

Diversification of Adaptive Policy for Effective Offline Reinforcement Learning

Yunseon Choi¹, Li Zhao², Chuheng Zhang², Lei Song², Jiang Bian² and Kee-Eung Kim¹

¹KAIST AI

²Microsoft Research Asia

{cys9506, kekim}@kaist.ac.kr, {lizo, chuhengzhang, lei.song, jiang.bian}@microsoft.com

Abstract

Offline Reinforcement Learning (RL) aims to learn policies from pre-collected datasets that capture only a subset of the environment’s dynamics. The predominant approach has been to solve a constrained optimization formulation, which ensures that the policy visits state-action pairs within the support of the offline dataset. However, this approach has limited the ability to make decisions when the agent faces unknown parts of the environment at deployment time. To address the challenge of decision-making in out-of-support regions, model-based Bayes-adaptive approaches have been proposed by considering all dynamics models that could potentially be the true environment. Since it is generally infeasible to compute the posterior of all dynamics models based on the offline dataset, these approaches usually approximate the posterior by using a finite ensemble of highly probable dynamics models. Hence, the diversity of these models is the key to obtaining good policies. In this work, we propose MoDAP (Model-based Diverse Adaptive Policy Learning), an algorithm to enable the adaptive policy to make informed decisions in previously unexplored states. MoDAP adopts an iterative strategy that simultaneously training the policy and dynamics models. The policy optimization seeks to maximize expected returns across dynamics models, while the dynamics models are trained to promote policy diversification through the proposed information-theoretic objective. We evaluate MoDAP through experiments on the D4RL and NeoRL benchmarks, showcasing its performance superiority over state-of-the-art algorithms.

1 Introduction

Reinforcement Learning (RL) has emerged as a powerful approach for decision-making in complex real-world scenarios, as evidenced by its successes in domains such as the games of chess, shogi, and go [Silver *et al.*, 2017], and real-time strategy video games [OpenAI *et al.*, 2019]. However, traditional

online RL methods involve exploratory interaction with the environment, which makes them impractical, costly, and potentially unsafe in various real-world applications.

Thereby, offline RL has recently garnered significant attention [Fujimoto *et al.*, 2019; Levine *et al.*, 2020]. Offline RL involves training a policy solely on a fixed dataset obtained from the environment, without any additional interaction with the environment. However, directly applying existing online RL algorithms [Mnih *et al.*, 2013; Haarnoja *et al.*, 2018] to the offline setting often results in poor performance due to the distribution shift between the learned policy and the data-collected policy. This distribution shift introduces extrapolation errors in the value function estimation for unseen actions. As a result, several model-free offline RL algorithms [Fujimoto *et al.*, 2019; Kumar *et al.*, 2019; Nair *et al.*, 2021] have focused on solving the constrained policy optimization problem, where the learned policy is encouraged to select actions from the dataset.

Alternately, there have been offline RL methods that adopt model-based RL approaches, which demonstrate superior generalizability by leveraging the learned dynamics model, in the offline setting. However, these methods still suffer from model overfitting issues and extrapolation errors in regions that are not explored by the data-collected policies [Kidambi *et al.*, 2020; Yu *et al.*, 2021]. This is because the dynamics model is learned solely from the offline dataset, which may not capture the full dynamics of the underlying environment. To address these challenges, previous works have proposed pessimistic framework [Yu *et al.*, 2020; Rigter *et al.*, 2022]. For example, model-based offline policy optimization (MOPO; [Yu *et al.*, 2020]) incorporates penalty terms to adjust the reward signals based on the degree of uncertainty inherent in the model.

Yet, these pessimistic approaches in model-based offline RL fall short in handling scenarios where the agent encounters out-of-support regions. Here, the Bayes-adaptive decision framework presents a valuable alternative method [Ross *et al.*, 2007; Ghavamzadeh *et al.*, 2015]. One of the primary benefits of the Bayesian method lies in its capacity to reason over all potential models by calculating their posterior distribution from the given dataset. A policy trained on this posterior distribution can naturally adapt during online execution, which may lead to enhanced performance in comparison to a policy trained by the pessimistic approach. AVE-P [Ghosh *et al.*

et al., 2022] capitalizes on this advantage by training the adaptive policy through belief state tracking, using an ensemble of value functions to interact indirectly with the models. In a similar vein, MAPLE [Chen *et al.*, 2021] employs a related approach by explicitly constructing an ensemble of dynamics models. This ensemble enables the generation of simulations for trajectories, which are subsequently utilized to augment the training of the adaptive policy.

These prior works used to construct the set of dynamics models with ensemble structures, facilitating the adaptive policy learning in practical applications. However, when the dynamics models are similar, they might not generate sufficiently diverse behaviors within the out-of-support region.

The lack of diversity in behaviors can impose limitations on the algorithm’s capacity for effective generalization, leading to sub-optimality in real-world scenarios. In practice, these methods often necessitate a considerable number of ensemble members to achieve the desirable level of generalization ability [Ghosh *et al.*, 2022; Ghosh *et al.*, 2021]. Consequently, the establishment of a more diverse set of models holds significant importance.

In this work, our goal is to enrich the diversity of the adaptive policy, all the while utilizing a limited number of models within the model-based framework. To do that, we first discuss which dynamics models are considered to encourage the diversification of the adaptive policy within an offline setting. These dynamics models should adequately represent the support region of the offline dataset, and also manifest distinct transitions in out-of-support regions. We particularly focus on the optimal trajectories generated by the optimal policies associated with different dynamics models. When certain optimal trajectories demonstrate similarities, it suggests that the corresponding MDPs do not lead to significantly different decision-making scenarios. Leveraging this insight, we introduce an information-theoretic objective that seeks to maximize the mutual information between the optimal trajectory and the identity of the underlying dynamics models.

We implement this idea and propose a Model-based Diverse Adaptive Policy Learning (MoDAP) algorithm for offline RL. MoDAP is an iterative training approach involving both the policy and dynamics models. We train the policy to maximize its expected return across dynamics models while concurrently training the dynamics models to facilitate policy diversification through the information-theoretic objective. We provide further details about MoDAP in Section 4. Lastly, we evaluate MoDAP across a range of D4RL [Fu *et al.*, 2020] and NeoRL [Qin *et al.*, 2022] datasets with a limited number of dynamics models, demonstrating its superior and competitive performance. Further details regarding these experiments are provided in Section 5.

2 Related Work

2.1 Offline RL

In model-free offline RL, the distribution shift between the learned policy and the data-collected policy introduces extrapolation errors [Kumar *et al.*, 2019] in the estimation of the value function for unseen actions in the dataset, presenting a challenge for naively adopting online algorithms. To

overcome this challenge, extensive research has been conducted to formulate offline RL as a constrained optimization problem [Kumar *et al.*, 2019; Fujimoto *et al.*, 2019; Nair *et al.*, 2021; Wang *et al.*, 2020; Kostrikov *et al.*, 2021]. The common approaches [Fujimoto and Gu, 2021; Fujimoto *et al.*, 2019; Kumar *et al.*, 2019] ensure that the optimized policy remains close to the behavior policy. Other strategies involve estimating conservative Q-values [Kumar *et al.*, 2020a] by incorporating regularization terms that assign lower values to unseen state-action pairs compared to observed ones. EDAC [An *et al.*, 2021], another notable technique, enforces the expected minimum action value through the use of diversified Q-ensemble and clipped Q-learning.

Moving to model-based approaches, these algorithms utilize the learned dynamics model derived from the dataset to simulate the policy. A significant challenge arises when the policy takes actions that were absent from the training dataset, leading to unseen action-state pairs. In such cases, the estimated model may provide transition dynamics that differ from the true environment, causing potential unreliability in the model’s predictions. To mitigate this issue, a prevalent strategy in model-based offline approaches introduces constructing a pessimistic MDP with a penalized reward function. For instance, MOPO [Yu *et al.*, 2020] introduces model ensemble uncertainty into reward signal, while MOREl [Kidambi *et al.*, 2020] utilizes the state-action detector to make penalized reward. Since these methods primarily concentrate on decision-making within the known portion of the MDP, they may have inherent limitations when the behavior policy is sub-optimal, as the constrained formulations do not guarantee optimal performance [Chen *et al.*, 2021; Ghosh *et al.*, 2022].

2.2 Bayesian RL and Offline Adaptive Policy

Bayesian RL has proven to be effective in striking a balance between exploration and exploitation in online RL, resulting in sample efficiency and optimality [Ross *et al.*, 2007; Asmuth *et al.*, 2009; Kolter and Ng, 2009; Poupart *et al.*, 2006]. To apply Bayesian principles to model-based offline RL, we assume the set of dynamics models that involve the true environment. The agent takes the action based on the posterior distribution of the models, or a sufficient statistic of interaction history, at deployment time, which naturally makes the policy adaptive. An illustrative example of this approach is MAPLE [Chen *et al.*, 2021], which employs a Recurrent Neural Network (RNN) policy to infer environmental contexts from historical information. Alternatively, in the model-free approach, APE-V [Ghosh *et al.*, 2022] takes an indirect strategy by engaging with the value function of each model, updating its belief state based on historical information.

2.3 Diversity and RL

There exists various range of research that leverages the concept of diversity in RL, often optimizing information-theoretic objectives. In the domain of unsupervised skill discovery, agents have to get valuable skills without explicit reward signals. A prominent example is DIAYN [Eysenbach *et al.*, 2019], which attains diverse skills by maximizing the

mutual information between states and skills. Additionally, DADS [Liu *et al.*, 2021] focuses on skill diversification by maximizing mutual information between the subsequent state and the skill, conditioned on the current state.

On the other hand, SMEARL [Kumar *et al.*, 2020b] aims to enhance adaptability to unseen tasks by generating a diverse set of sub-optimal policies within the specific MDP. This is achieved through the utilization of an augmented reward derived from a diversification objective. Even though our work also utilizes the information-theoretic objective, its purpose and detailed algorithm are clearly different as we concentrate on training a set of dynamics models to facilitate the diversification of adaptive policy.

3 Preliminaries

3.1 MDPs and Offline RL

We consider a Markov Decision Process (MDP) as a tuple $m = (\mathcal{S}, \mathcal{A}, P_m, r, \rho_0, \gamma)$, where \mathcal{S} is the state space, \mathcal{A} is the action space, $P_m : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ is the state transition probability function, and $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, $\rho_0 \in \Delta(\mathcal{S})$ is the initial state distribution, and $\gamma \in (0, 1]$ is the discount factor. The objective of RL is to obtain a policy, $\pi_m : \mathcal{S} \rightarrow \Delta(\mathcal{A})$, that maximizes the discounted cumulative reward:

$$J(\pi_m) = \mathbb{E}_{\substack{s_0 \sim \rho_0(\cdot), a_t \sim \pi_m(\cdot|s_t) \\ s_{t+1} \sim P_m(\cdot|s_t, a_t)}} \left[\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t) \right] \quad (1)$$

The solution to the optimization problem defined by Eq. (1) is denoted as π_m^* .

While the standard RL assumes online interaction with the environment during training, offline RL does not allow such interaction. Instead, we can only access a pre-collected dataset that is gathered by (unknown) mixed behavior policies. Since the pre-collected dataset usually covers only a subset of the states and actions, the goal of offline RL is to find the best policy given the limited information available in the dataset.

3.2 Model-based Offline RL Algorithm

Model-based Offline RL algorithms aim to train a policy by leveraging a learned dynamics model $P_\phi(s'|s, a)$. This model is typically trained using maximum likelihood estimation on the offline dataset, $\max_{\phi} \mathbb{E}_{(s, a, s') \sim \mathcal{D}_{\text{off}}} [\log P_\phi(s'|s, a)]$.¹ Once the model is learned, it is utilized to simulate rollouts of the policy, and the resulting rollout data is then used to optimize the policy.

Instead of focusing on building a single MDP, we want to work with a set of MDPs, each element representing an MDP with transition dynamics consistent with the dataset but different underlying transition dynamics in other regions of the state-action space. These MDPs comprise an ensemble of hypotheses about the true environment, and act as simulation environments to train the adaptive policy. The policy learns to switch from one MDP to another based on the iteration,

¹Since the transition and reward function are unknown, they are usually trained together, $\max_{\phi} \mathbb{E}_{(s, a, r, s') \sim \mathcal{D}_{\text{off}}} [\log P_\phi(r, s'|s, a)]$.

making it adaptive and hence generalize better, compared to learning from a single MDP.

One thing we need to notice is that the generalization capacity of the adaptive policy is significantly influenced by the characteristic of the MDPs within the ensemble. For instance, the performance of the approach is particularly sensitive to the number of ensemble models used. When working with a limited number of models, the policy’s performance tends to be poor, conversely, as the ensemble size increases, the policy’s effectiveness improves significantly. However, having a larger number of ensemble models for learning well-generalized policies might not always be feasible due to the practical constraint of computational resources. MAPLE [Chen *et al.*, 2021], despite its objective of improving decision-making in out-of-support regions through enhanced generalizability, adopts a pessimistic approach to work with a limited number of models. It involves uncertainty over the MDPs as a penalty term in the reward signal. Thus, the challenge lies in the construction of a set of MDPs that remains manageable in size while effectively enhancing policy learning and generalization.

In this work, we propose a novel method to construct an effective yet small number of MDPs in the set for training the adaptive policy and the value function. Similar to established methodologies, our approach employs model-based policy optimization (MBPO) [Jiang *et al.*, 2020], utilizing a standard actor-critic RL algorithm. To generate synthetic data for training, MBPO conducts k -step rollouts originating from states $s \in \mathcal{D}_{\text{off}}$, subsequently adding this data to the \mathcal{D}_{P_ϕ} . During policy training, minibatches of data are sampled from the combined dataset $\mathcal{D}_{\text{off}} \cup \mathcal{D}_{P_\phi}$. Each data point within this batch is drawn from either of the real data \mathcal{D} with a probability f or the synthetic data \mathcal{D}_{P_ϕ} with a probability $1 - f$. Although our methodology shares similarities with MBPO regarding the generation and usage of synthetic datasets for training, it distinguishes itself in the utilization of history for value function and policy, as described in the next section.

4 Method

Before introducing our method, we begin with discussing the dynamics models that are essential for effective learning of the adaptive policy in offline RL.

Which models do we need? To facilitate a clear understanding of our method, we consider a scenario where the MDPs share the same state space, action space, and reward function, but differ in their transition functions P_m with the subscript/identity $m \in [n]$ being the index of MDPs.² Instead of directly focusing on different transition dynamics $\{P_m\}_{m=1}^n$, we narrow our focus to the optimal trajectories, denoted as $\{\tau^{\pi_m^*}\}_{m=1}^n$, which are generated by executing the corresponding optimal policies $\{\pi_m^*\}_{m=1}^n$ for each MDP. When certain optimal trajectories exhibit similarities, it indicates that the corresponding MDPs do not yield distinct

²This assumption does not compromise the generality of our method, as it can be applied when the reward is MDP-dependent while the state space, action space, and transition function are shared.

decision-making scenarios. The presence of such dynamics models within the set is inefficient and restricts the diversity for effective policy adaptation and generalization. To address this, we emphasize the importance of constructing a set of MDPs in which each MDP exhibits significant differentiation from the others.

4.1 Promoting Diversity in Dynamics Models

To achieve this goal, we employ an information-theoretic approach that aims to maximize the Mutual Information (MI)

$$\begin{aligned} \mathcal{I}(\tau^{\pi^*M}; M) &= \mathcal{H}(\tau^{\pi^*M}) - \mathcal{H}(\tau^{\pi^*M} | M) \\ &= \mathbb{E} \left[\log \frac{p(\tau^{\pi^*M} | M)}{p(\tau^{\pi^*M})} \right], \end{aligned} \quad (2)$$

where M and τ^{π^*M} represent the random variables associated with the MDP's identity and its corresponding optimal trajectory, respectively. The equation quantifies the information gain concerning the optimal trajectory when the MDP is known, measuring trajectory-aware diversity.

To optimize Eq. (2), we decompose $p(\tau^{\pi^*M} | m)$ as follows:

$$p(\tau^{\pi^*M} | m) = p(s_0) \prod_{t=0}^{T-1} \pi_M^*(a_t | s_t) P_m(s_{t+1} | s_t, a_t) \quad (3)$$

where π_M^* is the optimal policy in the MDP M . Since M is a discrete random variable, we can compute the marginal $p(\tau^{\pi^*M})$ as $\sum_m p(m) p(\tau^{\pi^*M} | m)$ where $p(m)$ is prior.

From Eq. (2), maximizing this objective equates to increasing the entropy of optimal trajectories ($\mathcal{H}(\tau^{\pi^*M})$) while making M informative about the corresponding optimal trajectory ($-\mathcal{H}(\tau^{\pi^*M} | M)$). It promotes the diversity of policy behaviors among MDPs. We provide the detailed theoretical interpretation of Eq. (2) in Appendix A.

Adaptive policy. Instead of learning independent optimal policies $\{\pi_m^*\}$ of each MDPs, our approach aims to train a single, adaptive policy π that covers all MDPs. Specifically, it maps history $h_t = \{s_{0:t-1}, a_{1:t-1}, r_{1:t-1}\}$, a sequence of observed transitions up to timestep $t-1$, to action a_t . We approximate the optimal policy π_m^* with this history-based policy, π . It notes that as π interacts with the environment at test time, it can identify the underlying transition dynamics by tracking the history. Over time, it becomes gradually the optimal policy. As a consequence of this approach, we maximize $\mathcal{I}(\tau^\pi; M)$ instead of Eq. (2), where τ^π is the trajectory obtained from the adaptive policy π over MDPs.

Overall objective. We aim to determine the set of estimated dynamics models $\{P_{\phi_m}\}_{m=1}^n$ given a fixed adaptive policy π . To achieve this, we solve the following optimization objective for each model m :

$$\mathcal{L}_{\text{MI}}(\phi_m) = \mathbb{E}_{\tau \sim \pi, P_{\phi_m}} [\log p(\tau) - \log p(\tau | m)] \quad \forall m \quad (4)$$

where $p(\tau | m) = p(s_0) \prod_{t=0}^{T-1} \pi(a_t | s_t, h_t) P_{\phi_m}(s_{t+1} | s_t, a_t)$.

The set of dynamics models $\{P_{\phi_m}\}_{m=1}^n$ should be capable of modeling the transitions observed in the offline dataset, also. It becomes necessary to include MLE term as usual to ensure proper model training:

$$\mathcal{L}_{\text{MLE}}(\phi_m) = \mathbb{E}_{\{s, a, s'\} \sim \mathcal{D}} [-\log P_{\phi_m}(s' | s, a)] \quad \forall m \quad (5)$$

Algorithm 1 MoDAP

Require: the parameters θ, ψ of the policy π_θ and GRU_ψ , the offline dataset \mathcal{D}_{off} and constant, λ

- 1: $\{P_{\phi_i}\}_{i=1}^n, \hat{r} \leftarrow$ the set of dynamics models and reward function from \mathcal{D}_{off}
- 2: **for** $iter = 0, 1, 2, \dots$ **do**
- 3: Randomly sample s, z from \mathcal{D}_{off}
- 4: Set $s_0, z_0 \leftarrow s, z$
- 5: Randomly select dynamics model from $\{P_{\phi_i}\}_{i=1}^n$
- 6: **for** $t = 0, 1, 2, \dots, k-1$ **do**
- 7: Sample $a_t \sim \pi_\theta(a_t | s_t, z_t)$
- 8: Rollout one step
- 9: $s_{t+1} \sim P_{\phi_i}(\cdot | s_t, a_t), r_{t+1} \leftarrow \hat{r}(s_t, a_t)$
- 10: $z_{t+1} \leftarrow \text{GRU}_\psi(s_{t+1}, a_t, z_t)$
- 11: Add $\{s_t, z_t, a_t, r_t, s_{t+1}, z_{t+1}\}$ to \mathcal{D}_{P_ϕ}
- 12: **end for**
- 13: // Policy Update //
- 14: Update $\pi_\theta, \text{GRU}_\psi$ by SAC with $\mathcal{D}_{P_\phi} \cup \mathcal{D}_{\text{off}}$
- 15: // Model Update //
- 16: Estimate \mathcal{L}_{MI} for the generated k -step trajectory
- 17: Update $\{P_{\phi_i}\}_{i=1}^n$ by maximizing $\lambda \mathcal{L}_{\text{MI}} + \mathcal{L}_{\text{MLE}}$
- 18: **end for**

4.2 MoDAP

To train the adaptive policy, we utilize a recurrent neural network (RNN), which is known for its effectiveness in incorporating historical information in meta RL [Duan *et al.*, 2017; Chen *et al.*, 2021]. Specifically, we use GRU [Chung *et al.*, 2014] to encode h_t into the hidden state z_t such that $z_t = \text{GRU}(h_t)$. The policy takes action based on both the current state s_t and the hidden state z_t . Overall, we optimize the adaptive policy π_θ to maximize the expected returns over the MDPs:

$$\mathbb{E}_{m \sim p(\cdot)} [J(\pi_\theta)] \quad (6)$$

and we assume that $p(\cdot)$ is a uniform distribution.

We now present our algorithm, MoDAP (Model-based Diverse Adaptive Policy Learning) in Algorithm 2. Before training the adaptive policy, the dynamics models and reward function are pre-trained using MLE on offline dataset \mathcal{D}_{off} . Notably, \hat{r} is defined as the mean value over the ensemble's output. Due to the finite number of models in the ensemble, planning during a long-horizon results in the generation of unrealistic trajectories in the training dataset for policy, leading to performance degradation. To mitigate this issue, we adopt a fixed-horizon rollout approach for planning. We start planning from an arbitrary time-step t within the offline dataset, collecting trajectories for a fixed horizon of length k , and gathering the dataset required for training the policy.

To provide further detail, our approach involves the following steps during each iteration: In each iteration, we uniformly choose a dynamics model from our ensemble since we assume a uniform prior distribution $p(m)$. We employ the selected dynamics model and the adaptive policy to generate trajectories. These trajectories are added to the synthetic dataset \mathcal{D}_{P_ϕ} . For each step in the trajectory, we obtain the next state s_t using the chosen dynamics model. The reward r is determined using a shared reward function $\hat{r}(s, a)$. We

Task Name	Model-free				Model-based				
	BC	CQL	APE-V	EDAC	MOPO	COMBO	MAPLE	RAMBO	MoDAP
halfcheetah-medium-expert	44.0	95.0	101.4	106.3	90.8	90.0	63.5	95.4	103.4±4.3
halfcheetah-medium-replay	37.6	45.3	64.6	61.3	68.1	55.1	59.0	68.7	67.3±3.4
halfcheetah-medium	43.2	46.9	48.3	69.1	73.0	54.2	50.4	77.9	77.3±1.1
halfcheetah-random	2.2	31.3	29.9	28.4	38.5	38.8	38.4	39.5	36.5±1.8
walker2d-medium-expert	90.1	109.1	110.0	114.7	112.9	103.3	73.8	56.7	112.2±2.8
walker2d-medium-replay	20.3	76.8	82.9	87.1	85.6	56.0	76.7	89.2	88.4±4.2
walker2d-medium	70.9	79.5	90.3	92.5	79.2	81.9	56.3	84.9	81.1±6.5
walker2d-random	1.3	5.4	15.5	16.6	3.0	7.0	21.7	0.0	23.1±1.6
hopper-medium-expert	53.9	96.9	105.7	110.7	81.6	111.1	42.5	88.2	94.5±7.8
hopper-medium-replay	16.6	86.3	98.5	101.0	103.5	89.5	87.5	99.5	94.2±4.8
hopper-medium	54.1	61.9	-	101.6	62.5	97.2	21.1	87.0	106.6±1.9
hopper-random	3.7	5.3	31.3	25.3	3.0	17.9	10.6	25.4	8.9±1.1

Table 1: Results on MuJoCo D4RL benchmark. Normalized scores are calculated as (score - random policy score) / (expert policy score - random policy score), with the standard deviation indicated by \pm . The score of our algorithm is averaged over 4 random seeds.

learn the policy using SAC [Haarnoja *et al.*, 2018] on the combined dataset $\mathcal{D}_{\text{off}} \cup \mathcal{D}_{P_\phi}$. We estimate the objective defined in Eq. (4) by rolling out trajectories with k steps rather than rolling out till the end. In our training process, we apply gradient updates to the dynamics models, aiming to minimize both Eq. (4) and the regularization term presented in Eq. (5). For a more comprehensive understanding of our algorithm’s architecture and implementation, refer to the details provided in Appendix B.

5 Experiments

In this section, we focus on the following questions: (1) How is the performance of our method compared to that of previous approaches in the standard offline RL benchmark tasks, (2) To what extent is the performance improved when utilizing a limited number of dynamics models, as opposed to the scenario where the dynamics models are not re-trained, and (3) How the number of models and the rollout length influence the performance. We investigate these questions through extensive experiments on both the conventional D4RL [Fu *et al.*, 2020] offline RL benchmark and the near-real-world NeoRL [Qin *et al.*, 2022] benchmark within MuJoCo task.

5.1 Experiment Setup

In the initial phase of pre-training the dynamics models, we divide the offline dataset into a training set and a validation set using an 8:2 ratio. For each task, we construct a set of estimated models by training either 7 (for D4RL) or 15 (for NeoRL) models. After this training, we proceed to select the top 5 (for D4RL) or 10 (for NeoRL) models based on their predictive accuracy, which is evaluated on the validation set.

Baselines. We conduct a comparison of our algorithm against various state-of-the-art offline RL methods. These include model-free approaches as follows: (1) BC that simply mimics the policy that collected the dataset, (2) CQL [Kumar *et al.*, 2020a] that uniformly penalizes Q-values for out-

of-distribution samples, (3) APE-V [Ghosh *et al.*, 2022] that uses the Bayesian approach for decision-making within in-support region, and (4) EDAC [An *et al.*, 2021] that diversifies Q-value ensembles to effectively estimate the expected minimum Q-value for policy learning. Additionally, we consider model-based approaches, which include (5) MOPO [Yu *et al.*, 2020] that uses the uncertainty of the transition prediction as a penalty on reward function, (6) COMBO [Yu *et al.*, 2021] that applies the penalty function of CQL within the model-based regime, (7) MAPLE [Lee *et al.*, 2021] that also addresses decision-making in the out-of-support region by utilizing an ensemble of dynamics models, and (8) RAMBO [Rigter *et al.*, 2022] that trains the policy through maximizing the expected value while training the dynamics model to minimize the expected value.

5.2 D4RL Benchmark

Datasets. Our evaluation spans across twelve datasets encompassing three distinct environments (halfcheetah, walker2d, hopper) and four data types (random, medium, medium-replay, medium-expert) for each environment. We used the ‘v2’ version of the datasets, and the results are summarized in Table 1. The reported scores are obtained from the 10 episodes of online evaluation at the last iteration.

Performance comparison. As a notable model-free baseline, EDAC uses either 10 or 50 critics to estimate the pessimistic Q-value. Our approach in D4RL relies on a mere 5 dynamics models to enhance the synthetic dataset for fitting the value function. This implies that when we have a restricted number of neural networks, employing them to enhance generalization can yield better results compared to utilizing them for estimating pessimistic values. This underscores the efficiency of our proposed method.

Model-based baselines also utilize 5 ensemble models, except MAPLE ($n = 14$). These models are employed to create pessimistic MDP, through estimating uncertainty or regularizing values for state-action pairs that are unseen during

Task Name	100						1000					
	BC	CQL	EDAC	MOPO	COMBO	MoDAP	BC	CQL	EDAC	MOPO	COMBO	MoDAP
HalfCheetah-L	29.1	32.6	32.5	42.0	33.8	49.5±4.5	29.1	38.2	38.2	40.1	32.9	53.9±1.1
Walker2d-L	29.1	30.3	21.9	9.7	22.7	26.6±17.5	28.5	44.7	39.8	11.6	31.7	51.3±7.8
Hopper-L	16.1	16.5	17.6	5.0	16.4	21.8±5.0	15.1	16.8	18.6	6.2	17.9	26.1±4.7
HalfCheetah-M	48.9	51.6	52.0	63.1	47.1	70.8±2.7	49.0	54.6	57.5	62.3	50.8	81.0±2.3
Walker2d-M	50.2	53.2	51.6	20.1	53.1	65.9±2.7	48.7	57.3	56.7	39.9	53.8	70.8±3.1
Hopper-M	28.0	63.2	15.4	1.8	55.2	29.5±8.4	51.3	64.5	42.4	1.0	56.3	44.2±15.3
HalfCheetah-H	47.2	74.0	7.1	47.8	15.8	64.4±8.4	71.3	77.4	79.7	65.9	62.2	84.1±8.3
Walker2d-H	64.1	74.3	72.3	23.3	71.2	57.5±25.1	72.6	75.3	76.0	18.0	71.8	73.6±2.8
Hopper-H	44.4	69.7	25.7	7.6	37.0	20.2±1.3	43.1	76.6	53.5	11.5	63.2	28.5±10.8

Table 2: Results on MuJoCo NeoRL benchmark with (100, 1000) trajectories datasets. Normalized scores are calculated as (score - random policy score) / (expert policy score - random policy score), and the standard deviation is denoted by \pm . The score of our algorithm is an average across 4 random seeds.

training. The results of our experiments convincingly demonstrate that our algorithm’s heightened capacity for generalization can outperform or rival these model-based alternatives.

We also conducted a comparison with MAPLE that does not re-train the dynamics model during policy training. As we excluded the results of MAPLE ($n = 5$) due to the relatively lower performance, the detailed comparison results with MAPLE are provided in Appendix D.1. It is worth to note that both MoDAP and MAPLE initiate policy training using the same dynamics models across different seeds in case ($n = 5$). Here, MoDAP exhibits significant performance improvement compared to MAPLE (e.g., +54.3 on halfcheetah-medium-expert or +39.6 on halfcheetah-medium-replay). Interestingly, even though MAPLE uses a larger number of dynamics models ($n = 14$) as shown in Table 1, its performance does not surpass that of MoDAP, which uses only 5 models. This highlights the importance of the diverse composition of the set of dynamics models as a critical factor in learning adaptive policy for model-based offline RL.

5.3 NeoRL Benchmark

Datasets. NeoRL benchmark [Qin *et al.*, 2022] is designed to replicate real-world scenarios. They have collected datasets through the more conservative policy, which aligns more closely with data-collection procedures encountered in real-world settings. The constrained nature of its narrower datasets hinders the high performance of existing offline RL algorithms. We focused on nine datasets that encompass three environments (HalfCheetah-v3, Hopper-v3, Walker-v3) and are categorized into three quality levels (L, M, H), representing low, medium, and high, respectively. Notably, NeoRL offers varying numbers of trajectories for training data (100, 1000, 10000) for each task. We have specifically experimented on 100 and 1000 trajectories settings, in order to nuance the limited support region covered by the datasets.

Performance comparison. In Table 8, we compared the performance of our algorithm with five baselines, including both model-free and model-based offline RL algorithms. To accommodate the characteristics of the NeoRL benchmark, MoDAP employs 10 dynamics models, in contrast to the

5 models utilized in the D4RL experiments. Further details about the baseline results and hyperparameter settings can be found in Appendix C.2. Our MoDAP method consistently showed robust and superior performance especially when compared to the deteriorated performance of EDAC, which performed remarkably well in D4RL benchmark. As the dataset support of NeoRL is significantly constrained, our diversity policy learning demonstrates the potential benefits of making decisions in out-of-support regions.

Furthermore, MoDAP showed substantial performance improvement with the larger number of trajectories (increasing from 100 to 1000). The augmented dataset size significantly contributes to constructing a set of better dynamics models. Meanwhile, the performance of the CQL method, which demonstrated high scores on several NeoRL datasets, showed relatively marginal improvement. This can be attributed to the inherent nature of NeoRL datasets, where the introduction of more conservative data does not necessarily translate to performance improvement.

We have observed that the performance of MoDAP was lower than that of BC, especially in the Hopper-v3-high dataset. This suggests that incorporating synthetic data sometimes leads to worse results than relying solely on the offline dataset if there is a significant discrepancy between the synthetic data and the true environment. We believe that increasing the number of dynamics models can handle this discrepancy issues, potentially leading to performance improvement. Specifically, in the Hopper-v3-high dataset, the normalized scores of MoDAP with $n = 40$ demonstrate a significant improvement: 37.1 ± 0.8 for 100 trajectories (compared to 20.2 for $n=10$), and 52.4 ± 3.2 for 1000 trajectories (compared to 28.4 for $n=10$). These results clearly indicate that increasing the number of dynamics models leads to substantial improvements in performance.

5.4 Analysis on Hyperparameters

The number of models n employed plays a pivotal role in the performance of our algorithm, and rollout length k can impact our algorithm’s behavior. To investigate these considerations, we conduct experiments to analyze the asymptotic

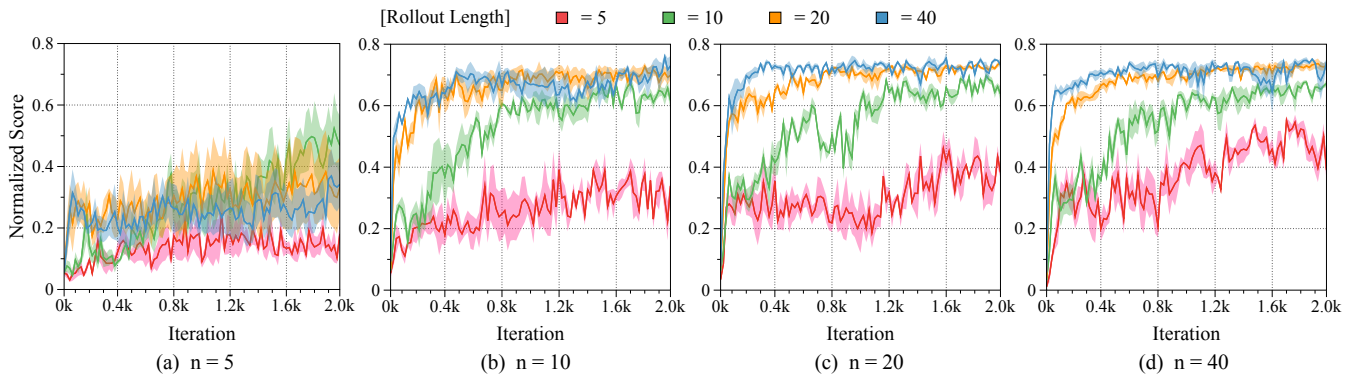


Figure 1: Investigation of performance variation with varying number of models and rollout lengths. The graph depicts the average score across 3 seeds, with standard error shading on the medium-quality dataset with 100 trajectories in the Walker2d-v3 task.

performance while varying the number of model n and the rollout length k . This decision was driven by the understanding that utilizing a comparatively small number of training trajectories would enable a clear observation of the distinct impact of varying the number of models. Furthermore, the use of a medium-quality dataset is particularly effective in showcasing the algorithm’s inherent capabilities. This is because other types of datasets could potentially yield performances that are either too low or too high, thus diminishing the significance of performance differences.

The results, illustrated in Figure 1, reveal several trends. Notably, for all values of n except when using 5 models, increasing the rollout length k leads to improved performance of the converged policies. However, in the case of 5 models, all tested rollout lengths show worse performance. This suggests that utilizing only 5 models is insufficient in the Walker-v3-medium-100 task.

Interestingly, when a rollout length of $k = 5$ is employed, increasing the number of models does not significantly improve performance. This observation implies that a rollout length of 5 may not be adequate to generate the necessary diversity across the dynamics models. However, as the number of models becomes sufficiently large, such as in the case of 20, 40 models, a rollout length of 40 exhibits rapid and superior performance. This outcome underscores the effectiveness of our algorithm in making decisions in out-of-support regions.

6 Conclusion

Offline RL has wrestled with the fundamental challenge of policy learning from limited datasets that capture only a fraction of the environment’s dynamics. While constrained optimization has been a prevailing approach to cope with this challenge, it tends to suffer when encountering unexplored region during test time. To tackle this dilemma and enable better decision-making in previously unexplored regions, the advent of adaptive methodologies has been instrumental. These approaches have the capacity to train highly generalizable adaptive policy by incorporating a broad range of possible dynamics models as potential candidates for representing the true environment. However, the practical construction of an

exhaustive set of dynamics models remains elusive, often resorting to ensemble structures to approximate this diversity. In this paper, we delve into the crucial question of which dynamics models should be embraced in training adaptive policies. To address this, we introduce a novel framework for assembling a set of dynamics models, driven by a mutual-information-based objective. Our method is put to the test through comprehensive experiments conducted on both the D4RL and NeoRL benchmarks. The results demonstrate the superiority and competitiveness of our approach when compared to state-of-the-art algorithms, affirming its prowess in enhancing policy learning from limited data.

Furthermore, there exist several potential extensions to our algorithm, particularly within the context of Offline Meta RL [Pong *et al.*, 2022; Lin *et al.*, 2022]. In such scenarios, a limited number of tasks are sampled from the task distribution. To facilitate adaptation to new tasks, it would be advantageous to construct novel tasks that can interact with the policy, ultimately enhancing its generalization capability.

Acknowledgments

This work was supported by IITP grant funded by MSIT (No.2019-0-00075 Artificial Intelligence Graduate School Program (KAIST); No.2020-0-00940 Foundations of Safe Reinforcement Learning and Its Applications to Natural Language Processing; No.2022-0-00311 Development of Goal-Oriented Reinforcement Learning Techniques for Contact-Rich Robotic Manipulation of Everyday Objects).

References

- [An *et al.*, 2021] Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 7436–7447. Curran Associates, Inc., 2021.
- [Asmuth *et al.*, 2009] John Asmuth, Lihong Li, Michael L. Littman, Ali Nouri, and David Wingate. A bayesian sampling approach to exploration in reinforcement learning.

- In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, page 19–26, Arlington, Virginia, USA, 2009. AUAI Press.
- [Chen *et al.*, 2021] Xiong-Hui Chen, Yang Yu, Qingyang Li, Fan-Ming Luo, Zhiwei Tony Qin, Shang Wenjie, and Jieping Ye. Offline model-based adaptable policy learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.
- [Chung *et al.*, 2014] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.
- [Duan *et al.*, 2017] Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, and Pieter Abbeel. RL²: Fast reinforcement learning via slow reinforcement learning, 2017.
- [Eysenbach *et al.*, 2019] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019.
- [Fu *et al.*, 2020] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2020.
- [Fujimoto and Gu, 2021] Scott Fujimoto and Shixiang (Shane) Gu. A minimalist approach to offline reinforcement learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 20132–20145. Curran Associates, Inc., 2021.
- [Fujimoto *et al.*, 2019] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2052–2062. PMLR, 09–15 Jun 2019.
- [Ghavamzadeh *et al.*, 2015] Mohammed Ghavamzadeh, Shie Mannor, Joelle Pineau, and Aviv Tamar. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015.
- [Ghosh *et al.*, 2021] Dibya Ghosh, Jad Rahme, Aviral Kumar, Amy Zhang, Ryan P Adams, and Sergey Levine. Why generalization in RL is difficult: Epistemic POMDPs and implicit partial observability. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.
- [Ghosh *et al.*, 2022] Dibya Ghosh, Anurag Ajay, Pulkit Agrawal, and Sergey Levine. Offline RL policies should be trained to be adaptive. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 7513–7530. PMLR, 17–23 Jul 2022.
- [Haarnoja *et al.*, 2018] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.
- [Jiang *et al.*, 2020] Xiaoyu Jiang, Qiuxuan Chen, Shiyi Han, Mingxuan Li, Jingyan Dong, and Ruochen Zhang. When to trust your model: Model-based policy optimization, 2020. Submitted to NeurIPS 2019 Reproducibility Challenge.
- [Kidambi *et al.*, 2020] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21810–21823. Curran Associates, Inc., 2020.
- [Kolter and Ng, 2009] J. Zico Kolter and Andrew Y. Ng. Near-bayesian exploration in polynomial time. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 513–520, New York, NY, USA, 2009. Association for Computing Machinery.
- [Kostrikov *et al.*, 2021] Ilya Kostrikov, Rob Fergus, Jonathan Tompson, and Ofir Nachum. Offline reinforcement learning with fisher divergence critic regularization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning Research*, pages 5774–5783. PMLR, 18–24 Jul 2021.
- [Kumar *et al.*, 2019] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [Kumar *et al.*, 2020a] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191. Curran Associates, Inc., 2020.
- [Kumar *et al.*, 2020b] Saurabh Kumar, Aviral Kumar, Sergey Levine, and Chelsea Finn. One solution is not all you need: Few-shot extrapolation via structured maxent rl. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8198–8210. Curran Associates, Inc., 2020.
- [Lee *et al.*, 2021] Byung-Jun Lee, Jongmin Lee, and Kee-Eung Kim. Representation balancing offline model-based reinforcement learning. In *International Conference on Learning Representations*, 2021.

- [Levine *et al.*, 2020] Sergey Levine, Aviral Kumar, G. Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *ArXiv*, 2020.
- [Lin *et al.*, 2022] Sen Lin, Jialin Wan, Tengyu Xu, Yingbin Liang, and Junshan Zhang. Model-based offline meta-reinforcement learning with regularization. In *International Conference on Learning Representations*, 2022.
- [Liu *et al.*, 2021] Jinxin Liu, Hao Shen, Donglin Wang, Yachen Kang, and Qiangxing Tian. Unsupervised domain adaptation with dynamics-aware rewards in reinforcement learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.
- [Mnih *et al.*, 2013] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.
- [Nair *et al.*, 2021] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets, 2021.
- [OpenAI *et al.*, 2019] OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d. O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning, 2019.
- [Pong *et al.*, 2022] Vitchyr H Pong, Ashvin V Nair, Laura M Smith, Catherine Huang, and Sergey Levine. Offline meta-reinforcement learning with online self-supervision. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 17811–17829. PMLR, 17–23 Jul 2022.
- [Poupart *et al.*, 2006] Pascal Poupart, Nikos Vlassis, Jesse Hoey, and Kevin Regan. An analytic solution to discrete bayesian reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 697–704, New York, NY, USA, 2006. Association for Computing Machinery.
- [Qin *et al.*, 2022] Rong-Jun Qin, Xingyuan Zhang, Songyi Gao, Xiong-Hui Chen, Zewen Li, Weinan Zhang, and Yang Yu. NeoRL: A near real-world benchmark for offline reinforcement learning. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.
- [Rigter *et al.*, 2022] Marc Rigter, Bruno Lacerda, and Nick Hawes. RAMBO-RL: Robust adversarial model-based offline reinforcement learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.
- [Ross *et al.*, 2007] Stephane Ross, Brahim Chaib-draa, and Joelle Pineau. Bayes-adaptive POMDPs. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.
- [Silver *et al.*, 2017] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharmhan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm, 2017.
- [Sun *et al.*, 2023] Yihao Sun, Jiayi Zhang, Chengxing Jia, Haoxin Lin, Junyin Ye, and Yang Yu. Model-Bellman inconsistency for model-based offline reinforcement learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 33177–33194. PMLR, 23–29 Jul 2023.
- [Wang *et al.*, 2020] Ziyu Wang, Alexander Novikov, Konrad Zolna, Josh S Merel, Jost Tobias Springenberg, Scott E Reed, Bobak Shahriari, Noah Siegel, Caglar Gulcehre, Nicolas Heess, and Nando de Freitas. Critic regularized regression. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7768–7778. Curran Associates, Inc., 2020.
- [Yu *et al.*, 2020] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14129–14142. Curran Associates, Inc., 2020.
- [Yu *et al.*, 2021] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 28954–28967. Curran Associates, Inc., 2021.