MOTO: Offline to Online Fine-tuning for Model-Based Reinforcement Learning

Anonymous Authors[1]

Abstract

We study the problem of offline-to-online reinforcement learning from high-dimensional pixel observations. While recent model-free approaches successfully use offline pre-training with online fine-tuning to either improve the performance of the data-collection policy or adapt to novel tasks, model-based approaches still remain underutilized in this setting. In this work, we argue that existing methods for high-dimensional model-based offline RL are not suitable for offline-to-online fine-tuning due to issues with representation learning shifts, off-dynamics data, and non-stationary rewards. We propose a simple on-policy model-based method with adaptive behavior regularization. In our simulation experiments, we find that our approach successfully solves long-horizon robot manipulation tasks completely from images by using a combination of offline data and online interactions.

1. Introduction

Offline reinforcement learning (Lange et al., 2012; Levine et al., 2020) is a promising approach for scaling reinforcement learning (RL) to broad datasets, and can serve as a pre-training step for efficiently learning a new task. In such an offline-to-online fine-tuning problem, the agent is provided with a static, pre-collected dataset and is tasked with leveraging this dataset and a small amount of online interaction to effectively solve a task. We study this problem with a focus on settings with high-dimensional pixel observations, as is common in real-world applications such as robotics. Prior works for offline-to-online fine-tuning often simply train a policy with model-free offline RL objectives throughout both the offline and online phases (Kostrikov et al., 2021; Kumar et al., 2020; Nair et al., 2020; Yang & Nachum, 2021; Chen et al., 2021; Reed et al., 2022). While this approach addresses the challenge of distribution shift in the offline phase, it leads to excessive conservatism in the online phase.

An alternative to prior model-free algorithms are model-based methods, in which the agent learns a representation and dynamics model from the provided offline dataset and uses model-generated rollouts for policy training or planning (Yu et al., 2020; Kidambi et al., 2020; Matsushima et al., 2020; Argenson & Dulac-Arnold, 2020; Yu et al., 2021c; Cang et al., 2021; Rafailov et al., 2020). These approaches have shown good performance on more diverse datasets and can also generalize to new within-distribution tasks (Yu et al., 2020; 2021c; Cang et al., 2021; Rafailov et al., 2021). Predictive models can also naturally learn stable representations, which makes them suitable for use in realistic high-dimensional domains (Hu et al., 2022; 2021; Akan & Güney, 2022; Rafailov et al., 2020). However, the pre-training and fine-tuning approaches for model-based RL are still under-explored, as the literature has mostly focused on model-free methods.

In this work, we argue that existing algorithms for offline model-based RL are not suitable to the pre-training and fine-tuning or continual learning regimes. In particular, algorithms that use replay buffers of model-generated data, such as (Yu et al., 2020; 2021c; Cang et al., 2021; Rafailov et al., 2020), create significant distributional shift issues, as the learned model dynamics and reward functions change with additional online interactions. Moreover, models with high-dimensional observations, such as (Rafailov et al., 2020; Yu et al., 2021c), deal with the additional complexity of representation shift of the latent data. These algorithms are also not feasible in large models with high-dimensional representation spaces, which are common in real-world applications (Hu et al., 2022; 2021; Akan & Güney, 2022). On the other hand, on-policy model-based RL methods such as (Kidambi et al., 2020; Matsushima et al., 2020) are amenable to fine-tuning but do not make efficient use of high-quality data in the policy training objective or are not scalable to models with changing representation spaces.

To alleviate these issues, we propose MOTO (**M**odel-based **O**ffline-**T**o-**O**nline) algorithm. MOTO is a model-based actor-critic algorithm which operates in high-dimensional observation spaces. Crucially, MOTO uses model-based

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the Reincarnating RL Workshop at ICLR 2023.

value expansion, which removes the need for large replay buffers, alleviates the distributional shift issue, and allows for the use of large-scale predictive models while still allowing us to use high-quality offline data in the critic learning. To prevent model exploitation, we additionally implement ensemble model-based uncertainty estimation and policy regularization. We evaluate MOTO on two tasks in the Franka Kitchen domain (Gupta et al., 2019; Fu et al., 2020), completely from vision. Our approach solves both tasks and, as far as we are aware, is the first method to solve this environment completely from vision. Moreover, by studying the fine-tuning regime, we empirically validate theoretical performance bounds from prior model-based offline RL, which have import implications for multi-task and transfer learning.

2. Related Work

Our work is at the intersection of offline RL, model-based RL and control from high-dimensional observations (i.e. images). We review related work from these fields below.

2.1. Model-Based Offline RL

Model-based offline RL algorithms (Kidambi et al., 2020; Yu et al., 2020; Argenson & Dulac-Arnold, 2020; Matsushima et al., 2020; Swazinna et al., 2020; Rafailov et al., 2020; Yu et al., 2021c) learn a predictive model from the offline dataset and use it for policy training. Several works (Yu et al., 2020; Kidambi et al., 2020; Cang et al., 2021; Rafailov et al., 2020) train ensemble of dynamics models and use model disagreement as a measure of model uncertainty. The RL policies are optimized by explicitly incorporating disagreement into the expected return, which encourages the agent to remain with the part of the state-action distribution with low model epistemic uncertainty. Another major approach is incorporating explicit data constraint into the policy optimization, such as (Matsushima et al., 2020; Argenson & Dulac-Arnold, 2020; Swazinna et al., 2020; Cang et al., 2021). The COMBO approach (Yu et al., 2021c) uses a single model to generate data and combines model-based RL with off-policy conservative optimization (Kumar et al., 2020). With the exception of (Rafailov et al., 2020) and limited experiments in (Yu et al., 2021c) the above works mostly focus on simple low-dimensional control problems with short planning horizons.

2.2. Variational Dynamics Models

Variational predictive models have demonstrated success in a variety of challenging applications. One line of research (Hu et al., 2022; 2021; Watter et al., 2015; Zhang et al., 2019; Lee et al., 2020a) utilizes the model for representation purposes only and use standard RL, control or imitation on top of it. Others such as (Hafner et al., 2019; 2020a;b; Ha & Schmidhuber, 2018) use the latent dynamics model either to learn a policy within the model or deploy shooting-based planning methods. However, most of those prior works focus on the online setting and do not make good use of highly-structured prior data or account for distribution shift. Our method utilizes model-based value expansion, which allows us to take advantage of the efficiency of model-based training, while also using offline data for critic supervision.

3. Preliminaries

We briefly review variational dynamics models and their prior use in offline RL.

3.1. Control As Inference

Our setting fits well within the control as inference problem (Levine, 2018; Lee et al., 2020b) in the standard POMDP setup. Let $M = (\mathcal{X}, \mathcal{S}, \mathcal{A}, T, p, r, \mu_0, \gamma)$ where $\mathcal{X}$ denotes the the high-dimensional observation space, $\mathcal{S}$ the unobserved state space, $\mathcal{A}$ the action space, $T(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$ the latent transition distribution, $p(\boldsymbol{x}|\boldsymbol{s})$ the observation model, $r(\boldsymbol{s}, \boldsymbol{a})$ the reward function, $\mu_0(\boldsymbol{s}_0)$ the initial latent state distribution, and $\gamma \in (0, 1)$ the discount factor. In this setting, the control problem is equivalent to an inference problem in a PGM with a binary random variable $p(\mathcal{O}_t = 1|\boldsymbol{s}_t, \boldsymbol{a}_t) \propto \exp(r(\boldsymbol{s}_t, \boldsymbol{a}_t))$, which depends on the latent state and action and indicates whether the agent behaviour is optimal or not. Under a suitable choice for a variational distribution, we can then optimize the representation and control problem jointly via the ELBO:

$$\log p_\theta(\boldsymbol{x}_{1:\tau+1}, \mathcal{O}_{\tau+1:T}|\boldsymbol{a}_{1:\tau}) \geqslant \mathbb{E}_{q_\theta}\Big[\underbrace{\sum_{t=1}^{\tau} \log p_\theta(\boldsymbol{x}_t|\boldsymbol{s}_t)}_{\text{reconstruction}} - \underbrace{\mathbb{D}_{KL}(q_\theta(\boldsymbol{s}_t|\boldsymbol{x}_t, \boldsymbol{s}_{t-1}, \boldsymbol{a}_{t-1})||T_\theta(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}, \boldsymbol{a}_{t-1}))}_{\text{latent forward model}} + \sum_{t=\tau}^{T} \underbrace{r(\boldsymbol{s}_t, \boldsymbol{a}_t) + \log p(\boldsymbol{a}_t) - \log \pi_\psi(\boldsymbol{a}_t|\boldsymbol{s}_t)}_{\text{policy optimisation}}\Big] \quad (1)$$

This choice of factorization has several nice properties: 1) the policy depends only on the belief state and can be optimized entirely within the model latent space, including model-based roll-outs and 2) the learned belief space represents an MDP and we can apply theoretical insights from the low-dimensional model-based literature to our problem. There is a rich literature on learning variational models, but we base our design implementation on the recurrent state-space model of (Hafner et al., 2020a) as it has demonstrated success across a variety of domains.

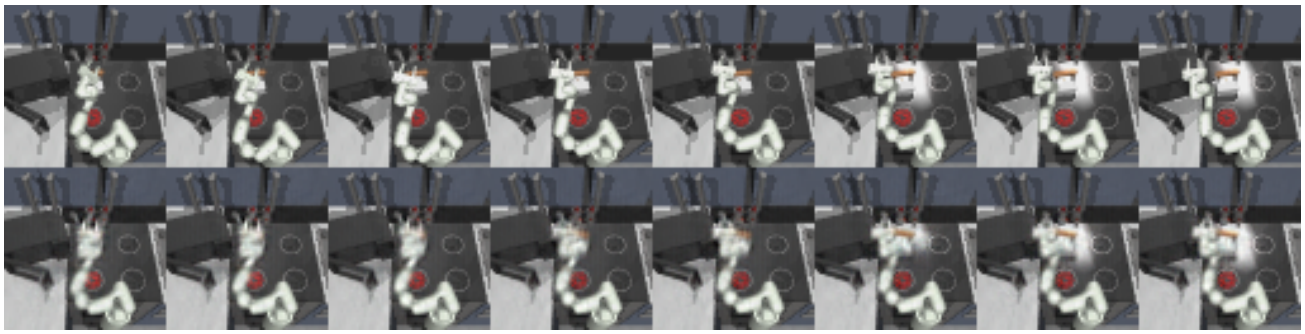

Figure 1. We evaluate the trained predictive model at the end of the offline pre-training phase on the "partial" task. We samle episodes from the trained exert agent and condition the model on the completion of the first three tasks (which are seen in the offline dataset), after which we rollout the expert actions. The model predicts the microwave, kettle, bottom burner and light switch in the correct configuration, even though the training dataset does not contain a configuration with those four objects.

3.2. Offline Model-Based RL From High-Dimensional Observations

Previous works have considered conservative-based models for offline RL from high-dimensional observations (Rafailov et al., 2020; Yu et al., 2021c). These approaches consider an MBPO-style approach (Janner et al., 2019) in the latent space of a variational model. In particular, they first train a variational dynamics model using Eq. 1 once from the available offline dataset. Rafailov et al. (2020) leverages the approach from Yu et al. (2020) and trains an ensemble of latent models $\{T_{\theta^i}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)\}_{i=1}^M$ by randomly selecting an ensemble member at each time step to optimize in Eq. 1. During policy training it maintains a latent replay buffer which consist of real transitions sampled from the inference model q_θ as well as off-policy model-generated data from the latent policy rollouts. Similarly to MOPO model-based conservatism is implemented by penalizing the model-sampled rewards using a measure of model-disagreement. All experiments in this work use the penalty

$$u_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) = \text{std}(\{l_{\theta^i}(\boldsymbol{s}_t, \boldsymbol{a}_t)\}_{i=1}^M) \tag{2}$$

where $l_\theta^i(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is the logit outputs of the discrete distribution $T_{\theta^i}(\cdot|\boldsymbol{s}_t, \boldsymbol{a}_t)$. And the final reward function is

$$\widehat{r}_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}) = r_\theta(\boldsymbol{s}_{t+1}) - \alpha u_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) \tag{3}$$

Follow-up work (Yu et al., 2021c) uses the same modeling approach but disposes of model ensembles and adapts the conservative Q-learning approach of (Kumar et al., 2020) into an MBPO critic optimization.

4. Model-based Offline to Online Fine-tuning (MOTO)

We would like to design a model-based reinforcement learning algorithm that can efficiently utilize offline datasets, while being easily amenable to continual learning and online fine-tuning. A line of prior works (Yu et al., 2020; 2021c; Cang et al., 2021) use MBPO-style optimization (Janner et al., 2019), which mixes real and model-generated data in a replay buffer used for policy training. (Rafailov et al., 2020) generalizes this approach to more realistic domains using variational models and latent ensembles and manages to solve a real robot task involving desk manipulation. However, these methods are not well-suited to the fine-tuning tasks, since the data in the replay buffer is sampled form the model's internal representation space, which suffers from significant distribution shift as the model is fine-tuned. Moreover, the need for replay buffers limits the scalability of these algorithms, as state-of-the-art predictive models in many realistic applications (such as autonomous driving (Hu et al., 2022; 2021; Akan & Güney, 2022)) require very large model and representation sizes. These issues make it impractical to carry large replay buffer of states, which have to be re-populated every time the predictive model

Algorithm 1 MOTO: Model-based Offline to Online Fine-tuning

Require: Offline dataset D, initialized policy π_ψ and critics Q_ψ, initialized prediction and reward model M_θ, policy rollout length H, number of offline training steps N_{offline}.

for $i = 1, 2, 3, \cdots$, **do**
  Sample a batch of trajectories $B \sim D$.
  Update M_θ on B according to Eq. 1.
  Generate H-step latent policy rollouts where the rewards are computed according to Eq. 3.
  Update π_ψ according to Eq. 10 and update Q_ψ according to Eq. 8 on the real and model data.
  if $i > N_{\text{offline}}$ **then**
    Rollout the policy π_θ in the environment to collect new trajectories D'
    $D = D \cup D'$
  end if
end for

is updated. Several algorithms such as MOREL (Kidambi et al., 2020) and BREMEN (Matsushima et al., 2020) use on-policy training within the learned model without the need for large replay buffers, which make them well-suited to continual learning settings. However, in this case the training procedure is unable to use potentially high-quality data from the offline dataset to supervise the critic training. In this work we train a model using Eq. 1 similarly to (Rafailov et al., 2020; Yu et al., 2021c), but instead we build the MOTO policy optimization on the three main design choices: 1) model-based value expansion, 2) uncertainty-aware predictive modelling, and 3) behaviour-regularized policy optimization. The full algorithm training outline is presented in Algorithm 1.

4.1. Variational Model-Based Value Expansion

To alleviate the issues discussed in Section 3.2, we would like to train a policy using model-based training, that can utilize previously collected data without the use of latent replay buffers. To this goal we adapt ideas from the model-based value expansion literature (Feinberg et al., 2018; Buckman et al., 2018; Amos et al., 2021; Clavera et al., 2020). We will still train a policy inside the latent space of our variational dynamics model using on-policy model-based rollouts. We consider sequences of data of the form $\tau = (\boldsymbol{x}_{1:T}, \boldsymbol{a}_{1:T}, \boldsymbol{r}_{1:T})$. At each agent training step, we infer latent states $\boldsymbol{s}^0_{1:T} \sim q_\theta(\boldsymbol{s}_{1:T}|\boldsymbol{x}_{1:T}, \boldsymbol{a}_{1:T})$. We also denote $\boldsymbol{a}_{1:T}$ and $\boldsymbol{r}_{1:T}$ as $\boldsymbol{a}^0_{1:T}$ and $\boldsymbol{r}^0_{1:T}$ respectively. Using these states as starting points, we use the policy π_ψ to generate H-step rollouts steps with the following notation: $\hat{\boldsymbol{a}}^t_j \sim \pi_\psi(\boldsymbol{a}|\hat{\boldsymbol{s}}^{t-1}_j)$, $\hat{\boldsymbol{s}}^{t+1}_j \sim p_\theta(\boldsymbol{s}|\hat{\boldsymbol{a}}^t_j, \hat{\boldsymbol{s}}^t_j)$ and $\hat{\boldsymbol{r}}^t_j \sim p_\theta(\boldsymbol{r}|\hat{\boldsymbol{s}}^t_j)$, where the rewards are computed according to Eq. 3. We also set $\hat{\boldsymbol{s}}^0_j = \boldsymbol{s}^0_j$. Following standard off-policy learning algorithms, we use critics $\{Q_{\psi^i}\}^m_{i=1}$ and and target networks $\{\bar{Q}_{\psi^i}\}^m_{i=1}$. We can then use our model to estimate Monte-Carlo base policy returns:

$$V^{\pi_\psi}_0(\hat{\boldsymbol{s}}^t_j) = \min_{i=1:m}\{Q_{\psi^i}(\hat{\boldsymbol{s}}^t_j, \hat{\boldsymbol{a}}^t_j)\}$$
$$V^{\pi_\psi}_K(\hat{\boldsymbol{s}}^t_j) = \sum_{k=1}^{K} \gamma^{k-1}\hat{\boldsymbol{r}}^{k+t}_j + \gamma^K V^{\pi_\psi}_0(\hat{\boldsymbol{s}}^{t+K}_j)$$

And evaluate the compute the standard TD(λ) estimate:

$$V^{\pi_\psi}(\hat{\boldsymbol{s}}^t_j) = (1-\lambda)\sum_{k=1}^{H-t-1}\lambda^{k-1}V^{\pi_\psi}_k(\hat{\boldsymbol{s}}^t_j) + \lambda^{H-t-1}V^{\pi_\psi}_{H-t}(\hat{\boldsymbol{s}}^t_j) \quad (4)$$

We use the the actor objective is then:

$$\mathcal{L}^{\text{model}}_{\pi_\psi} = -\frac{1}{HT}\left[\sum_{t=0,j=1}^{H-1,T}\lambda V^{\pi_\psi}(\hat{\boldsymbol{s}}^t_j) + (1-\lambda)V^{\pi_\psi}_0(\hat{\boldsymbol{s}}^t_j)\right] \quad (5)$$

which maximizes the expected MC return at the dataset and rollout states together. Notice that this fully differentiable function of the policy parameters, by using straight-through gradient estimation.

We can use MC return estimates to train the critics as well. We recompute the critic target values $\bar{V}^k(\hat{\boldsymbol{s}}^t_j)$ for all states similarly to Eq. 4 using the target networks $\{\bar{Q}_{\psi^i}\}^m_{i=1}$. Critics are trained on both the model-generated and real data with two losses:

$$\mathcal{L}^{\text{model}}_{Q_{\psi^i}} = \frac{1}{HT}\left[\sum_{t=0,j=1}^{H-1,T}(Q_{\psi^i}(\hat{\boldsymbol{s}}^t_j, \hat{\boldsymbol{a}}^t_j) - \bar{V}^{\pi_\psi}(\hat{\boldsymbol{s}}^t_j))^2\right] \quad (6)$$

$$\mathcal{L}^{\text{data}}_{Q_{\psi^i}} = \frac{1}{T-1}\left[\sum_{j=1}^{T-1}\Big(Q_{\psi^i}(\boldsymbol{s}^0_j, \boldsymbol{a}^0_j) - (\boldsymbol{r}^0_{j+1} + \gamma\big((1-\lambda)\bar{V}^{\pi_\psi}_0(\boldsymbol{s}^0_{j+1}) + \lambda\bar{V}^{\pi_\psi}(\boldsymbol{s}^0_{j+1})\big)\Big)^2\right] \quad (7)$$

The final critic loss is a combination of the two losses:

$$\mathcal{L}_{Q_{\psi^i}} = \mathcal{L}^{\text{model}}_{Q_{\psi^i}} + \mathcal{L}^{\text{data}}_{Q_{\psi^i}} \quad (8)$$

The loss $\mathcal{L}^{\text{data}}_{Q_{\psi^i}}$ is computed on transitions sampled form the dataset trajectories through the inference model q_θ and have ground-truth environment rewards. Training the critic networks on the available offline data serves as a strong supervision when the dataset already contains trajectories with high returns.

4.2. Model Uncertainty Corrections

Similarly to other model-based RL algorithms, without any regularization, the proposed method would also suffer from the model exploitation issue, which is especially prevalent in the offline training/pre-training case. The conservative critic optimization approach of (Kumar et al., 2020; Yu et al., 2021c) is not applicable to the policy optimization outlined in Section 4.1 as it is incompatible with multi-step returns and thus require the use of replay buffers (which we aim to avoid). Instead we opt to use model-based uncertainty estimates: (Chua et al., 2018; Clavera et al., 2018; Deisenroth & Rasmussen, 2011; Kurutach et al., 2018; Nagabandi et al., 2020; Luo et al., 2018; Strehl & Littman, 2008; Zanette & Brunskill, 2019). We utilize ensemble-based uncertainty estimates, which have proven efficient in many previous works.

In particular, we follow the reward penalty formulation from Section 3.2. We apply the correction as in Eq 3 only to the model-generated rewards used in Eq. 6, and we still use the ground-truth rewards in Eq. 7. Since the uncertainty penalty $u_\theta(\boldsymbol{s}, \boldsymbol{a}) \geqslant 0$, this implicitly builds conservatism into the critic itself. Note that the uncertainty penalty in Eq. 3 is a fully differentiable function, hence the Eq. 5 is still a differentiable function of the policy parameters.

4.3. Behaviour Prior Policy Regularization

Given the success of model-free behaviour priors in previous works (Swazinna et al., 2020; Cang et al., 2021; Argenson & Dulac-Arnold, 2020; Matsushima et al., 2020) we also evaluate such an approach in this method as well. One could use a general policy-training regularizer with different choices of divergence and priors yielding different algorithms. We choose to not explicitly parameterize the behaviour prior since that would require training an additional policy during the fine-tuning phase. Instead, we choose the standard forward KL divergence, which is equivalent to a behaviour cloning regularization

$$\mathcal{L}^{\text{reg}}_{\pi_\psi} = -\mathbb{E}_{\rho^{\pi_E}(\boldsymbol{s},\boldsymbol{a})}[\log \pi_\psi(\boldsymbol{a}|\boldsymbol{s})] \tag{9}$$

and the final policy loss is

$$\mathcal{L}_{\pi_\psi} = \mathcal{L}^{\text{model}}_{\pi_\psi} + \beta \mathcal{L}^{\text{reg}}_{\pi_\psi} \tag{10}$$

This method works surprisingly well in model-free RL (Fujimoto & Gu, 2021; Emmons et al., 2021; Lu et al., 2022) and does not require any additional computational overhead. In addition, however, we apply the regularization only to trajectories that successfully complete the tasks, an approach known as "filtered BC", (Emmons et al., 2021; Nguyen et al., 2022). During the fine-tuning phase new trajectories are added to the BC buffer if they are successful.

4.4. Theoretical Results for Uncertainty-Aware Model-based Training

Given our choice of variational parameterization and model uncertainty estimation we can directly adapt certain theoretical guarantees from prior model-based RL literature (Yu et al., 2020; Kidambi et al., 2020; Rafailov et al., 2020). We consider the following result in particular: let $T_\theta(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$ and $T(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$ be the learned and true latent dynamics models respectively. We define the discounted state-action distribution $\rho^\pi_{\mathcal{T},\mu_0}(\boldsymbol{s}, \boldsymbol{a}) \propto \sum_{t=0}^{\infty} \gamma^t \mathbb{P}^\pi_{\mathcal{T},\mu_0}(\boldsymbol{s}_t = \boldsymbol{s})\pi(\boldsymbol{a}|\boldsymbol{s})$ in the standard way. The function $u(\boldsymbol{s}, \boldsymbol{a})$ is an admissible error estimator if $d_{\mathcal{F}}[T(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})||T_\theta(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})] \leqslant u(\boldsymbol{s}, \boldsymbol{a})$. For any policy π we can then define $\epsilon_u(\pi) = \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim\rho^\pi_{T_\theta,\mu_0}}[u(\boldsymbol{s}, \boldsymbol{a})]$. Then the following Theorem holds:

Theorem 4.1. *(Informal) Let $\widehat{\pi}^*(\boldsymbol{s})$ be the optimal policy under the learned model $T_\theta(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$ with an uncertainty-penalized reward and π^* the optimal policy in the ground-truth MDP. Under certain mild assumptions, then the following inequality holds:*

$$2\alpha\epsilon_u(\pi^*) \geqslant \mathbb{E}_{\pi^*,T}\Big[\sum_{t=0}^{\infty} r_t\Big] - \mathbb{E}_{\widehat{\pi}^*,T}\Big[\sum_{t=0}^{\infty} r_t\Big] \tag{11}$$

Proof. Consult (Yu et al., 2020). □

The policy under-performance is upper bounded by the discounted model-based uncertainty over the state-action distribution induced by the expert policy under the learned model. In practice we do not have access to an oracle estimator $u(\boldsymbol{s}, \boldsymbol{a})$ and we use the ensemble disagreement from Eq. 2. While these results are not new, empirical verification is difficult in the fully offline case, since we have a static dataset, and all values are point estimates. However, in the online fine-tuning case, we have a continuum of datasets and we can empirically verify the claims of Theorem 4.1.

5. Experiments

For our experiments we use the Franka Kitchen environment from (Gupta et al., 2019) (RPL). This is a challenging long-range control problem, which involves a simulated 9-DOF Franka Emika Robot in a kitchen setting. The robot uses joint-space control and the observation is a single 64x64 RGB image; we do not assume access to object states or robot proprioception. The goal of the agent is to manipulate a set of 4 pre-defined objects and receives a reward of 1.0 for each object in right configuration at each time step. This is a very challenging environment due to 1) high-dimensional observation spaces; 2) partial observability with non-trivial object and robot state estimation; 3) need for very-fine-grained control in order to operate the small elements of the environment, such as turning knobs and flipping the light-switch; 4) the long-range nature of the tasks; 5) the use of sparse rewards, which provide limited intermediate supervision to the policy, and finally 6) the use of high-dimensional control which requires learning forward kinematics from images alone. For our experiments we render the original RPL datasets and consider two environments from the D4RL benchmark (Fu et al., 2020). The "mixed" task requires operating the microwave, kettle, light switch and slide cabinet and has a small set of successful demos in the offline dataset. The "partial" task, which requires manipulating the microwave, kettle, bottom burner and light switch does not have any trajectories that successfully complete all four objects, but has demonstrations for several configurations which complete up to three objects. We will release this dataset with our project to facilitate the development and testing of vision-based offline RL algorithms.

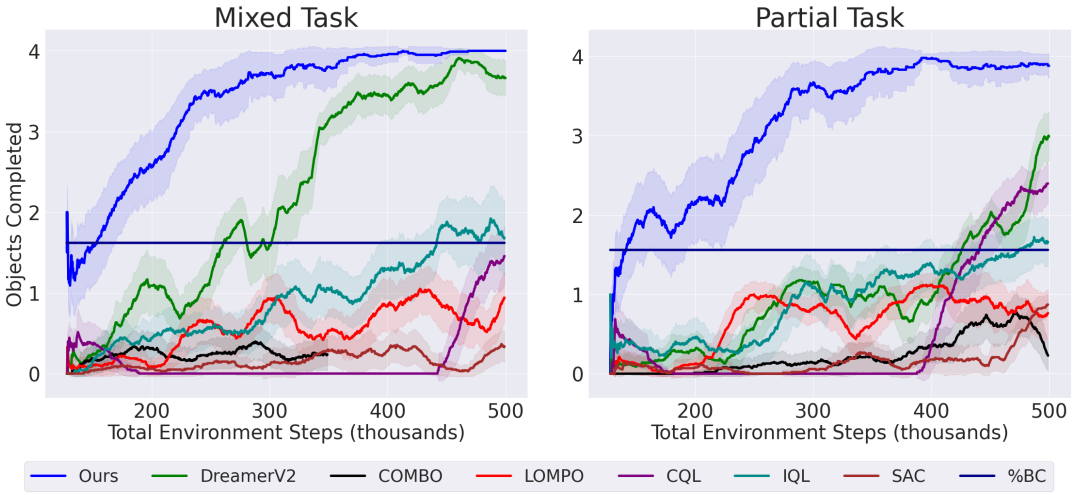

Figure 2. We pre-train offline RL algorithms on the offline dataset and fine-tune them with additional online interactions. MOTO solves both the "mixed" and "partial" tasks. Model-free methods struggle due to partial observability of the environment and the need for combinatorial generalization on the "partial" task. Offline model-based RL algorithms have poor online performance due to distributional shift issues in the latent replay buffers. DreamerV2 is the only other algorithm to achieve non-trivial success, but learns potentially unsafe behaviours.

5.1. Results

We compare our method to prior vision-based offline model-based RL algorithms LOMPO (Rafailov et al., 2020) and COMBO (Yu et al., 2021c) as well as DreamerV2 (Hafner et al., 2020b), a state-of-the art online model-based learning algorithm. We also compare our approach against CQL (Kumar et al., 2020) a successful model-free offline RL algorithm, IQL (Kostrikov et al., 2021) a state-of-the art model-free regression-based fine-tuning algorithm, SAC (Haarnoja et al., 2018) and behaviour cloning. All methods are pre-trained offline for 10 thousand gradient steps and fine-tuned with online interactions for a total of 500 thousand environment steps. Since the "partial" task does not contain successful trajectories for all four objects, we only regularize policy training with respect to the first three objects. Results are shown in Fig. 2. Our method successfully solves both the "mixed" and "partial" tasks with 100% and 90.5% final success rates respectively, and it's the only method besides DreamerV2 to get non-trivial success rates. While model-free methods make some progress, ultimately they stagnate and cannot successfully complete all four objects on either task. This is likely due to the partial observability nature of the environment, since the robot can occlude manipulated objects and also requires joint state estimation directly from images. In contrast variational models serve as Bayesian filters and naturally build state estimations of the environment in the latent space. Model-based methods LOMPO and COMBO achieve very limited progress, due to the non-stationarity issues described in the beginning of Section 4. The DreamerV2 algorithm is the only other method that achieves some success but learns more slowly and only reaches final success rates of 77.5% and 13.5% versus 100% and 90.5% for our method on the "mixed" and "partial" task respectively. As far as we know ours model is the first method to solve the Franka Kitchen environment from images.

5.2. Ablation Studies

We also carry out a number of ablations to study how each component of our model affects performance. In particular, we test our method ("Full Model"), remove the behaviour cloning regularization ("No Reg."), remove the model uncertainty reward penalty ("No Unc.") and the final version which removed both ("No Reg., No Unc."), which corresponds to a standard variational RL model where the critic is trained on both the real and model-based data. We also include the standard DreamerV2 algorithm for direct model-based comparison. While all ablations make significant progress on the "mixed" task, only the full model manages to fully solve it. Using our model-based value expansion approach still outperforms the DreamerV2 algorithm, which trains the actor and critic fully within the learned model. This demonstrates that we can inject meaningful supervision into the critic learning, when we have access to high-quality data. We also note that without any data regularization both the "No Reg., No Unc." and DreamerV2 methods learn unsafe behaviours such as hitting the kettle into the goal position or smashing the light switch with the robot head, instead of using its gripper to grasp and place the objects. These policies would be quite unsafe for both the hardware and environment in a real setting. Such behaviours are not

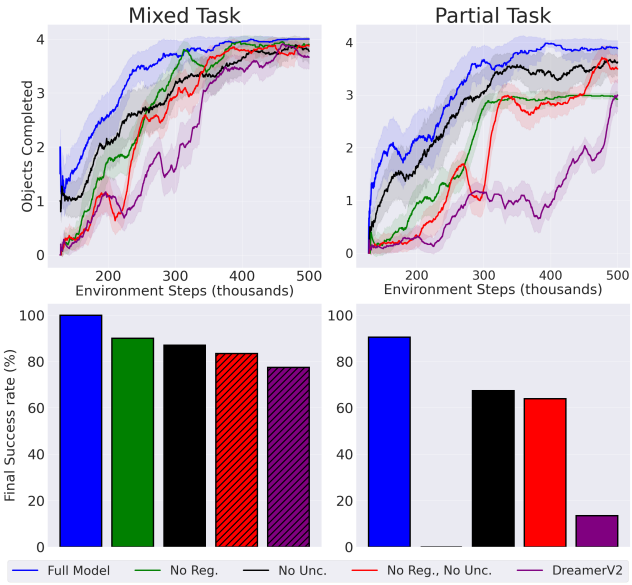

Figure 3. We carry out a number of ablations on the MOTO design, such as not uncertainty penalties "No Unc.", not using policy regularization "No. Reg", and removing both "No Reg., No Unc.". The top graph presents the learning curves over the online fine-tuning phase, while the bottom bar chart shows successful completions in the last 200 evaluation episodes. We see that all model component contribute to learning and final performance.

present in any of the regularized methods that either use model uncertainty or behaviour cloning regularization. Our method also significantly outperforms ablation baselines on the "partial" task as well. The "No Reg." version of our model manages to manipulate three objects. This is because the offline dataset contains several configurations which manipulate three out of the four objects, but are mutually inconsistent. The model-based uncertainty penalty encourages the policy to stay closer to the data distribution, which, given it's multi-modality, creates a difficult optimization problem. In contrast, versions of the model that use behaviour cloning regularization ("Full model", "No Unc.") are grounded in a particular sequence of objects, removing the multi-modality problem and models without uncertainty penalties ("No Reg., No Unc.", DreamerV2) are not as constrained to remain within distribution.

5.3. Model-Based Generalization

The "partial" task also provides a good test bed for an algorithm's generalization capabilities, since the offline dataset does not contain full solutions for it. This is a different problem than the standard dynamic programming ("stitching") issue of data-centric reinforcement learning since the dataset does not contain a sequence of state-action pairs that lead from the initial state to the goal state. Instead, to solve this task, a learning agent must understand the compositional nature of the scene and do combinatorial generalization over the objects. In this section we seek to answer whether 1) the learned model can do combinatorial generalization of within distribution tasks and 2) whether policy optimization can take advantage of the model's capabilities. We evaluate the agent at the end of the offline pre-training phase. To answer the first question, we consider episodes that successfully complete the "partial" task from the trained agent. We condition our model on the frames that solves the first three tasks (which are covered in the offline dataset) and rollout the expert actions to predict the following frames. Results are shown in Fig. 1. The model successfully predicts a combination of the microwave, kettle, bottom burner and light switch in the correct configuration, despite never encountering these four objects together in the offline dataset. Moreover, we evaluate the model-predicted rewards on these expert trajectories, plotted in Fig. 4 (left). We see that the model predicts rewards of up to 4 with an average reward of 3.63, despite only being trained on trajectories with maximum reward of 3. This results show that the learned model is capable of compositional generalization. To evaluate whether the learned policy can take advantage of the model generalization capabilities, we rollout the trained agent under the model and evaluate the predicted rewards, results are shown in Fig. 4 (right). The agent achieves an average final reward of 3.52 under the learned model and solves all four tasks. This suggest that the model-based RL agent is able to do combinatorial generalization, but the offline dataset is not enough to adequately learn the environment dynamics.

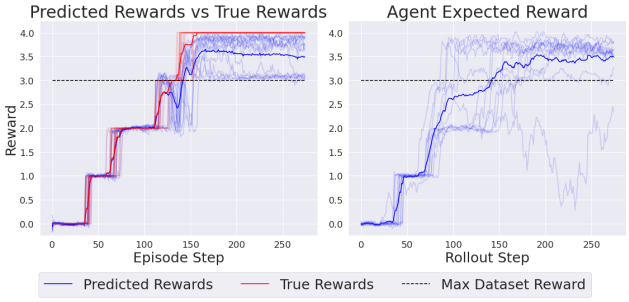

Figure 4. We evaluate the model's generalization capabilities at the end of the offline pre-training phase. The model correctly predicts rewards of up to 4 on successful episodes in the "partial" task, even though the maximum dataset reward is 3. (left). When doing rollouts in the learned model, the policy solves all four objects in the "partial" task and reaches rewards of up to 4 (right).

5.4. Performance Gap Empirical Verification

It is not possible to verify the bound from Eq. 11 empirically in the offline case, as we only have the point estimates of the static dataset. However, in the online fine-tuning setting, we have a continuum of datasets as we aggregate additional data. We can then periodically evaluate $\epsilon_u(\pi^*)$ and the expected model uncertainty induced under the expert state-action distribution in the learned model. At each epoch E, we cannot generate model rollouts from the ex-

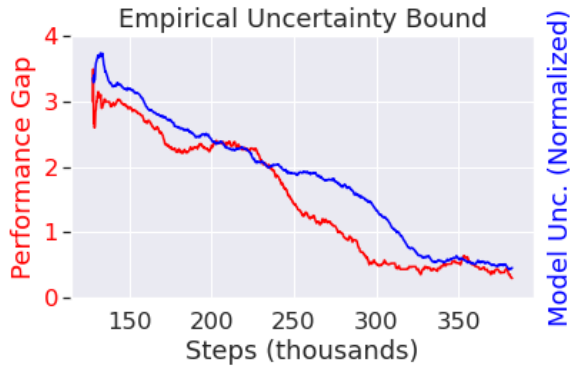

Figure 5. Empirical evaluation of Theorem 4.1. We plot the performance gap versus the the empirical estimates of (normalized) expected model uncertainty using Eq. 12.

pert, since that would require training an expert policy under the current inference model q_{θ_E}. However, we can sample expert episodes from the trained expert and the environment. Given an expert trajectory $\tau^{\text{exp}} = \boldsymbol{x}_{1:T}, \boldsymbol{a}_{1:T}$ we sample latent belief states from the first $T - H$ steps to obtain $\boldsymbol{s}_{1:(T-H)} \sim q_{\theta_E}(\cdot|\boldsymbol{x}_{1:T-H}, \boldsymbol{a}_{1:T-H})$. From each state $\boldsymbol{s}_j$ we then rollout the expert actions $\boldsymbol{a}_{j:j+H}$ using the current iteration of the dynamics model T_{θ_E} and obtain states $\{(\hat{\boldsymbol{s}}_j^t, \boldsymbol{a}_j^t\}_{j=1,t=0}^{T-H,H}$ as in Section 4.1 (here $\boldsymbol{a}_j^t = \boldsymbol{a}_{j+t}$ from the expert dataset. We can then obtain the empirical estimate of

$$\epsilon_u(\pi^*) \approx \mathbb{E}_{q_{\theta_E}(\boldsymbol{s}_j^0|\tau^{\text{exp}}), T_{\theta_E}}\Big[\frac{1}{H(T-H)}\sum u_\theta(\hat{\boldsymbol{s}}_j^t, \boldsymbol{a}_j^t)\Big] \tag{12}$$

Empirical results evaluated on the "partial" task are shown in Fig. 5 We see that the performance gap is strongly bounded (up to a choice of the penalty scale) by the estimate from Eq. 12, which verifies the claim of Theorem 4.1.

6. Discussion

We present a model-based RL algorithm which can efficiently train offline and fine-tune online or continually. The algorithm design does not require large replay buffers of intermediate representations, while still allowing the use of high-quality data to supervise the critic learning and bootstrap the policy optimization. We believe that these qualities make MOTO very suitable for realistic applications such as autonomous driving. Large scale recurrent geometric prediction models are standard in these applications (Hu et al., 2022; 2021; Akan & Güney, 2022) and can be adapted to our setting. Recently such work (Hu et al., 2022) reported state-of-the-art performance on the CARLA driving benchmark (Dosovitskiy et al., 2017) using a combination of model-based representation and imitation learning. In the same time concurrent work (Lu et al., 2022) demonstrated that direct imitation learning can be sub-optimal in real world driving cases, especially in scenarios with low data coverage and high-stakes. Similarly to our work the authors use a reinforcement learning approach with a behaviour cloning regularization and show nearly two fold decrease in sub-optimal behaviour in critical driving scenarios. We believe that MOTO is well-suited for AV applications and plan to pursue this direction in following work.

MOTO is also well-suited to the model-based imitation learning setting (Baram et al., 2017; Rafailov et al., 2021; Chang et al., 2021; Zhang et al., 2022), which has recently been successfully applied to real world scenarios as well (Lu et al., 2022; Bronstein et al., 2022). By using on-policy roll-outs MOTO can maintain the stability and theoretical guarantees of adversarial imitation learning (Ho & Ermon, 2016; Finn et al., 2016; Rafailov et al., 2020), while still using the high-quality expert data to both provide supervision to the critic, as well as regularize the policy. Moreover, MOTO can be applied to both the offline and online case.

Previous work (Yu et al., 2021a;b) has also shown that model-free offline RL agents can struggle to utilize multi-task data without specific adjustments, while model-based works (Yu et al., 2020; 2021c; Argenson & Dulac-Arnold, 2020; Cang et al., 2021) have demonstrated good transfer between related tasks in the low-dimensional domain. Our generalization experiments from Section 5.3 as well as theoretical and empirical results from Section 4.4 and Section 5.4 indicate that MOTO can be well suited to multi-task or the pre-training and downstream task fine-tuning regime as long as these tasks share overlapping dynamics or behaviours. We plan to evaluate such capabilities on realistic robotic tasks (Dasari et al., 2019; Ebert et al., 2021) in later work.

7. Conclusion

We presented MOTO, a model-based reinforcement learning algorithm specifically designed for the offline pre-training and down-stream fine-tuning regime. MOTO learns a variational model directly from pixels, and trains and actor-critic agent within the learned latent dynamics model using model-based value expansion, ensembles and behaviour policy regularization. MOTO performs well on the established Franka Kitchen benchmark and demonstrates capability for combinatorial generalization. As far as we are aware, MOTO is the first algorithm to solve the environment directly from images. Also, by studying the offline pre-training and fine-tuning regime, we empirically verify long-standing theoretical results on the offline model-based RL problem, which have implications for multi-task and transfer applications. Finally, the structure of the algorithm makes it suitable for use in very large scale models, which prior work was not able to utilize, as well as a backbone for model-based imitation, multi-task and transfer learning. We plan to further pursue these directions in follow-up works.

References

Akan, A. K. and Güney, F. Stretchbev: Stretching future instance prediction spatially and temporally. *arXiv preprint arXiv:2203.13641*, 2022.

Amos, B., Stanton, S., Yarats, D., and Wilson, A. G. On the model-based stochastic value gradient for continuous reinforcement learning. In *Learning for Dynamics and Control*, pp. 6–20. PMLR, 2021.

Argenson, A. and Dulac-Arnold, G. Model-based offline planning. *arXiv preprint arXiv:2008.05556*, 2020.

Baram, N., Anschel, O., Caspi, I., and Mannor, S. End-to-end differentiable adversarial imitation learning. In *International Conference on Machine Learning*, pp. 390–399. PMLR, 2017.

Bronstein, E., Palatucci, M., Notz, D., White, B., Kuefler, A., Lu, Y., Paul, S., Nikdel, P., Mougin, P., Chen, H., et al. Hierarchical model-based imitation learning for planning in autonomous driving. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8652–8659. IEEE, 2022.

Buckman, J., Hafner, D., Tucker, G., Brevdo, E., and Lee, H. Sample-efficient reinforcement learning with stochastic ensemble value expansion. *Advances in neural information processing systems*, 31, 2018.

Cang, C., Rajeswaran, A., Abbeel, P., and Laskin, M. Behavioral priors and dynamics models: Improving performance and domain transfer in offline rl. *arXiv preprint arXiv:2106.09119*, 2021.

Chang, J. D., Uehara, M., Sreenivas, D., Kidambi, R., and Sun, W. Mitigating covariate shift in imitation learning via offline data without great coverage. *arXiv preprint arXiv:2106.03207*, 2021.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

Clavera, I., Rothfuss, J., Schulman, J., Fujita, Y., Asfour, T., and Abbeel, P. Model-based reinforcement learning via meta-policy optimization. In *Conference on Robot Learning*, pp. 617–629. PMLR, 2018.

Clavera, I., Fu, V., and Abbeel, P. Model-augmented actor-critic: Backpropagating through paths. *arXiv preprint arXiv:2005.08068*, 2020.

Dasari, S., Ebert, F., Tian, S., Nair, S., Bucher, B., Schmeckpeper, K., Singh, S., Levine, S., and Finn, C. Robonet: Large-scale multi-robot learning. In *Conference on Robot Learning*, pp. 885–897, 2019.

Deisenroth, M. and Rasmussen, C. E. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pp. 465–472, 2011.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

Ebert, F., Yang, Y., Schmeckpeper, K., Bucher, B., Georgakis, G., Daniilidis, K., Finn, C., and Levine, S. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

Emmons, S., Eysenbach, B., Kostrikov, I., and Levine, S. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751*, 2021.

Feinberg, V., Wan, A., Stoica, I., Jordan, M. I., Gonzalez, J. E., and Levine, S. Model-based value estimation for efficient model-free reinforcement learning. *arXiv preprint arXiv:1803.00101*, 2018.

Finn, C., Tan, X. Y., Duan, Y., Darrell, T., Levine, S., and Abbeel, P. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 512–519. IEEE, 2016.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019.

Ha, D. and Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. *International Conference on Learning Representations*, 2019.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. *International Conference on Learning Representations*, 2020a.

Hafner, D., Lillicrap, T., Norouzi, M., and Ba, J. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020b.

Ho, J. and Ermon, S. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Hu, A., Murez, Z., Mohan, N., Dudas, S., Hawke, J., Badrinarayanan, V., Cipolla, R., and Kendall, A. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15273–15282, 2021.

Hu, A., Corrado, G., Griffiths, N., Murez, Z., Gurau, C., Yeo, H., Kendall, A., Cipolla, R., and Shotton, J. Model-based imitation learning for urban driving. *arXiv preprint arXiv:2210.07729*, 2022.

Janner, M., Fu, J., Zhang, M., and Levine, S. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, pp. 12498–12509, 2019.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. Morel: Model-based offline reinforcement learning. *arXiv preprint arXiv:2005.05951*, 2020.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*, 2020.

Kurutach, T., Clavera, I., Duan, Y., Tamar, A., and Abbeel, P. Model-ensemble trust-region policy optimization. *arXiv preprint arXiv:1802.10592*, 2018.

Lange, S., Gabel, T., and Riedmiller, M. A. Batch reinforcement learning. In *Reinforcement Learning*, volume 12. Springer, 2012.

Lee, A. X., Nagabandi, A., Abbeel, P., and Levine, S. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *arXiv preprint arxiv:1907.00953.pdf*, 2020a.

Lee, A. X., Nagabandi, A., Abbeel, P., and Levine, S. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. In *Advances in Neural Information Processing Systems*, 2020b.

Levine, S. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

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

Lu, Y., Fu, J., Tucker, G., Pan, X., Bronstein, E., Roelofs, B., Sapp, B., White, B., Faust, A., Whiteson, S., et al. Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios. *arXiv preprint arXiv:2212.11419*, 2022.

Luo, Y., Xu, H., Li, Y., Tian, Y., Darrell, T., and Ma, T. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. *arXiv preprint arXiv:1807.03858*, 2018.

Matsushima, T., Furuta, H., Matsuo, Y., Nachum, O., and Gu, S. Deployment-efficient reinforcement learning via model-based offline optimization. *arXiv preprint arXiv:2006.03647*, 2020.

Nagabandi, A., Konolige, K., Levine, S., and Kumar, V. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pp. 1101–1112. PMLR, 2020.

Nair, A., Dalal, M., Gupta, A., and Levine, S. Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Nguyen, T., Zheng, Q., and Grover, A. Conserweightive behavioral cloning for reliable offline reinforcement learning. *arXiv preprint arXiv:2210.05158*, 2022.

Rafailov, R., Yu, T., Rajeswaran, A., and Finn, C. Offline reinforcement learning from images with latent space models. *ArXiv*, abs/2012.11547, 2020.

Rafailov, R., Yu, T., Rajeswaran, A., and Finn, C. Visual adversarial imitation learning using variational models. *Advances in Neural Information Processing Systems*, 34: 3016–3028, 2021.

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J. T., et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Strehl, A. L. and Littman, M. L. An analysis of model-based interval estimation for markov decision processes.

Journal of Computer and System Sciences, 74(8):1309–1331, 2008.

Swazinna, P., Udluft, S., and Runkler, T. Overcoming model bias for robust offline deep reinforcement learning. *arXiv preprint arXiv:2008.05533*, 2020.

Watter, M., Springenberg, J., Boedecker, J., and Riedmiller, M. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in neural information processing systems*, pp. 2746–2754, 2015.

Yang, M. and Nachum, O. Representation matters: offline pretraining for sequential decision making. In *International Conference on Machine Learning*, pp. 11784–11794. PMLR, 2021.

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J., Levine, S., Finn, C., and Ma, T. Mopo: Model-based offline policy optimization. *arXiv preprint arXiv:2005.13239*, 2020.

Yu, T., Kumar, A., Chebotar, Y., Finn, C., Levine, S., and Hausman, K. Data sharing without rewards in multi-task offline reinforcement learning. 2021a.

Yu, T., Kumar, A., Chebotar, Y., Hausman, K., Levine, S., and Finn, C. Conservative data sharing for multi-task offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34:11501–11516, 2021b.

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

Zanette, A. and Brunskill, E. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In *International Conference on Machine Learning*, pp. 7304–7312. PMLR, 2019.

Zhang, M., Vikram, S., Smith, L., Abbeel, P., Johnson, M., and Levine, S. Solar: Deep structured representations for model-based reinforcement learning. In *International Conference on Machine Learning*, pp. 7444–7453. PMLR, 2019.

Zhang, W., Xu, H., Niu, H., Cheng, P., Li, M., Zhang, H., Zhou, G., and Zhan, X. Discriminator-guided model-based offline imitation learning. *arXiv preprint arXiv:2207.00244*, 2022.