

# Model-Based Offline Reinforcement Learning with Local Misspecification

Kefan Dong\*, Yannis Flet-Berliac\*, Allen Nie\*, Emma Brunskill

Stanford University  
{kefandong,yfletberliac,anie,ebrun}@stanford.edu

## Abstract

We present a model-based offline reinforcement learning policy performance lower bound that explicitly captures dynamics model misspecification and distribution mismatch and we propose an empirical algorithm for optimal offline policy selection. Theoretically, we prove a novel safe policy improvement theorem by establishing pessimism approximations to the value function. Our key insight is to jointly consider selecting over dynamics models and policies: as long as a dynamics model can accurately represent the dynamics of the state-action pairs *visited* by a given policy, it is possible to approximate the value of that particular policy. We analyze our lower bound in the LQR setting and also show competitive performance to previous lower bounds on policy selection across a set of D4RL tasks.

## Introduction

Offline reinforcement learning (RL) could leverage historical decisions made and their outcomes to improve data-driven decision-making in areas like marketing (Thomas et al. 2017), robotics (Quillen et al. 2018; Yu et al. 2020, 2021; Swazinna, Udluft, and Runkler 2020; Singh et al. 2020), recommendation systems (Swaminathan and Joachims 2015), etc. Offline RL is particularly useful when it is possible to deploy context-specific decision policies, but it is costly or infeasible to do online reinforcement learning.

Prior work on offline RL for large state and/or action spaces has primarily focused on one of two extreme settings. One line of work makes minimal assumptions on the underlying stochastic process, requiring only no confounding, and leverages importance-sampling estimators of potential policies (e.g., Thomas, Theodorou, and Ghavamzadeh (2015); Thomas et al. (2019)). Unfortunately, such estimators have a variance that scales exponentially with the horizon (Liu et al. 2018b) and are often ill-suited to long horizon problems<sup>1</sup>.

An alternative, which is the majority of work in offline RL, is to make a number of assumptions on the domain,

behavior data generation process and the expressiveness of the function classes employed. The work in this space typically assumes the domain satisfies the Markov assumption, which has been recently shown in the off-policy evaluation setting to enable provably more efficient policy value estimation (Kallus and Uehara 2020). Historically, most work (e.g., Munos (2003); Farahmand, Munos, and Szepesvári (2010); Xie and Jiang (2020); Chen and Jiang (2019)) assumes the batch data set has coverage on any state-action pairs that could be visited under any possible policy. More recent work relaxes this strong requirement using a pessimism under uncertainty approach that is model-based (Yu et al. 2020, 2021; Kidambi et al. 2020), model-free (Liu et al. 2020) or uses policy search (Curi, Berkenkamp, and Krause 2020; van Hasselt, Hessel, and Aslanides 2019). Such work still relies on realizability/lack of misspecification assumptions. For model-free approaches, a common assumption is that the value function class can represent all policies. Liu et al. (2020) assume that the value function class is closed under (modified) Bellman backups. A recent exception is Xie and Jiang (2020), which only requires the optimal  $Q$ -function to be representable by the value function class. However, their sample complexity scales non-optimally (Xie and Jiang 2020, Theorem 2), and they also make strong assumptions on the data coverage – essentially the dataset must visit all states with sufficient probability. Model-based approaches such as Malik et al. (2019); Yu et al. (2020) assume the dynamics class has no misspecification.

These two lines of work hint at possibilities in the middle: can we leverage the sample-efficient benefits of Markov structure and allow for minimal assumptions on the data-gathering process and potential model misspecification? This can be viewed as one step towards more best-in-class results for offline RL. Such results are relatively rare in RL, which tends to focus on obtaining optimal or near-optimal policies for the underlying domain. Yet in many important applications, it may be much more practical to hope to identify a strong policy within a particular policy class.

Our insight is that the algorithm may be able to leverage misspecified models and still leverage the Markov assumption for increased data efficiency. In particular, we take a model-based offline RL approach to leverage dynamics models that can accurately fit the space of state-action pairs visited under a particular policy (*local small misspecifica-*

\*These authors contributed equally.

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

<sup>1</sup>Marginalized importance sampling (MIS) methods (Liu et al. 2018a; Xie, Ma, and Wang 2019; Yin and Wang 2020; Liu, Bacon, and Brunskill 2020) help address this but rely on the system being Markov in the underlying state space

tion), rather than being a good model of the entire possible state-action space (*global small misspecification*). Our work is most closely related to the recently proposed Minimax Model Learning (MML) algorithm (Voloshin, Jiang, and Yue 2021): MML optimizes for the model that minimizes a value-aware error which upper bounds the difference of policy value in learned and real models. If the considered model class includes the true model, this can work very well, but when the models are misspecified, this can become overly conservative since it optimizes with respect to a worst-case potential state-action distribution shift.

The key feature of our algorithm is to jointly optimize policy and dynamics. Prior model-based offline RL algorithms typically estimate dynamics first, and then optimize a policy w.r.t. the learned dynamics (Yu et al. 2020, 2021; Voloshin, Jiang, and Yue 2021). But when the dynamics model class is misspecified, there may not exist a unique “good dynamics” that can approximate the value of every policy. As a result, the learned policy may have a good estimated value under the learned dynamics, but a poor performance in the real environment, or the learned policy may be overly conservative due to the misestimated dynamics.

Our paper makes the following contributions. First, we provide a finite sample bound that assumes a Markov model, leverages the pessimism principle to work with many data-gathering distributions, accounts for estimation error in the behavior policy and, most importantly, directly accounts for dynamics and value function model misspecification (see Lemma 3). We prove the misspecification error of our method is much tighter than other approaches because we only look at the models’ ability to represent visited state-action pairs for a particular policy. In that sense, we say our algorithm relies on small *local* model dynamics misspecification. In Theorem 6, we show that when the dynamics model class does not satisfy realizability, decoupling the learning of policy and dynamics is suboptimal. This motivates our algorithm which jointly optimizes the policy and model dynamics across a finite set. Because of the tighter pessimistic estimation, we can prove a novel safe policy improvement theorem (see Theorem 4) for offline policy optimization (OPO). While our primary contribution is theoretical, our proposed method for policy selection improves over the state-of-the-art MML Voloshin, Jiang, and Yue (2021) in a simple linear Gaussian setting, and has solid performance on policy selection on a set of D4RL benchmarks.

## Related Works

There is an extensive and growing body of research on offline RL and we focus here on methods that also assume a Markov domain. Many papers focus on model-free methods (e.g., Fujimoto et al. (2018); Kumar et al. (2019, 2020)). Nachum et al. (2019) and their follow-ups (Zhang et al. 2019; Zhang, Liu, and Whiteson 2020) learn a distribution correction term, on top of which they perform evaluation or policy optimization tasks. Uehara, Huang, and Jiang (2020); Jiang and Huang (2020) study the duality between learning  $Q$ -functions and learning importance weights. Liu et al. (2020) explicitly consider the distribution shift in offline RL and propose conservative Bellman equations.

Algorithm	Statistical Error	Misspecification	SPI
VAML	$\tilde{O}\left(\frac{p}{\sqrt{n}}\right)^2$	✓(global)	✗
MBS-PI	$\tilde{O}\left(\frac{V_{\max}\zeta}{(1-\gamma)^2\sqrt{n}}\right)$	✓(global)	✓
MML	$\mathfrak{R}_n^3$	✓(global)	✗
Ours	$\tilde{O}\left(\frac{V_{\max}}{1-\gamma}\sqrt{\frac{\zeta}{n}}\right)$	✓(local)	✓

Table 1: Comparison of error bounds with prior works.

Another line of research uses model-based methods (Kidambi et al. 2020; Yu et al. 2020, 2021; Matsushima et al. 2020; Swazinna, Udluft, and Runkler 2020; Fu and Levine 2021; Farahmand, Barreto, and Nikovski 2017). Gelada et al. (2019); Delgrange, Nowe, and Pérez (2022); Voloshin, Jiang, and Yue (2021) learn the dynamics using different loss functions. Yu et al. (2020) build an uncertainty quantification on top of the learned dynamics and select a policy that optimizes the lower confidence bound. (Argenson and Dulac-Arnold 2020; Zhan, Zhu, and Xu 2021) focus on policy optimization instead of model learning.

In Table 1, we compare our error bounds with existing results. Our statistical error (introduced by finite dataset) is comparable with VAML (Farahmand, Barreto, and Nikovski 2017), MBS-PI (Liu et al. 2020) and MML (Voloshin, Jiang, and Yue 2021). In addition, we consider misspecification errors and safe policy improvement (SPI).

## Problem Setup

A Markov Decision Process (MDP) is defined by a tuple  $\langle T, r, \mathcal{S}, \mathcal{A}, \gamma \rangle$ .  $\mathcal{S}$  and  $\mathcal{A}$  denote the state and action spaces.  $T : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$  is the transition and  $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}_+$  is the reward.  $\gamma \in [0, 1)$  is the discount factor. For a policy  $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ , the value function is defined as

$$V_T^\pi(s) = \mathbb{E}_{s_0=s, a_t \sim \pi(s_t), s_{t+1} \sim T(s_t, a_t)} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right].$$

Let  $R_{\max} \triangleq \max_{s,a} r(s, a)$  be the maximal reward and  $V_{\max} \triangleq R_{\max}/(1-\gamma)$ . Without loss of generality, we assume that the initial state is fixed as  $s_0$ . We use  $\eta(T, \pi) \triangleq V_T^\pi(s_0)$  to denote the expected value of policy  $\pi$ . Let  $\rho_T^\pi(s, a) \triangleq (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \Pr_T^\pi(s_t = s, a_t = a \mid s_0)$  be the normalized state-action distribution when we execute policy  $\pi$  in a domain with dynamics model  $T$ . For simplicity in this paper we assume the reward function is known.

An offline RL algorithm takes a dataset  $\mathcal{D} = \{(s_i, a_i, s'_i)\}_{i=1}^n$  as input, where  $n$  is the size of the dataset. Each  $(s_i, a_i, s'_i)$  tuple is drawn independently from a behavior distribution  $\mu$ . We assume that  $\mu$  is consistent with the MDP in the sense that  $\mu(\cdot \mid s, a) = T(s, a)$  for all  $(s, a)$ . For simplicity, we use  $\hat{\mathbb{E}}$  to denote the empirical distribution over the dataset  $\mathcal{D}$ . In this paper, we assume that the

<sup>2</sup>VAML only considers linear function approximation and  $p$  is the dimension of the feature vector.

<sup>3</sup>The Rademacher complexity. For the finite hypothesis, the best-known upper bound is in the same order of ours.

algorithm has access to an estimated behavior distribution  $\hat{\mu}$  such that  $\text{TV}(\mu, \hat{\mu})$  is small. This estimation can be easily obtained using a separate dataset (e.g., Liu et al. (2020)).

The algorithm can access three (finite) function classes  $\mathcal{G}, \mathcal{T}, \Pi$ .  $\mathcal{G}$  is a class of value functions,  $\mathcal{T}$  a class of dynamics and  $\Pi$  a class of policies. We assume that  $g(s, a) \in [0, V_{\max}]$  for all  $g \in \mathcal{G}$ . We use  $T^*$  to denote the ground-truth dynamics. Note that  $T^*$  may not be in  $\mathcal{T}$ . Our goal is to return a policy  $\pi \in \Pi$  that maximizes  $\eta(T^*, \pi)$ .

## Main Results

A standard model-based RL algorithm learns the dynamics models first, and then uses the learned dynamics to estimate the value of a policy, or optimize it. In this approach, it is crucial to link the estimation error of the dynamics to the estimation error of the value. Therefore, as a starting point, we invoke the simulation lemma.

**Lemma 1** (Simulation Lemma (Yu et al. 2020; Kakade and Langford 2002)). *Consider two MDPs with dynamics  $T, T^*$ , and the same reward function. Then,*

$$\eta(T, \pi) - \eta(T^*, \pi) = \frac{\gamma}{1 - \gamma} \mathbb{E}_{(s,a) \sim \rho_T^\pi} \left[ \mathbb{E}_{s' \sim T(s,a)} [V_{T^*}^\pi(s')] - \mathbb{E}_{s' \sim T^*(s,a)} [V_{T^*}^\pi(s')] \right]. \quad (1)$$

For a fixed ground-truth dynamics  $T^*$ , we define  $G_T^\pi(s, a) = \mathbb{E}_{s' \sim T(s,a)} [V_{T^*}^\pi(s')] - \mathbb{E}_{s' \sim T^*(s,a)} [V_{T^*}^\pi(s')]$ . The simulation lemma states that the dynamics will provide an accurate estimate of the policy value if  $\mathbb{E}_{s' \sim T(s,a)} [V_{T^*}^\pi(s')]$  matches  $\mathbb{E}_{s' \sim T^*(s,a)} [V_{T^*}^\pi(s')]$ . In other words, to obtain a good estimate of a policy value, it is sufficient to minimize the model error  $G_T^\pi(s, a)$ .

Since the value function  $V_{T^*}^\pi$  is unknown, Yu et al. (2020) upper bound the model error by introducing a class of test functions  $\mathcal{G} : \mathcal{S} \rightarrow \mathbb{R}$ . When  $V_{T^*}^\pi \in \mathcal{G}$ , we have

$$|G_T^\pi(s, a)| \leq \sup_{g \in \mathcal{G}} \left| \mathbb{E}_{s' \sim T(s,a)} g(s') - \mathbb{E}_{s' \sim T^*(s,a)} g(s') \right|.$$

In an offline dataset  $\mathcal{D}$ , typically we can only observe one sample from  $T^*(s, a)$  per state-action pair. Hence the algorithm cannot compute this upper bound exactly. In addition, the distribution of the dataset  $\mathcal{D}$  is also different from the one required by the simulation lemma  $\rho_T^\pi$ . To address these issues, we explicitly introduce a density ratio  $w : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}_+$ . For a test function  $g \in \mathcal{G}$  and a dynamics model  $T$ , let  $f_T^g(s, a) \triangleq \mathbb{E}_{s' \sim T(s,a)} [g(s')]$ . Recall that  $\hat{\mathbb{E}}$  denotes the empirical expectation over dataset  $\mathcal{D}$ . Then our model loss is defined as

$$\ell_w(T, g) = \left| \hat{\mathbb{E}}[w(s, a)(f_T^g(s, a) - g(s'))] \right|. \quad (2)$$

**Distribution mismatch.** We aim to upper bound policy evaluation error by the loss function even if there are state action pairs with small probability mass under behavior distribution  $\mu$  (i.e., the offline dataset does not have a perfect coverage). Following Liu et al. (2020), we treat the unknown state-action pairs pessimistically. Let  $\zeta$  be a fixed cutoff threshold. Recall that  $\hat{\mu}$  is an estimation of the behavior distribution. For a policy  $\pi$  and dynamics  $T$ , we define

$w_{\pi, T}(s, a) \triangleq \mathbb{I} \left[ \frac{\rho_T^\pi(s, a)}{\hat{\mu}(s, a)} \leq \zeta \right] \frac{\rho_T^\pi(s, a)}{\hat{\mu}(s, a)}$  as the truncated density ratio. For a fixed policy  $\pi$ , when  $w = w_{\pi, T}$ ,

$$\begin{aligned} & \left| \mathbb{E}_{(s,a) \sim \rho_T^\pi} \left[ G_T^\pi(s, a) \right] \right| \\ & \leq \left| \mathbb{E}_{(s,a) \sim \rho_T^\pi} \left[ \mathbb{I} \left[ \frac{\rho_T^\pi(s, a)}{\hat{\mu}(s, a)} \leq \zeta \right] G_T^\pi(s, a) \right] \right| \\ & \quad + \left| \mathbb{E}_{(s,a) \sim \rho_T^\pi} \left[ \mathbb{I} \left[ \frac{\rho_T^\pi(s, a)}{\hat{\mu}(s, a)} > \zeta \right] G_T^\pi(s, a) \right] \right| \\ & \leq \mathbb{E}_{(s,a) \sim \hat{\mu}} \left[ w(s, a) G_T^\pi(s, a) \right] \\ & \quad + V_{\max} \left| \mathbb{E}_{(s,a) \sim \rho_T^\pi} \left[ \mathbb{I} \left[ \frac{\rho_T^\pi(s, a)}{\hat{\mu}(s, a)} > \zeta \right] \right] \right| \\ & \leq \mathbb{E}_{(s,a) \sim \mu} \left[ w(s, a) G_T^\pi(s, a) \right] + \zeta V_{\max} \text{TV}(\hat{\mu}, \mu) \\ & \quad + V_{\max} \left| \mathbb{E}_{(s,a) \sim \rho_T^\pi} \left[ \mathbb{I} \left[ \frac{\rho_T^\pi(s, a)}{\hat{\mu}(s, a)} > \zeta \right] \right] \right|. \end{aligned}$$

Hence, ignoring statistical error due to finite dataset, we can upper bound the estimation error  $|\eta(T^*, \pi) - \eta(T, \pi)|$  by

$$\begin{aligned} & \frac{\gamma}{1 - \gamma} \left( \sup_{g \in \mathcal{G}} \left| \ell_{w_{\pi, T}}(g, T) \right| + \zeta V_{\max} \text{TV}(\hat{\mu}, \mu) \right. \\ & \quad \left. + V_{\max} \mathbb{E}_{(s,a) \sim \rho_T^\pi} \left[ \mathbb{I} \left[ \frac{\rho_T^\pi(s, a)}{\hat{\mu}(s, a)} > \zeta \right] \right] \right). \quad (3) \end{aligned}$$

Intuitively, the first term measures the error caused by imperfect dynamics  $T$ , the second term captures the estimation error of the behavior distribution, and the last term comes from truncating the density ratios.

## Pessimistic Policy Optimization with Model Misspecification

In this section, we explicitly consider misspecifications of the function classes used for representing the value function and dynamics models ( $\mathcal{G}$  and  $\mathcal{T}$ , respectively). Most prior theoretical work on model-based RL make strong assumptions on the realizability of the dynamics model class. For example, in the offline setting, Voloshin, Jiang, and Yue (2021) focus on exact realizability of the dynamics model (that is,  $T^* \in \mathcal{T}$ ). In the online setting, Jin et al. (2020) provide bounds where there is a linear regret term due to global model misspecification. Their bounds require a  $T \in \mathcal{T}$  such that  $\text{TV}(T(s, a), T^*(s, a)) \leq \epsilon$  for all  $(s, a)$ , even if the state-action pair  $(s, a)$  is only visited under some poorly performing policies. We now show that offline RL tasks can need much weaker realizability assumptions on the dynamics model class.

Our key observation is that for a given dynamics  $T$  and policy  $\pi$ , computing the density ratio  $w_{\pi, T}$  is *statistically efficient*. Note that to compute  $w_{\pi, T}$  we do not need any samples from the true dynamics: instead, we only need to be able to estimate the state-action density under a dynamics model  $T$  for policy  $\pi$ . This allows us to explicitly utilize the density ratio to get a relaxed realizability assumption.

**Definition 2.** *The local value function error for a particular*

---

**Algorithm 1: Model-based Offline RL with Local Misspecification Error**


---

**Require:** estimated behavior distribution  $\hat{\mu}$ , truncation threshold  $\zeta$ .

**for**  $\pi \in \Pi, T \in \mathcal{T}$  **do**

Compute  $w_{\pi, T}(s, a) = \mathbb{I} \left[ \frac{\rho_T^\pi(s, a)}{\hat{\mu}(s, a)} \leq \zeta \right] \frac{\rho_T^\pi(s, a)}{\hat{\mu}(s, a)}$ .

Compute  $\text{lb}(T, \pi)$  by Eq. (4).

**end for**

$\pi \leftarrow \text{argmax}_{\pi \in \Pi} \max_{T \in \mathcal{T}} \text{lb}(T, \pi)$ .

---

dynamics model  $T$  and policy  $\pi$  is defined as

$$\epsilon_V(T, \pi) \triangleq \inf_{g \in \mathcal{G}} |\mathbb{E}_{(s, a) \sim \mu} [w_{\pi, T}(s, a) (\mathbb{E}_{s' \sim T(s, a)} [(g - V_{T^*}^\pi)(s']) + \mathbb{E}_{s' \sim T^*(s, a)} [(g - V_{T^*}^\pi)(s'])]|.$$

The term  $\epsilon_V$  measures the local misspecification of the value function class – that is, the error between the true value of the policy  $V_{T^*}^\pi$  and the best fitting value function in the class  $\mathcal{G}$  – only on the state-action pairs that policy  $\pi$  visits under a particular potential dynamics model  $T$ . In contrast, previous results (Jin et al. 2020; Nachum et al. 2019; Voloshin, Jiang, and Yue 2021) take the global maximum error over all (reachable)  $(s, a)$ , which can be much larger than the local misspecification error  $\epsilon_V(T, \pi)$ .

With this local misspecification error, we can establish a pessimistic estimation of the true reward. Let  $\mathcal{E}$  be a high probability event under which the loss function  $\ell_{w_{\pi, T}}(T, g)$  is close to its expectation (randomness comes from the dataset  $\mathcal{D}$ ). In the Appendix, we define this event formally and prove that  $\Pr(\mathcal{E}) \geq 1 - \delta$ . The following lemma gives a lower bound on the true reward. Proofs, when omitted, are in the Appendix.

**Lemma 3.** Let  $\iota = \log(2|\mathcal{G}||\mathcal{T}||\Pi|/\delta)$ . For any dynamics model  $T$  and policy  $\pi$ , we define

$$\begin{aligned} \text{lb}(T, \pi) &= \eta(T, \pi) - \frac{1}{1 - \gamma} \left( \sup_{g \in \mathcal{G}} \ell_{w_{\pi, T}}(g, T) \right. \\ &\quad \left. + V_{\max} \mathbb{E}_{(s, a) \sim \rho_T^\pi} \left[ \mathbb{I} \left[ \frac{\rho_T^\pi(s, a)}{\hat{\mu}(s, a)} > \zeta \right] \right] \right). \end{aligned} \quad (4)$$

Then, under the event  $\mathcal{E}$ , we have

$$\begin{aligned} \eta(T^*, \pi) &\geq \text{lb}(T, \pi) - \frac{1}{1 - \gamma} \left( \epsilon_V(T, \pi) \right. \\ &\quad \left. - 2V_{\max} \sqrt{\zeta \iota / n} - \zeta V_{\max} \text{TV}(\hat{\mu}, \mu) \right). \end{aligned} \quad (5)$$

We use this to define our offline policy selection Alg. 1.

In contrast to existing offline model-based algorithms (Yu et al. 2020; Voloshin, Jiang, and Yue 2021), our algorithm optimizes the dynamics and policy jointly. For a given dynamics model, policy pair, our Alg. 1 computes the truncated density ratio  $w_{\pi, T}$  which does not require collecting new samples and then uses this to compute a lower bound  $\text{lb}(T, \pi)$  (Eq. (4)). Finally, it outputs a policy that maximizes the lower bound. We will shortly show this joint optimization can lead to better offline learning.

Parameter  $\zeta$  controls the truncation of the stationary importance weights. Increasing  $\zeta$  decreases the last term in the lower bound objective  $\text{lb}(T, \pi)$ , but it may also increase the variance given the finite dataset size. Note that by setting  $\zeta = \log(n)$  and letting  $n \rightarrow \infty$  (i.e., with infinite data), the last term in Eq. (4) and the statistical error converge to zero.

### Safe Policy Improvement

We now derive a novel safe policy improvement result, up to the error terms given below. Intuitively this guarantees that the policy returned by Alg. 1 will improve over the behavior policy when possible, which is an attractive property in many applied settings. Note that recent work (Voloshin, Jiang, and Yue 2021; Yu et al. 2020) on model-based offline RL does not provide this guarantee when the dynamics model class is misspecified. For a fixed policy  $\pi$ , define

$$\epsilon_\rho(\pi) \triangleq \inf_{T \in \mathcal{T}} \mathbb{E}_{(s, a) \sim \rho_{T^*}^\pi} [\text{TV}(T(s, a), T^*(s, a))], \quad (6)$$

$$\epsilon_\mu(\pi) \triangleq \mathbb{E}_{(s, a) \sim \rho_{T^*}^\pi} \left[ \mathbb{I} \left[ \frac{\rho_{T^*}^\pi(s, a)}{\hat{\mu}(s, a)} > \zeta / 2 \right] \right]. \quad (7)$$

The term  $\epsilon_\rho$  measures the local misspecification error of the dynamics model class in being able to represent the dynamics for state-action pairs encountered for policy  $\pi$ .  $\epsilon_\mu$  represents that overlap of the dataset for an alternate policy  $\pi$ : such a quantity is common in much of offline RL. In the following theorem, we prove that the true value of the policy computed by Alg. 1 is lower bounded by that of the optimal policy in the function class with some error terms.

**Theorem 4.** Consider a fixed parameter  $\zeta$ . Let  $\hat{\pi}$  be the policy computed by Alg. 1 and  $\hat{T} = \text{argmax}_T \text{lb}(T, \hat{\pi})$ . Let  $\iota = \log(2|\mathcal{G}||\mathcal{T}||\Pi|/\delta)$ . Then, with probability at least  $1 - \delta$ , we have

$$\begin{aligned} \eta(T^*, \hat{\pi}) &\geq \sup_{\pi} \left\{ \eta(T^*, \pi) - \frac{6V_{\max} \epsilon_\rho(\pi)}{(1 - \gamma)^2} - \frac{V_{\max} \epsilon_\mu(\pi)}{1 - \gamma} \right\} \\ &\quad - \frac{\epsilon_V(\hat{T}, \hat{\pi})}{1 - \gamma} - \frac{4V_{\max}}{1 - \gamma} \sqrt{\frac{\zeta \iota}{n}} - \frac{2\zeta V_{\max} \text{TV}(\hat{\mu}, \mu)}{1 - \gamma}. \end{aligned} \quad (8)$$

To prove Theorem 4, we prove the tightness of  $\text{lb}(T, \pi)$  – the lower bound  $\max_T \text{lb}(T, \pi)$  is at least as high as the true value of the policy with some errors. Consequently, maximizing the lower bound also maximizes the true value of the policy. Formally speaking, we have the following Lemma.

**Lemma 5.** For any policy  $\pi \in \Pi$ , under the event  $\mathcal{E}$  we have

$$\begin{aligned} \max_{T \in \mathcal{T}} \text{lb}(T, \pi) &\geq \eta(T^*, \pi) - 6V_{\max} \epsilon_\rho(\pi) / (1 - \gamma)^2 \\ &\quad - \frac{1}{1 - \gamma} \left( V_{\max} \epsilon_\mu(\pi) - 2V_{\max} \sqrt{\zeta \iota / n} - \zeta V_{\max} \text{TV}(\hat{\mu}, \mu) \right). \end{aligned}$$

In the sequel, we present a proof sketch for Lemma 5. In this proof sketch, we hide  $1/(1 - \gamma)$  factors in the big-O notation. For a fixed policy  $\pi$ , let  $\hat{T}$  be the minimizer of Eq. (6). We prove Lemma 5 by analyzing the terms in the definition of  $\text{lb}(\hat{T}, \pi)$  (Eq. (4)) separately.

- i. Following the definition of Eq. (6), we can show that  $\|\rho_{\hat{T}}^\pi - \rho_{T^*}^\pi\|_1 \leq \mathcal{O}(\epsilon_\rho(\pi))$ . Consequently we get  $\eta(\hat{T}, \pi) \geq \eta(T^*, \pi) - \mathcal{O}(\epsilon_\rho(\pi))$ .

- ii. Recall that  $0 \leq g(s, a) \leq V_{\max}$  for all  $g \in \mathcal{G}$ . Then for any  $(s, a)$  we have  $\sup_{g \in \mathcal{G}} |\mathbb{E}_{s' \sim \hat{T}(s, a)} g(s') - \mathbb{E}_{s' \sim T^*(s, a)} g(s')| \leq V_{\max} \text{TV}(\hat{T}(s, a), T^*(s, a))$ . Combining the definition of  $\ell_w(g, T)$ , Eq. (6) and statistical error we get  $\sup_{g \in \mathcal{G}} \ell_{w, \pi, T}(g, T) \leq \tilde{\mathcal{O}}(\epsilon_\rho(\pi) + 1/\sqrt{n} + V_{\max} \text{TV}(\hat{\mu}, \mu))$  under event  $\mathcal{E}$ .
- iii. For the last term regarding distribution mismatch, we combine Eq. (7) and Lemma 8. We can upper bound this term by  $\mathcal{O}(\epsilon_\rho(\pi) + \epsilon_\mu(\pi))$ .
- iv. The final term arises due to the potential estimation error in the behavior policy distribution.

Theorem 4 follows directly from combining Lemma 3 and Lemma 5. Note that Theorem 4 accounts for estimation error in the behavior policy, misspecification in the dynamics model class, and misspecification in the value function class, the latter two in a more local, tighter form than prior work.

### Illustrative Example

To build intuition of where our approach may yield benefits, we provide an illustrative example where Alg. 1 has better performance than existing approaches: an MDP whose state space is partitioned into several parts. The model class is restricted so that every model can only be accurate on one part of the state space. When each deterministic policy only visits one part of the state space, the local misspecification error is small — for each policy, there exists a dynamics model in the set which can accurately estimate the distribution of states and actions visited under that policy. In contrast, if the dynamics are learned to fit the whole state space, the estimation error will be large.

More precisely, for a fixed parameter  $d$ , consider a MDP where  $\mathcal{S} = \{s_0, \dots, s_d\} \cup \{s_g, s_b\}$ . There are  $d$  actions denoted by  $a_1, \dots, a_d$ . The true dynamics are deterministic and given by

$$T^*(s_0, a_i) = s_i, \quad T^*(s_i, a_j) = \begin{cases} s_g, & \text{if } \mathbb{I}[i = j], \\ s_b, & \text{if } \mathbb{I}[i \neq j], \end{cases} \quad (9)$$

$$T^*(s_g, a_i) = s_g, \quad T^*(s_b, a_i) = s_b, \quad \forall i \in [d]. \quad (10)$$

And the reward is  $r(s, a_i) = \mathbb{I}[s = s_g], \forall i \in [d]$ .

The transition function class  $\mathcal{T}$  is parameterized by  $\theta \in \mathbb{R}^d$ . For a fixed  $\theta$ , the transition for states  $s_1, \dots, s_d$  is

$$T_\theta(s_i, a_j) = \begin{cases} s_g, & \text{w.p. } \frac{1}{2}(1 + e_j^\top \theta), \\ s_b, & \text{w.p. } \frac{1}{2}(1 - e_j^\top \theta), \end{cases} \quad (11)$$

where  $e_j$  is the  $j$ -th standard basis of  $\mathbb{R}^d$ . The transitions for states  $s_0, s_g, s_b$  is identical to the true dynamics  $T^*$ . But the transition model  $T_\theta$  in the function class must use the same parameter  $\theta$  to approximate the dynamics in states  $s_1, \dots, s_d$ , which makes it misspecified.

**Decoupling learning the dynamics model and policy is suboptimal.** Most prior algorithms first learn a dynamics model and then do planning with that model. However, note here that the optimal action induced by MDP planning given a particular  $T_\theta$  is suboptimal (assuming a uniformly random tie-breaking). This is because, for any given  $\theta$ , that dynamics model will estimate the dynamics of states  $s_1, \dots, s_d$

as being identical, with identical resulting value functions. Note this is suboptimality will occur in this example even if the dataset is large and covers the state-action pairs visited by any possible policy ( $\epsilon_\mu(\pi) = 0$ ), the value function class is tabular and can represent any value function  $\epsilon_V = 0$ , the behavior policy is known or the resulting estimation error is small ( $\text{TV}(\hat{\mu}, \mu) = 0$ , and  $\zeta = 0$ ). In such a case, Theorem 4 guarantees that with high probability, our algorithm will learn the optimal policy because there exist couplings of the dynamics models and optimal policies such that the local misspecification error  $\epsilon_\rho = 0$ . This demonstrates that prior algorithms (including MML (Voloshin, Jiang, and Yue 2021)) that decouple the learning of dynamics and policy can be suboptimal. We now state this more formally:

**Theorem 6.** *Consider any (possibly stochastic) algorithm that outputs an estimated dynamics  $T_\theta \in \mathcal{T}$ . Let  $\pi_\theta$  be the greedy policy w.r.t.  $T_\theta$  (with ties breaking uniformly at random). Then*

$$\max_{\pi} \eta(T^*, \pi) - \eta(T^*, \pi_\theta) \geq \frac{(A-1)\gamma^2}{A(1-\gamma)}. \quad (12)$$

As a side point, we also show that the off-policy estimation error in Voloshin, Jiang, and Yue (2021) is large when the dynamics model class is misspecified in Proposition 7. We defer this result to the Appendix.

## Experiments

While our primary contribution is theoretical, we now investigate how our method can be used for offline model-based policy selection with dynamics model misspecification. We first empirically evaluate our method on Linear-Quadratic Regulator (LQR), a commonly used environment in optimal control theory (Bertsekas et al. 2000), in order to assess: *Can Algorithm 1 return the optimal policy when we have both model and distribution mismatch?* We also evaluate our approach using D4RL (Fu et al. 2020), a standard offline RL benchmark for continuous control tasks. Here we consider: *Given policies and dynamics pairs obtained using state-of-the-art offline model-based RL methods with ensemble dynamics, does Alg. 1 allow picking the best policy, outperforming previous methods?*

### Linear-Quadratic Regulator (LQR)

LQR is defined by a linear transition dynamics  $s_{t+1} = A s_t + B a_t + \eta$ , where  $s_t \in \mathbb{R}^n$  and  $a_t \in \mathbb{R}^m$  are state and action at time step  $t$ , respectively.  $\eta \sim \mathcal{N}(0, \sigma^2 I)$  is random noise. LQR has a quadratic reward function  $\mathcal{R}(s, a) = -(s^T Q s + a^T R a)$  with  $Q \in \mathbb{R}^{n \times n}$  and  $R \in \mathbb{R}^{m \times m}$  being positive semi-definite matrices,  $Q, R \succeq 0$ . The optimal controller to maximize the sum of future rewards  $\sum_{t=1}^H -(s_t^T Q s_t + a_t^T R a_t)$  until the end of horizon  $H$  has the form  $a_t = -K s_t$  ( $K \in \mathbb{R}^{m \times n}$ ) (Bertsekas et al. 2000). The value function is also a quadratic function,  $V(s) = s^T U s + q$  for some constant  $q$  and positive semi-definite matrix  $U \succeq 0$ . In the experiment, the state space is  $[-1, 1]$ .

**Misspecified transition classes.** Consider a 1D version of LQR with  $A(x) = (1 + x/10), B(x) = (-0.5 - x/10)$ ,

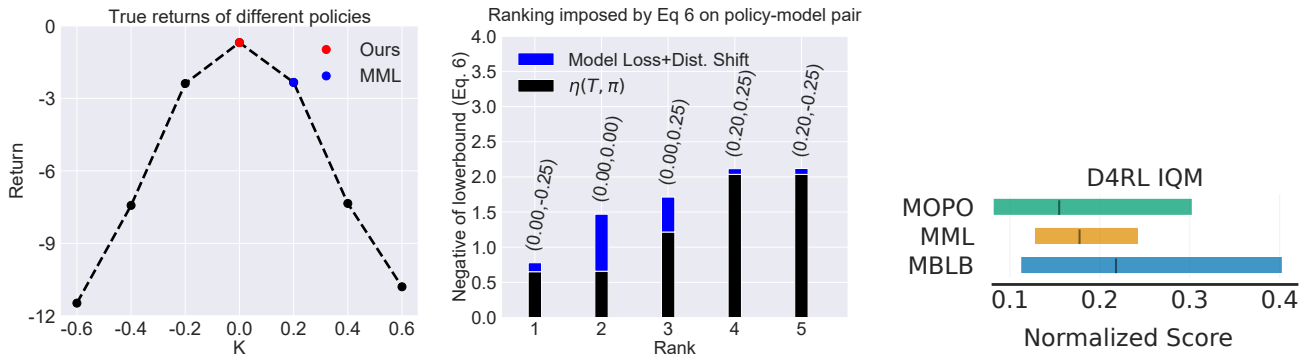


Figure 1: Left: Visualization of true policy value  $\eta(T^*, \pi)$ . Our algorithm picks the optimal policy, whereas MML picks a suboptimal policy. Middle: Visualization of negative lower bounds  $\text{lb}(T, \pi)$  for different policies and models (indexed by the values of  $(v, u)$ ). Right: We show the interquartile mean (IQM) scores of two model-based lower bounds (MML and MBLB) and a recent model-based policy learning algorithm (MOPO) on D4RL.

$Q = 1, R = 1$  and noise  $\eta \sim \mathcal{N}(0, 0.05)$ . Our true dynamics is given by  $x^* = 6$ , and the corresponding optimal policy has  $K = -1.1$ . Function classes used by Alg. 1 are finite and computed as follows: (i) the value function class  $\mathcal{G}$  contains the value functions of 1D LQR with parameters  $x \in \{2, 4, 10\}$  and  $K \in \{-1.1, -0.9, -0.7\}$ ; (ii) the transition class  $\mathcal{T}$  is misspecified. We use the following transition class  $T_u \in \mathcal{T}$  parametrized by  $u$ ,

$$T_u = \begin{cases} s_{t+1} = A(x^*)s_t - B(x^*)a_t, & s_t \in [u, u + 1], \\ s_{t+1} = s_t, & \text{otherwise,} \end{cases}$$

with  $u \in \{-0.75, -0.5, -0.25, 0, 0.25\}$ . In other words, the capacity of the transition class is limited – each function can only model the true dynamics of a part of the states; (iii) the policy class is given by  $\pi_v$  parameterized by  $v$ , and  $\pi_v(s) = -1.1(s - v) + \mathcal{N}(0, 0.01)$  with  $v \in \{-0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6\}$ . Intuitively,  $\pi_v$  tries to push the state toward  $s = v$ .

Since the state and action spaces are one dimensional, we can compute the density ratio  $w_{\pi, T}$  efficiently by discretization. The implementation details are deferred to Appendix.

**Baseline.** We compare our algorithm to minimizing MML loss as described in the OPO algorithm of Voloshin, Jiang, and Yue (2021, Algorithm 2). MML strictly outperformed VAML (Farahmand, Barreto, and Nikovski 2017) as shown in the experiments of (Voloshin, Jiang, and Yue 2021); hence, we only compare to MML in our experiments.

**Results.** Figure 1 (Left) shows the return of different policies under the true environment. Our method picks the optimal policy for the true model, whereas MML picks the wrong policy. In Figure 1 (Middle), we also visualize different terms in the definition of  $\text{lb}(T, \pi)$  (Eq. (5)). Note that the model loss for different policy is different (model loss for  $(v, u) = (0, 0)$  is significantly larger than  $(0.0, -0.25)$ , even if the dynamics are the same). This is because the model loss is evaluated with a different density ratio.

This highlights the main benefit of our method over the baseline. Since the model class is misspecified, maximizing over the weight function  $w$  in the MML loss results in an

unrealistically large loss value for some models. However, if the chosen policy does not visit the part of the state space with a large error, there is no need to incur a high penalty.

## D4RL

D4RL (Fu et al. 2020) is an offline RL standardized benchmark designed and commonly used to evaluate the progress of offline RL algorithms. This benchmark is standard for evaluating offline policy learning algorithms. Here, we use a state-of-the-art policy learning algorithm MOPO (Yu et al. 2020) to propose a set of policy-transition model tuples – for  $N$  policy hyperparameters and  $K$  transition models, we can get  $M \times K$  tuples:  $\{(\pi_1, T_1), (\pi_1, T_2), \dots, (\pi_N, T_K)\}$ . The MOPO algorithm learns an ensemble of transition models and randomly chooses one to sample trajectories during each episode of training. Instead, we choose one transition model to generate trajectories for the policy throughout the entire training. In our experiment, we choose  $M = 1$  and  $K = 5$ , and train each tuple for 5 random seeds on Hopper and HalfCheetah tasks (see Appendix). We then compute the model-based lower bound for each  $(\pi_i, T_j)$ , and select the optimal policy that has the highest lower bound. We learn the dynamics using 300k iterations and we train each policy using 100k gradient iterations steps with SAC (Haarnoja et al. 2018) as the policy gradient algorithm, imitating MOPO (Yu et al. 2020) policy gradient update.

**MML.** Voloshin, Jiang, and Yue (2021) recommended two practical implementations for computing MML lower bounds. The implementation parametrizes  $w(s, a)V(s')$  jointly via a new function  $h(s, a, s')$ . We refer readers to Prop 3.5 from Voloshin, Jiang, and Yue (2021) for a detailed explanation. We describe how we parametrize this function as follows:

- **Linear:** Voloshin, Jiang, and Yue (2021) showed that if  $T, V, \mu$  are all from the linear function classes, then a model  $T$  that minimizes MML loss is both unique and identifiable. This provides a linear parametrization of  $h(s, a, s') = \psi(s, a, s')^T \theta$ , where  $\psi$  is a basis function. We choose  $\psi$  to be either a *squared* basis function or a

Dataset Type	Env	MOPO	MML (Squared)	MML (Polynomial)	MML (RKHS)	MBLB (Linear)	MBLB (Quad)
medium	hopper	175.4 (95.3)	379.4 (466.4)	375.6 (459.5)	375.0 (459.9)	591.7 (523.1)	<b>808.5</b> (502.7)
med-expert	hopper	183.8 (94.4)	160.9 (131.5)	116.5 (148.4)	61.4 (35.0)	<b>261.1</b> (157.9)	242.5 (134.0)
expert	hopper	80.4 (63.4)	93.8 (87.9)	61.6 (61.9)	70.0 (56.2)	118.2 (61.6)	<b>121.0</b> (72.5)
medium	halfcheetah	599.8 (668.4)	1967.6 (1707.5)	2625.1 (937.2)	<b>3858.2</b> (1231.1)	3290.4 (1753.1)	2484.2 (1526.8)
med-expert	halfcheetah	-486.6 (48.1)	-188.5 (137.2)	-77.0 (252.5)	-343.2 (225.2)	<b>207.4</b> (509.5)	192.8 (432.0)

Table 2: We report the mean and (standard deviation) of selected policy’s simulator environment performance across 5 random seeds. MML and MBLB are used as model-selection procedures where they select the best policy for each seed. Our method is choosing the most near-optimal policy across the datasets.

*polynomial* basis function with degree 2.

- **Kernel:** Using a radial basis function (RBF) over  $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$  and computing  $K((s, a, s'), (\tilde{s}, \tilde{a}, \tilde{s}'))$ , Voloshin, Jiang, and Yue (2021) showed that there exists a closed-form solution to compute the maxima of the MML loss (RKHS). Here, there is no need for any gradient update, we only sample  $s'$  from  $T$ .

**MBLB (Ours).** For a continuous control task, we compute our model-based lower bound (MBLB) as follows:

**Compute**  $\eta(T, \pi)$ . Although it is reasonable to directly use a value function  $V_T^\pi$  trained during policy learning to compute  $\eta(T, \pi)$ , Paine et al. (2020); Kumar et al. (2021) points out how this value function often severely over-estimates the actual discounted return. Therefore, we estimate the expected value of policy  $\pi$  using the generalized advantage estimator (GAE) (Schulman et al. 2016). For a sequence of transitions  $\{s_t, a_t, r(s_t, a_t), s_{t+1}\}_{t \in [0, N]}$ , it is defined as:  $A_t = \sum_{t'=t}^{t+N} (\gamma\lambda)^{t'-t} (r(s_{t'}, a_{t'}) + \gamma V_\phi(s_{t'+1}) - V_\phi(s_{t'}))$ , with  $\lambda$  a fixed hyperparameter and  $V_\phi$  the value function estimator at the previous optimization iteration. Then, to estimate the value function, we solve the non-linear regression problem  $\text{minimize}_\phi \sum_{t'=t}^{t+N} (V_\phi(s_{t'}) - \hat{V}_{t'})^2$  where  $\hat{V}_{t'} = A_t + V_\phi(s_{t'})$ . We also provide a comparison to using the standard TD-1 Fitted Q Evaluation (FQE) (Le, Voloshin, and Yue 2019) instead in Table A1 in the Appendix. We find that using GAE provides better policy evaluation estimations.

**Behavior density modeling.** We use a state-of-the-art normalizing flow probability model to estimate the density of state-action pairs (Papamakarios et al. 2021). For  $\rho_T^\pi$ , we sample 10,000 trajectories from  $T, \pi$ , and estimate the corresponding density; for the behavior distribution  $\mu$ , we use the given dataset  $\mathcal{D}$ . We empirically decide the number of training epochs that will give the model the best fit.

**Compute**  $\sup_{g \in \mathcal{G}} |\ell_{w_{\pi, T}}(g, T)|$ . We parametrize  $g$  either as a linear function of state:  $g(s) = m^T s$ , or a quadratic function of the state:  $g(s) = s^T M s + b$ . We use gradient ascent on  $\ell_{w_{\pi, T}}(g, T)$  to maximize this objective.

**Results.** We report the results in Table 2. There is general overlap across seeds for the performance between various methods, but our approach has the best average performance or is within the standard deviation of the best. We also show that for different choices of how we parameterize the  $w(s, a)V(s')$  distribution (MML) and how we choose the family of  $g$  test function (MBLB), we are selecting different final policies. However, overall, MBLB can pick better-performing final policies with two different parametrizations while MML is choosing lower-performing policies with its three parametrizations. We find that our approach of selecting among the set of policies computed from each of the models used by MOPO consistently outperforms the policy produced by MOPO in the considered tasks.

To summarize these results, we report the interquartile mean (IQM) scores of each method in Figure 1 (Right). IQM is an outlier robust metric proposed by Agarwal et al. (2021) to compare deep RL algorithms. We create the plot by sampling with replacement over all runs on all datasets 50000 times. Though there is significant overlap, our method generally outperforms policies learned from MOPO.

## Conclusion

There are many directions for future work. The current  $\text{lb}(T, \pi)$  implementation with density ratio  $w_{\pi, T}(s, a)$  is not differentiable: an interesting question is to make this differentiable so that we can directly optimize a policy. Another interesting question would be to construct estimators for the local misspecification errors  $\epsilon_\rho, \epsilon_\mu$  and  $\epsilon_V$ , which could be used to refine the model class to optimize performance.

To conclude, this paper studies model-based offline reinforcement learning with local model misspecification errors, and proves a novel safe policy improvement theorem. Our theoretical analysis shows the benefit of this tighter analysis and approach. We illustrate the advantage of our method over prior work in a small linear quadratic example and also demonstrate that it is competitive or has stronger performance than recent model-based offline RL methods on policy selection in a set of D4RL tasks.

## Acknowledgments

Research reported in this paper was sponsored in part by NSF grant #2112926, the DEVCOM Army Research Laboratory under Cooperative Agreement W911NF-17-2-0196 (ARL IoBT CRA) and a Stanford Hoffman-Yee grant. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

## References

- Agarwal, R.; Schwarzler, M.; Castro, P. S.; Courville, A. C.; and Bellemare, M. 2021. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34: 29304–29320.
- Argenson, A.; and Dulac-Arnold, G. 2020. Model-based offline planning. *arXiv preprint arXiv:2008.05556*.
- Bertsekas, D. P.; et al. 2000. *Dynamic programming and optimal control: Vol. 1*. Athena scientific Belmont.
- Chen, J.; and Jiang, N. 2019. Information-Theoretic Considerations in Batch Reinforcement Learning. In *International Conference on Machine Learning*, 1042–1051.
- Curi, S.; Berkenkamp, F.; and Krause, A. 2020. Efficient Model-Based Reinforcement Learning through Optimistic Policy Search and Planning. *Advances in Neural Information Processing Systems*, 33.
- Delgrange, F.; Nowe, A.; and Pérez, G. A. 2022. Distillation of rl policies with formal guarantees via variational abstraction of markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 6497–6505.
- Farahmand, A.-m.; Barreto, A.; and Nikovski, D. 2017. Value-aware loss function for model-based reinforcement learning. In *Artificial Intelligence and Statistics*, 1486–1494. PMLR.
- Farahmand, A. M.; Munos, R.; and Szepesvári, C. 2010. Error propagation for approximate policy and value iteration. In *Advances in Neural Information Processing Systems*.
- Fu, J.; Kumar, A.; Nachum, O.; Tucker, G.; and Levine, S. 2020. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*.
- Fu, J.; and Levine, S. 2021. Offline Model-Based Optimization via Normalized Maximum Likelihood Estimation. *arXiv preprint arXiv:2102.07970*.
- Fujimoto, S.; van Hoof, H.; Meger, D.; et al. 2018. Addressing function approximation error in actor-critic methods. *Proceedings of Machine Learning Research*, 80.
- Gelada, C.; Kumar, S.; Buckman, J.; Nachum, O.; and Bellemare, M. G. 2019. Deepmdp: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, 2170–2179. PMLR.
- Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*.
- Jiang, N.; and Huang, J. 2020. Minimax confidence interval for off-policy evaluation and policy optimization. *arXiv preprint arXiv:2002.02081*.
- Jin, C.; Yang, Z.; Wang, Z.; and Jordan, M. I. 2020. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, 2137–2143.
- Kakade, S.; and Langford, J. 2002. Approximately Optimal Approximate Reinforcement Learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, 267–274. Morgan Kaufmann Publishers Inc.
- Kallus, N.; and Uehara, M. 2020. Double Reinforcement Learning for Efficient Off-Policy Evaluation in Markov Decision Processes. *J. Mach. Learn. Res.*, 21: 167–1.
- Kidambi, R.; Rajeswaran, A.; Netrapalli, P.; and Joachims, T. 2020. Morel: Model-based offline reinforcement learning. *arXiv preprint arXiv:2005.05951*.
- Kumar, A.; Fu, J.; Soh, M.; Tucker, G.; and Levine, S. 2019. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. *Advances in Neural Information Processing Systems*, 32: 11784–11794.
- Kumar, A.; Singh, A.; Tian, S.; Finn, C.; and Levine, S. 2021. A Workflow for Offline Model-Free Robotic Reinforcement Learning. In *5th Annual Conference on Robot Learning*.
- Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative Q-Learning for Offline Reinforcement Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1179–1191. Curran Associates, Inc.
- Le, H.; Voloshin, C.; and Yue, Y. 2019. Batch policy learning under constraints. In *International Conference on Machine Learning*, 3703–3712. PMLR.
- Liu, Q.; Li, L.; Tang, Z.; and Zhou, D. 2018a. Breaking the curse of horizon: infinite-horizon off-policy estimation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 5361–5371.
- Liu, Y.; Bacon, P.-L.; and Brunskill, E. 2020. Understanding the curse of horizon in off-policy evaluation via conditional importance sampling. In *International Conference on Machine Learning*, 6184–6193. PMLR.
- Liu, Y.; Gottesman, O.; Raghu, A.; Komorowski, M.; Faisal, A.; Doshi-Velez, F.; and Brunskill, E. 2018b. Representation Balancing MDPs for Off-Policy Policy Evaluation. *Advances in neural information processing systems*.
- Liu, Y.; Swaminathan, A.; Agarwal, A.; and Brunskill, E. 2020. Provably Good Batch Off-Policy Reinforcement Learning Without Great Exploration. *Advances in Neural Information Processing Systems*, 33.
- Malik, A.; Kuleshov, V.; Song, J.; Nemer, D.; Seymour, H.; and Ermon, S. 2019. Calibrated model-based deep reinforcement learning. In *International Conference on Machine Learning*, 4314–4323. PMLR.



- Matsushima, T.; Furuta, H.; Matsuo, Y.; Nachum, O.; and Gu, S. 2020. Deployment-efficient reinforcement learning via model-based offline optimization. *arXiv preprint arXiv:2006.03647*.
- Munos, R. 2003. Error bounds for approximate policy iteration. In *ICML*, volume 3, 560–567.
- Nachum, O.; Chow, Y.; Dai, B.; and Li, L. 2019. DualDICE: Behavior-Agnostic Estimation of Discounted Stationary Distribution Corrections.
- Paine, T. L.; Paduraru, C.; Michi, A.; Gulcehre, C.; Zolna, K.; Novikov, A.; Wang, Z.; and de Freitas, N. 2020. Hyperparameter selection for offline reinforcement learning. *arXiv preprint arXiv:2007.09055*.
- Papamakarios, G.; Nalisnick, E. T.; Rezende, D. J.; Mohamed, S.; and Lakshminarayanan, B. 2021. Normalizing Flows for Probabilistic Modeling and Inference. *J. Mach. Learn. Res.*, 22(57): 1–64.
- Quillen, D.; Jang, E.; Nachum, O.; Finn, C.; Ibarz, J.; and Levine, S. 2018. Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 6284–6291. IEEE.
- Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; and Abbeel, P. 2016. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*.
- Singh, A.; Yu, A.; Yang, J.; Zhang, J.; Kumar, A.; and Levine, S. 2020. COG: Connecting New Skills to Past Experience with Offline Reinforcement Learning. *arXiv preprint arXiv:2010.14500*.
- Swaminathan, A.; and Joachims, T. 2015. Batch learning from logged bandit feedback through counterfactual risk minimization. *The Journal of Machine Learning Research*, 16(1): 1731–1755.
- Swazinna, P.; Udluft, S.; and Runkler, T. 2020. Overcoming Model Bias for Robust Offline Deep Reinforcement Learning. *arXiv preprint arXiv:2008.05533*.
- Thomas, P.; Theocharous, G.; and Ghavamzadeh, M. 2015. High confidence policy improvement. In *International Conference on Machine Learning*, 2380–2388. PMLR.
- Thomas, P. S.; da Silva, B. C.; Barto, A. G.; Giguere, S.; Brun, Y.; and Brunskill, E. 2019. Preventing undesirable behavior of intelligent machines. *Science*, 366(6468): 999–1004.
- Thomas, P. S.; Theocharous, G.; Ghavamzadeh, M.; Durgkar, I.; and Brunskill, E. 2017. Predictive Off-Policy Policy Evaluation for Nonstationary Decision Problems, with Applications to Digital Marketing. In *AAAI*, 4740–4745.
- Uehara, M.; Huang, J.; and Jiang, N. 2020. Minimax weight and q-function learning for off-policy evaluation. In *International Conference on Machine Learning*, 9659–9668. PMLR.
- van Hasselt, H. P.; Hessel, M.; and Aslanides, J. 2019. When to use parametric models in reinforcement learning? In *Walach, H.; Larochelle, H.; Beygelzimer, A.; d’Alché-Buc, F.; Fox, E.; and Garnett, R., eds., Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Voloshin, C.; Jiang, N.; and Yue, Y. 2021. Minimax Model Learning. In *International Conference on Artificial Intelligence and Statistics*, 1612–1620. PMLR.
- Xie, T.; and Jiang, N. 2020. Batch value-function approximation with only realizability. *arXiv preprint arXiv:2008.04990*.
- Xie, T.; Ma, Y.; and Wang, Y. 2019. Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. *Advances in neural information processing systems*.
- Yin, M.; and Wang, Y.-X. 2020. Asymptotically efficient off-policy evaluation for tabular reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, 3948–3958. PMLR.
- Yu, T.; Kumar, A.; Rafailov, R.; Rajeswaran, A.; Levine, S.; and Finn, C. 2021. Combo: Conservative offline model-based policy optimization. *arXiv preprint arXiv:2102.08363*.
- Yu, T.; Thomas, G.; Yu, L.; Ermon, S.; Zou, J.; Levine, S.; Finn, C.; and Ma, T. 2020. MOPO: Model-based Offline Policy Optimization. *arXiv preprint arXiv:2005.13239*.
- Zhan, X.; Zhu, X.; and Xu, H. 2021. Model-Based Offline Planning with Trajectory Pruning. *arXiv preprint arXiv:2105.07351*.
- Zhang, R.; Dai, B.; Li, L.; and Schuurmans, D. 2019. GenDICE: Generalized Offline Estimation of Stationary Values. In *International Conference on Learning Representations*.
- Zhang, S.; Liu, B.; and Whiteson, S. 2020. Gradientdice: Rethinking generalized offline estimation of stationary values. In *International Conference on Machine Learning*, 11194–11203. PMLR.