OFFLINE REINFORCEMENT LEARNING FROM IMAGES WITH LATENT SPACE MODELS

Rafael Rafailov[*1], **Tianhe Yu**[*1], **Aravind Rajeswaran** [2] **& Chelsea Finn** [1]
[1]Stanford University, [2]University of Washington
{rafailov,tianheyu,cbfinn}@stanford.edu,aravraj@cs.washington.edu

ABSTRACT

Offline reinforcement learning (RL) refers to the task of learning policies from a static dataset of environment interactions. Offline RL enables extensive utilization and re-use of historical datasets, while also alleviating safety concerns associated with online exploration, thereby expanding the real-world applicability of RL. Most prior work in offline RL has focused on tasks with compact state representations. However, the ability to learn directly from rich observation spaces like images is critical for real-world applications like robotics. In this work, we build on recent advances in model-based algorithms for offline RL, and extend them to high-dimensional visual observation spaces. Model-based offline RL algorithms have achieved state of the art results in state based tasks and are minimax optimal. However, they rely crucially on the ability to quantify uncertainty in the model predictions. This is particularly challenging with image observations. To overcome this challenge, we propose to learn a latent-state dynamics model, and represent the uncertainty in the latent space. Our approach is both tractable in practice and corresponds to maximizing a lower bound of the ELBO in the unknown POMDP. Through experiments on a range of challenging image-based locomotion and robotic manipulation tasks, we find that our algorithm significantly outperforms previous offline model-free RL methods as well as state-of-the-art online visual model-based RL methods. Moreover, we also find that our approach excels on an image-based drawer closing task on a real robot using a pre-existing dataset. All results including videos can be found online at https://sites.google.com/view/lompo/.

1 INTRODUCTION

For robots and artificial agents to be competent in a wide variety of dynamic and uncertain environments, they require the ability to perceive the world and act based on rich sensory observations like vision. In most real-world scenarios like homes or disaster management, it is difficult to hand-design state representations or simulators, let alone instrument the world to estimate the states. This suggests the need for an end-to-end integration of sensing and control. Despite recent advances (Yarats et al., 2019; Laskin et al., 2020b; Kostrikov et al., 2020; Hafner et al., 2020), the interactive sample complexity for learning control policies from vision is prohibitively high. Furthermore, interactive reinforcement learning (RL) with physical systems such as robots is fraught with safety challenges that limit widespread applicability. Our goal in this work is to develop algorithmic approaches for overcoming these challenges by utilizing offline datasets.

Offline RL (Lange et al., 2012) involves the learning of control policies from static pre-collected data. By utilizing already collected historical data, we alleviate the safety challenges associated with online exploration. Large offline datasets are also available in domains like autonomous driving (Caesar et al., 2020), recommendation systems (Harper & Konstan, 2015), and robotic manipulation (Finn & Levine, 2017; Sharma et al., 2018; Dasari et al., 2019; Mandlekar et al., 2019), typically in image (or video) format. Prior work in offline RL (see Section 2) has typically focused on environments with compact state representations, which are not representative of challenges faced in real-world applications. In this work, we focus on control from pixels using offline datasets. We

[*]denotes equal contribution.

believe the ability to train vision based policies using offline data can reduce interactive sample complexity, enhance safety, and greatly expand the applicability of RL.

Our work builds on recent advances in model-based offline RL (Kidambi et al., 2020; Yu et al., 2020). Model-based RL algorithms have demonstrated impressive sample efficiency results in interactive RL (Janner et al., 2019; Rajeswaran et al., 2020; Hafner et al., 2020). For offline RL, model-based algorithms (Kidambi et al., 2020; Yu et al., 2020) have been shown to be mimimax optimal, obtain state of the art results in a variety of benchmark tasks, as well as generalize to new out-of-distribution tasks. Uncertainty quantification and pessimism have emerged as key principles and requirements for successful offline RL, with both model-based and model-free approaches. This represents a unique challenge for control from pixels, since modeling uncertainty in high-dimensional image spaces such as using ensembles of video prediction models is challenging and computationally expensive.

Our Contributions. The main contribution of our work is an algorithm, latent offline model-based policy optimization (LOMPO), which enables learning of visuomotor policies using offline datasets. LOMPO efficient learns a dynamics model using latent state spaces, and represents undertainty in the low-dimensional latent state space. Our algorithm can be summarized as follows. (i) We learn an ensemble of latent state dynamics models from pixels using the offline dataset. Specifically, we learn a single encoder and decoder, but ensembles of dynamics models in the low-dimensional latent state space. (ii) We construct an uncertainty penalized MDP in the latent space and the corresponding uncertainty-penalized POMDP, where we quantify uncertainty based on disagreement between models in the latent state space. (iii) We learn a control policy using the offline dataset by optimizing an uncertainty-penalized objective. The learned uncertainty penalized MDP provides a pessimistic regularizing effect for policy learning and guards against major challenges like distributional shift and model exploitation (Yu et al., 2020). We evaluate our algorithm on extensive simulated visuomotor control tasks and one real-world robotic manipulation task. We find that LOMPO outperforms or matches other model-based methods across the board on these challenging tasks, and also outperforms model-free methods on most tasks.

2 Related Work

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

Offline RL. Offline RL has recently emerged as a prominent paradigm for learning control policies (Lange et al., 2012; Levine et al., 2020). Most offline RL algorithms augment well known RL algorithms with various forms of regularization. These include regularized variants of importance sampling based algorithms (Liu et al., 2019; Swaminathan & Joachims, 2015; Nachum et al., 2019; Zhang et al.), actor-critic algorithms (Wu et al., 2019; Jaques et al., 2019; Siegel et al., 2020; Peng et al., 2019), approximate dynamic programming algorithms (Fujimoto et al., 2018; Kumar et al., 2019; 2020; Liu et al., 2020; Agarwal et al., 2019), and model-based RL algorithms (Kidambi et al., 2020; Yu et al., 2020; Argenson & Dulac-Arnold, 2020; Matsushima et al., 2020; Swazinna et al., 2020). However, most of these prior works focus on problems with low-dimensional compact state information except that a few of them consider Atari games (Fujimoto et al., 2018; Agarwal et al., 2019; Kumar et al., 2020), whereas our focus in this work is on control from high-dimensional realistic perception in both simulation and the real world.

Control from Pixels. Control from high-dimensional observation inputs has become an important problem within control and robotics as it makes real world applications more practical. Prior works tackle this problem by either learning policies from pixel inputs end-to-end with model-free RL methods (Lee et al., 2020; Haarnoja et al., 2018; Lange & Riedmiller, 2010; Ghadirzadeh et al., 2017; Wayne et al., 2018; Kostrikov et al., 2020; Gelada et al., 2019; Nair et al., 2018; Singh et al., 2019; Srinivas et al., 2020; Laskin et al., 2020a; Han et al., 2020) or using model-based RL by learning a dynamics model either in the pixel space (Finn & Levine, 2017; Ebert et al., 2018) or in a latent space (Levine et al., 2016; Finn et al., 2016; Watter et al., 2015; Banijamali et al., 2018; Zhang et al., 2019; Hafner et al., 2019; 2020; Ha & Schmidhuber, 2018; Kipf et al., 2019; Chen et al., 2020). However, most of those prior works rely crucially on online data collection to be

successful. Visual foresight algorithms (Finn & Levine, 2017; Ebert et al., 2018; Chen et al., 2020) handle control from pixels in a fully offline setting, but do not explicitly tackle the distributional shift issue that arises in the offline RL problem, while our method address such an issue. As a result, we find in Section 5.2 that our approach significantly outperforms visual foresight.

3 PRELIMINARIES

POMDPs. We consider a class of partially observable Markov decision processes (POMDPs), whose transition dynamics can be described using a compact MDP, and observations using an emmision model. Concretely, we consider POMDPs of the form $M = (\mathcal{X}, \mathcal{S}, \mathcal{A}, T, D, r, \mu_0, \gamma)$ where $\mathcal{X}$ denotes the the visual observation space, $\mathcal{S}$ the latent state space, $\mathcal{A}$ the action space, $T(s'|s,a)$ the latent transition distribution, $D(x|s)$ the observation model, $r(s,a)$ the reward function, $\mu_0(s_0)$ the initial latent state distribution, and $\gamma \in (0,1)$ the discount factor. The goal of the POMDP is to learn a policy $\pi(a_t|s_t)$ that maximizes the discounted expected return $\eta_M := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$.

Model-based RL with latent dynamics. Model-based RL leverages a learned dynamics model to accelerate the RL training (Sutton, 1991; Watter et al., 2015). However, when the sensory inputs consist of high-dimensional observations such as images, it is both more memory-efficient and more computationally-efficient to learn the dynamics model in a compact latent space. Since the high-dimensional observations do not fully capture the full state of the environment, prior works such as (Hafner et al., 2019), (Hafner et al., 2020) and Lee et al. (2020) consider the POMDP setting, which they solve by learning a deep variational model. Given a learned latent dynamics model, PlanNet (Hafner et al. (2019)) carries out model-predictive control in the latent space, while Dreamer (Hafner et al., 2020) learns an actor-critic policy in the latent space by directly differentiating through the model dynamics and rewards. On the other hand Lee et al. (2020) uses the model as a deep filter and separately learns an SAC-based policy on top of the latent representation.

Control as inference. For MDPs with directly observable states s_t, actions a_t, rewards r_t, initial state distribution $\mu_0(s_1)$, and the stochastic transition dynamics $T(s_{t+1}|s_t, a_t)$, the control problem is equivalent to an inference problem in the graphical model with a binary random variable $\mathcal{O}_t$, which denotes if the agent is optimal at time step t. When $p(\mathcal{O}_t = 1|s_t, a_t) = \exp(r(s_t, a_t))$, maximizing $p(\mathcal{O}_{1:T})$ for some finite horizon T via approximate inference is equivalent to optimizing the maximum entropy RL objective $\mathbb{E}[\sum_{t=1}^{T}(r(s_t, a_t) + \alpha\mathcal{H}(\pi(\cdot|s_t)))]$(Levine, 2018). Lee et al. (2020) extend the framework to POMDPs by factorizing the variational distribution $q(s_{1:T}, a_{t+1:T}|x_{1:t+1}, a_{1:t})$ into a product of inference terms $q(s_{t+1}|x_{t+1}, s_t, a_t)$, latent dynamics terms $T(s_{t+1}|s_t, a_t)$ and the policy terms $\pi(a_t|x_{1:t}, a_{1:t-1})$. Optimizing the evidence lower bound (ELBO) of $p(x_{1:t+1}, \mathcal{O}_{t+1:T}|a_{1:t})$, the likelihood of observed data and future optimality of the agent is also equivalent to the objective of maximum entropy RL.

Offline RL is the setting where an agent has to learn the control policy using only a fixed dataset of interactions. In our work, we focus on offline RL in high-dimensional POMDPs with a latent MDP, where the agent has access to the fixed dataset $\mathcal{D}_{\text{env}} = (x_t, a_t, r_t, x_{t+1})$. The dataset can be collected by a single or a mixture of behavior policies π^{B}. No additional interactions with the environment are possible. We call the distribution induced by $\mathcal{D}_{\text{env}}$ as the *behavioral distribution*.

Model-Based Offline RL. Model-based RL in conjunction with the key idea of pessimism or conservatism, has emerged as a promising paradigm for offline RL (Kidambi et al., 2020; Yu et al., 2020; Matsushima et al., 2020; Argenson & Dulac-Arnold, 2020). Conservative versions of model-based RL have been shown to be minimax optimal and have also obtained state of the art results in many benchmark tasks. In this work, we build on the MOPO (Yu et al., 2020) framework. Given a dataset $\mathcal{D}_{\text{env}}$ from an MDP M, MOPO learns an uncertainty penalized MDP with learned transition dynamics $\hat{T}(\cdot|s,a)$ and the penalized reward $\widetilde{r}(s,a) = \widehat{r}(s,a) - u(s,a)$ where $\widehat{r}$ is the learned reward function and u is an admissible uncertainty estimator such as $u(s,a) \geq \frac{r_{\max}}{1-\gamma} D_{TV}(T(s,a), \hat{T}(s,a))$. MOPO theoretically shows that optimizing a policy under the *uncertainty-penalized* MDP $\widetilde{M} = (\mathcal{S}, \mathcal{A}, \widehat{T}, \widetilde{r}, \mu_0, \gamma)$ is equivalent to optimize a lower bound of the return under the learned policy in the true MDP M. While MOPO achieve impressive results from low-dimensional observation spaces, they haven't shown successes in visual domains where the inputs are pixels. In the next section, we will extend MOPO to the setting with high-dimensional observations.

4 LATENT OFFLINE MODEL-BASED POLICY OPTIMIZATION

The goal of our method is to design an offline model-based RL method that handles high-dimensional inputs. Since we need to learn a model from the fixed dataset without further interaction with the environment, the model prediction will become less trustworthy as the model rollouts move further from the behavioral distribution. Such inaccurate model predictions would generate states that could negatively impact the policy optimization. Therefore, quantifying the uncertainty of the states generated by the learned model is important for offline model-based RL to avoid large extrapolation error on out-of-distribution states. However, estimating the model uncertainty in the high-dimensional observation space is challenging as the common approach in uncertainty quantification such as learning an ensemble of models is very memory-intensive and time-consuming when applied to visual dynamics models.

In this section, we present our offline visual model-based RL algorithm that address the above challenges by learning a latent dynamics model and estimating the model uncertainty in the compact latent space. Specifically, we first assume that the control problem in the latent space is an MDP and then construct an uncertainty-penalized latent MDP with the reward penalized by the uncertainty of the latent dynamics model (Section 4.1). Second, we construct the corresponding uncertainty-penalized POMDP and optimize the policy and the latent dynamics model by maximizing the ELBO in the uncertainty-penalized POMDP, which is a lower bound of the ELBO in the true POMDP (Section 4.2). Finally, we discuss our overall practical algorithm LOMPO (Section 4.3).

4.1 QUANTIFYING MODEL UNCERTAINTY IN THE LATENT SPACE

In order to quantify the uncertainty in the latent state space, we first make the assumption that the latent state space $\mathcal{S}$ forms an MDP, which we define as the latent MDP $M_{\mathcal{S}} = (\mathcal{S}, \mathcal{A}, T, \mu_0, \mu_0, \gamma)$ with the same notations in Section 3. Similarly, we define the estimated latent MDP $\widehat{M}_{\mathcal{S}} = (\mathcal{S}, \mathcal{A}, \widehat{T}, r, \mu_0, \gamma)$ where $\widehat{T}(s'|s,a)$ denotes the learned latent dynamics model. The objective of our algorithm is to learn an optimal policy $\pi(a|s)$ in $\widehat{M}_{\mathcal{S}}$ that also maximizes expected return in $M_{\mathcal{S}}$. With the above definitions of $M_{\mathcal{S}}$ and $\widehat{M}_{\mathcal{S}}$, we can construct the *uncertainty-penalized* latent MDP $\widetilde{M}_{\mathcal{S}} = (\mathcal{S}, \mathcal{A}, \widehat{T}, \widetilde{r}, \mu_0, \gamma)$ where $\widetilde{r}(s,a) = r(s,a) - \lambda u(s,a)$ where $u(s,a)$ is an admissible uncertainty estimator as defined in Section 3. Under the uncertainty-penalized latent MDP, we first define $\epsilon_u(\pi) = \mathop{\bar{\mathbb{E}}}_{(s,a)\sim\rho^{\pi}_{\widehat{T},\mu_0}}[u(s,a)]$ where $\rho^{\pi}_{T,\mu_0}(s,a) := \sum_{t=0}^{\infty}\gamma^t \mathbb{P}^{\pi}_{T,\mu,t}(s)\pi(a\,|\,s)$ denotes the discounted state-action distribution. Then we proceed to show that the total return of a policy π under the uncertainty-penalized latent MDP is a lower bound of the the total return under the true latent state MDP and the gap between the learned policy under the uncertainty-penalized latent MDP and the optimal policy π^* depends on the latent dynamics model error $\epsilon_u(\pi^*)$. Directly following Theorem 4.4 in Yu et al. (2020), we can show the above claims as following:

$$\eta_{M_{\mathcal{S}}}(\pi) \geq \eta_{\widetilde{M}_{\mathcal{S}}}(\pi), \tag{1}$$

$$\eta_{M_{\mathcal{S}}}(\hat{\pi}) \geq \sup_{\pi}\{\eta_{M_{\mathcal{S}}}(\pi) - 2\lambda\epsilon_u(\pi)\} \tag{2}$$

where $\hat{\pi}$ is the policy learned via maximizing the return under $\widetilde{M}_{\mathcal{S}}$. In practice, we do not have access to the uncertainty quantification oracle $u(s,a)$ that upper bounds the latent model error, and we estimate the uncertainty of the latent model using heuristics as discussed in Section 4.3.

4.2 LATENT MODEL TRAINING AND POLICY OPTIMIZATION WITH UNCERTAINTY-PENALIZED ELBO

With the uncertainty-penalized latent MDP, we can similarly construct the corresponding uncertainty-penalized POMDP $\widetilde{M} = (\mathcal{X}, \mathcal{S}, \mathcal{A}, \widehat{T}, D, \widetilde{r}, \mu_0, \gamma)$. We define the learned variational distribution $\widehat{q}(s_{1:T}, a_{t+1:T}|x_{1:t+1}, a_{1:t})$ as a product of inference terms $q(s_{t+1}|x_{t+1}, s_t, a_t)$, the learned latent dynamics terms $\widehat{T}(s_{t+1}|s_t, a_t)$ and the policy terms $\pi(a_t|x_{1:t}, a_{1:t-1})$ as follows

$$\widehat{q}(s_{1:T}, a_{t+1:T}|x_{1:t+1}, a_{1:t}) = \prod_{\tau=0}^{t} q(s_{\tau+1}|x_{\tau+1}, s_\tau, a_\tau) \prod_{\tau=t+1}^{T-1} \widehat{T}(s_{\tau+1}|s_\tau, a_\tau) \prod_{\tau=t+1}^{T} \pi(a_\tau|x_{1:\tau}, a_{1:\tau-1}).$$

With $\widehat{q}$, we can bound the expected return term from below as follows:

$$\mathbb{E}_{s_{t+1:T}, a_{t+1:T} \sim q} \left[\sum_{\tau=t+1}^{T} r(s_\tau, a_\tau) \right] = \bar{\mathbb{E}}_{s_{t+1:T}, a_{t+1:T} \sim \rho^{\pi}_{T, q(s_{t+1}|x_{t+1}, s_t, a_t)}} [r(s_\tau, a_\tau)] \tag{3}$$

$$\geq \bar{\mathbb{E}}_{s_{t+1:T}, a_{t+1:T} \sim \rho^{\pi}_{\widehat{T}, q(s_{t+1}|x_{t+1}, s_t, a_t)}} [\widetilde{r}(s_\tau, a_\tau)] = \mathbb{E}_{s_{t+1:T}, a_{t+1:T} \sim \widehat{q}} \left[\sum_{\tau=t+1}^{T} \widetilde{r}(s_\tau, a_\tau) \right] \tag{4}$$

where $r(s_\tau, a_\tau) = \log p(\mathcal{O}_\tau = 1 | s_\tau, a_\tau)$ and Eq. 3 follows from the definition of $\widehat{q}$ and the discounted state-action distribution with the initial state distribution being $q(s_{t+1}|x_{t+1}, s_t, a_t)$ and the inequality in Eq. 4 follows from Eq. 1 and the latent MDP assumption. Now we can derive the ELBO in the uncertainty-penalized POMDP as follows:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{s_{1:T}, a_{t+1:T} \sim q} \Bigg[\sum_{\tau=0}^{t} (\log D(x_{\tau+1}|s_{\tau+1}) - D_{KL}\left(q(s_{\tau+1}|x_{\tau+1}, s_\tau, a_\tau) \| T(s_{\tau+1}|s_\tau, a_\tau)\right)) \tag{5}$$

$$+ \sum_{\tau=t+1}^{T} (r(s_\tau, a_\tau) + \log p(a_\tau) - \log \pi(a_\tau | x_{1:\tau}, a_{1:\tau-1})) \Bigg] \tag{6}$$

$$\geq \mathbb{E}_{s_{1:t}, a_{t+1:T} \sim q} \left[\sum_{\tau=0}^{t} \left(\underbrace{\log D(x_{\tau+1}|s_{\tau+1})}_{\text{reconstruction}} - \underbrace{D_{KL}\left(q(s_{\tau+1}|x_{\tau+1}, s_\tau, a_\tau) \| T(s_{\tau+1}|s_\tau, a_\tau)\right)}_{\text{consistency}} \right) \right] \tag{7}$$

$$+ \mathbb{E}_{s_{t+1:T}, a_{t+1:T} \sim \widehat{q}} \left[\sum_{\tau=t+1}^{T} (\widetilde{r}(s_\tau, a_\tau) + \log p(a_\tau) - \log \pi(a_\tau | x_{1:\tau}, a_{1:\tau-1})) \right] := \widetilde{\mathcal{L}}_{\text{ELBO}} \tag{8}$$

where $p(a_\tau)$ is the prior action distribution and $\tilde{\mathcal{L}}_{\text{ELBO}}$ denotes the ELBO in the uncertainty-penalized POMDP, which turns out to be a lower bound of $\mathcal{L}_{\text{ELBO}}$, the ELBO in the original POMDP. With the uncertainty-penalized ELBO, we optimize the latent dynamics model and the inference model using Eq. 7 (the reconstruction and consistency terms), which can be viewed as offline latent model training, and optimize the policy with Eq. 8, which uses an uncertainty penalized reward similar to MOPO. Next, we will discuss the practical implementation of our model training and policy optimization in Section 4.3.

4.3 Practical Implementation of the Main Algorithm

We now present our practical LOMPO algorithm, outlined in Algorithm 1 in Appendix E, and visualize the whole training pipeline in Figure 1.

Variational Model Training. Since we would like to be able to estimate the uncertainty term in the uncertainty-penalized POMDP, we train an ensemble of latent transition models and use model-disagreement as a proxy. In designing the optimization approach we have two main considerations: (1) since we are training latent models in a learned representation space, we need all of the members of the ensemble to be grounded within the same latent space, (2) minimize additional model complexity and training overhead. Considering these, we optimize the following objective:

$$\sum_{\tau=0}^{T-1} \left[\mathbb{E}_q[\ln D(x_{\tau+1}|s_{\tau+1})] - \mathbb{E}_q D_{KL}[q(s_{\tau+1}|x_{\tau+1}, s_\tau, a_\tau) \| T_\tau(s_{\tau+1}|s_\tau, a_\tau)] \right] \tag{9}$$

Here at each step we sample a random forward transition model T_τ from a fixed set of K models $\{T_1, \ldots, T_K\}$. The inference distribution q is modeled via the standard mean-field approximation as a uni-modal Gaussian distribution and is shared across all time steps. This explicitly grounds all forward models within the same latent space satte representation induced by the inference distribution addressing concern (1) above. Moreover since we use a single forward model at each time step this procedure has the same computational overhead as the regular training of a single variational model, which addresses concern (2).

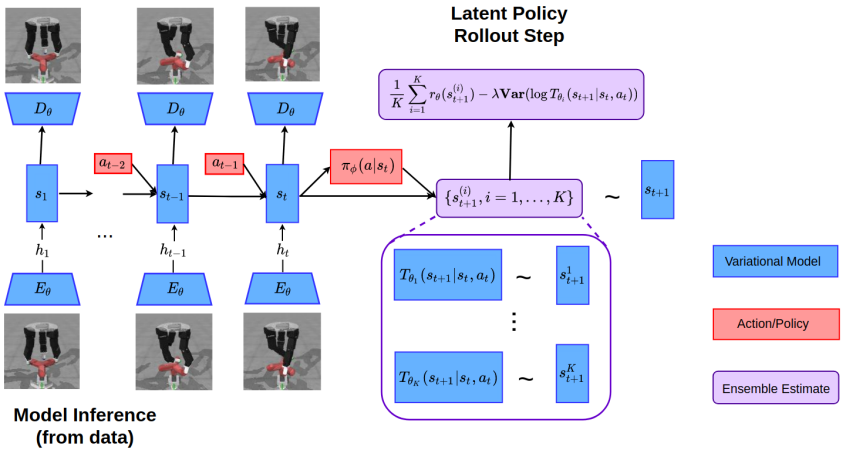

Figure 1: Images are passed through a convolutional encoder E_θ to form a compact representation which are then used along with previous state to infer the current state s_t. The model is trained by reconstructing the images from the latent states through the decoder network D_θ. Latent rollouts are carried by choosing a random transition model $T_{\theta_j}(s_{t+1}|s_t, a_t)$ and rewards are penalized based on ensemble disagreement.

Model-based Policy Optimization. We train a policy and a critic with components $\pi_\phi(a_t|s_t)$ and $Q_\phi(s_t, a_t)$ on top of the latent space representation similar to Lee et al. (2020), however we deploy model inference for the policy as well as the critic, as we cannot do model rollouts in pixel space during training. We maintain two replay buffers $\mathcal{B}_{real}, \mathcal{B}_{sample}$. The real data replay buffer contains transition tuples s_t, a_t, r_t, s_{t+1} from the latent MDP, where states are sampled from the inference distribution $s_{1:T} \sim q(s_{1:T}|x_{1:T}, a_{1:T-1})$ over trajectories from the real dataset $x_{1:T}, r_{1:T}, a_{1:T-1} \sim \mathcal{D}_{env}$. The latent data buffer contains transitions from rolling-out the policy in the model latent space utilizing the ensemble of learned forward models. During rollouts transitions are carried out at each step by picking a a random forward model from the ensemble. As discussed in Section 4.1, the final rewards for the rollout use ensemble estimates and the uncertainty penalty term and are computed as:

$$\widetilde{r}_t(s_t, a_t) = \frac{1}{K}\sum_{i=1}^{K} r_\theta(s_t^{(i)}, a_t) - \lambda u(s_t, a_t) \tag{10}$$

where $s_t^{(i)} \sim T_{\theta_i}(s_{t-1}, a_{i-1})$ are sampled from each forward model and s_t is sampled from $\{s_t^{(i)}, i = 1, \ldots, K\}$. Here $u(s_t, a_t)$ is an estimate of model uncertainty and λ is a penalty parameter. In particular, we pick $u(s_t, a_t)$ as disagreement of the latent model predictions in the ensemble, i.e. the variance of log-likelihoods under the ensemble $u(s_t, a_t) = \mathbf{Var}(\{\log T_{\theta_i}(s_t|s_{t-1}, a_{t-1}), i = 1, \ldots, K\})$, since ensembles have been shown to capture the epistemic uncertainty (Bickel & Freedman, 1981) and also work well for model-based RL in practice (Yu et al., 2020; Kidambi et al., 2020). We used this heuristic to estimate uncertainty as it estimates disagreement across the means of the forward models, as well as the variances. Finally the actor and critic $\pi_\phi(a_t|s_t), Q_\phi(s_t, a_t)$ are trained using standard off-policy training algorithm using batches of equal data mixed from the real and sampled replay buffers, as we find that maintaining a fixed sampling proportion is important in preventing distributional shift in the actor-critic training.

5 Experiments

The goal of our experimentation evaluation is to answer the following questions. (1) Can offline RL reliable scale to realistic robot environments with complex dynamics and interactions? (2) How does LOMPO compare to prior offline model-free RL algorithms and online model-based RL algorithms when learning vision-based control tasks from offline data? (3) How does the quality and size of the dataset affect performance? (4) Can LOMPO be applied to an offline RL task on real robot with raw camera image observations? We answer questions (1), (2) and (3) in Section 5.1 and

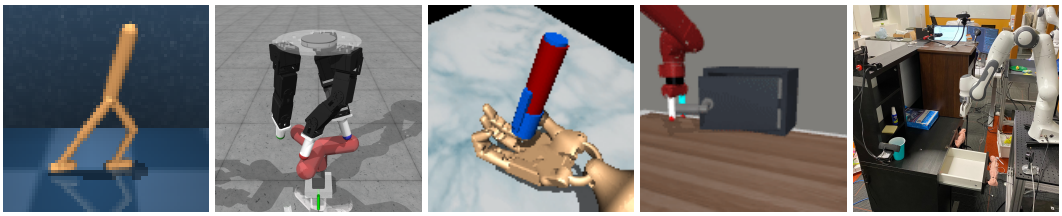

Figure 2: Test environments: Deep Mind Control Walker Walk task - the observations are raw 64×64 images. Robel D'Claw Screw and Adroit Pen tasks observations are raw 128×128 images and robot proprioception. Sawyer Door open environment - the observation space is raw 128×128 images. The observations for the real robot environment are raw 64×64 images from the overhead camera.

Environment	Dataset	LOMPO (ours)	LMBRL	Offline SLAC	CQL	BC
Walker Walk	medium-replay	**74.9**	44.7	-0.1	8.0	5.3
Walker Walk	medium-expert	**91.7**	76.3	32.8	14.1	15.6
Walker Walk	expert	**75.8**	24.5	11.3	10.9	11.8
D'Claw Screw	medium-replay	**71.8**	**72.4**	65.9	16.9	11.7
D'Claw Screw	medium-expert	**100.4**	**96.2**	76.3	20.2	27.6
D'Claw Screw	expert	**99.2**	90.8	63.4	18.8	25.2
Adroit Pen	medium-replay	**82.8**	5.2	5.4	10.0	46.7
Adroit Pen	medium-expert	**94.6**	0.0	-1.7	59.3	41.8
Adroit Pen	expert	**96.1**	0.2	-0.4	27.6	45.4
Door Open	medium-expert	**95.7**	0.0	0.0	0.0	72.2
Door Open	expert	0.0	0.0	0.0	0.0	**97.4**

Table 1: Results for the DeepMind Control Walker task, the Robel D'Claw Screw task, Adroit Pen task and the Sawyer Door Open task. The scores are undiscounted average returns normalized to roughly lie between 0 and 100, where a score of 0 corresponds to a random policy, and 100 corresponds to an expert. LOMPO consistently outperforms LMBRL, offline SLAC, CQL and behavioral cloning in almost all settings.

address question (4) in Section 5.2. All implementation details such as model architectures and hyperparameter choices are included in Appendix D.

5.1 Simulated Experiments

Previous offline RL benchmarks are not well-suited for answering questions (1), (2) and (3) above as they largely lack image-based control problems. Thus, to answer those three questions, we design a suite of four simulated image-based offline RL problems, focusing on robotics applications, described in Appendix A and visualized in the four pictures on the left in Figure 2. We also include the visualization of the samples generated from our learned variational model on the four environments in Appendix C. All of the environments and datasets will be open-sourced to allow future work to also study this problem and make direct comparisons.

Comparisons. We compare our proposed method to both model-free and model-based learning algorithms. Our first benchmark is direct behaviour cloning (BC), which has proved to be a strong benchmark in the state-based case (Fu et al. (2020)), from raw observations. We also benchmark against the Conservative Q-Learning model (Kumar et al. (2020)), which is a state of the art offline learning algorithms in the low-dimensional case (Fu et al. (2020)), however we again train it from raw observations. We also train the Stochastic Latent Actor Critic (SLAC) model Lee et al. (2020), a state of the art online learning algorithm from images, however we train fully online.

Results. Results are reported in Table 1. We see that LOMPO achieves high-scores across most high-fidelity simulation environments, using raw observations. Moreover, our proposed model outperforms other model-based learning algorithms across the board and is the only model-based learning algorithm that achieves any success on several environments. Comparing to model-free algorithms, LOMPO still outperforms CQL and behaviour cloning across most environments, with the exception of learning on the expert dataset on the Door Open task. This is a well-known phenomenon when learning dynamics models from narrow expert data. On the other hand, given the thin data distribution and relatively simple dynamics of the task, direct behaviour cloning from images performs well on both the medium-expert and expert dataset. We hypothesize that LMOPO performs well on the D'Claw and Adroit expert datasets, as these environments are relatively stationary, as compared to a robot arm manipulation task, and even actions from a stochastic expert

cover a wide range of the environment dynamics. The question of dataset size (question (3)) has not been extensively studied in offline RL, however we found that this almost as important as the data distribution itself. We believe this is an important question as collection large robot dataset can still be complex and expensive task. We carried additional ablation experiments (Appendix B) and discover that performance for regular model-based and model-free methods can decline drastically as data size decrease, while LMOPO performance remains relatively stable.

5.2 Real Robot Experiments

To answer question (4), we deploy our approach on a desk drawer-closing task with a real Franka Emika Panda robot arm.

Task. The environment consists of a mounted Panda robot arm in front of an Ikea desk cluttered with random distractor objects. The robot arm is initialized randomly above the desk and the drawer is initialized randomly in a open position. The goal of the robot is to navigate to the handle, hook it, and close the drawer. Observations are raw RGB images from a single overhead camera. The complete setup is shown in the rightmost picture in Figure 2.

Dataset. Due to difficulty in training online reinforcement learning on the real robot, we utilize a pre-existing dataset of 1000 trajectories that were collected using a semi-supervised batch exploration algorithm, BEE (Chen et al., 2020). A small balanced dataset of 200 images (0.2% of the full dataset) was manually labeled with whether the drawer is open or closed. Following the set up by Chen et al. (2020), we used this dataset to train a classifier to predict whether the drawer is closed or not. We use the classifier probability as a reward for RL, which leads to a noisy and unstable reward signal, but is indicative of RL in the real world. The dataset was collected and labeled in the context of a different paper (Chen et al., 2020), allowing us to evaluate the ability to reuse existing offline datasets, which is also indicative of real-world problems.

Comparisons. We compare LOMPO, LMBRL, Offline SLAC, and visual foresight. As CQL did not achieve competitive performance across previous environments we did not deploy it on the real robot. Moreover the unsupervised exploration dataset has high variance and consist of mostly non-task centric exploration, which is not suitable for imitation learning, hence we did not experiment with behavioural cloning either.

Table 2: Success rates for the desk drawer-closing task with a real Franka Emika Panda robot.

LOMPO (ours)	LMBRL	Offline SLAC
76.0%	0.0%	0.0%

Results. We carry out 25 rollout evaluations on the real robot and summarize the results in Table 2. Overall 24/25 of the LOMPO agent rollouts successfully navigate to the drawer handle, hook it and push the drawer in, however the agent fully closes the drawer in 19 of the rollouts for a final success rate of **76%**. We hypothesize that the agent does not always close the door as the classifier, used to generate the reward provides wrong predictions when the drawer is only slightly open. As a comparison both the LMBRL and Offline SLAC agents do not manage to even successfully navigate to the correct handle location and achieve a success rate of **0%**. Finally, visual foresight in the same environment was evaluated by Chen et al. (2020). The evaluation set-up by Chen et al. (2020) was identical to that in this paper, except that the robot arm was initialized at the same level as the drawer, which makes the task significantly easier as it shortens the task horizon. In this easier setting, visual foresight achieves a success rate of **65%** (Figure. 8 of Chen et al. (2020)). These experiments suggest that LOMPO's uncertainty estimation and pessimism are critical for good offline RL performance.

6 Conclusion

We present an offline model-based RL algorithm that handles high-dimensional observations with latent dynamics models and uncertainty quantification. We noted that learning a visual dynamics model is challenging and quantifying uncertainty in the pixel space is extremely costly. We address such challenges by learning a latent space and quantify the model uncertainty in the latent space. Our algorithm, LOMPO, penalizes latent states with latent model uncertainty implemented as the latent model ensemble disagreement. LOMPO empirically outperforms previous latent model-based and model-free models in the offline setting on four simulated locomotion and manipulation tasks and one real-world robotic manipulation task.

REFERENCES

Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. Striving for simplicity in off-policy deep reinforcement learning. *arXiv preprint arXiv:1907.04543*, 2019.

Michael Ahn, Henry Zhu, Kristian Hartikainen, Hugo Ponte, Abhishek Gupta, Sergey Levine, and Vikash Kumar. Robel: Robotics benchmarks for learning with low-cost robots, 2019.

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

Ershad Banijamali, Rui Shu, Hung Bui, Ali Ghodsi, et al. Robust locally-linear controllable embedding. In *International Conference on Artificial Intelligence and Statistics*, pp. 1751–1759. PMLR, 2018.

Peter J Bickel and David A Freedman. Some asymptotic theory for the bootstrap. *The annals of statistics*, pp. 1196–1217, 1981.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11621–11631, 2020.

Annie S. Chen, HyunJi Nam, Suraj Nair, and Chelsea Finn. Batch exploration with examples for scalable robotic reinforcement learning, 2020.

Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. In *Conference on Robot Learning*, pp. 885–897, 2019.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2786–2793. IEEE, 2017.

Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 512–519. IEEE, 2016.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. *arXiv preprint arXiv:1812.02900*, 2018.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. *arXiv preprint arXiv:1906.02736*, 2019.

Ali Ghadirzadeh, Atsuto Maki, Danica Kragic, and Mårten Björkman. Deep predictive policy training using reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2351–2358. IEEE, 2017.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

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

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

Dongqi Han, Kenji Doya, and Jun Tani. Variational recurrent models for solving partially observable control tasks. *International Conference on Learning Representations*, 2020.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

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

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.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *arXiv preprint arXiv:2005.05951*, 2020.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. *arXiv preprint arXiv:1911.12247*, 2019.

Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*, pp. 11761–11771, 2019.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*, 2020.

Sascha Lange and Martin Riedmiller. Deep auto-encoder neural networks in reinforcement learning. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2010.

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

Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *arXiv preprint arXiv:2004.14990*, 2020a.

Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *arXiv preprint arXiv:2004.14990*, 2020b.

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

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

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

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

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

Ajay Mandlekar, Jonathan Booher, Max Spero, Albert Tung, Anchit Gupta, Yuke Zhu, Animesh Garg, Silvio Savarese, and Li Fei-Fei. Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1048–1055. IEEE, 2019.

Tatsuya Matsushima, Hiroki Furuta, Yutaka Matsuo, Ofir Nachum, and Shixiang Gu. Deployment-efficient reinforcement learning via model-based offline optimization. *arXiv preprint arXiv:2006.03647*, 2020.

Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. Algaedice: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*, 2019.

Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. In *Advances in Neural Information Processing Systems*, pp. 9191–9200, 2018.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

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

Pratyusha Sharma, Lekha Mohan, Lerrel Pinto, and Abhinav Gupta. Multiple interactions made easy (mime): Large scale demonstrations data for imitation. *arXiv preprint arXiv:1810.07121*, 2018.

Noah Y Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, and Martin Riedmiller. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *arXiv preprint arXiv:2002.08396*, 2020.

Avi Singh, Larry Yang, Kristian Hartikainen, Chelsea Finn, and Sergey Levine. End-to-end robotic reinforcement learning without reward engineering. *arXiv preprint arXiv:1904.07854*, 2019.

Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

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

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

Phillip Swazinna, Steffen Udluft, and Thomas Runkler. Overcoming model bias for robust offline deep reinforcement learning. *arXiv preprint arXiv:2008.05533*, 2020.

Yuval Tassa, Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, and Nicolas Heess. dm$_{c}ontrol$: $Softwareandtasksforcontinuouscontrol$, 2020.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. 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.

Greg Wayne, Chia-Chun Hung, David Amos, Mehdi Mirza, Arun Ahuja1, Agnieszka, Grabska-Barwinska, Jack Rae1, Piotr Mirowski, Joel Z. Leibo, Adam Santoro, Mevlana Gemici, Malcolm Reynolds, Tim Harley, Josh Abramson, Shakir Mohamed, Danilo Rezende, David Saxton, Adam Cain, Chloe Hillier, David Silver, Koray Kavukcuoglu, Matt Botvinick, Demis Hassabis, and Timothy Lillicrap. Unsupervised predictive memory in a goal-directed agent. *arXiv preprint arxiv:1803.10760*, 2018.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. *arXiv preprint arXiv:1910.01741*, 2019.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *arXiv preprint arXiv:2005.13239*, 2020.

Marvin Zhang, Sharad Vikram, Laura Smith, Pieter Abbeel, Matthew Johnson, and Sergey Levine. Solar: Deep structured representations for model-based reinforcement learning. In *International Conference on Machine Learning*, pp. 7444–7453. PMLR, 2019.

Ruiyi Zhang, Bo Dai, Li Lihong, and Dale Schuurmans. Gendice: Generalized offline estimation of stationary values, 2020. *Preprint*.