

---

# Behavioral Priors and Dynamics Models: Improving Performance and Domain Transfer in Offline RL

---

Anonymous Author(s)

Affiliation  
Address  
email

## Abstract

1 Offline Reinforcement Learning (RL) aims to extract near-optimal policies from  
2 imperfect offline data without additional environment interactions. Extracting  
3 policies from diverse offline datasets has the potential to expand the range of  
4 applicability of RL by making the training process safer, faster, and more stream-  
5 lined. We investigate how to improve the performance of offline RL algorithms,  
6 its robustness to the quality of offline data, as well as its generalization capabili-  
7 ties. To this end, we introduce Offline Model-based RL with Adaptive Behavioral  
8 Priors (MABE). Our algorithm is based on the finding that dynamics models,  
9 which support within-domain generalization, and behavioral priors, which support  
10 cross-domain generalization, are complementary. When combined together, they  
11 substantially improve the performance and generalization of offline RL policies.  
12 In the widely studied D4RL offline RL benchmark, we find that MABE achieves  
13 higher average performance compared to prior model-free and model-based algo-  
14 rithms. In experiments that require cross-domain generalization, we find that  
15 MABE outperforms prior methods.

## 16 1 Introduction

17 Over the last five years advances in Deep Reinforcement Learning (RL) have been at the  
18 source of a number of impressive results in autonomous control, including the ability to solve  
19 video games from pixels [2], master the game of Go [3], play multi-agent large scale video  
20 games [4], and control robots [5]. Most advances in RL were achieved in simulated envi-  
21 ronments where data was cheap to collect and mistakes during policy training were harmless.  
22 However, two substantial problems stand in the way from utilizing the above approaches to de-  
23 ploy RL algorithms in real-world settings. First, since RL algorithms require millions and some-  
24 times billions of environment interactions, learning policies with RL in the real world is costly  
25 in terms of time and resources. Second, since RL algorithms stochastically explore their envi-  
26 ronment, the resulting agents are not safe and can harm the environment, themselves, or other  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36

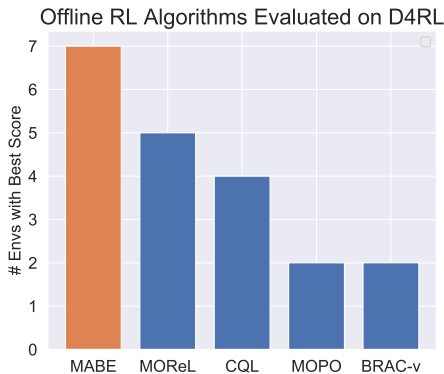


Figure 1: Our proposed algorithm, MABE, when compared to prior work, achieves the top score in **7 out of 9** D4RL datasets [1] we study. We consider multiple algorithms to achieve the top score if they are within 2% points of each other.

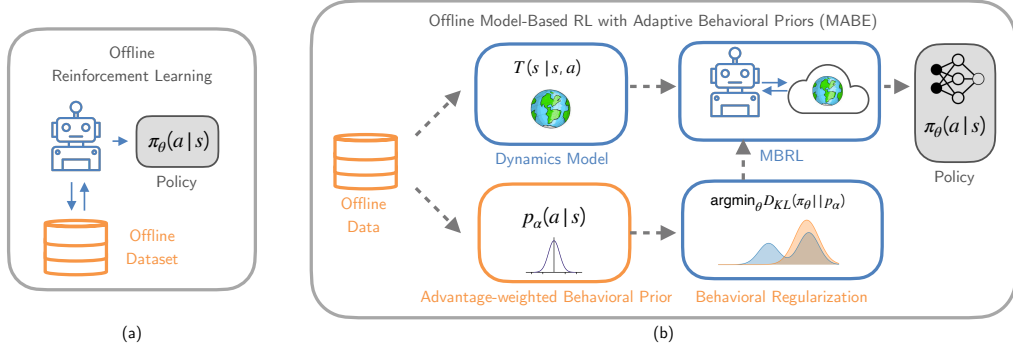


Figure 2: (a) The offline RL paradigm. Rather than interacting with the environment directly, an agent extracts a policy from an offline dataset. (b) A schematic architecture of our proposed algorithm (MABE). First, by interaction with the the offline dataset, the agent learns a dynamics model and an advantage weighted behavioral prior. Then, the dynamics model generates a synthetic dataset which is used alongside the original offline data to train the policy  $\pi_\theta(a|s)$ . Finally, the behavioral prior regularizes the learned policy to keep the agent within the support of the original dataset.

37 agents if trained in the real world. How can we overcome the challenges of data efficiency and safety  
 38 to enable RL algorithms that can be deployed in real world settings?

39 Offline or Batch RL [6, 7] has recently been proposed as a promising paradigm to tackle these  
 40 challenges. Offline RL agents use logged or previously collected data by humans or other agents  
 41 for learning. Importantly, the offline data does not have to consist of expert demonstrations like in  
 42 the case of imitation learning [8, 9, 10], but can be collected with policies that are sub-optimal or  
 43 noisy. Such policies may already be in deployment for a variety of applications like autonomous  
 44 driving, warehouse automation, dialogue systems [11, 12] and recommendation systems [13, 14]. By  
 45 learning policies only using offline datasets and perhaps fine-tuning the policy using a small dataset  
 46 of subsequent interactions, offline RL has the potential to be highly sample efficient and safe. The  
 47 primary challenge with extracting policies from offline data comes from the distribution mismatch  
 48 between transitions seen during training and those encountered during evaluation. Conservatism  
 49 or pessimism has emerged as a core principle in offline RL to deal with distribution mismatch.  
 50 Conservatism encourages the offline RL agent to improve the policy while also staying close to  
 51 the dataset distribution, thereby minimizing distribution shift between training and deployment.  
 52 A number of algorithms, both model-free and model-based, have been proposed that incorporate  
 53 conservatism in various forms like importance weights [15], value functions [16, 17, 18, 19], and  
 54 dynamics models [20, 21, 22, 23].

55 Recently, model-based offline RL algorithms like MOREL [20] and MOPO [21] have demonstrated  
 56 impressive results in benchmark tasks and also the ability to re-purpose the learned dynamics model  
 57 to solve downstream tasks that are different from those encountered in the offline dataset. They  
 58 incorporate conservatism in the learning process by learning pessimistic dynamics models using  
 59 uncertainty quantification. However, uncertainty quantification with deep neural networks can pose  
 60 challenges in many domains, such as those with high dimensional input-output spaces or multiple  
 61 confounding factors [24, 25, 26, 27, 28]. Since offline RL views uncertainty quantification as  
 62 a means to the end of incorporating conservatism, and since uncertainty quantification by itself  
 63 can be a difficult exercise, we are motivated to develop offline RL algorithms that do not require  
 64 uncertainty quantification. In this work, we develop an algorithm that achieves this goal. Our  
 65 algorithm outperforms prior approaches in the widely studied D4RL benchmark [1] as well as in  
 66 tasks that require domain adaptation and generalization. Thus, our algorithm has potentially wider  
 67 applicability, especially in settings where uncertainty estimation can be difficult.

68 **Our Contribution** Our principal contribution in this work is the development of a new algorithm –  
 69 offline model-based RL with Adaptive Behavioral Regularization (MABE). Using the offline dataset,  
 70 MABE learns an approximate dynamics model, reward function, as well as an adaptive behavioral  
 71 prior. By adaptive behavioral prior, we mean a policy that approximates the behavior in the offline  
 72 dataset while giving more importance to trajectories with high rewards. Using the learned dynamics  
 73 model and reward function, MABE performs model-based RL with an objective to maximize the

74 rewards along with a KL-divergence penalty that encourages the agent to stay close to adaptive  
 75 behavioral prior. This divergence penalty provides the necessary conservatism needed to succeed in  
 76 offline RL. Our major findings in this work are listed below.

- 77 1. Our algorithm, MABE, achieves the highest scores in 7 out of 9 D4RL [1] benchmark tasks  
 78 we study, as well as the highest average normalized score.
- 79 2. MABE is flexible and can benefit from uncertainty estimation if available or forgo it  
 80 altogether. Our empirical ablations suggest that uncertainty estimation contributes only  
 81 minor improvements compared to the other components of dynamics models and behavioral  
 82 priors. Thus, MABE can be used in a wider set of application domains, especially those  
 83 where uncertainty estimation is difficult.
- 84 3. We demonstrate that MABE has favorable generalization capabilities to new tasks by  
 85 leveraging the learned dynamics model and transferring of behavioral priors across datasets,  
 86 a capability that is only possible when both model-based and behavioral priors are combined.

## 87 2 Preliminaries

88 We operate in the standard RL setting of infinite horizon discounted Markov Decision Process (MDPs),  
 89 defined as the tuple  $\mathcal{M} = (S; A; R; T; \gamma; s_0)$ . The MDP tuple has states  $s \in S$ , actions  $a \in A$ ,  
 90 rewards  $r = R(s; a)$ , transition dynamics  $s^0 \sim T(js; a)$ , an initial state distribution  $s_0 \sim s_0(\cdot)$ , and a  
 91 discount factor  $\gamma \in [0; 1)$ . A policy defines a mapping from states to actions, typically in the form of  
 92 a probability distribution:  $a \sim \pi(js)$ . The value  $V(s)$  and action-value function  $Q(s; a)$  describe  
 93 the long term reward behavior of policy  $\pi$ .

$$V(s) := E_{\mathcal{M}; \pi} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t; a_t) \mid s_0 = s \right]; Q(s; a) := R(s; a) + E_{s^0 \sim T(js; a)} [V(s^0)]$$

94 where the first expectation  $E_{\mathcal{M}; \pi}$  denotes actions are sampled according to  $\pi$  and future states are  
 95 sampled according to the MDP dynamics  $T(js; a)$ . The goal in RL is to learn the optimal policy:

$$\pi^* \in \arg \max_{\pi} J(\pi; \mathcal{M}) := E_{s \sim s_0} [V(s)]; \quad (1)$$

96 When the MDP (especially  $T$ ) is unknown, exploration is important to learn the optimal policy.

97 **Model-Based RL (MBRL)** is an approach to learning in MDPs that involves learning an approxi-  
 98 mate MDP  $\hat{\mathcal{M}} = (S; A; \hat{R}; \hat{T}; \hat{s}_0)$ . The learned MDP has the same state and action spaces, but  
 99 uses the learned approximate dynamics and reward models. Generating samples from  $\hat{\mathcal{M}}$  is cheap  
 100 and does not require environment interaction. As a result, various algorithms based on policy gradient  
 101 and dynamic programming [29] can be used to efficiently improve the policy, with intermittent data  
 102 collection to improve model approximation quality. Recently, MBRL algorithms have demonstrated  
 103 strong results in a variety of RL tasks [30, 31, 32, 33], including offline RL [20, 21, 22].

104 **Offline RL** is a setting in RL where we must learn a policy using a fixed dataset of environment  
 105 interactions. Specifically, we are given a dataset of interactions  $D = \{s_i; a_i; s_i^0; r_i; g_{i=1}^N\}$  of  $N$   
 106 environment interactions collected using one or more behavioral policies. If the behavioral policies  
 107 do not induce sufficient exploration, it is not possible to learn an optimal policy for the underlying  
 108 MDP even as  $N \rightarrow \infty$  [34, 20]. Thus, the goal in offline RL is typically to learn the best possible  
 109 policy using the provided dataset.

110 **Model-Based Offline RL** algorithms like MOPO [21] and MOREL [20] leverage MBRL to learn  
 111 in the offline RL setting. They learn an approximate MDP using the offline dataset. Simulation with  
 112 the learned MDP allows the offline RL agent to ask counterfactual questions about actions that are  
 113 unseen in the dataset by leveraging the generalization capabilities of the learned dynamics model.  
 114 However, since the model cannot be iteratively refined or improved like in the case of online RL,  
 115 the learned MDP is likely erroneous on out-of-distribution states. As a result, policy learning in the  
 116 learned MDP may exploit the errors in the model to optimize rewards, leading to poor performance in  
 117 the true MDP. To guard against this exploitation, MOPO and MOREL penalize the agent for visiting  
 118 out-of-distribution states in the learned MDP, with uncertainty in the dynamics model being used to  
 119 detect out-of-distribution states.

### 120 3 Model-Based Offline RL with Adaptive Behavioral Regularization

121 Given an offline dataset  $D$ , our goal is to learn a parameterized policy that achieves high rewards,  
 122 without any additional interaction with the environment. We assume  $D$  consists of  $(s; a; s'; r; g)$  tuples  
 123 which we use to learn  $\hat{\pi}$  along with a behavioral prior  $p(a_t|s_t)$ . This dataset can be collected  
 124 using one or more structured behavioral policies interacting with test environment. We now present  
 125 our algorithm MABE (Model-Based Offline RL with Adaptive Behavioral Regularization), which  
 126 consists of three components described below.

127 **Dynamics Model Learning** MABE is a model-based RL algorithm, and thus we use the offline  
 128 dataset to learn a neural network dynamics model. This can be accomplished using maximum  
 129 likelihood estimation or other generative modeling techniques such as variational models [32]. Let  
 130  $\hat{\pi}(s; a)$  represent the generative model for the conditional next state distribution. Similar to prior  
 131 offline MBRL works [21, 20, 22, 23], we learn the generative dynamics model with maximum-  
 132 likelihood learning as:

$$\max_{\hat{\pi}(s; a)} \mathbb{E}_{(s; a; s')} \log \hat{\pi}(s'; a) \quad (2)$$

133 **Learning Behavioral Priors** Our main insight is the use of adaptive behavioral priors as a  
 134 form of regularization in offline MBRL. Building on prior work [35, 36], we utilize behavioral  
 135 regularization within the MBRL framework. Our experimental results suggest that combining MBRL  
 136 with behavioral regularization can incorporate sufficient conservatism to succeed in offline RL. This  
 137 is in contrast to prior offline MBRL works that rely crucially on uncertainty estimation which may  
 138 prove difficult in various applications.

139 We consider a parameterized generative model  $p(a_t|s_t)$  that represents our behavioral prior. A  
 140 straightforward option is to learn a behavior model that replicates the statistics in the dataset.

$$p^{\text{eq}} \triangleq \arg \max_p \mathbb{E}_D \sum_{t=0}^J \log p(a_t|s_t) \quad (3)$$

141 Alternatively, we can consider an adaptive behavioral prior that is biased towards trajectories that  
 142 achieve higher rewards. This can be particularly useful in diverse datasets collected with multiple  
 143 policies – some of which perform better at the task while other policies may exhibit behaviors that  
 144 may hinder the task we want the offline RL agent to learn. Similar to Siegel et al. [37], we seek a  
 145 behavioral prior that is biased towards the high reward trajectories in the dataset while also staying  
 146 close to the average statistics in the dataset. We formulate this as:

$$p^{\text{adapt}} \triangleq \arg \max_p \mathbb{E}_D \sum_{t=0}^J \omega(s_t; a_t) \log p(a_t|s_t) \quad (4)$$

$$\text{subject to } \mathbb{E}_s \mathbb{E}_D [D_{KL}(p \| p)] \leq \epsilon;$$

147 where  $p$  denotes the empirical behavioral policy and  $\omega(s_t; a_t)$  is the weighting function. The non-  
 148 parametric solution to the above optimization is given by:

$$p^{\text{adapt}}(a_t|s_t) \propto p(a_t|s_t) \exp(\omega(s_t; a_t) / \beta);$$

149 where we have used  $\propto$  to avoid specification of the normalization factor, and  $\beta$  represents a tem-  
 150 perature parameter that is related to the constraint level  $\epsilon$ . The above non-parametric policy can be  
 151 projected into the space of parametric neural network policies as [38, 37]:

$$p^{\text{adapt}} \triangleq \arg \max_p \mathbb{E}_D \sum_{t=0}^J \exp(\omega(s_t; a_t) / \beta) \log p(a_t|s_t) \quad (5)$$

152 For the choice of the weighting function, we use

$$\omega(s_t; a_t) := \hat{Q}(s_t; a_t) - (1 - \gamma)r_{\max};$$

153 where  $\hat{Q}$  is learned using TD-error minimization and  $r_{\max}$  is the maximum reward observed in the  
 154 dataset. In this process, we treat the temperature  $\beta$  as the hyper-parameter choice. This implicitly  
 155 defines the constraint threshold  $\epsilon$ , and makes the problem specification and optimization more  
 156 straightforward.

157 **Behavior Regularized Model-Based RL** Equipped with a dynamics model and adaptive behav-  
 158 ioral prior, our algorithm MABE, performs model-based RL with a regularized objective given by:  
 159

$$\max_{\pi} \mathbb{E}_{\mu} [F(s; a)]; \text{ with } F(s; a) := \hat{r}(s; a) - (\log \hat{p}(a|s) + \log p(a|s)) : \quad (6)$$

160 We use  $\hat{\mu}$  to denote the discounted state visitation distribution induced by executing  $\pi$  in the  
 161 learned MDP model. This objective encourages the agent to increase the rewards along with entropy  
 162 and behavioral regularization. We learn a policy to solve this optimization using SAC [39], resulting  
 163 in an algorithm that is similar to a behavior regularized version of Dyna [40] and MBPO [31].  
 164 Algorithm 3 presents the full details of our learning approach.

---

**Algorithm 1** MABE: Model-Based Offline RL with Adaptive Behavioral Regularization

---

- 1: **Inputs:** Offline dataset  $D$ , learned dynamics and reward models  $\hat{T}$  and  $\hat{r}$ , adaptive behavioral prior  $p(a|s)$ , target divergence  $\epsilon$ , learning rates  $\alpha, \beta, \beta_c$ , rollout length  $h$
  - 2: Initialize policy  $\pi$ , critic  $Q$ , target network  $Q_c$ , and  $D_{\text{aug}} = D$
  - 3: **for**  $N$  epochs **do**
  - 4:   **for**  $K$  trajectories **do**
  - 5:     Collect rollouts  $(s_t; a_t; r_t; s_{t+1})$  of length  $h$  using  $\hat{T}$  and  $\hat{r}$  starting from a randomly chosen state from the offline dataset.
  - 6:      $D_{\text{aug}} \leftarrow D_{\text{aug}} \cup \{(s_t; a_t; r_t; s_{t+1})\}$
  - 7:   **for** each gradient step **do**
  - 8:     Sample a batch of  $(s_t; a_t; r_t; s_t^j)$  tuples from  $D_{\text{aug}}$
  - 9:      $Q = r(s_t; a_t) + \beta [Q_c(s_t; a_t) - D_{\text{KL}}(\hat{p}(a|s_t); p(a|s_t))]$
  - 10:      $r = [Q_c(s_t; a_t) - D_{\text{KL}}(\hat{p}(a|s_t); p(a|s_t))]$
  - 11:      $\alpha r - \frac{1}{2}(Q(s_t; a_t) - Q_c)^2$
  - 12:      $(D_{\text{KL}}(\hat{p}(a|s_t); p(a|s_t)) - \epsilon)$
  - 13:      $+ (1 - \beta)$
- 

165 **Optional use of uncertainty quantification** MABE is a flexible framework that can additionally  
 166 incorporate uncertainty quantification if available, in addition to the behavioral prior regularization.  
 167 Let  $u(s; a) = D_{TV}(\hat{T}(j; s; a); T(j; s; a))$  be an estimate of the dynamics model uncertainty in state  
 168  $(s; a)$ . Analogous to prior work like MOPO and MOREL, we can additionally incorporate uncertainty  
 169 into the MABE objective given by Eq. 6 as:

$$F(s; a) = \hat{r}(s; a) - (\log \hat{p}(a|s) + \log p(a|s)) - u(s; a):$$

170 We emphasize again that additional reward penalty based on uncertainty is optional, and our experi-  
 171 ment results suggest that it only offers marginal benefits compared to our other components.

## 172 4 Results

173 **MABE design choices** We first outline the main decision choices and implementation details used  
 174 for our experiments. Our implementation of MABE is built on MOPO. We parameterize the policy,  
 175 behavioral prior, and dynamics model as a Gaussian distributions, with the mean being parameterized  
 176 by an MLP network, and the covariance is also learned. For example, the dynamics is represented as

$$\hat{T}(s^j; s; a) = \mathcal{N}(\text{MLP}(s; a); \Sigma):$$

177 The reward and Q-function are modeled using deterministic MLP networks. We learn the policy  
 178 and Q-function using MBPO [31] (which itself uses SAC [39] internally), similar to MOPO. MBPO  
 179 is a model-based RL algorithm that augments Additional implementation details of MABE and  
 180 hyperparameters are provided in the Appendix.

181 **Experiments in D4RL offline RL benchmark tasks** Our first goal is to study the performance of  
 182 MABE on the widely studied D4RL [1] benchmark. We consider a total of nine domains involving  
 183 three simulated locomotion tasks and three datasets per task: medium, medium-replay (or mixed),  
 184 and medium-expert. The medium dataset is collected with partially trained SAC agent, the mixed

185 dataset is the entire replay buffer of a SAC agent throughout training, and the medium-expert is  
 186 a mix between trajectories from the medium dataset and an expert policy. These represent three  
 187 distinct types of imperfect data - one imperfect policy, many changing policies, and a mixture of  
 188 expert and suboptimal policies respectively. We compare our method to published leading offline  
 189 RL algorithms which include: (a) MOREL [20] and MOPO [21] – model-based algorithms that  
 190 rely on uncertainty quantification; (b) CQL [17], a model-free algorithm that learns a conservative  
 191 Q-function, and (c) BRAC-v [35], which regularizes a model-free actor-critic algorithm with an  
 192 unweighted (or equally-weighted) behavioral prior. Please see appendix for more details.

193 Evaluation scores on D4RL are shown in Table 1. We find that MABE achieves the highest score  
 194 on the majority (7 out of 9) environments as well as the highest average score of 77.5. Crucially,  
 195 MABE’s performance is robust across the three dataset types, achieving a leading score on at least 2  
 196 out of 3 environments for each dataset. Finally, we note that MABE substantially outperforms its the  
 197 two most directly competing baselines: MOPO, an uncertainty-based MBRL method; and BRAC-v, a  
 198 model-free method with explicit behavioral prior regularization. This suggests that a combination of  
 199 MBRL and behavioral priors can substantially benefit offline RL.

Table 1: Normalized scores for the D4RL environments we consider. Scores for MABE are calculated from the average scores of the last 10 evaluation steps, over 3 seeds. Baseline results for MOPO, MOREL, and CQL are reproduced from their respective papers. SAC, BC, and BRAC-v numbers are reproduced from Fu et al. [1]. We observe that MABE either matches or outperforms prior methods in a majority of the tasks, and achieves the highest average score.

Dataset	Environment	BC	MABE (ours)		MOPO		MOREL	SAC	CQL	BRAC-v
medium	halfcheetah	36.1	<b>46.8</b>	0.8	42.3	1.6	42.1	-4.3	44.4	45.5
medium	hopper	29.0	<b>94.1</b>	5.8	28.0	12.4	<b>95.4</b>	0.8	58.0	32.3
medium	walker2d	6.6	65.7	8.5	17.8	19.3	77.8	0.9	79.2	<b>81.3</b>
med-replay	halfcheetah	38.4	<b>53.5</b>	0.5	<b>53.1</b>	2.0	40.2	-2.4	46.2	45.9
med-replay	hopper	11.8	71.7	12.5	67.5	24.7	<b>93.6</b>	1.9	48.6	0.9
med-replay	walker2d	11.3	<b>51.0</b>	2.4	39.0	9.6	49.8	3.5	26.7	0.8
med-expert	halfcheetah	35.8	<b>100.6</b>	1.3	63.3	38.0	53.3	1.8	62.4	45.3
med-expert	hopper	111.9	<b>110.5</b>	0.8	23.7	6.0	108.7	1.6	<b>111.0</b>	0.8
med-expert	walker2d	6.4	<b>103.3</b>	1.3	44.6	12.9	95.6	-0.1	98.7	66.6
Average	Average	31.7	<b>77.5</b>		42.1		72.9	0.4	63.9	35.5

200 In the remainder of this section, we investigate in detail why MABE performs well and what new  
 201 capabilities are enabled by MABE.

202 **Which components of MABE contribute most to performance?** MABE consists of several components that each play a part in the final agent. The full MABE algorithm consists of three components: 203 (a) adaptive behavioral prior regularization; (b) policy learning (improvement) using model-based RL; 204 and (c) the optional use of uncertainty quantification through model ensembles [41, 30, 20, 21] to 205

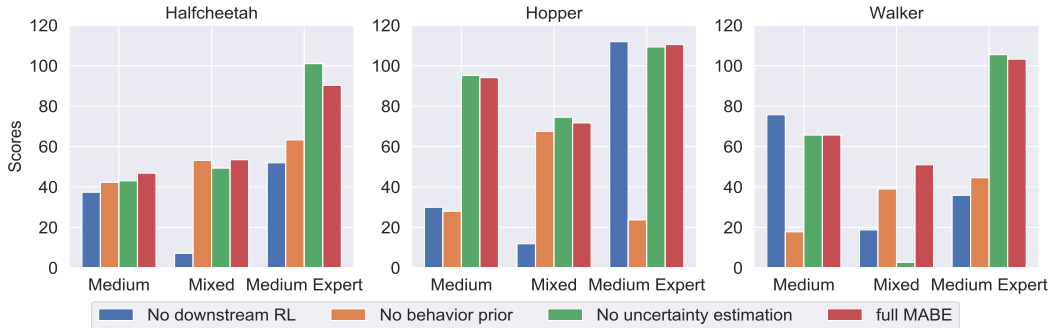


Figure 3: Ablation over the three components of MABE. In most environments, the best performance is achieved from having all three components of MABE, but the component that gives the biggest boost in performance on average is the behavior prior. In most environments, uncertainty estimation does not materially affect the policy’s performance.



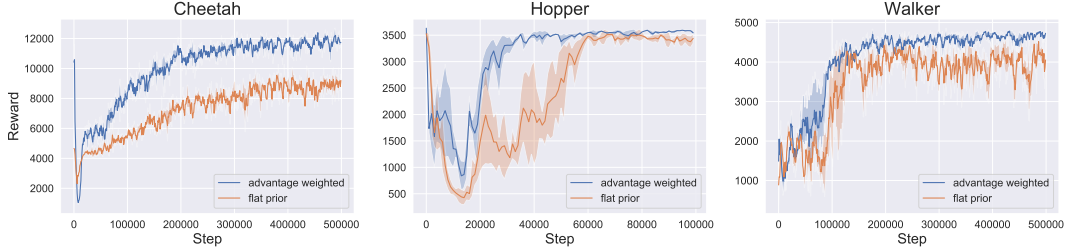


Figure 4: Comparison of MABE run with an advantage weighted behavioral as well as a non-weighted “flat” prior. We find that advantage-weighting improves the stability and performance of the policy.

206 incorporate additional conservatism. In this ablation study, we investigate the importance of each of  
 207 these components by removing one while keeping all others fixed. Results shown in Figure 3, indicate  
 208 that RL and behavioral priors are the largest contributors to MABE’s performance, while the optional  
 209 uncertainty penalty only incrementally improves the final policies. Removing the uncertainty penalty  
 210 leads to an observable drop in performance in only 2 out of the 9 environments. In contrast, removing  
 211 behavioral priors drops performance in 8 environments, and removing RL drops performance in 7.  
 212 Aggregated across the datasets, we find that removing behavioral priors results and RL result in a  
 213 41% and 48% drop in performance respectively. At the same time, removing uncertainty estimation  
 214 only marginally degrades MABE performance by 9%. This suggests that MABE has the potential to  
 215 find wider applicability, especially in situations where uncertainty estimation can be difficult, but can  
 216 also benefit from uncertainty estimation where available.

217 In Figure 3, no downstream RL refers to the direct use of the adaptive behavioral prior, without any  
 218 finetuning with MBRL. This can be viewed as a baseline inspired by imitation learning. The ablation  
 219 study of no-behavioral prior corresponds to MOPO and incorporates conservatism through the use  
 220 of uncertainty estimation. The no uncertainty estimation ablation utilizes adaptive behavior prior  
 221 regularization to incorporate conservatism when learning the policy using MBRL. This utilizes all  
 222 the components of the full MABE algorithm except the optional uncertainty-based reward penalties.  
 223 Finally, the full MABE algorithm uses all the three aforementioned components of behavioral priors,  
 224 policy learning with MBRL, and additional conservatism through uncertainty penalized rewards.

225 **Weighted vs Unweighted Behavioral Prior Regularization** Finally, we ablate the importance of  
 226 adaptive or weighted behavioral priors as used in MABE. In particular, we compare MABE with the  
 227 unweighted behavioral prior in Eq. 3 against the full MABE algorithm that uses the adaptive prior  
 228 in Eq. 5. We show learning curves for MABE trained with the two priors in Figure 4 and find that  
 229 adaptive priors help with training stability as well as asymptotic performance.

230 **Cross-domain and cross-task generalization capability of MABE** A unique capability enabled  
 231 by the use of behavioral priors is the possibility of transferring behaviors from one environment  
 232 (or domain) to another. Prior work has explored the use of offline datasets and RL to acquire new  
 233 behaviors in the same environment [21, 42]. For example, Yu et al. [21] demonstrates that offline RL  
 234 using a dataset that primarily consists of an agent walking forward can be used to learn a jumping

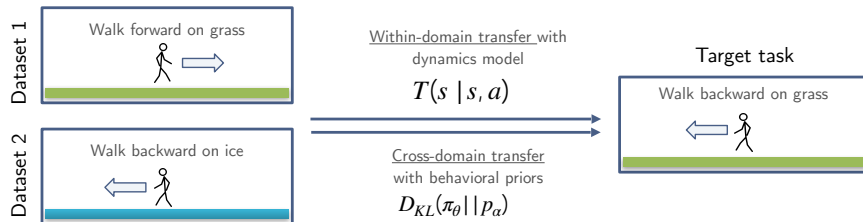


Figure 5: A schematic visualization of MABE’s domain transfer capabilities. In prior work it was shown that offline MBRL is capable of in-domain generalization to new tasks [21]. Here, we investigate cross-domain transfer capabilities of MABE and MOPO. Given multiple datasets with different dynamics and behaviors, can we generalize to a new task in the target domain that was not present in the offline data for the target domain? We hypothesize that behaviors are transferable across domains even if the dynamics are different.

235 behavior. In contrast, we seek for the agent to learn the same behavior but in a different environmental  
 236 condition. This is particularly useful in robotics applications, like for instance home robots that  
 237 operate in kitchens. While the environmental scene and physical dynamics would vary across different  
 238 kitchens depending on the types of cabinets, stoves, plates, floor etc. we would often want to robot  
 239 to exhibit similar behaviors in different kitchens like loading plates in a dishwasher. By utilizing  
 240 behavioral priors that can potentially capture the core concepts of manipulation like force closure for  
 241 grasping, robots can learn to become competent quickly in the home of a target user.

242 To test the generalization capabilities of MABE,  
 243 we setup the following simple experiment.  
 244 We use simulated locomotion agents (Hopper,  
 245 Walker, HalfCheetah), and collect two datasets:  
 246  $D_1$  containing medium-replay forward walking  
 247 data in normal terrain; and  $D_2$  containing expert  
 248 backwards walking data in low friction terrain in-  
 249 tended to simulate ice. In this behavior transfer  
 250 test, we use these two datasets to train an agent  
 251 to run backward on normal terrain. A schematic  
 252 illustration of our setting can be found in Fig-  
 253 ure 5. In our experiments, we consider the fol-  
 254 lowing approaches: (i) *task transfer only* where  
 255 we use the forwards walking dataset to learn a  
 256 backwards walking policy using offline MBRL.  
 257 (ii) *domain transfer only* where we train a policy  
 258 in source domain and directly deploy it in the  
 259 target domain. (iii) *task transfer with behavior*  
 260 *initialization* where we initialize the task transfer  
 261 approach with the adaptive behavioral prior;  
 262 (iv) *task + domain transfer with MABE* where we run MABE using the dataset corresponding to the  
 263 target dynamics ( $D_1$ ) and behavioral prior corresponding to the desired behavior ( $D_2$ ). We show  
 264 the resulting expert normalized scores in Figure 6 and find that MABE is the only algorithm that is  
 265 able to successfully solve the target task through cross-domain behavior transfer. This suggests that  
 266 dynamics models and behavioral priors are complementary and can be used to acquire a wide range  
 267 of behaviors from offline data using domain and task transfer.

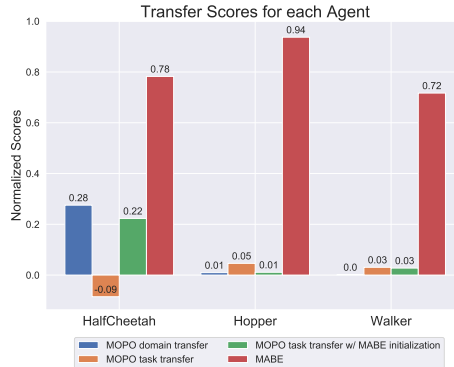


Figure 6: Experiments comparing MABE domain transfer to MOPO domain transfer and MOPO task transfer. MABE is the only offline MBRL approach that is able to successfully transfer behaviors across domains.

## 268 5 Related Work

269 Our method, MABE, is at the intersection of model-based reinforcement learning, offline reinforcement  
 270 learning, and behavioral prior regularization. There are a number of related algorithms that  
 271 utilize dynamics models or behavioral priors in the context of offline RL [21, 35, 23, 36, 37], which  
 272 we describe in Table 2 with a comprehensive overview. While MABE is similar to prior work,  
 273 our primary contribution is identifying a unique mixture of components that enable robust offline  
 274 RL on the D4RL benchmark. Recently, concurrent work COMBO [43] has also investigated an  
 275 uncertainty-free approach to offline MBRL. The difference is that COMBO combines offline MBRL  
 276 with conservative Q-functions whereas MABE utilizes adaptive behavioral priors, which helps with  
 277 cross domain generalization capability as demonstrated in Section 4.

278 **Model-based Reinforcement Learning:** Reinforcement learning algorithms can be broadly clas-  
 279 sified into model-based and model-free categories. Model-based reinforcement learning (MBRL)  
 280 algorithms build an explicit dynamics model of the environment for use with policy search. Model-  
 281 based approaches can be further categorized into Dyna-style algorithms, policy search with temporal  
 282 backpropagation, and shooting methods. In dyna-style approaches [44, 40, 45], interactions with  
 283 the environment are used to update the dynamics model and the RL policy is trained on synthetic  
 284 rollouts from the dynamics model, often using a model-free RL algorithm like policy gradients  
 285 or actor-critic. Some representative examples of Dyna-style algorithms include MBPO [31], ME-  
 286 TRPO [46], PAL/MAL [30], and Dreamer [32]. Policy search with temporal backpropagation and  
 287 differential dynamic programming methods [47, 48, 49, 50, 51] utilize gradients through the model  
 288 to help compute the policy gradient. Shooting methods [41, 52, 53, 54, 55] extract an implicit policy  
 289 from the learned model by performing real-time planning using the learned model. For simplicity and  
 290 to build on prior work in the area of offline RL, we implemented MABE with MBPO, a Dyna-style  
 291 algorithm. However, MABE can in principle be implemented with any MBRL algorithm.



Table 2: A comparison between MABE and similar algorithms.

	MABE (ours)	MOPO/MOReL	BRAC-v [35]	BREMEN [23]	ABM [37]	AWR [36]
Model-Based	Yes	Yes	No	Yes	No	No
Behavior Prior	Adaptive	None	Unweighted	Unweighted	Adaptive	Adaptive
Policy Regularization	Explicit KL	None	Explicit KL	Implicit KL	Implicit KL	Implicit KL
Policy Optimization	SAC	SAC/NPG	SAC	TRPO	MPO [59]	Imitation
Uncertainty	Optional	Yes	No	No	No	No

292 **Offline Reinforcement Learning:** Offline RL [7] has recently received much attention due to its  
 293 potential for applicability in a wide range of applications, and consequently many algorithms have  
 294 been developed recently. Among them include importance sampling based algorithms [56, 15, 14],  
 295 dynamic programming and actor-critic based algorithms [35, 18, 19, 37, 17], and model-based  
 296 algorithms [20, 21, 23, 22]. These algorithms are primarily evaluated using recently proposed  
 297 benchmarks including D4RL [1], Atari [19, 57] and RL-Unplugged [58]. We outline the contrasts  
 298 between MABE and prior work in the remainder of the section.

299 **Relationship to prior offline MBRL algorithms** In terms of the policy learning, our work is closest  
 300 to prior offline MBRL algorithms – MOPO [21] and MOReL [20], which rely on uncertainty  
 301 quantification to estimate model prediction error to incorporate conservatism. In contrast, MABE  
 302 can benefit from uncertainty estimation, but even in its absence demonstrates strong performance  
 303 and thus has wider applicability. BREMEN [23] is another MBRL algorithm that was primarily  
 304 developed for a different setting of deployment efficient RL but can be re-purposed for offline RL.  
 305 Like MABE, it uses a behavioral prior instead of uncertainty driven conservatism. However, it uses  
 306 an unweighted behavioral prior and performs only a small number of policy updates with implicit  
 307 KL regularization. As a result, it may not benefit from the full potential of policy learning for many  
 308 iterations with an explicit KL regularization. Furthermore, in our experiments (Section 4), we find  
 309 that adaptive behavioral prior helps learning stability and improves asymptotic performance.

310 **Relationship to prior work with behavioral priors:** An alternate class of offline RL algorithms  
 311 incorporate conservatism to prevent over-fitting by regularizing the policy learning towards a be-  
 312 havioral prior. Some representative algorithms are BRAC [35], ABM [37], and AWR [36], which  
 313 are all model-free algorithms. Among these, BRAC uses an unweighted behavioral prior and learns  
 314 the policy using an actor-critic algorithm like SAC [39]. AWR was primarily developed for online  
 315 RL but can be re-purposed for offline RL. It is analogous to our learning of adaptive behavioral  
 316 prior, but without any RL based fine-tuning. In our ablation experiments, we find that RL finetuning  
 317 significantly improves the performance of MABE. ABM learns an adaptive behavior prior similar  
 318 to MABE, but learns the policy using the model-free MPO algorithm. In contrast, model-based  
 319 algorithms that train on a broader data distribution by incorporating synthetic model rollouts, can  
 320 unlock better generalization capabilities, including to new tasks.

321 In summary, we note that MABE presents a novel combination of MBRL and adaptive behav-  
 322 ioral priors for offline RL. Through this combination, MABE can serve as an attractive choice for  
 323 uncertainty-free offline MRBL. MABE also achieves state of the art results in benchmark offline RL  
 324 tasks, and also demonstrates strong results in transferring behaviors across different domains.

## 325 6 Broader Impacts and Limitations

326 Robust offline RL has the potential to make RL as widely applicable for decision making problems  
 327 as supervised learning is today for vision and language. Applications include domains where offline  
 328 data is ample but exploration can be harmful such as controlling autonomous vehicles, digital  
 329 assistants, and recommender systems. Negative potential impacts of MABE and RL algorithms  
 330 more generally is the lack of explainability. Since MABE is simply optimizing a reward function  
 331 while regularizing against a behavioral prior it can learn policies with undesired consequences that  
 332 exploit the reward function. Future work on explainability of RL policies as well as constrained  
 333 policy optimization could help alleviate these concerns. While we extensively evaluate our method  
 334 using D4RL benchmark tasks, and also study cross-domain transfer, our experimental evaluation  
 335 is in continuous control tasks. Although continuous control is representative of many applications  
 336 in robotics, offline RL is a broad and vibrant field with applications involving language [11, 12]  
 337 and visual modalities [19, 60, 32]. We hope to extend MABE to different offline RL tasks and  
 338 high-dimensional observation modalities in future work.

339 **References**

- 340 [1] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: datasets for  
341 deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020.
- 342 [2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G  
343 Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al.  
344 Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- 345 [3] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur  
346 Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of  
347 go without human knowledge. *Nature*, 550(7676):354–359, 2017.
- 348 [4] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Jun-  
349 young Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster  
350 level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.
- 351 [5] OpenAI et al. Solving rubik’s cube with a robot hand. *ArXiv*, abs/1910.07113, 2019.
- 352 [6] Sascha Lange, Thomas Gabel, and Martin A. Riedmiller. Batch reinforcement learning. In  
353 *Reinforcement Learning*, volume 12. Springer, 2012.
- 354 [7] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning:  
355 Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020.
- 356 [8] Dean A Pomerleau. Alvin: an autonomous land vehicle in a neural network. In *Proceedings of*  
357 *the 1st International Conference on Neural Information Processing Systems*, pages 305–313,  
358 1988.
- 359 [9] P. Abbeel and A. Ng. Apprenticeship learning via inverse reinforcement learning. *Proceedings*  
360 *of the twenty-first international conference on Machine learning*, 2004.
- 361 [10] Brian D. Ziebart, Andrew L. Maas, J. Bagnell, and A. Dey. Maximum entropy inverse reinforce-  
362 ment learning. In *AAAI*, 2008.
- 363 [11] Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza,  
364 Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement  
365 learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.
- 366 [12] Li Zhou, Kevin Small, Oleg Rokhlenko, and Charles Elkan. End-to-end offline goal-oriented  
367 dialog policy learning via policy gradient. *CoRR*, abs/1712.02838, 2017.
- 368 [13] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommenda-  
369 tions. In *RecSys*. ACM, 2016.
- 370 [14] Adith Swaminathan and Thorsten Joachims. Batch learning from logged bandit feedback  
371 through counterfactual risk minimization. *J. Mach. Learn. Res.*, 16:1731–1755, 2015.
- 372 [15] Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Off-policy policy gradient  
373 with state distribution correction. *CoRR*, abs/1904.08473, 2019.
- 374 [16] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy  
375 q-learning via bootstrapping error reduction. In Hanna M. Wallach, Hugo Larochelle, Alina  
376 Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances*  
377 *in Neural Information Processing Systems 32: Annual Conference on Neural Information*  
378 *Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages  
379 11761–11771, 2019.
- 380 [17] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning  
381 for offline reinforcement learning. In Hugo Larochelle, Marc’ Aurelio Ranzato, Raia Hadsell,  
382 Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing*  
383 *Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*  
384 *2020, December 6-12, 2020, virtual*, 2020.
- 385 [18] Scott Fujimoto, Herke Van Hoof, and David Meger. Addressing function approximation error  
386 in actor-critic methods. In *International Conference on Machine Learning*, 2018.
- 387 [19] Rishabh Agarwal, D. Schuurmans, and Mohammad Norouzi. An optimistic perspective on  
388 offline reinforcement learning. In *ICML*, 2020.

- 389 [20] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel:  
390 Model-based offline reinforcement learning. In Hugo Larochelle, Marc’ Aurelio Ranzato, Raia  
391 Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information  
392 Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020,  
393 NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- 394 [21] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, Sergey Levine, Chelsea  
395 Finn, and Tengyu Ma. MOPO: model-based offline policy optimization. In Hugo Larochelle,  
396 Marc’ Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Ad-  
397 vances in Neural Information Processing Systems 33: Annual Conference on Neural Information  
398 Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- 399 [22] Arthur Argenson and Gabriel Dulac-Arnold. Model-based offline planning. *ArXiv*,  
400 abs/2008.05556, 2020.
- 401 [23] T. Matsushima, H. Furuta, Y. Matsuo, Ofir Nachum, and Shixiang Gu. Deployment-efficient  
402 reinforcement learning via model-based offline optimization. *ArXiv*, abs/2006.03647, 2020.
- 403 [24] Yaniv Ovadia, E. Fertig, J. Ren, Zachary Nado, D. Sculley, S. Nowozin, Joshua V. Dillon, Balaji  
404 Lakshminarayanan, and Jasper Snoek. Can you trust your model’s uncertainty? evaluating  
405 predictive uncertainty under dataset shift. In *NeurIPS*, 2019.
- 406 [25] Edmon Begoli, Tanmoy Bhattacharya, and D. Kusnezov. The need for uncertainty quantification  
407 in machine-assisted medical decision making. *Nature Machine Intelligence*, 1:20–23, 2019.
- 408 [26] Heinrich Jiang, Been Kim, and M. Gupta. To trust or not to trust a classifier. In *NeurIPS*, 2018.
- 409 [27] M. Abdar, Farhad Pourpanah, Sadiq Hussain, D. Rezazadegan, Li Liu, M. Ghavamzadeh,  
410 P. Fieguth, Xiaochun Cao, A. Khosravi, U. Acharya, V. Makarenkov, and S. Nahavandi. A  
411 review of uncertainty quantification in deep learning: Techniques, applications and challenges.  
412 *ArXiv*, abs/2011.06225, 2020.
- 413 [28] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explain-  
414 ing the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International  
415 Conference on Knowledge Discovery and Data Mining*, 2016.
- 416 [29] Richard S Sutton et al. *Introduction to reinforcement learning*, volume 135. 1998.
- 417 [30] Aravind Rajeswaran, Igor Mordatch, and Vikash Kumar. A game theoretic framework for  
418 model-based reinforcement learning. In *ICML*, 2020.
- 419 [31] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-  
420 based policy optimization. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence  
421 d’ Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information  
422 Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019,  
423 NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12498–12509, 2019.
- 424 [32] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control:  
425 Learning behaviors by latent imagination. In *International Conference on Learning Representa-  
426 tions*, 2020.
- 427 [33] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Si-  
428 mon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering  
429 atari, go, chess and shogi by planning with a learned model. *arXiv preprint arXiv:1911.08265*,  
430 2019.
- 431 [34] Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement  
432 learning. In *ICML*, 2019.
- 433 [35] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement  
434 learning. *CoRR*, abs/1911.11361, 2019.
- 435 [36] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression:  
436 Simple and scalable off-policy reinforcement learning. *CoRR*, abs/1910.00177, 2019.
- 437 [37] Noah Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael  
438 Neunert, Thomas Lampe, Roland Hafner, and Martin A. Riedmiller. Keep doing what worked:  
439 Behavioral modelling priors for offline reinforcement learning. *ArXiv*, abs/2002.08396, 2020.
- 440 [38] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression:  
441 Simple and scalable off-policy reinforcement learning, 2019.

- 442 [39] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-  
443 policy maximum entropy deep reinforcement learning with a stochastic actor. In *International*  
444 *Conference on Machine Learning*, 2018.
- 445 [40] Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM*  
446 *Sigart Bulletin*, 2(4):160–163, 1991.
- 447 [41] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement  
448 learning in a handful of trials using probabilistic dynamics models. In Samy Bengio, Hanna M.  
449 Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, edi-  
450 tors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural*  
451 *Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*,  
452 pages 4759–4770, 2018.
- 453 [42] Avi Singh, Huihan Liu, Gaoyue Zhou, Albert Yu, Nicholas Rhinehart, and Sergey Levine.  
454 Parrot: Data-driven behavioral priors for reinforcement learning. In *International Conference*  
455 *on Learning Representations*, 2021.
- 456 [43] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea  
457 Finn. Combo: Conservative offline model-based policy optimization, 2021.
- 458 [44] Richard S Sutton. Integrated architectures for learning, planning, and reacting based on  
459 approximating dynamic programming. In *Machine learning proceedings 1990*, pages 216–224.  
460 Elsevier, 1990.
- 461 [45] Richard S Sutton. Planning by incremental dynamic programming. In *Machine Learning*  
462 *Proceedings 1991*, pages 353–357. Elsevier, 1991.
- 463 [46] Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble  
464 trust-region policy optimization. In *International Conference on Learning Representations*,  
465 2018.
- 466 [47] H. Rosenbrock, D. Jacobson, and D. Mayne. Differential dynamic programming. *The Mathe-*  
467 *matical Gazette*, 56:78, 1972.
- 468 [48] Marc Peter Deisenroth and Carl Edward Rasmussen. PILCO: A model-based and data-efficient  
469 approach to policy search. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th*  
470 *International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June*  
471 *28 - July 2, 2011*, pages 465–472. Omnipress, 2011.
- 472 [49] N. Heess, Greg Wayne, D. Silver, T. Lillicrap, T. Erez, and Yuval Tassa. Learning continuous  
473 control policies by stochastic value gradients. In *NIPS*, 2015.
- 474 [50] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behav-  
475 iors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on*  
476 *Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*,  
477 pages 4906–4913. IEEE, 2012.
- 478 [51] E. Todorov and W. Li. A generalized iterative lqg method for locally-optimal feedback control  
479 of constrained nonlinear stochastic systems. *Proceedings of the 2005, American Control*  
480 *Conference, 2005.*, pages 300–306 vol. 1, 2005.
- 481 [52] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee,  
482 and James Davidson. Learning latent dynamics for planning from pixels. In *International*  
483 *Conference on Machine Learning*, 2019.
- 484 [53] Grady Williams, Nolan Wagener, Brian Goldfain, P. Drews, James M. Rehg, Byron Boots, and  
485 E. Theodorou. Information theoretic mpc for model-based reinforcement learning. *2017 IEEE*  
486 *International Conference on Robotics and Automation (ICRA)*, pages 1714–1721, 2017.
- 487 [54] Anusha Nagabandi, K. Konolige, Sergey Levine, and V. Kumar. Deep dynamics models for  
488 learning dexterous manipulation. *ArXiv*, abs/1909.11652, 2019.
- 489 [55] Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch.  
490 Plan Online, Learn Offline: Efficient Learning and Exploration via Model-Based Control. In  
491 *International Conference on Learning Representations (ICLR)*, 2019.
- 492 [56] Yao Liu, A. Swaminathan, A. Agarwal, and Emma Brunskill. Provably good batch reinforcement  
493 learning without great exploration. *ArXiv*, abs/2007.08202, 2020.

- 494 [57] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning  
 495 environment: An evaluation platform for general agents. *Journal of Artificial Intelligence*  
 496 *Research*, 47:253–279, 2013.
- 497 [58] Caglar Gulcehre, Ziyu Wang, Alexander Novikov, T. Paine, Sergio Gomez Colmenarejo, Konrad  
 498 Zolna, Rishabh Agarwal, J. Merel, Daniel J. Mankowitz, Cosmin Paduraru, Gabriel Dulac-  
 499 Arnold, J. Li, Mohammad Norouzi, Matthew W. Hoffman, Ofir Nachum, G. Tucker, N. Heess,  
 500 and N. D. Freitas. RL unplugged: A suite of benchmarks for offline reinforcement learning.  
 501 2020.
- 502 [59] Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, R. Munos, N. Heess, and Martin A.  
 503 Riedmiller. Maximum a posteriori policy optimisation. *ArXiv*, abs/1806.06920, 2018.
- 504 [60] Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. Offline reinforcement  
 505 learning from images with latent space models. In *L4DC*, 2021.

## 506 Checklist

- 507 1. For all authors...
- 508 (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s  
 509 contributions and scope? [Yes]
- 510 (b) Did you describe the limitations of your work? [Yes] , See section 6 for discussion of  
 511 limitations of our work.
- 512 (c) Did you discuss any potential negative societal impacts of your work? [Yes] , See  
 513 section 6 for discussion of broader impacts.
- 514 (d) Have you read the ethics review guidelines and ensured that your paper conforms to  
 515 them? [Yes]
- 516 2. If you are including theoretical results...
- 517 (a) Did you state the full set of assumptions of all theoretical results? [N/A]
- 518 (b) Did you include complete proofs of all theoretical results? [N/A]
- 519 3. If you ran experiments...
- 520 (a) Did you include the code, data, and instructions needed to reproduce the main ex-  
 521 perimental results (either in the supplemental material or as a URL)? [Yes] , See  
 522 supplemental material.
- 523 (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they  
 524 were chosen)? [Yes] , See appendix for all training details.
- 525 (c) Did you report error bars (e.g., with respect to the random seed after running experi-  
 526 ments multiple times)? [Yes] , See Section 1 for reported standard deviations.
- 527 (d) Did you include the total amount of compute and the type of resources used (e.g., type  
 528 of GPUs, internal cluster, or cloud provider)? [Yes] , see appendix for all training  
 529 details.
- 530 4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
- 531 (a) If your work uses existing assets, did you cite the creators? [Yes] , See section 4 for all  
 532 citations.
- 533 (b) Did you mention the license of the assets? [Yes] , see appendix.
- 534 (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]  
 535 , see supplemental material.
- 536 (d) Did you discuss whether and how consent was obtained from people whose data you’re  
 537 using/curating? [Yes] , see appendix for discussion of data.
- 538 (e) Did you discuss whether the data you are using/curating contains personally identifiable  
 539 information or offensive content? [Yes] , see appendix for discussion of data.
- 540 5. If you used crowdsourcing or conducted research with human subjects...
- 541 (a) Did you include the full text of instructions given to participants and screenshots, if  
 542 applicable? [N/A]
- 543 (b) Did you describe any potential participant risks, with links to Institutional Review  
 544 Board (IRB) approvals, if applicable? [N/A]



545  
546

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