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

Anonymous Author(s)
Affiliation
Address
email

Abstract

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

1 Introduction

Over the last five years advances in Deep Reinforcement Learning (RL) have been at the source of a number of impressive results in autonomous control, including the ability to solve video games from pixels [2], master the game of Go [3], play multi-agent large scale video games [4], and control robots [5]. Most advances in RL were achieved in simulated environments where data was cheap to collect and mistakes during policy training were harmless. However, two substantial problems stand in the way from utilizing the above approaches to deploy RL algorithms in real-world settings. First, since RL algorithms require millions and sometimes billions of environment interactions, learning policies with RL in the real world is costly in terms of time and resources. Second, since RL algorithms stochastically explore their environment, the resulting agents are not safe and can harm the environment, themselves, or other

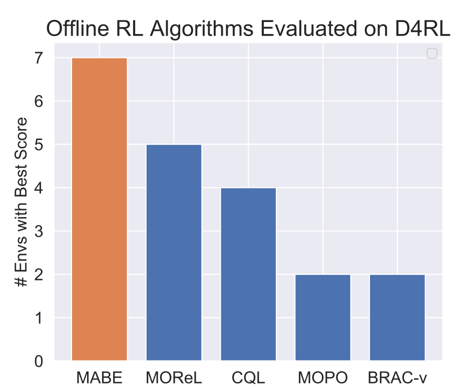

Figure 1: Our proposed algorithm, MABE, when compared to prior work, achieves the top score in **7 out of 9** D4RL datasets [1] we study. We consider multiple algorithms to achieve the top score if they are within 2% points of each other.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

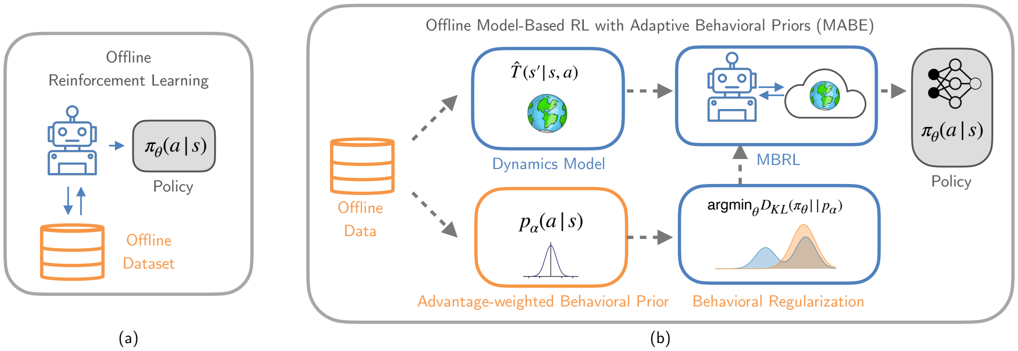

Figure 2: (a) The offline RL paradigm. Rather than interacting with the environment directly, an agent extracts a policy from an offline dataset. (b) A schematic architecture of our proposed algorithm (MABE). First, by interaction with the the offline dataset, the agent learns a dynamics model and an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

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

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

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

Our Contribution Our principal contribution in this work is the development of a new algorithm – offline model-based RL with Adaptive Behavioral Regularization (MABE). Using the offline dataset, MABE learns an approximate dynamics model, reward function, as well as an adaptive behavioral prior. By adaptive behavioral prior, we mean a policy that approximates the behavior in the offline dataset while giving more importance to trajectories with high rewards. Using the learned dynamics model and reward function, MABE performs model-based RL with an objective to maximize the rewards along with a KL-divergence penalty that encourages the agent to stay close to adaptive behavioral prior. This divergence penalty provides the necessary conservatism needed to succeed in offline RL. Our major findings in this work are listed below.

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

2 Preliminaries

We operate in the standard RL setting of infinite horizon discounted Markov Decision Process (MDPs), defined as the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, T, \rho_0, \gamma)$. The MDP tuple has states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, rewards $r = R(s, a)$, transition dynamics $s' \sim T(\cdot|s, a)$, an initial state distribution $s_0 \sim \rho_0(\cdot)$, and a discount factor $\gamma \in [0, 1)$. A policy defines a mapping from states to actions, typically in the form of a probability distribution: $a \sim \pi(\cdot|s)$. The value $V^\pi(s)$ and action-value function $Q^\pi(s, a)$ describe the long term reward behavior of policy π.

$$V^\pi(s) := \mathbb{E}_{\mathcal{M},\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s\right], \quad Q^\pi(s, a) := R(s, a) + \gamma \mathbb{E}_{s' \sim T(\cdot|s,a)}\left[V^\pi(s')\right]$$

where the first expectation $\mathbb{E}_{\mathcal{M},\pi}$ denotes actions are sampled according to π and future states are sampled according to the MDP dynamics $T(\cdot|s, a)$. The goal in RL is the learn the optimal policy:

$$\pi^* \in \arg\max_\pi \; J(\pi, \mathcal{M}) := \mathbb{E}_{s \sim \rho_0}\left[V^\pi(s)\right]. \tag{1}$$

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

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

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

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

3 Model-Based Offline RL with Adaptive Behavioral Regularization

Given an offline dataset $\mathcal{D}$, our goal is to learn a parameterized policy π_θ that achieves high rewards, without any additional interaction with the environment. We assume $\mathcal{D}$ consists of $\{s, a, s', r\}$ tuples which we use to learn $\hat{T}$ along with a behavioral prior $p_\alpha(a_t|s_t)$. This dataset can be collected using one or more structured behavioral policies interacting with test environment. We now present our algorithm MABE (Model-Based Offline RL with Adaptive Behavioral Regularization), which consists of three components described below.

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

$$\max_{\hat{T}_\psi(\cdot|s,a)} \ \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\log\left(\hat{T}_\psi(s'|s,a)\right)\right]. \tag{2}$$

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

We consider a parameterized generative model $p_\alpha(a|s)$ that represents our behavioral prior. A straightforward option is to learn a behavior model that replicates the statistics in the dataset.

$$p_\alpha^{\text{eq}} \ \in \ \arg\max_{p_\alpha} \ \mathbb{E}_{\tau\sim\mathcal{D}}\left[\sum_{t=0}^{|\tau|}\log\left(p_\alpha(a_t|s_t)\right)\right] \tag{3}$$

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

$$\begin{aligned} p_\alpha^{\text{adapt}} \ &\in \arg\max_{p_\alpha} \ \mathbb{E}_{\tau\sim\mathcal{D}}\left[\sum_{t=0}^{|\tau|}\omega(s_t,a_t)\cdot p_\alpha(a_t|s_t)\right] \\ &\text{subject to } \ \mathbb{E}_{s\sim\mathcal{D}}\left[D_{KL}\left(p_\alpha \| \bar{p}\right)\right] \le \delta, \end{aligned} \tag{4}$$

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

$$p_\alpha^{\text{adapt}}(a_t|s_t) \propto \bar{p}(a_t|s_t)\cdot\exp\left(\omega(t,\tau)/\eta\right),$$

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

$$p_\alpha^{\text{adapt}} \in \ \arg\max_{p_\alpha} \ \mathbb{E}_{\tau\sim\mathcal{D}}\left[\sum_{t=0}^{|\tau|}\exp\left(\omega(s_t,a_t)/\eta\right)\cdot\log\left(p_\alpha(a_t|s_t)\right)\right]. \tag{5}$$

For the choice of the weighting function, we use

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

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

Behavior Regularized Model-Based RL Equipped with a dynamics model and adaptive behavioral prior, our algorithm MABE, performs model-based RL with a regularized objective given by:

$$\max_{\pi_\theta} \underset{(s,a)\sim\rho^{\pi_\theta}_{\hat{\mathcal{M}}}}{\mathbb{E}} \left[\tilde{r}(s,a)\right], \quad \text{with } \tilde{r}(s,a) := \hat{r}_\psi(s,a) - \beta \left(\log \pi_\theta(a|s) + \log p_\alpha(a|s)\right). \tag{6}$$

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

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

---

Inputs: Offline dataset $\mathcal{D}$, learned dynamics and reward models $\hat{T}_\psi$ and $\hat{r}_\psi$, adaptive behavioral prior $p_\alpha(a_t|s_t)$, target divergence δ, learning rates λ_π, λ_Q, λ_β, λ_ϕ, rollout length h
Initialize policy π_θ, critic Q_ϕ, target network $Q_{\bar{\phi}}$, and $\mathcal{D}_{\text{aug}} = \mathcal{D}$
for N epochs **do**
    for K trajectories **do**
        Collect rollouts (s_t, a_t, r_t, s_{t+1}) of length h using $\hat{T}_\psi$ and $\hat{r}_\psi$ starting from a randomly chosen state from the offline dataset.
        $\mathcal{D}_{\text{aug}} \leftarrow \mathcal{D}_{\text{aug}} \cup (s_t, a_t, r_t, s_{t+1})$
    for each gradient step **do**
        Sample a batch of (s_t, a_t, r_t, s'_t) tuples from $\mathcal{D}_{\text{aug}}$
        $\bar{Q} = r(s_t, a_t) + \gamma[Q_{\bar{\phi}}(s'_t, \pi_\theta(\cdot \mid s'_t)) - \beta D_{\mathrm{KL}}(\pi_\theta(\cdot \mid s'_t), p_\alpha(\cdot \mid s'_t))]$
        $\theta \leftarrow \theta - \lambda_\pi \nabla_\theta \left[Q_\phi(s_t, \pi_\theta(\cdot \mid s_t)) - \beta D_{\mathrm{KL}}(\pi_\theta(\cdot \mid s_t), p_\alpha(\cdot \mid s_t))\right]$
        $\phi \leftarrow \phi - \lambda_Q \nabla_\phi \left[\frac{1}{2}(Q_\phi(s_t, a_t) - \bar{Q})^2\right]$
        $\beta \leftarrow \beta - \lambda_\beta \cdot (D_{\mathrm{KL}}(\pi_\theta(\cdot \mid s_t), p_\alpha(\cdot \mid s_t)) - \delta)$
        $\bar{\phi} \leftarrow \lambda_\phi \phi + (1 - \lambda_\phi)\bar{\phi}$

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

$$\tilde{r}(s,a) = \hat{r}_\psi(s,a) - \beta \left(\log \pi_\theta(a|s) + \log p_\alpha(a|s)\right) - \xi u(s,a).$$

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

4 Results

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

$$\hat{T}_\psi(s'|s,a) = \mathcal{N}\left(\mathrm{MLP}_\psi(s,a),\ \Sigma_\psi\right).$$

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

Experiments in D4RL offline RL benchmark tasks Our first goal is to study the performance of MABE on the widely studied D4RL [1] benchmark. We consider a total of nine domains involving three simulated locomotion tasks and three datasets per task: medium, medium-replay (or mixed), and medium-expert. The medium dataset is collected with partially trained SAC agent, the mixed dataset is the entire replay buffer of a SAC agent throughout training, and the medium-expert is a mix between trajectories from the medium dataset and an expert policy. These represent three distinct types of imperfect data - one imperfect policy, many changing policies, and a mixture of expert and suboptimal policies respectively. We compare our method to published leading offline RL algorithms which include: (a) MOReL [20] and MOPO [21] – model-based algorithms that rely on uncertainty quantification; (b) CQL [17], a model-free algorithm that learns a conservative Q-function, and (c) BRAC-v [35], which regularizes a model-free actor-critic algorithm with an unweighted (or equally-weighted) behavioral prior. Please see appendix for more details.

Evaluation scores on D4RL are shown in Table 1. We find that MABE achieves the highest score on the majority (7 out of 9) environments as well as the highest average score of 77.5. Crucially, MABE's performance is robust across the three dataset types, achieving a leading score on at least 2 out 3 environments for each dataset. Finally, we note that MABE substantially outperforms its the two most directly competing baselines: MOPO, an uncertianty-based MBRL method; and BRAC-v, a model-free method with explicit behavioral prior regularization. This suggests that a combination of 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	**46.8** $\pm$ 0.8	42.3 $\pm$ 1.6	42.1	-4.3	44.4	45.5
medium	hopper	29.0	**94.1** $\pm$ 5.8	28.0 $\pm$ 12.4	**95.4**	0.8	58.0	32.3
medium	walker2d	6.6	65.7 $\pm$ 8.5	17.8 $\pm$ 19.3	77.8	0.9	79.2	**81.3**
med-replay	halfcheetah	38.4	**53.5** $\pm$ 0.5	**53.1** $\pm$ 2.0	40.2	-2.4	46.2	45.9
med-replay	hopper	11.8	71.7 $\pm$ 12.5	67.5 $\pm$ 24.7	**93.6**	1.9	48.6	0.9
med-replay	walker2d	11.3	**51.0** $\pm$ 2.4	39.0 $\pm$ 9.6	49.8	3.5	26.7	0.8
med-expert	halfcheetah	35.8	**100.6** $\pm$ 1.3	63.3 $\pm$ 38.0	53.3	1.8	62.4	45.3
med-expert	hopper	111.9	**110.5** $\pm$ 0.8	23.7 $\pm$ 6.0	108.7	1.6	**111.0**	0.8
med-expert	walker2d	6.4	**103.3** $\pm$ 1.3	44.6 $\pm$ 12.9	95.6	-0.1	98.7	66.6
Average	Average	31.7	**77.5**	42.1	72.9	0.4	63.9	35.5

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

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: (a) adaptive behavioral prior regularization; (b) policy learning (improvement) using model-based RL, and (b) the optional use of uncertainty quantification through model ensembles [41, 30, 20, 21] to

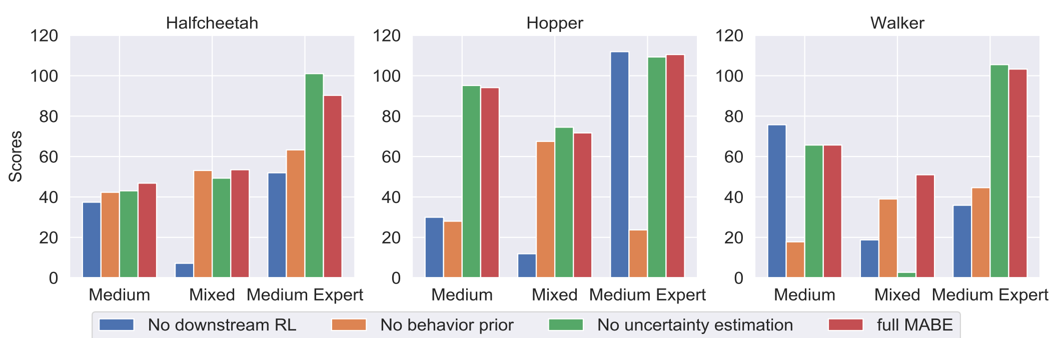

Figure 3: Ablation over the three components of MABE. In most environments, the best performance is achieved from having all three components of MABE, but the component that gives the biggest boost in performance on average is the behavior prior. In most environments, uncertainty estimation does not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

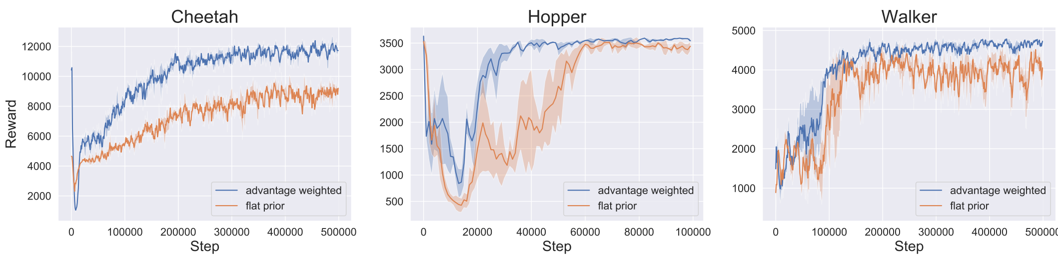

Figure 4: Comparison of MABE run with an advantage weighted behavioral as well as a non-weighted “flat" prior. We find that advantage-weighing improves the stability and performance of the policy.

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

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

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

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

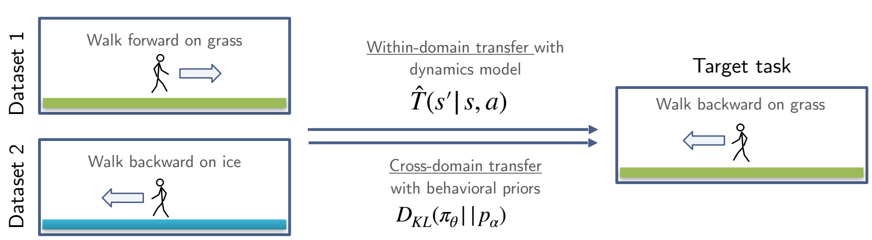

Figure 5: A schematic visualization of MABE's domain transfer capabilities. In prior work it was shown that offline MBRL is capable of in-domain generalization to new tasks [21]. Here, we investigate cross-domain transfer capabilities of MABE and MOPO. Given multiple datasets with different dynamics 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.

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

To test the generalization capabilities of MABE, we setup the following simple experiment. We use simulated locomotion agents (Hopper, Walker, HalfCheetah), and collect two datasets: $\mathcal{D}_1$ containing medium-replay forward walking data in normal terrain; and $\mathcal{D}_2$ containing expert backwards walking data in low friction terrain intended to simulate ice. In this behavior transfer test, we use these two datasets to train an agent to run backward on normal terrain. A schematic illustration of our setting can be found in Figure 5. In our experiments, we consider the following approaches: (i) *task transfer only* where we use the forwards walking dataset to learn a backwards walking policy using offline MBRL. (ii) *domain transfer only* where we train a policy in source domain and directly deploy it in the target domain. (iii) *task transfer with behavior initialization* where we initialize the task transfer approach with the adaptive behavioral prior; (iv) *task + domain transfer with MABE* where we run MABE using the dataset corresponding to the target dynamics $(\mathcal{D}_1)$ and behavioral prior corresponding to the desired behavior $(\mathcal{D}_2)$. We show the resulting expert normalized scores in Figure 6 and find that MABE is the only algorithm that is able to successfully solve the target task through cross-domain behavior transfer. This suggests that dynamics models and behavioral priors are complementary and can be used to acquire a wide range of behaviors from offline data using domain and task transfer.

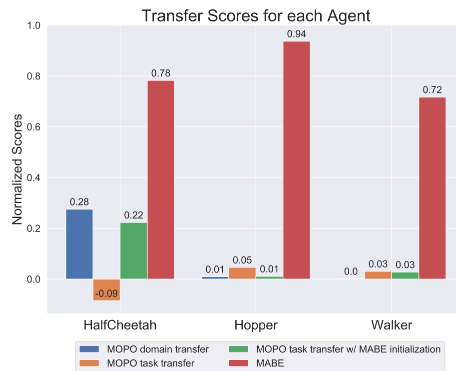

Figure 6: Experiments comparing MABE domain transfer to MOPO domain transfer and MOPO task transfer. MABE is the only offline MBRL approach that is able to successfully transfer behaviors across domains.

5 Related Work

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

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

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

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

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

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

6 Broader Impacts and Limitations

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

References

[1] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

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

[4] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[5] OpenAI et al. Solving rubik's cube with a robot hand. *ArXiv*, abs/1910.07113, 2019.

[6] Sascha Lange, Thomas Gabel, and Martin A. Riedmiller. Batch reinforcement learning. In *Reinforcement Learning*, volume 12. Springer, 2012.

[7] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020.

[8] Dean A Pomerleau. Alvinn: an autonomous land vehicle in a neural network. In *Proceedings of the 1st International Conference on Neural Information Processing Systems*, pages 305–313, 1988.

[9] P. Abbeel and A. Ng. Apprenticeship learning via inverse reinforcement learning. *Proceedings of the twenty-first international conference on Machine learning*, 2004.

[10] Brian D. Ziebart, Andrew L. Maas, J. Bagnell, and A. Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, 2008.

[11] Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

[12] Li Zhou, Kevin Small, Oleg Rokhlenko, and Charles Elkan. End-to-end offline goal-oriented dialog policy learning via policy gradient. *CoRR*, abs/1712.02838, 2017.

[13] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *RecSys*. ACM, 2016.

[14] Adith Swaminathan and Thorsten Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *J. Mach. Learn. Res*, 16:1731–1755, 2015.

[15] Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Off-policy policy gradient with state distribution correction. *CoRR*, abs/1904.08473, 2019.

[16] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11761–11771, 2019.

[17] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[18] Scott Fujimoto, Herke Van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, 2018.

[19] Rishabh Agarwal, D. Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *ICML*, 2020.

[20] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[21] 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 Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[22] Arthur Argenson and Gabriel Dulac-Arnold. Model-based offline planning. *ArXiv*, abs/2008.05556, 2020.

[23] T. Matsushima, H. Furuta, Y. Matsuo, Ofir Nachum, and Shixiang Gu. Deployment-efficient reinforcement learning via model-based offline optimization. *ArXiv*, abs/2006.03647, 2020.

[24] Yaniv Ovadia, E. Fertig, J. Ren, Zachary Nado, D. Sculley, S. Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019.

[25] Edmon Begoli, Tanmoy Bhattacharya, and D. Kusnezov. The need for uncertainty quantification in machine-assisted medical decision making. *Nature Machine Intelligence*, 1:20–23, 2019.

[26] Heinrich Jiang, Been Kim, and M. Gupta. To trust or not to trust a classifier. In *NeurIPS*, 2018.

[27] M. Abdar, Farhad Pourpanah, Sadiq Hussain, D. Rezazadegan, Li Liu, M. Ghavamzadeh, P. Fieguth, Xiaochun Cao, A. Khosravi, U. Acharya, V. Makarenkov, and S. Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *ArXiv*, abs/2011.06225, 2020.

[28] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[29] Richard S Sutton et al. *Introduction to reinforcement learning*, volume 135. 1998.

[30] Aravind Rajeswaran, Igor Mordatch, and Vikash Kumar. A game theoretic framework for model-based reinforcement learning. In *ICML*, 2020.

[31] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12498–12509, 2019.

[32] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

[33] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *arXiv preprint arXiv:1911.08265*, 2019.

[34] Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *ICML*, 2019.

[35] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *CoRR*, abs/1911.11361, 2019.

[36] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *CoRR*, abs/1910.00177, 2019.

[37] Noah Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, and Martin A. Riedmiller. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *ArXiv*, abs/2002.08396, 2020.

[38] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019.

[39] 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*, 2018.

[40] Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

[41] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 4759–4770, 2018.

[42] Avi Singh, Huihan Liu, Gaoyue Zhou, Albert Yu, Nicholas Rhinehart, and Sergey Levine. Parrot: Data-driven behavioral priors for reinforcement learning. In *International Conference on Learning Representations*, 2021.

[43] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization, 2021.

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

[45] Richard S Sutton. Planning by incremental dynamic programming. In *Machine Learning Proceedings 1991*, pages 353–357. Elsevier, 1991.

[46] Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. In *International Conference on Learning Representations*, 2018.

[47] H. Rosenbrock, D. Jacobson, and D. Mayne. Differential dynamic programming. *The Mathematical Gazette*, 56:78, 1972.

[48] Marc Peter Deisenroth and Carl Edward Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 465–472. Omnipress, 2011.

[49] N. Heess, Greg Wayne, D. Silver, T. Lillicrap, T. Erez, and Yuval Tassa. Learning continuous control policies by stochastic value gradients. In *NIPS*, 2015.

[50] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, pages 4906–4913. IEEE, 2012.

[51] E. Todorov and W. Li. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. *Proceedings of the 2005, American Control Conference, 2005.*, pages 300–306 vol. 1, 2005.

[52] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, 2019.

[53] Grady Williams, Nolan Wagener, Brian Goldfain, P. Drews, James M. Rehg, Byron Boots, and E. Theodorou. Information theoretic mpc for model-based reinforcement learning. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1714–1721, 2017.

[54] Anusha Nagabandi, K. Konolige, Sergey Levine, and V. Kumar. Deep dynamics models for learning dexterous manipulation. *ArXiv*, abs/1909.11652, 2019.

[55] Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch. Plan Online, Learn Offline: Efficient Learning and Exploration via Model-Based Control. In *International Conference on Learning Representations (ICLR)*, 2019.

[56] Yao Liu, A. Swaminathan, A. Agarwal, and Emma Brunskill. Provably good batch reinforcement learning without great exploration. *ArXiv*, abs/2007.08202, 2020.

[57] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[58] Caglar Gulcehre, Ziyu Wang, Alexander Novikov, T. Paine, Sergio Gomez Colmenarejo, Konrad Zolna, Rishabh Agarwal, J. Merel, Daniel J. Mankowitz, Cosmin Paduraru, Gabriel Dulac-Arnold, J. Li, Mohammad Norouzi, Matthew W. Hoffman, Ofir Nachum, G. Tucker, N. Heess, and N. D. Freitas. Rl unplugged: A suite of benchmarks for offline reinforcement learning. 2020.

[59] Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, R. Munos, N. Heess, and Martin A. Riedmiller. Maximum a posteriori policy optimisation. *ArXiv*, abs/1806.06920, 2018.

[60] Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. Offline reinforcement learning from images with latent space models. In *L4DC*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] , See section 6 for discussion of limitations of our work.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] , See section 6 for discussion of broader impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] , See supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] , See appendix for all training details.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] , See Section 1 for reported standard deviations.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] , see appendix for all training details.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] , See section 4 for all citations.
   (b) Did you mention the license of the assets? [Yes] , see appendix.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] , see supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] , see appendix for discussion of data.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] , see appendix for discussion of data.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

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