Optimistic Model Rollouts for Pessimistic Offline Policy Optimization

Yuanzhao Zhai[1,2], Yiying Li[3], Zijian Gao[1,2], Xudong Gong[1,2], Kele Xu[1,2], Dawei Feng[1,2*], Ding Bo[1,2], Huaimin Wang[1,2]

[1]National University of Defense Technology, Changsha, China [2]State Key Laboratory of Complex & Critical Software Environment [3]Artificial Intelligence Research Center, DII, Beijing, China
{yuanzhaozhai, liyiying10, gaozijian19, gongxudong09, dingbo, hmwang}@nudt.edu.cn, kelele.xu@gmail.com, davyfeng.c@gmail.com

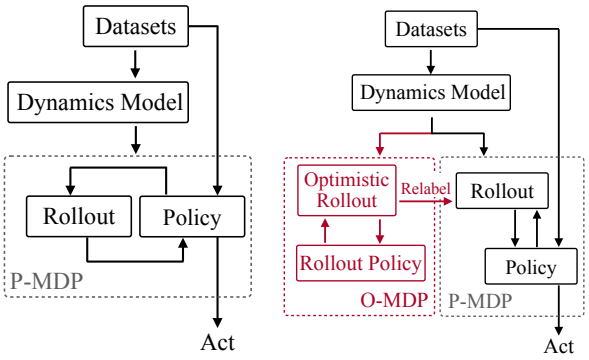
Abstract

Model-based offline reinforcement learning (RL) has made remarkable progress, offering a promising avenue for improving generalization with synthetic model rollouts. Existing works primarily focus on incorporating pessimism for policy optimization, usually via constructing a Pessimistic Markov Decision Process (P-MDP). However, the P-MDP discourages the policies from learning in out-of-distribution (OOD) regions beyond the support of offline datasets, which can under-utilize the generalization ability of dynamics models. In contrast, we propose constructing an Optimistic MDP (O-MDP). We initially observed the potential benefits of optimism brought by encouraging more OOD rollouts. Motivated by this observation, we present ORPO, a simple yet effective model-based offline RL framework. ORPO generates Optimistic model Rollouts for Pessimistic offline policy Optimization. Specifically, we train an optimistic rollout policy in the O-MDP to sample more OOD model rollouts. Then we relabel the sampled state-action pairs with penalized rewards and optimize the output policy in the P-MDP. Theoretically, we demonstrate that the performance of policies trained with ORPO can be lower-bounded in linear MDPs. Experimental results show that our framework significantly outperforms P-MDP baselines by a margin of 30%, achieving state-of-the-art performance on the widely-used benchmark. Moreover, ORPO exhibits notable advantages in problems that require generalization.

1 Introduction

In scenarios where online trial-and-error are too costly or prohibited, such as autonomous driving (Yu et al. 2018), healthcare (Gottesman et al. 2019), and robotics (Mandlekar et al. 2020), offline RL (Levine et al. 2020) has emerged as a solution to leverage previously-collected datasets. While successful, recent research (Wang et al. 2021; Lee et al. 2022) demonstrates that model-free offline RL methods typically learn overly conservative policies and lack generalization beyond the datasets.

Model-based approach, which leverages a learned dynamics model to generate rollouts for policy optimization (Sutton 1990), has been introduced to offline RL, achieving remarkable progress (Yu et al. 2020; Rigter et al. 2022). In the context of offline RL, the dynamic models may exhibit inaccuracies due to limited datasets. To avoid over-estimation on out-of-distribution (OOD) data, prior methods construct a Pessimistic Markov Decision Process (P-MDP) based on uncertainty quantification of dynamics models (Yu et al. 2020; Kidambi et al. 2020), which lower-bounds the real MDP.



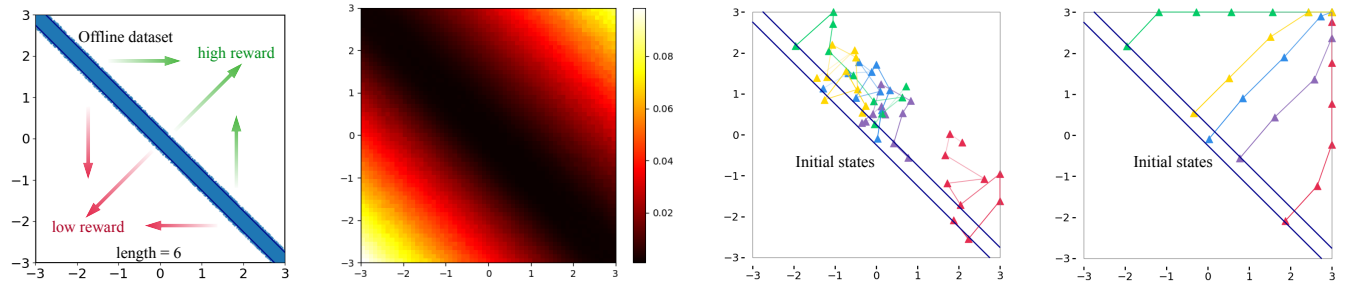
(a) Prior approach. (b) ORPO framework.

Figure 1: (a) Previous model-based offline RL generates model rollouts and optimizes the policy within the P-MDP. (b) We decouple the training of optimistic rollout policies from the pessimistic policy optimization.

Dynamics models trained in a supervised manner can exhibit refined generalization capacity for some near-distribution OOD state-action pairs, which is mainly studied and utilized in the model-based online RL (Janner et al. 2019; Moerland et al. 2023). Recent works (An et al. 2021; Bai et al. 2022) have demonstrated that OOD sampling can effectively regularize behaviors and enhance generalization for offline policy optimization. However, P-MDP used in prior model-based offline RL penalizes OOD state-action pairs that have high uncertainty (Figure 1(a)), discouraging policies from sampling in OOD regions. Therefore, utilizing only pessimistic rollouts may under-utilize the dynamics model, thereby limiting generalization. We verify this by designing a toy task in Figure 2.

This paper delves into the efficacy of optimism in the context of model-based offline RL, aiming to take full advantage of the learned dynamics model. To introduce optimism when

*Corresponding author.



(a) Scenario of our toy experiment.

(b) Uncertainty value of the learned dynamics model.

(c) Evaluation trajectories of the policy trained in the P-MDP.

(d) Evaluation trajectories of the policy trained with ORPO.

Figure 2: (a) In toy experiments with a 2-dimensional continuous state space and action space, the coordinate origin (0, 0) is taken as the central point of the square region. The agent starts at the region between lines $y = -x - 0.25$ and $y = -x + 0.25$, and the goal is to move upper right to obtain high rewards. The offline dataset only contains transitions whose state is in the initial area. (b) The further the states are from the offline dataset, the higher the estimated uncertainty value by the dynamics model. (c) The policy trained with MOPO (Yu et al. 2020) with only P-MDP can not reach regions with high reward but high uncertainty. (d) With more optimistic model rollouts but optimization in the same P-MDP, ORPO agents can learn to reach states with high rewards and avoid regions with low rewards. Please refer to Appendix C.1 for the detailed experimental setup.

generating model rollouts, we can flip the sign of uncertainty penalties of the Pessimistic MDP (P-MDP), resulting in an Optimistic MDP (O-MDP). However, using only O-MDP in model-based offline RL contrasts the provably efficient pessimism (Jin, Yang, and Wang 2021) in offline RL, and may lead policies to risky state-action regions with large dynamics model errors. Hence, the central question that this work is trying to answer is: can we train an offline policy that exploits the generalization ability of dynamics models while still adopting provably efficient pessimism?

To this end, we present a novel model-based offline RL algorithmic framework called ORPO (Figure 1(b)), which decouples the training of rollout policies from the pessimistic policy optimization. Specifically, we construct an O-MDP to train optimistic rollout policies, which have a higher probability of accessing OOD state-action regions based on the generalization ability of dynamics models. Subsequently, we relabel the optimistic model rollouts by assigning them penalized rewards in the P-MDP. The agent trained by the relabeled optimistic rollouts is more likely to select OOD actions when their value estimations are high, while avoiding low-value risky regions, as shown in Figure 2(d). In summary, our main contributions are:

- We introduce the construction of an O-MDP in the model-based offline RL framework, highlighting its potential benefits derived from encouraging increased OOD sampling.
- We present ORPO, a novel framework that generates optimistic model rollouts for pessimistic offline policy optimization. We theoretically provide the lower bound of the expected return of policies trained with ORPO.
- Through empirical evaluations, ORPO policies outperform the P-MDP baseline by a substantial margin of 30%, and achieve competitive or superior scores compared to baseline methods in 8 out of 12 datasets from the D4RL benchmark (Fu et al. 2020). Furthermore, our method has better performance compared to the state-of-the-art in two datasets requiring policy to generalize.

2 Related Works

Off-policy online RL algorithms (Fujimoto, Hoof, and Meger 2018; Haarnoja et al. 2018) often suffer from inefficiency due to extrapolation errors (Fujimoto, Meger, and Precup 2019). These errors arise from overestimating the values of out-of-distribution (OOD) state-action pairs beyond the support of offline datasets. Offline RL is proposed to learn effective policies from a logged dataset without interacting with the environment (Levine et al. 2020), which can generally be categorized into two types: model-free and model-based. Model-free offline RL methods learn conservative value functions (Kumar et al. 2020; Kostrikov, Nair, and Levine 2022) or directly constrain the policy (Fujimoto, Meger, and Precup 2019; Kumar et al. 2019; Fujimoto and Gu 2021) to preclude OOD actions. However, policy trained by such methods may be overly conservative (Lee et al. 2022), lacking generalization ability beyond the offline dataset (Wang et al. 2021).

Model-based Offline RL. Model-based offline RL algorithms first train a dynamics model using supervised learning with the logged dataset. Then the dynamics model can be used to optimize policies, in which Dyna-style algorithm (Sutton 1990) is adopted by a number of recent methods (Yu et al. 2020, 2021; Clavera, Fu, and Abbeel 2020; Rafailov et al. 2021). By utilizing the additional synthetic data generated by the learned dynamics model, model-based offline RL methods have the potential to exhibit better generalization abilities compared to model-free (Kumar et al. 2020; Kostrikov, Nair, and Levine 2022; Fujimoto and Gu 2021).

Since the limitation of the logged dataset, it is essential to quantify how trustable the model is for specific rollouts. Both MOPO (Yu et al. 2020) and MOReL (Kidambi et al. 2020) construct the P-MDP to optimize the policy, where

rewards are penalized according to uncertainty quantification. Many recent works aim to incorporate pessimism into policy optimization, via backward dynamics model (Wang et al. 2021; Lyu, Li, and Lu 2022), uncertainty-free conservatism (Yu et al. 2021) or robust MDPs (Guo, Yunfeng, and Geng 2022; Rigter et al. 2022). In contrast, we investigate the potential benefits of optimism for training rollout policies. We adopt the P-MDP from MOPO for pessimistic policy optimization and introduce the O-MDP for generating model rollouts. Importantly, our proposed framework is not limited to MOPO and can be easily combined with other model-based offline RL methods.

Uncertainty Aware Reinforcement Learning. Uncertainty plays a crucial role in RL. Optimism in the Face of Uncertainty (OFU) (Abbasi-Yadkori, Pál, and Szepesvári 2011) principle is commonly employed in online RL for active and efficient environment exploration (Lockwood and Si 2022), which is provably efficient (Abbasi-Yadkori, Pál, and Szepesvári 2011; Jin et al. 2020). Uncertainty is also widely used in model-based online RL for controlling the model usage (Luo et al. 2019; Janner et al. 2019; Pan et al. 2020).

In offline RL, uncertainty is typically utilized for pessimism. As aforementioned, certain model-based offline RL methods (Lu et al. 2022) estimate the uncertainty of the dynamics model to construct P-MDPs. Additionally, recent model-free methods (An et al. 2021; Bai et al. 2022; Wu et al. 2021) employ the uncertainty quantification of Q-functions to penalize OOD state-action pairs. Our proposed framework is closely related to both the provably efficient designs for exploration in online RL and pessimism in offline RL.

3 Preliminaries

We define a Markov Decision Process (MDP) as the tuple $M = (\mathcal{S}, \mathcal{A}, T, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action space, $T(s'|s,a)$ represents the dynamics or transition distribution, $r(s,a)$ is the reward function, and $\gamma \in (0,1)$ is the discount factor. Let $\mathbb{P}^{\pi}_{T,t}(s)$ denote the probability of being in state s at time step t if actions are sampled according to π and transitions according to T. Let $\rho^{\pi}_{T}(s,a)$ be the discounted occupancy measure of policy π under dynamics T: $\rho^{\pi}_{T}(s,a) := \sum_{t=0}^{\infty} \gamma^t \mathbb{P}^{\pi}_{T,t}(s)\pi(a|s)$. The goal is to find a policy $\pi(a|s)$ that maximizes the expected discounted return $\eta_M(\pi) = \bar{\mathbb{E}}_{(s,a)\sim\rho^{\pi}_{T}}[r(s,a)]$. The value function $V_M(s) := \mathbb{E}_{\pi,T}[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t)|s_0 = s]$ gives the expected discounted return when starting from state s.

For offline RL where agents can not interact with the environment, we have a previously-collected static dataset $\mathcal{D}^{env} = \{(s_j, a_j, r_j, s_{j+1})\}_{j=1}^{J}$, which consists of J transition tuples from trajectories collected by a behavior policy. Canonical model-based offline RL methods typically train an ensemble of N probabilistic networks as the dynamics model $\widehat{T} = \{\widehat{T}_i(\hat{s}'|s,a) = \mathcal{N}(\mu_i(s,a), \Sigma_i(s,a))\}_{i=1}^{N}$ to predict the next state s' from a state-action pair. Following previous works (Yu et al. 2020; Kidambi et al. 2020), we assume the reward function r is known. If $r(\cdot)$ is unknown, it can also be learned from data. The learned dynamics model $\widehat{T}$ define a model MDP $\widehat{M} = (\mathcal{S}, \mathcal{A}, \widehat{T}, r, \gamma)$. Then the goal switches to find a policy $\pi(a|s)$ that maximizes the expected discounted return with respect to $\rho^{\pi}_{\widehat{T}}$, as in $\eta_{\widehat{M}}(\pi) = \bar{\mathbb{E}}_{(s,a)\sim\rho^{\pi}_{\widehat{T}}}[r(s,a)]$.

Model-based offline RL methods often construct a P-MDP for pessimistic offline policy optimization. Notably, based on the model error between the true and learned dynamics,

$$G_{\widehat{M}}(s,a) := \mathbb{E}_{s'\sim\widehat{T}(s,a)}[V_M(s')] - \mathbb{E}_{s'\sim T(s,a)}[V_M(s')], \quad (1)$$

MOPO assumes that there is an admissible model uncertainty $u(s,a)$ that can upper-bound the model error $|G^{\pi}_{\widehat{M}}(s,a)|$:

$$u(s,a) \geq |G^{\pi}_{\widehat{M}}(s,a)|, \forall s \in \mathcal{S}, a \in \mathcal{A}. \quad (2)$$

Penalized by the estimator, pessimistic reward $r^p(s,a) = r(s,a) - \lambda^p u(s,a)$ can be used to construct P-MDP as $M^p = (\mathcal{S}, \mathcal{A}, \widehat{T}, r^p, \gamma)$, where $\lambda^p := \gamma c$ denotes the degree of pessimism and c is a constant. Define the average model uncertainty $\epsilon_u(\pi)$ as:

$$\epsilon_u(\pi) := \bar{\mathbb{E}}_{(s,a)\sim\rho^{\pi}_{\widehat{T}}} u(s,a). \quad (3)$$

Then the lower bound of performance in the real MDP can be established by the P-MDP (Yu et al. 2020) as:

$$\begin{aligned}\eta_M(\pi) =& \bar{\mathbb{E}}_{(s,a)\sim\rho^{\pi}_{\widehat{T}}}\left[r(s,a) - \gamma|G^{\pi}_{\widehat{M}}(s,a)|\right] \\ \geq& \bar{\mathbb{E}}_{(s,a)\sim\rho^{\pi}_{\widehat{T}}}[r(s,a) - \lambda^p u(s,a)] \\ =& \eta_{\widehat{M}}(\pi) - \lambda^p \epsilon_u(\pi) = \eta_{M^p}(\pi). \end{aligned} \quad (4)$$

According to Equation 4, we can optimize policies in the real MDP by improving their performance in the P-MDP.

4 Proposed Method

In this section, we present the construction of an O-MDP and discuss its potential benefits in Section 4.1. Next, we provide an overview of the ORPO framework in Section 4.2. To establish a solid theoretical foundation for ORPO, we delve into the theoretical analysis in Section 4.3.

4.1 Optimistic MDP Construction

To introduce optimism when generating model rollouts, we flip the sign of the uncertainty penalty in the Pessimistic MDP to construct an Optimistic MDP (O-MDP) $M^o = (\mathcal{S}, \mathcal{A}, \widehat{T}, r^o, \gamma)$, where $r^o(s,a) = r(s,a) + \lambda^o u(s,a)$. Subsequently, we train the rollout policy within the O-MDP, optimizing the following objective (Equation 5), where λ^o serves as a coefficient to regulate the level of optimism.

$$\pi^o = \arg\max_{\pi} \bar{\mathbb{E}}_{(s,a)\sim\rho^{\pi}_{\widehat{T}}}[r(s,a) + \lambda^o u(s,a)]. \quad (5)$$

To analyze the impact of the O-MDP, we begin by comparing the model rollouts generated under the P-MDP and O-MDP settings. We measure the distance between the rollout actions and the offline dataset actions using the ℓ_2 norm, calculated as $\mathbb{E}_{(s,a)\sim\mathcal{D}^{env}}[\|\pi(\cdot|s) - a\|_2]$, where π represents the rollout policies. As shown in Figure 3, we observe that optimistic

rollouts exhibit larger distances from the offline dataset compared to pessimistic rollouts, indicating that optimistic rollouts involve more OOD sampling. While prior works (Kumar et al. 2020; Bai et al. 2022) typically sample OOD actions using random policies, we contend that OOD model rollouts generated within the O-MDP framework possess greater value. On the one hand, optimistic rollout policies guided by the objective in Equation 5 selectively sample actions with high uncertainty and high estimated values, as opposed to random policies. This targeted sampling strategy can lead to more informative OOD actions. On the other hand, dynamics models with generalization capacity can generate better characterization for OOD model rollouts.

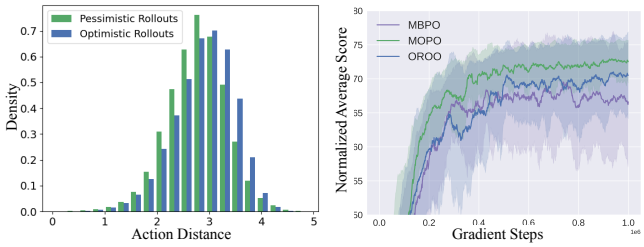


Figure 3: A case study of methods using model MDP (MBPO), P-MDP (MOPO), and O-MDP (OROO) on "Halfcheetah-medium-v2" datasets over 5 different seeds. **Left:** Histograms of distances between actions from different model rollouts and the offline dataset. **Right:** Learning curves of different methods.

To showcase the effectiveness of OOD model rollouts, we introduce a simple baseline called Optimistic model Rollouts for Optimistic policy Optimization (OROO), which is derived from MOPO by replacing the P-MDP with the O-MDP. We make the intriguing observation that OROO utilizing the O-MDP, surpasses the performance of the model MDP baseline, MBPO (Janner et al. 2019). Interestingly, our experiments in Section 5.3 reveal that OROO can even outperform MOPO on certain offline RL datasets. This finding highlights the additional benefits brought by O-MDP and the potential of optimism in model-based offline RL.

However, it is important to acknowledge that optimizing policies in the O-MDP can result in a larger model error $\epsilon_u(\pi^o)$, which in turn reduces the performance lower bound as depicted in Equation 4. This is consistent with the previous conclusion that pessimism is provably efficient in the offline setting (Jin, Yang, and Wang 2021). In fact, the choice between P-MDP and O-MDP involves a trade-off between the generalization capacity of the dynamics model and the introduced model error. Building upon the aforementioned analysis, our objective is to train optimistic rollout policies that encourage more OOD sampling while still utilizing the P-MDP to control the model error within an admissible range.

4.2 Algorithmic Framework

We now present our framework, ORPO, which is designed to generate optimistic model rollouts for offline policy optimization in a pessimistic manner. In ORPO, we decouple the training of the rollout policy from the pessimistic optimization of the output policy π^p. Instead, we focus on learning a more optimistic rollout policy denoted as π^o, which is optimized under the O-MDP constructed in Section 4.1.

Algorithm 1: Framework for Optimistic Rollout for Pessimistic Policy Optimization (ORPO)

Require: Offline dataset $\mathcal{D}^{env}$, initialized rollout policy π^o and output policy π^p.
Train the dynamics model $\widehat{T}$ with uncertainty quantifier $u(s,a)$.
Initialize the replay buffers $\mathcal{D}^o_{\pi^o} \leftarrow \varnothing, \mathcal{D}^p_{\pi^o} \leftarrow \varnothing, \mathcal{D}^p_{\pi^p} \leftarrow \varnothing$.
for epoch $1, 2, \ldots$ **do**
    Run any online RL algorithm in M^o to optimize rollout policy π^o, and add the rollouts in replay buffer to $\mathcal{D}^o_{\pi^o}$.
    Relabel $\mathcal{D}^o_{\pi^o}$ with penalized rewards according to P-MDP, obtaining $\mathcal{D}^p_{\pi^o}$.
    Collect model rollouts by sampling from π^p in M^p starting from states in $\mathcal{D}^{env}$, and add the rollouts to $\mathcal{D}^p_{\pi^p}$.
    Run any offline RL algorithm on $\mathcal{D}^{env} \cup \mathcal{D}^p_{\pi^o} \cup \mathcal{D}^p_{\pi^p}$ to optimize policy π^p.
end for
Return: Optimized output policy π^p.

The optimistic rollout policy π^o is capable of interacting with the dynamics model, allowing us to optimize it using online RL algorithms. During the training of π^o, we collect and store the optimistic rollouts $(s, \pi^o(a|s), r^o, \hat{s}')$ in a buffer denoted as $\mathcal{D}^o_{\pi^o}$. Then we directly relabel the optimistic rollouts using the penalized reward according to the P-MDP. This relabeling process transforms the rollouts into $(s, \pi^o(a|s), r^p, \hat{s}')$. We then store these relabeled optimistic rollouts into another buffer $\mathcal{D}^p_{\pi^o}$ to be used for pessimistic policy optimization. Besides, we also store pessimistic rollouts $(s, \pi^p(a|s), r^p, \hat{s}')$ which are sampled by the output policy in P-MDP, denoted as $\mathcal{D}^p_{\pi^p}$. Note that previous model-based offline RL methods (Lu et al. 2022) typically utilize $\mathcal{D}^p_{\pi^p}$ and $\mathcal{D}^{env}$ for pessimistic policy optimization. In our framework, with the inclusion of the rollout policy π^o, we can leverage the additional dataset $\mathcal{D}^p_{\pi^o}$ to introduce more OOD state-action pairs. Given the datasets $\mathcal{D}^p_{\pi^p}$, $\mathcal{D}^p_{\pi^o}$, and the offline dataset $\mathcal{D}^{env}$, our objective is to derive a policy π^p that maximizes the expected discounted return in the real MDP, i.e.,

$$\pi^p = \arg\max_{\pi}[\eta_M(\pi)]. \tag{6}$$

The behavior policy used to collect our synthetic dataset, which includes $\mathcal{D}^p_{\pi^o}$, $\mathcal{D}^p_{\pi^p}$, and $\mathcal{D}^{env}$, differs significantly from the desired output policy π^p. Therefore, we employ offline RL algorithms for pessimistic optimization. The training of the rollout policy and pessimistic policy optimization is conducted iteratively in an alternating fashion. The overall framework of ORPO is outlined in Algorithm 1. For practical implementation details, please refer to Appendix B.2.

4.3 Theoretical Analysis

Denote the optimal policy in the P-MDP as $\hat{\pi}^p$. MOPO has demonstrated that the expected return of $\hat{\pi}^p$ in the real MDP,

		Model-free methods				Model-based methods					
		2020 NeurIPS CQL	2021 NeurIPS TD3+BC	2022 ICLR IQL	2022 ICLR PBRL	2020 NeurIPS MOPO	2020 NeurIPS MOReL	2021 NeurIPS COMBO	2022 NeurIPS RAMBO	2022 NeurIPS CABI	(Ours) ORPO
Random	HalfCheetah	27.0±0.6	11.3±0.5	7.8±0.3	11.0	20.7±1.8	25.6	38.8	40.0	15.1	**40.8**±1.6
	Hopper	16.2±2.5	12.7±3.9	8.5±0.0	26.8	31.7±0.3	**53.6**	17.8	21.6	11.9	9.2±1.4
	Walker2d	1.2±0.5	2.1±1.2	5.6±0.0	8.1	1.7±0.5	**37.3**	7.0	11.5	6.4	10.8±9.3
Medium	HalfCheetah	52.6±0.3	48.4±0.3	47.7±0.3	57.9	71.1±2.6	42.1	54.2	**77.6**	45.1	**73.4**±0.5
	Hopper	78.9±6.4	56.4±4.9	54.3±4.3	75.3	20.7±12.9	95.4	94.9	92.8	**100.4**	30.4±37.4
	Walker2d	82.2±2.6	80.8±2.9	76.1±5.1	**89.6**	16.8±15.0	17.8	77.8	86.9	82.0	55.5±23.4
Medium Replay	HalfCheetah	49.5±0.5	44.2±0.5	44.5±0.5	45.1	62.5±10.4	40.2	55.1	68.9	44.4	**72.8**±0.9
	Hopper	99.2±1.6	56.3±20.8	78.1±5.3	88.8	100.8±4.9	93.6	73.1	96.6	31.3	**104.6**±1.5
	Walker2d	80.7±10.7	75.7±7.6	68.6±9.9	77.7	80.0±8.9	49.8	56.0	85.0	29.4	**91.1**±2.0
Medium Expert	HalfCheetah	64.2±11.5	86.0±6.7	81.2±6.0	92.3	80.8±11.4	53.3	90.0	93.7	**105.0**	**101.5**±3.1
	Hopper	68.2±25.1	100.0±9.8	5.1±1.6	110.8	21.1±20.0	108.7	111.1	83.3	**112.7**	**111.0**±0.6
	Walker2d	109.6±0.3	**110.3**±0.5	107.8±4.0	110.1	102.1±8.5	95.6	96.1	68.3	108.4	**108.8**±3.2

Table 1: Average normalized score and the standard deviation with the 'v2' dataset of D4RL. The highest-performing and competitive scores of our method are highlighted. We run CQL, TD3+BC, IQL, MOPO, and ORPO over 5 different seeds and take the average scores. The scores of PBRL, MOReL, COMBO, and CABI are taken from their papers.

denoted as $\eta_M(\hat{\pi}^p)$, has a lower bound. However, how to train optimal policies in the P-MDP has not been thoroughly investigated. To bridge this gap, we analyze the optimality of ORPO under the linear-MDPs assumption, which is widely adopted by previous theoretical works (Melo and Ribeiro 2007; Jin et al. 2020; Jin, Yang, and Wang 2021).

We initially learn a dynamics model and subsequently employ this model to conduct online RL for generating optimistic rollouts. Based on this point, ORPO aligns closely with the theoretical investigations in online RL, which explore the environment through Upper Confidence Bound (UCB) (Audibert, Munos, and Szepesvári 2009). From the theoretical perspective, appropriate uncertainty quantification is essential to the provable efficiency in our framework. We utilize the standard deviation of the dynamics model ensembles for uncertainty quantification, i.e., $u(s,a) := \mathrm{Std}\big(\{\widehat{T}_i(s,a)\}_{i=1}^N\big)$. Then we can make the following proposition:

Proposition 1. Under the assumption of linear MDPs, the uncertainty of dynamics models can form a UCB bonus.

We train an optimistic rollout policy for generating model rollouts in the O-MDP. Since the P-MDP and O-MDP share the same transition distribution, from the view of P-MDP, the reward bonus for training optimistic rollout policy is $(\lambda^p + \lambda^o)u(s,a)$, which can be a UCB bonus for an appropriately selected tuning λ^o and λ^p. Then we use the samples (model rollouts) to optimize the output policy π^p in the P-MDP.

Theorem 1. Under linear model MDPs and the same assumptions to MOPO, with at least constant probability, the output policy of ORPO π^p can be ϵ-optimal in the P-MDP, and satisfies

$$\eta_M(\pi^p) \geq \sup_{\pi}\{\eta_M(\pi) - 2\lambda\epsilon_u(\pi) - \epsilon\}. \tag{7}$$

Theorem 1 shows that the performance of ORPO in the real MDP can be guaranteed. Note that we omit the sample complexity because within our framework, samples to optimize the policy can be generated by the learned dynamics model instead of the real environment, which is much cheaper and easier. We refer to Appendix A for details.

5 Experiments

In our experiment, we aim to investigate three primary research questions (RQs):

RQ1 (Performance): How does ORPO perform on standard offline RL benchmarks and tasks requiring generalization compared to state-of-the-art baselines?

RQ2 (Effectiveness of optimistic rollout policy): How does the proposed optimistic rollout policy compare to various other rollout policies?

RQ3 (Ablation study): How does each design in ORPO affect performance?

To answer the above questions, we conducted our experiments on the D4RL benchmark suite (Fu et al. 2020) as well as two datasets that require generalization to related but previously unseen tasks using the MuJoCo simulator (Todorov, Erez, and Tassa 2012). For the practical implementation of the ORPO algorithm, we utilized the SAC (Haarnoja et al. 2018) to train the optimistic rollout policy, and for pessimistic offline policy optimization, we used TD3+BC (Fujimoto and Gu 2021). Most of the hyper-parameters were inherited from the optimized MOPO (Lu et al. 2022).

5.1 Performance (RQ1)

To answer RQ1, we compared ORPO with several state-of-the-art algorithms, including: 1) CQL (Kumar et al. 2020): A conservative Q-learning algorithm that minimizes Q-values of OOD actions. 2) TD3+BC (Fujimoto and Gu 2021): A model-free algorithm that incorporates an adaptive behavior cloning (BC) constraint to regularize the policy. 3)

Environments	CQL	TD3+BC	MOPO	COMBO	ORPO
Halfcheetah-jump	1287.8±40.4	-4733.3±746.7	4411.8±642.9	4595.2±405.6	**5218.0**±128.5
Halfcheetah-jump-hard	-2989.8±2.0	-2484.4±383.3	-1881.8±1342.2	2782.8±206.7	**4867.9**±381.6

Table 2: Average returns over 5 random seeds on tasks that require OOD policy.

IQL (Kostrikov, Nair, and Levine 2022), an implicit conservative Q-learning algorithm to avoid using Q-values of OOD actions. 4) PBRL (Bai et al. 2022): An uncertainty-based algorithm that uses OOD sampling. 5) MOPO (Yu et al. 2020): A model-based algorithm that penalizes rewards based on uncertainty. 6) COMBO (Yu et al. 2021): A model-based variant of CQL. 7) RAMBO (Rigter et al. 2022): A model-based algorithm using robust adversarial RL. 8) CABI (Lyu, Li, and Lu 2022): An algorithm that utilizes forward and backward CVAE rollout policies to generate trustworthy rollouts.

Results on D4RL benchmarks: We summarized the average normalized scores in Table 1, which includes three environments (HalfCheetah, Hopper, and Walker2d), each with four datasets. Our ORPO achieved competitive or better results compared to state-of-the-art methods in 8 out of 12 datasets. Overall, ORPO demonstrated significant advantages, particularly when the offline datasets were more diverse, such as in the "random" and "medium-replay" types. This can be attributed to the improved generalization abilities of the dynamics models trained on such datasets.

We observed that implementing ORPO based on MOPO resulted in a significant performance boost in 11 out of the 12 datasets, increasing the total average normalized score from 610.0 to 809.9, with an improvement of more than 30%. The only exception is the "hopper-random-v2" datasets. This may be because the "halfcheetah" and "walker2d" tasks are more resilient to OOD actions, while the hopper tasks are more prone to terminating the episode when encountering OOD actions, [1] making most OOD model rollouts useless.

Results on tasks requiring generalization: To further demonstrate the generalization ability of the output policy, we evaluated on "Halfcheetah-jump" dataset proposed by Yu et al. (Yu et al. 2020). This dataset was collected by storing the entire training replay buffer from training SAC for 1 million steps in the HalfCheetah task. The state-action pairs in the dataset were then assigned new rewards that incentivized the halfcheetah to jump. Based on the "Halfcheetah-jump" dataset, we constructed a more challenging dataset, "Halfcheetah-jump-hard". This dataset consists of trajectories sampled by a random policy, and the assigned new rewards are further penalized if the halfcheetah is unhealthy.

We observe that model-based methods show great advantages over model-free. Notably, ORPO outperforms all the baseline methods by a large margin, highlighting its effectiveness in terms of generalization ability. In the "Halfcheetah-jump-hard" dataset, due to the additional unhealthy penalization on rewards, the policy trained by MOPO is too conservative to run. ORPO is the only method that achieves satisfactory performance, which suggests that our method can not only generalize to OOD regions but also preclude some of them with low values.

[1] https://www.gymlibrary.dev/environments/mujoco/

5.2 Effectiveness of optimistic rollout policy

To answer RQ2, we compare rollout policies in our framework with various rollout policies including: 1) Random rollout policy, which generates actions from the uniform distribution in the action space. 2) Conditional variational autoencoder (CVAE) rollout policy (Lyu, Li, and Lu 2022), which offers diverse actions while staying within the span of the dataset. 3) Trained optimistic rollout policy, which uses well-trained fixed optimistic rollout policy to generate optimistic rollouts for pessimistic policy optimization.

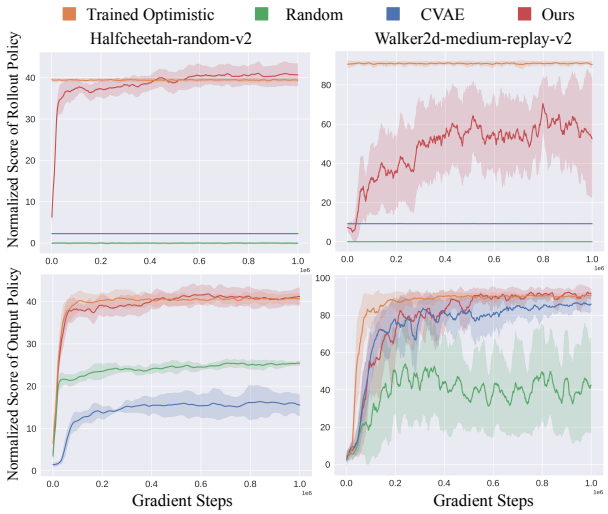


Figure 4: Learning curves of rollout policies and corresponding output policies in two datasets over 5 different seeds.

In Figure 4, we report the normalized average scores of the rollout policies and output policies in two datasets. Except for ours, all baseline rollout policies have fixed parameters. So the scores of them are constant. The score of CVAE rollout policies is only slightly higher than that of the random policies. This is because CVAE policies are trained to generate rollouts within the support of the offline dataset, while these two datasets conclude many low-value transitions. In contrast, our optimistic rollout policy can achieve the highest scores due to more valuable model rollouts.

Considering our goal is to achieve high scores for the output policies, the CVAE rollout policy is more effective than the random rollout policy in "Walker2d-medium-replay-v2" dataset and vice verse in "Halfcheetah-random-v2". This is because the CVAE rollout policies trained by "medium-replay" datasets can generate more valuable rollouts with

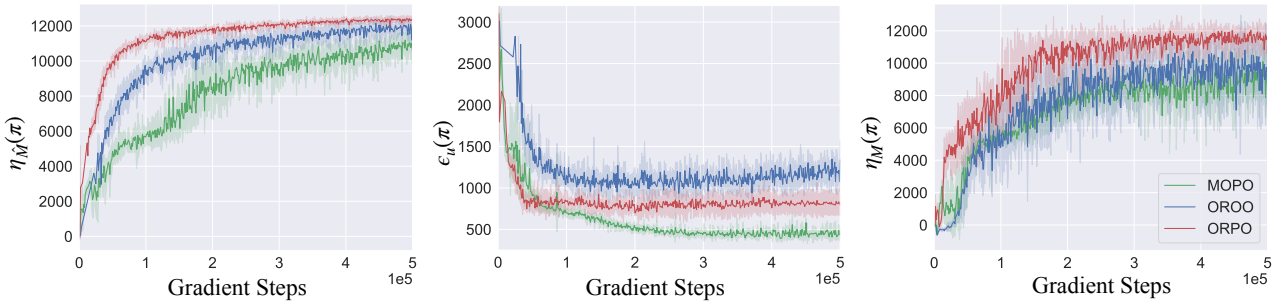


Figure 5: Learning curves of OROO, MOPO, and ORPO over 5 different seeds on "Halfcheetah-medium-expert-v2". We report the expected discounted returns in the model MDP $\eta_{\widehat{M}}(\pi)$ and the real MDP $\eta_M(\pi)$ as well as the average model error $\epsilon_u(\pi)$.

high-value state-action pairs, but not in "random" datasets. The output policies of ORPO can obtain the highest scores in both datasets. Though our rollout policy on the "Walker2d-medium-replay-v2" dataset can not achieve satisfactory performance, it is still beneficial for optimizing the output policy. While using fixed well-trained rollouts policy can match our performance, we notice it can be affected by the selected checkpoint. Therefore, we train optimistic rollout policy and pessimistic output policy iteratively and alternately.

5.3 Ablation study

Effect of P-MDP and O-MDP: We conducted an analysis comparing the expected discounted returns ($\eta_M(\pi)$ and $\eta_{\widehat{M}}(\pi)$) and the average model uncertainty ($\epsilon_u(\pi)$) of our output policy with the O-MDP and P-MDP baselines, i.e., OROO and MOPO methods. As shown in Figure 5, we observed that with more OOD sampling, OROO achieves a higher $\eta_{\widehat{M}}(\pi)$ compared to MOPO. However, when evaluating OROO in the real environment, we observed a performance significant deterioration due to the noticeable increase in $\epsilon_u(\pi)$, indicating larger model errors. As a result, MOPO remains comparable to OROO in terms of $\eta_M(\pi)$ on this dataset.

ORPO effectively prevents the agent from accessing risky or potentially dangerous areas. This explains why ORPO achieves higher $\eta_{\widehat{M}}(\pi)$ and lower $\epsilon_u(\pi)$ than OROO. Consequently, our method achieves better performance in the real environment ($\eta_M(\pi)$) compared to the baselines that use either P-MDP or O-MDP. Thus, we conclude that ORPO achieves a better trade-off between the generalization ability and estimation errors of the learned dynamics model.

Sensitivity of the hyper-parameter λ^o: We also conduct experiments to evaluate the sensitivity of ORPO to the hyper-parameter λ^o, which is used to construct the O-MDPs. As shown in Table 3, our results indicate that ORPO achieves satisfactory performance across a wide range of λ^o values spanning three orders of magnitude. The values of λ^p for the two configurations were 4.56 and 2.48, respectively. Since we optimize the output policies in the P-MDP, the rollout policy is optimistic as long as $\lambda^o > -\lambda^p$. Results show that incorporating optimism can bring significant performance gains. Due to the robustness of ORPO to the choice of λ^o, there is no need to finely tune this parameter for each environment-dataset configuration, and we set $\lambda^o = 0.015$ by default for 9 out of the 12 datasets used in the D4RL benchmarks.

	-10	-1	-0.1	-0.01	0.01	0.1	1	10
H-R	27.7	39.1	40.6	38.5	40.2	41.2	41.6	26.0
W-M-R	43.7	85.5	91.1	91.3	91.8	88.6	8.6	0.1

Table 3: Ablation of the different optimism hyper-parameter λ^o for ORPO. "H-R" represents HalfCheetah-Random-v2 and "W-M-R" represents Walker2d-Medium-Replay-v2.

Other ablation studies: We briefly report the results compared to other baselines. 1) We compare to MOPO (TD3+BC) which replaces SAC in MOPO with TD3+BC for policy optimization. The results migrate the effect of different RL algorithms on performance gain over MOPO and suggest the effectiveness of optimistic rollouts. 2) We compare ORPO to ORPO (SAC), which use SAC to optimize both rollout policies and output policies, and demonstrate the effectiveness of utilizing offline RL algorithms for pessimistic policy optimization. 3) We compare to ORPO without pessimism which replaces the P-MDP used in ORPO with the model MDP, and demonstrate the necessity of pessimism in ORPO. Complete results can be found in Appendix D.2.

6 Conclusion and Limitations

In this paper, we started with the observation that incorporating optimism when generating model rollouts can yield benefits for model-based offline RL. Building upon this insight, we have introduced ORPO, a novel framework that leverages optimistic model rollouts for pessimistic policy optimization. The theoretical analysis of ORPO demonstrates its efficiency in addressing the challenges of offline RL. Through extensive empirical evaluations, we have demonstrated that ORPO significantly enhances the performance of the P-MDP baseline and surpasses state-of-the-art methods on both the D4RL benchmark and tasks demanding generalization.

Our work has limitations. One limitation is the additional time overhead required for training optimistic rollout policies. Additionally, in tasks where OOD actions are strictly prohibited, ORPO may have negative effects compared to existing P-MDP baselines. Therefore, one future direction is to explore adaptive degree of optimism when evaluation.

7 Acknowledgements

This work was supported by the Major Science and Technology Innovation 2030 "New Generation Artificial Intelligence" project 2021ZD0112904.

References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24.

An, G.; Moon, S.; Kim, J.-H.; and Song, H. O. 2021. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34: 7436–7447.

Audibert, J.-Y.; Munos, R.; and Szepesvári, C. 2009. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19): 1876–1902.

Bai, C.; Wang, L.; Yang, Z.; Deng, Z.; Garg, A.; Liu, P.; and Wang, Z. 2022. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. *International Conference on Learning Representations*.

Bradtke, S. J.; and Barto, A. G. 1996. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1-3): 33–57.

Clavera, I.; Fu, Y.; and Abbeel, P. 2020. Model-Augmented Actor-Critic: Backpropagating through Paths. In *International Conference on Learning Representations*.

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*.

Fujimoto, S.; and Gu, S. S. 2021. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34: 20132–20145.

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, 1587–1596. PMLR.

Fujimoto, S.; Meger, D.; and Precup, D. 2019. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, 2052–2062. PMLR.

Gottesman, O.; Johansson, F.; Komorowski, M.; Faisal, A.; Sontag, D.; Doshi-Velez, F.; and Celi, L. A. 2019. Guidelines for reinforcement learning in healthcare. *Nature medicine*, 25(1): 16–18.

Guo, K.; Yunfeng, S.; and Geng, Y. 2022. Model-based offline reinforcement learning with pessimism-modulated dynamics belief. *Advances in Neural Information Processing Systems*, 35: 449–461.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, 1861–1870. PMLR.

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

Jin, C.; Allen-Zhu, Z.; Bubeck, S.; and Jordan, M. I. 2018. Is Q-learning provably efficient? *NeurIPS*, 31.

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. PMLR.

Jin, Y.; Yang, Z.; and Wang, Z. 2021. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, 5084–5096. PMLR.

Kidambi, R.; Rajeswaran, A.; Netrapalli, P.; and Joachims, T. 2020. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33: 21810–21823.

Kostrikov, I.; Nair, A.; and Levine, S. 2022. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*.

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.

Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191.

Lee, S.; Seo, Y.; Lee, K.; Abbeel, P.; and Shin, J. 2022. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Conference on Robot Learning*, 1702–1712. PMLR.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Lockwood, O.; and Si, M. 2022. A Review of Uncertainty for Deep Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 18, 155–162.

Lu, C.; Ball, P.; Parker-Holder, J.; Osborne, M.; and Roberts, S. J. 2022. Revisiting design choices in offline model based reinforcement learning. In *International Conference on Learning Representations*.

Luo, Y.; Xu, H.; Li, Y.; Tian, Y.; Darrell, T.; and Ma, T. 2019. Algorithmic Framework for Model-based Deep Reinforcement Learning with Theoretical Guarantees. In *International Conference on Learning Representations*.

Lyu, J.; Li, X.; and Lu, Z. 2022. Double Check Your State Before Trusting It: Confidence-Aware Bidirectional Offline Model-Based Imagination. In *Advances in Neural Information Processing Systems*.

Mandlekar, A.; Ramos, F.; Boots, B.; Savarese, S.; Fei-Fei, L.; Garg, A.; and Fox, D. 2020. Iris: Implicit reinforcement without interaction at scale for learning control from offline robot manipulation data. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 4414–4420. IEEE.

Melo, F. S.; and Ribeiro, M. I. 2007. Q-learning with linear function approximation. In *Learning Theory: 20th Annual Conference on Learning Theory, COLT 2007, San Diego, CA, USA; June 13-15, 2007. Proceedings 20*, 308–322. Springer.

Moerland, T. M.; Broekens, J.; Plaat, A.; Jonker, C. M.; et al. 2023. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1): 1–118.

Pan, F.; He, J.; Tu, D.; and He, Q. 2020. Trust the model when it is confident: Masked model-based actor-critic. *Advances in neural information processing systems*, 33: 10537–10546.

Rafailov, R.; Yu, T.; Rajeswaran, A.; and Finn, C. 2021. Offline reinforcement learning from images with latent space models. In *Learning for Dynamics and Control*, 1154–1168. PMLR.

Rigter, M.; Lacerda, B.; ; and Hawes, N. 2022. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *Advances in neural information processing systems*, 35: 16082–16097.

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

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

Wang, J.; Li, W.; Jiang, H.; Zhu, G.; Li, S.; and Zhang, C. 2021. Offline reinforcement learning with reverse model-based imagination. *Advances in Neural Information Processing Systems*, 34: 29420–29432.

Wu, Y.; Zhai, S.; Srivastava, N.; Susskind, J. M.; Zhang, J.; Salakhutdinov, R.; and Goh, H. 2021. Uncertainty Weighted Actor-Critic for Offline Reinforcement Learning. In *International Conference on Machine Learning*, 11319–11328. PMLR.

Yu, F.; Xian, W.; Chen, Y.; Liu, F.; Liao, M.; Madhavan, V.; and Darrell, T. 2018. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5): 6.

Yu, T.; Kumar, A.; Rafailov, R.; Rajeswaran, A.; Levine, S.; and Finn, C. 2021. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34: 28954–28967.

Yu, T.; Thomas, G.; Yu, L.; Ermon, S.; Zou, J. Y.; Levine, S.; Finn, C.; and Ma, T. 2020. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33: 14129–14142.