_\pi^{f_t}$
    for timestep $h = 1, ..., H$ do
        Execute $\pi_t$ in real environment
        Update $\mathcal{D}_t = \mathcal{D}_t \cup (s_h, a_h, r_h, s_{h+1})$
    end for
end for
return policy $\pi_T$
```

Algorithm 8 DCPU Algorithm

```
for iteration $t = 1, ..., T$ do
    Compute $q_t$ following Eq. (8)
    Compute $\pi_t$ following Eq. (9)
    for timestep $h = 1, ..., H$ do
        Execute $\pi_t$ in real environment
        Update $\mathcal{D}_t = \mathcal{D}_t \cup (s_h, a_h, r_h, s_{h+1})$
    end for
end for
return policy $\pi_T$
```