Efficient Imitation Learning with Conservative World Models

Victor Kolev*   VKOLEV@CS.STANFORD.EDU
Stanford University

Rafael Rafailov*   RAFAILOV@CS.STANFORD.EDU
Stanford University

Kyle Hatch   KYLE.HATCH@TRI.GLOBAL
Toyota Research Institute

Jiajun Wu   JIAJUNWU@CS.STANFORD.EDU
Stanford University

Chelsea Finn   CBFINN@CS.STANFORD.EDU
Stanford University

Abstract

We tackle the problem of policy learning from expert demonstrations without a reward function. A central challenge in this space is that these policies fail upon deployment due to issues of distributional shift, environment stochasticity, or compounding errors. Adversarial imitation learning alleviates this issue but requires additional on-policy training samples for stability, which presents a challenge in realistic domains due to inefficient learning and high sample complexity. One approach to this issue is to learn a world model of the environment, and use synthetic data for policy training. While successful in prior works, we argue that this is sub-optimal due to additional distribution shifts between the learned model and the real environment. Instead, we re-frame imitation learning as a fine-tuning problem, rather than a pure reinforcement learning one. Drawing theoretical connections to offline RL and fine-tuning algorithms, we argue that standard online world model algorithms are not well suited to the imitation learning problem. We derive a principled conservative optimization bound and demonstrate empirically that it leads to improved performance on two very challenging manipulation environments from high-dimensional raw pixel observations. We set a new state-of-the-art performance on the Franka Kitchen environment from images, requiring only 10 demos on no reward labels, as well as solving a complex dexterity manipulation task.

Keywords: Imitation learning, Model-based learning, World models, Fine-tuning

1. Introduction

Learning by demonstration is a natural way for agents to learn complex behaviors. A small set of demonstrations is often easy to obtain via tele-operation from a human expert, while dense reward functions are non-trivial to design, and sparse rewards are challenging for reinforcement learning methods. Demonstrations further alleviate the need for exploration, as they forego both the search problem and the exploration-exploitation trade-off, and instead reduce the task to distribution matching. The enduring approach to learning from demonstrations is Behavior Cloning (BC) (Pomerleau, 1988), which, however, suffers from compounding policy errors, instability due to environment stochasticity, and covariate shift (Ross et al., 2011). Ways to alleviate these shortcomings involve

---

*. Equal contribution.

more expert data – wider expert coverage (Spencer et al., 2021) or directly querying the expert (Ross et al., 2011).

Alternative approaches to BC are inverse RL (IRL) (Finn et al., 2016b; Fu et al., 2018) and adversarial imitation learning (AIL) (Ho and Ermon, 2016; Finn et al., 2016a). Instead of reducing the problem to supervised learning, these algorithms use reinforcement learning with online interactions to match the long-term visitation distribution of the expert, and agents learn to self-correct when they deviate from the expert support. In particular, AIL formulates this problem as a GAN (Goodfellow et al., 2014): a discriminator is trained to distinguish between the expert trajectories and those produced by the policy. The policy acts as a generator, producing rollouts from the environment, and is optimized with RL to maximize likelihood produced by the discriminator that the rollout is from the expert distribution. Both model-free (Ho and Ermon, 2016; Kostrikov et al., 2019; Blondé and Kalousis, 2019) and model-based approaches (Baram et al., 2016; Rafailov et al., 2021) to the RL optimization problem have been developed. In general, these methods deploy existing online policy optimization algorithms with the discriminator-based reward-learning framework described above.

A crucial difference between the RL setting, for which these online policy optimization algorithms have been developed, and imitation learning is the role of exploration. We argue that RL methods carry out excessive exploration when used in the imitation learning setting, and are thus sub-optimal. As the expert distribution is already empirically known and the goal of the agent is to match that distribution, we argue that this setting is much closer to offline pre-training with online fine-tuning, where the agents learns from offline data and fine-tunes with few online interactions. Moreover, this argument is supported by a recent imitation learning method (Garg et al., 2021), which draws connections between IL and conservative Q-learning (Kumar et al., 2020), an offline RL method.

In this work, we develop a conservative model-based policy optimization algorithm for adversarial imitation learning from pixel observations, which we dub *CMIL*, drawing on ideas from offline RL. We argue that conservatism is inherently well-suited to the imitation learning problem. We justify theoretically that a model-based conservative algorithm is appropriate, and show significant improvements on efficiency and stability over prior imitation learning algorithms. We evaluate our method on a set of challenging environments featuring long horizon manipulation and complex dexterity (fig. 1). We further show direct empirical evidence from the environments above for our theoretical bounds and approximations.

We can summarize our contributions as follows: (i) we re-frame imitation learning as offline pretraining with online fine-tuning; (ii) we augment adversarial imitation learning with conservatism, which improves performance; (iii) we present a theoretical derivation of imitation learning as conservative model-based policy optimization; (iv) we provide a framework for reward-free fine-tuning of world model agents.

2. Related Work

Our work sits at the intersection of imitation learning, high-dimensional model-based RL, and offline RL. We will review relevant works from these fields below.

World models and RL   Variational predictive models have demonstrated success in a variety of challenging applications. These works model the high-dimensional observation space as a POMDP

and use a recurrent VAE model to jointly learn a compact latent representation and a forward dynamics model. One line of research (Hu et al., 2022, 2021; Watter et al., 2015; Zhang et al., 2019; Lee et al., 2020) utilizes the model for representation purposes only and uses standard RL, control, or imitation on top of it. Others such as (Ha and Schmidhuber, 2018; Hafner et al., 2020a,b, 2023) use the learned latent dynamics model to train policies entirely on imagined rollouts. A more recent line of work (Hansen et al., 2022b, 2023) learns a dynamics model only using a reward, critic and latent consistency objectives without reconstruction. They then deploy planning-based methods for the actual control part. These line of work was further extended to efficient learning from demonstrations and sparse rewards (Hansen et al., 2022a) and deployed on real robot systems (Lancaster et al., 2023; Feng et al., 2023).

Conservative MBRL   When learning from prior data model-based algorithms need explicit regularization due to model-hacking issues. Model-based offline RL algorithms (Kidambi et al., 2020b; Yu et al., 2020b; Argenson and Dulac-Arnold, 2020; Matsushima et al., 2020; Swazinna et al., 2020; Rafailov et al., 2020a; Yu et al., 2021) also start by pre-training a dynamics model from prior data. However, many of these approaches (Yu et al., 2020b, 2021; Cang et al., 2021) use large off-policy replay buffers, which make them unsuitable for online fine-tuning. Other such as (Kidambi et al., 2020a; Matsushima et al., 2020) use on-policy optimization, which makes them more suitable for fine-tuning, but also does not allow them to make efficient use of prior data for policy optimization. MoDem (Hansen et al., 2022a) uses the core approach of (Hansen et al., 2022b) and makes several modifications to the training pipeline, such as behaviour-cloning pre-training, seed rollouts and prioritized replay to make the fine-tuning approach more efficient. These ideas were further developed in (Lancaster et al., 2023; Feng et al., 2023) which also consider epistemic uncertainty through a critic ensemble to combat model hacking. Alternatively (Rafailov et al., 2023) builds on the DreamerV2 architecture for the online-fine-tuning. It combined online-model based rollouts with Q learning, policy regularization and ensemble-based epistemic uncertainty.

Model-based imitation learning   Many works have tried combining the benefits of imitation learning and the stable, efficient and predictable nature of model-based learning or planning. A line of works have scaled the approach from (Hafner et al., 2019) to realistic driving scenarios (Hu et al., 2021). In particular (Hu et al., 2022) combined model-based representation learning and pure behaviour cloning to achieve state of the art results on the CARLA challenge (Dosovitskiy et al., 2017). Other works (Baram et al., 2016) use model-based optimization with adversarial imitation learning (Ho and Ermon, 2016). This approach was also scaled to realistic driving applications in (Bronstein et al., 2022), which train fully offline, but adds a behaviour cloning regularization term to the policy training. Other works (Yue et al., 2023; DeMoss et al., 2023; Zhang et al., 2023) use techniques from inverse RL and offline RL to train fully offline imitation algorithms with world models. Similar to our setting (Yin et al., 2022) uses online model learning in combination with behaviour cloning and planning with a learned discriminator as a cost function and achieves significant improvement in learning efficiency over prior work (Rafailov et al., 2021).

3. Preliminaries

In this section we introduce the preliminaries of of variational dynamics models and adversarial imitation learning.

POMDP   We consider the problem setting of learning in partially observed Markov decision processes (POMDPs), which can be described with the tuple: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{X}, \mathcal{R}, \mathcal{T}, \mathcal{U}, \gamma)$, where $s \in \mathcal{S}$ is the state space, $a \in \mathcal{A}$ is the action space, $x \in \mathcal{X}$ is the observation space and $r = \mathcal{R}(s, a)$ is a reward function. The state transition dynamics is Markovian and given by the distribution $s' \sim \mathcal{T}(\cdot|s, a)$. Finally, the observations are generated through the observation model $x \sim \mathcal{U}(\cdot|s)$. We do not have access to the underlying dynamics, the true state representation of the POMDP, or the reward function. Following the standard inverse RL framework, the agent is provided with a fixed set of expert demonstrations, which we assume are optimal under the unknown reward function. The agent can interact with the environment and learns a policy $\pi(a_t|x_{\leqslant t})$ that mimics the expert.

Variational Dynamics Models   A common approach in model-based learning for POMDPs is to use a variational recurrent state-space model (RSSM) jointly model the belief space and the (latent) transition dynamics as $\mathcal{T}_\theta$ (Hafner et al., 2019, 2020a). The RSSM is optimized via the standard ELBO bound:

$$\mathbb{E}_{q_\theta}\Big[\sum_{t=1}^{\tau} \underbrace{\log p_\theta(x_t|s_t)}_{\text{reconstruction}} - \underbrace{\mathbb{D}_{KL}(q_\theta(s_t|x_t, s_{t-1}, a_{t-1})||\mathcal{T}_\theta(s_t|s_{t-1}, a_{t-1}))}_{\text{latent forward model}} \Big] \qquad (1)$$

where both the inference distribution q_θ and the latent dynamics model $\mathcal{T}_\theta$ are Gaussian distributions parameterized by neural networks. Following (Rafailov et al., 2020b), we train a latent ensemble of dynamics models $\{\mathcal{T}_{\theta^i}\}_{i=1}^{K}$ by selecting a different member of the ensemble to evaluate the above loss at every time step in the trajectory.

Imitation learning as divergence minimization   In line with prior work, we interpret imitation learning as a divergence minimization problem (Ho and Ermon, 2016; Ghasemipour et al., 2019; Ke et al., 2019). We begin by analyzing the MDP case. Let $\rho_\mathcal{M}^\pi(s, a) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}_\mathcal{M}^\pi(s_t{=}s, a_t{=}a)$ be the discounted state-action visitation distribution of a policy π in MDP $\mathcal{M}$. We can notice by marginalizing over all possible states and actions that this is indeed a valid probability distribution. Then, a divergence minimization objective for imitation learning corresponds to

$$\min_\pi \ \mathbb{D}(\rho_\mathcal{M}^\pi, \rho_\mathcal{M}^E), \qquad (2)$$

where $\rho_\mathcal{M}^E$ is the discounted visitation distribution of the expert policy π^E, and $\mathbb{D}$ is a divergence measure between probability distributions. Essentially, we would like to minimize the divergence between the distributions of expert and policy-induced trajectories. To see why this is a reasonable objective, consider the following proposition:

Proposition 1 *Let $V_\mathcal{M}^\pi$ denote the expected return of a policy π in $\mathcal{M}$. We we can then bound the sub-optimality of any policy π as:*

$$\left| V_\mathcal{M}^\pi - V_{\mathcal{M}'}^{\pi'} \right| \leqslant \frac{2R_{\max}}{1-\gamma} \, \mathbb{D}_{TV}(\rho_\mathcal{M}^\pi, \rho_{\mathcal{M}'}^{\pi'})$$

where $R_{\max}$ is the maximum reward in the underlying MDP and $\mathbb{D}_{TV}$ is total variation distance.

By optimizing the objective in Eq. 2, we directly upper bound the policy under-performance.

Adversarial Imitation Learning   With the divergence minimization viewpoint, we follow the reasoning in (Ho and Ermon, 2016) to optimize the bound in Eq. 2 using the following max-min objective.

$$\max_{\pi} \min_{D_\psi} \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim\rho_{\mathcal{M}}^E} \left[-\log D_\psi(\boldsymbol{s},\boldsymbol{a}) \right] + \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim\rho_{\mathcal{M}}^\pi} \left[-\log\left(1 - D_\psi(\boldsymbol{s},\boldsymbol{a})\right) \right] \tag{3}$$

where D_ψ is a classifier used to distinguish between samples from the expert distribution and the policy generated distribution, also referred to as a "discriminator". Results from Goodfellow et al. (2014) and Ho and Ermon (2016) suggest that the learning objective in Eq. 3 corresponds to the divergence minimization objective in Eq. 2 with Jensen-Shannon divergence. This is a valid optimization objective, since the total variation distance is upper bounded by the Jensen-Shanon divergence through Pinsker's inequality. A challenge of this objective is that the second term requires on-policy samples, which is inefficient and hard to scale to high dimensions. Off-policy algorithms have been proposed, but can no longer guarantee that the induced visitation distribution of the learned policy will match that of the expert.

As an alternative, model-based algorithms can utilize a large number of *imagined* rollouts using the learned dynamics model, with periodic model correction. In addition, learning the dynamics model serves as a rich auxiliary task for state representation learning, making policy learning easier and more sample efficient. Below we justify the model-based approach.

4. Conservative Model-Based Adversarial Imitation Learning

In this section we will justify our choice of model-based RL algorithms and our main theoretical results. Model-based algorithms for RL and IL involve learning an approximate dynamics model $\widehat{\mathcal{T}}$ using environment interactions. The learned dynamics model can be used to construct an approximate MDP $\widehat{\mathcal{M}}$. In our context of imitation learning, learning a dynamics model allows us to generate on-policy samples from $\widehat{\mathcal{M}}$ as a surrogate for samples from $\mathcal{M}$, leading to the objective:

$$\min_{\pi} \quad \mathbb{D}(\rho_{\widehat{\mathcal{M}}}^\pi, \rho_{\mathcal{M}}^E), \tag{4}$$

which can serve as a good proxy to Eq. 2. In other words, we can train the imitation learning algorithm on imagined rollouts inside a learned model, which is trained in an off-policy manner. However, this comes at the cost of potential under-performance, due to model mismatch. This intuition can be captured using the following Theorem.

Theorem 2   *Let $R_{\max}$ be the maximum of the unknown reward in the MDP with unknown dynamics $\mathcal{T}$. For any policy π, we can bound the sub-optimality with respect to the expert policy π^E as:*

$$\left| V_{\mathcal{M}}^{\pi^E} - V_{\mathcal{M}}^\pi \right| \leqslant \frac{2R_{\max}}{1-\gamma} \underbrace{\mathbb{D}_{TV}(\rho_{\widehat{\mathcal{M}}}^\pi, \rho_{\mathcal{M}}^E)}_{\text{distribution mathcing}} + \frac{\gamma \cdot R_{\max}}{(1-\gamma)^2} \underbrace{\mathbb{E}_{\rho_{\widehat{\mathcal{M}}}^\pi} \left[\mathbb{D}_{TV}(\mathcal{T}(s'|s,a), \widehat{\mathcal{T}}(s'|s,a)) \right]}_{\text{model mismatch}} \tag{5}$$

Proof   The proof combines several techniques from prior works on imitation learning and offline model-based RL. We begin with the left-hand side

$$\left| V_{\mathcal{M}}^{\pi^E} - V_{\mathcal{M}}^\pi \right| \leqslant \underbrace{\left| V_{\mathcal{M}}^{\pi^E} - V_{\widehat{\mathcal{M}}}^\pi \right|}_{\text{Term I}} + \underbrace{\left| V_{\widehat{\mathcal{M}}}^\pi - V_{\mathcal{M}}^\pi \right|}_{\text{Term II}}$$

which is a simple application of the triangle inequality. Through Proposition 1 the first term above produces the distribution matching component of the objective in Eq. 5. The second term yields the model mismatch part of the objective via the "simulation lemma" (Abbeel and Ng, 2005):

$$
\begin{aligned}
\left| V_{\widehat{\mathcal{M}}}^{\pi}(\boldsymbol{s}_0) - V_{\mathcal{M}}^{\pi}(\boldsymbol{s}_0) \right| &= \left| \mathbb{E}_{\pi,\widehat{\mathcal{M}}} \left[r(\boldsymbol{s}_0, \boldsymbol{a}_0) + \gamma V_{\widehat{\mathcal{M}}}^{\pi}(\boldsymbol{s}_1) \right] - \mathbb{E}_{\pi,\mathcal{M}} \left[r(\boldsymbol{s}_0, \boldsymbol{a}_0) + \gamma V_{\mathcal{M}}^{\pi}(\boldsymbol{s}_1) \right] \right| \\
&= \gamma \left| \mathbb{E}_{\pi,\widehat{\mathcal{M}}} \left[V_{\widehat{\mathcal{M}}}^{\pi}(\boldsymbol{s}_1) \right] - \mathbb{E}_{\pi,\mathcal{M}} \left[V_{\mathcal{M}}^{\pi}(\boldsymbol{s}_1) \right] \right| \\
&\leqslant \gamma \left| \mathbb{E}_{\pi,\mathcal{M}} \left[V_{\mathcal{M}}^{\pi}(\boldsymbol{s}_1) \right] - \mathbb{E}_{\pi,\widehat{\mathcal{M}}} \left[V_{\mathcal{M}}^{\pi}(\boldsymbol{s}_1) \right] \right| + \gamma \left| \mathbb{E}_{\pi,\widehat{\mathcal{M}}} \left[V_{\mathcal{M}}^{\pi}(\boldsymbol{s}_1) \right] - \mathbb{E}_{\pi,\widehat{\mathcal{M}}} \left[V_{\widehat{\mathcal{M}}}^{\pi}(\boldsymbol{s}_1) \right] \right| \\
&\leqslant \gamma \frac{R_{\max}}{1-\gamma} \mathbb{E}_{\pi} \left[\mathbb{D}_{TV}(\mathcal{T}(\boldsymbol{s}_1|\boldsymbol{s}_0,\boldsymbol{a}_0), \widehat{\mathcal{T}}(\boldsymbol{s}_1|\boldsymbol{s}_0,\boldsymbol{a}_0)) \right] + \gamma \mathbb{E}_{\pi,\widehat{\mathcal{M}}} \left[\left| V_{\mathcal{M}}^{\pi}(\boldsymbol{s}_1) - V_{\widehat{\mathcal{M}}}^{\pi}(\boldsymbol{s}_1) \right| \right]
\end{aligned}
$$

The first equation follows from the definition of the value function, the next inequality is a direct application of the triangle inequality. The final inequality follows from Proposition 1. We can then recursively apply the same reasoning to the final term to obtain the bound:

$$
\left| V_{\widehat{\mathcal{M}}}^{\pi}(\boldsymbol{s}_0) - V_{\mathcal{M}}^{\pi}(\boldsymbol{s}_0) \right| \leqslant \frac{R_{\max}}{1-\gamma} \mathbb{E}_{\pi,\widehat{\mathcal{M}}} \left[\sum_{t=0}^{\infty} \gamma^{t+1} \mathbb{D}_{TV}(\mathcal{T}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t), \widehat{\mathcal{T}}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t)) \right] = \\
\frac{\gamma \cdot R_{\max}}{(1-\gamma)^2} \mathbb{E}_{\rho_{\widehat{\mathcal{M}}}^{\pi}} \left[\mathbb{D}_{TV}(\mathcal{T}(s'|s,a), \widehat{\mathcal{T}}(s'|s,a)) \right]
$$

∎

That is, if we want to use imagined rollouts for on-policy training of the distribution matching objective, as outlined in Section 3, we need to pay the model mismatch cost in potential under-performance. Prior works (Kidambi et al., 2020a; Yu et al., 2020a; Rafailov et al., 2020b) consider the model mismatch objective in the context of fully offline model-based reinforcement learning. Usually such approaches are not applied in online scenarios, or can even be flipped and used as exploration algorithms (Sekar et al., 2020) as they explicitly encourage the agent to remain within parts of the state space with high confidence and hinder exploration. However, in the imitation learning case, we can treat the problem as online fine-tuning, since we have explicit access to data from the expert. This makes the problem similar to the fine-tuning setting in that it does not require extensive exploration, where conservative MBRL has show meaningful improvement recently (Rafailov et al., 2023; Feng et al., 2023).

4.1. Extension to POMDPs

The results of Section 4 were derived under the MDP formulation. In this section we will formulate how they can be translated into the more general POMDP setting, which we explore in this work. Consider the upper bound under Theorem 2. Theorem 1 in (Rafailov et al., 2021) justifies using the belief representation for bounding distribution matching objective in Eq. 5. Here we will extend this interpretation to justify using latent transition models for the model mismatch objective as well. We formulate this in the below Theorem:

Theorem 3 *Consider two POMDPs $\mathcal{M}$ and $\mathcal{M}'$ with identical components, besides potentially different transition models $\mathcal{T}$ and $\mathcal{T}'$. We denote the density over observations at time $t+1$ as $P_{\mathcal{M}}(\cdot|x_{\leqslant t}, a_{\leqslant t})$. The, the following result holds*

$$\mathbb{D}_f(P_{\mathcal{M}}(x_{t+1}|x_{\leqslant t}, a_{\leqslant t}), P_{\mathcal{M}'}(x_{t+1}|x_{\leqslant t}, a_{\leqslant t})) \leqslant \mathbb{D}_f(\mathcal{T}(s'|s,a), \mathcal{T}'(s'|s,a)) \tag{6}$$

where $\mathbb{D}_f$ is an arbitrary f-divergence.

Proof *From the data processing inequality* Cover *(1999), and the definition of a POMDP, it follows that*

$$\mathbb{D}_f(P_{\mathcal{M}}(x_{t+1}|x_{\leqslant t}, a_{\leqslant t}), P_{\mathcal{M}'}(x_{t+1}|x_{\leqslant t}, a_{\leqslant t})) \leqslant \mathbb{D}_f(P_{\mathcal{M}}(s_{t+1}|x_{\leqslant t}, a_{\leqslant t}), P_{\mathcal{M}'}(s_{t+1}|x_{\leqslant t}, a_{\leqslant t})) \tag{7}$$

However, we also have that $P_{\mathcal{M}}(s_{t+1}|x_{\leqslant t}, a_{\leqslant t}) = \mathcal{T}(s_{t+1}|s_t, a_t)$ and $P_{\mathcal{M}'}(s_{t+1}|x_{\leqslant t}, a_{\leqslant t}) = \mathcal{T}'(s_{t+1}|s_t, a_t)$ which gives us the final result. ∎

This result, allows us to justify latent model uncertainty in as an upper bound on the

5. Our Method

Our full algorithm has several components: (i) variational dynamics model (as described in sec. 3), (ii) state-action discriminator, and (iii) the actor-critic policy optimization. We will discuss all of these in more detail. We train the model, discriminator, actor and critic simultaneously.

Adversarial Formulation Directly applying Theorem 1 in (Rafailov et al., 2021), we can bound the objective of Eq. 5 by optimizing the same bound in the learned models' latent belief space. In more detail, we consider sequences of data of the form $\tau = (\boldsymbol{x}_{1:T}, \boldsymbol{a}_{1:T})$. At each agent training step, we infer latent states $\boldsymbol{s}_{1:T}^0 \sim q_\theta(\boldsymbol{s}_{1:T}|\boldsymbol{x}_{1:T}, \boldsymbol{a}_{1:T})$. We also denote $\boldsymbol{a}_{1:T}$ as $\boldsymbol{a}_{1:T}^0$. Using these states as starting points, we use the policy π_ψ to generate H-step rollouts steps with the following notation: $\hat{\boldsymbol{a}}_j^t \sim \pi_\psi(\boldsymbol{a}|\hat{\boldsymbol{s}}_j^t)$ and $\hat{\boldsymbol{s}}_j^{t+1} \sim p_\theta(\boldsymbol{s}|\hat{\boldsymbol{a}}_j^t, \hat{\boldsymbol{s}}_j^t)$. Following standard off-policy learning algorithms, we use critics $\{Q_{\psi^i}\}_{i=1}^2$ and and target networks $\{\bar{Q}_{\psi^i}\}_{i=1}^2$.

We can bound the distribution matching component of Eq. 5 through Pinsker's inequality and follow a standard adversarial approach by training a discriminator (Ghasemipour et al., 2019):

$$\min_{D_\psi} -\frac{1}{T}\mathbb{E}_{\tau \sim \mathcal{D}^E, q_\theta}\left[\sum_{i=1}^T \log D_\psi(\boldsymbol{s}_i^E, \boldsymbol{a}_i^E)\right] - \frac{1}{HT}\mathbb{E}_{\mathcal{T}_\theta, \pi_\psi}\left[\sum_{i=1,t=0}^{T,H-1} \log(1 - D_\psi(\boldsymbol{s}_i^t, \boldsymbol{a}_i^t))\right] \tag{8}$$

where $\boldsymbol{s}_{1:T}^E \sim q_\theta(\boldsymbol{s}_{1:T}|\boldsymbol{x}_{1:T}^T, \boldsymbol{a}_{1:T}^E)$ are the expert's inferred latent states and actions. On-policy samples from the model also give theoretical justification for this discriminator learning objective as the expectation is taken under the current policy. We cannot directly optimize the model mismatch component of Eq. 5, as we cannot directly estimate divergence factor. Instead, following prior work (Yu et al., 2020a; Kidambi et al., 2020a; Rafailov et al., 2020b), we optimize a surrogate objective based on ensemble model disagreement. In particular, we use

$$\mathbb{E}_{\rho_{\widehat{\mathcal{M}}}^\pi}\left[\mathbb{D}_{TV}(\mathcal{T}(s'|s,a), \widehat{\mathcal{T}}(s'|s,a)\right] \approx \mathbb{E}_{\pi, \widehat{\mathcal{T}}}[\text{std}(\{\mu_{\theta^i}(\boldsymbol{s}, \boldsymbol{a})\}_{i=1}^K)] \tag{9}$$

where $\mu_{\theta^i}(s, a)$ is the parameterization of the mean of i-th Gaussian latent model $\mathcal{T}_{\theta^i}(\cdot|s, a)$. The combined final reward for the agent is then:

$$r(s, a) = \log D_\psi(s, a) - \log(1 - D_\psi(s, a)) - \alpha \text{std}(\{\mu_{\theta^i}(s, a)\}_{i=1}^K) \tag{10}$$

where α is a tunable hyper-parameter. We note that this still a fully differentiable function of the state. We can then label all data and model-sampled latent transitions $r_i^t = r(s_i^t, a_i^t)$. However, we use the uncertainty regularization only on the model-generated rollouts and not on the replay buffer data.

Actor-Critic Policy Optimization   Once we have rewards, we can use them to train an actor-critic model. We follow the actor and critic objectives in Rafailov et al. (2023), which work well in the offline pre-training to online fine-tuning setting. In short, the actor is optimized on a joint objective of maximizing the TD(λ) estimates of both model imagined and real environment rollouts (using the reward formulation above): The critic is trained via bellman backup using model-generated rollouts and real data sampled from the environment. Notably we do not use the epistemic uncertainty penalty on real rollouts. Incorporating real environment data serves as strong supervision for the actor and critic, especially since we have access to expert trajectories, and aids in minimizing shift from the expert-induced state-action distribution.

Practical Implementation Notes   We share some notable implementation[1] details that we have employed. We build on the DreamerV2 architecture, keeping all hyperparameters default. Beyond the optimization procedure outlined above, we add a few details to help with training. As in (Hansen et al., 2022a), we initialize the policy with behavior cloning, and then roll out "seed" episodes to add to the replay buffer. Additionally, we add behaviour cloning loss with a hyperparameter β to the actor loss to regularize actor training (Rafailov et al., 2023). Crucially, we add Gaussian noise $\mathcal{N}(0, \sigma^2)$ to the state-action input vector in training the discriminator (eq. 3), which serves to regularize training. We found this to be important in training stable and robust discriminator, and hence an integral part of the algorithm's success. We apply the same regularization to the expert and the policy. While we have experimented with different discriminator training regimes (e.g. using Wasserstein distance), they did not yield a significant improvement in results.

6. Experiments & results

We devise a diverse experimental setup to evaluate our approach, *CMIL*, on the relevant domains – dexterous manipulation (ShadowHand Baoding balls) and long-horizon planning (Franka Kitchen). The first environment consists of manipulating a 20-DoF 24-joint Shadow Hand robot that has to revolve two Baoding balls, and in the latter environment the objective of the task is to use a 7-DoF Franka Emica Panda robot to manipulate four objects in the kitchen in a certain order (open the microwave, move the kettle, turn on the light, open a cabinet); this happens over ≈ 250 time steps, and hence demands long-horizon planning, in addition to learning dexterity. We focus on the high-dimensional learning setting and use only pixel observations for both environments. Note that we have access to action information of the expert demonstrations, i.e. we are targeting the setting where we can extract actions, but it is non-trivial to define a complete state and form an MDP. We

---

1. https://www.github.com/victorkolev/cmil

Figure 1: The suite of environments. **Left**: ShadowHand Baoding Balls. **Right**: Franka Kitchen.

use 10-20 expert demonstrations for each task, which are collected from expert policies[2]. We use the reward functions only as oracles for evaluation, but not at any stage during training.

In the experimental results, we will be looking at performance measured by success rate and sample efficiency of learning. We choose success rate as a metric instead of total reward to be faithful to the reward-free setting, and also because it is more representative of the agent's success in learning the expert distribution. We run three random seeds and present training curves with 95% bootstrap confidence intervals.

Baselines We compare CMIL with three baselines: (i) standard behavior cloning, (ii) P-DAC, an AIL model-free algorithm designed to work from pixel observations, and (iii) VMAIL, a model-based imitation learning algorithm, which does not use conservatism. We also attempt to benchmark with IQL (Garg et al., 2021), but could not stabilize the algorithm for continuous control from pixel observations. With this set of baselines, we cover both standard approaches (BC), and leading model-free and model-based imitation learning algorithms. We use default hyperparameters for the baselines.

Results: performance & sample efficiency Training curves are presented in figure 2. We see CMIL out-performing baselines in both environments, notably achieving close to expert-level performance with few environment interactions (250K steps), demonstrating improved sample efficiency to previous methods which are typically trained on the order of 1M steps.

We note that our algorithm is the only one to solve the ShadowHand Baoding Balls environment. We postulate this is due to the higher dimensionality of the action space, which demands conservatism to constrain the search space. While baselines also solve the Kitchen environment, we note that there is a tangible difference in both stability and sample efficiency between our algorithm and baselines. All architectural hyperparameters have been kept at default from the DreamerV2 architecture (Hafner et al., 2020a), and we only tune the Gaussian noise added to the discriminator in training (σ^2=2.5 tends to work well across the board), the disagreement penalty α (we keep α=10) and the behaviour cloning regularization parameter (β=10). We note the algorithm is quite resilient to the choice of the latter two, but the former is integral to successful discriminator training and hence accurate rewards.

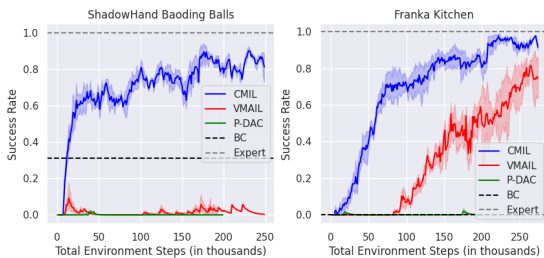2. Demonstration sets are published along with code

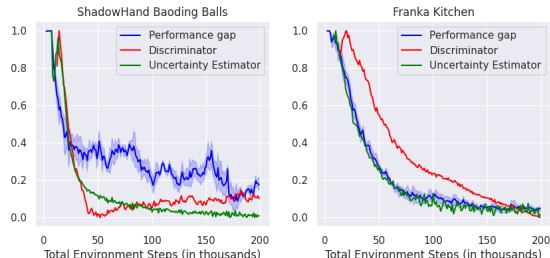Figure 2: Training curve of success rate of our approach, CMIL, vs. baselines.

Figure 3: Empirical estimates of elements from performance gap bound.

Empirical verification of theoretical results   We aim to validate empirically the claims of Theorem 2 and to qualitatively evaluate the discriminator as an approximation of the reward. Theorem 2 states that the performance gap between the expert and the current policy is upper bounded by the sum of (i) the distribution match between the expert and the policy learned in the model, and (ii) the error of the model. We can approximate the distribution matching component as $D_\psi(\hat{\rho}_{\mathcal{M}}^E) - D_\psi(\hat{\rho}_{\hat{\mathcal{M}}}^\pi)$, where $\hat{\rho}^\pi$ is the empirical state-action distribution. On the other hand, we approximate the model mismatch component via an admissible error estimator as in Eq. 9, which we also calculate. We plot those two quantities (normalized min=0, max=1), as well as the oracle reward performance gap, in fig. 3.

Noting the plots for the Franka Kitchen, we see the discriminator is a strong upper bound to the performance gap, while providing a reasonably accurate (dense) reward approximation. On the other hand, in the ShadowHand environment, we see the discriminator diverging initially and then converging back to the performance gap. In both environments, the uncertainty estimator has a similar behavior, indicating that the model mismatch is stably decreasing.

7. Conclusion

In this work we argue that policy optimization algorithms designed for online RL are not well suited to the IRL/AIL setting as they carry out excessive exploration and induce additional distributional shifts. We focus on the model-based case, and argue that conservative models used for offline RL are better suited to the AIL setting. We pose imitation learning as a fine-tuning problem, rather than a purely RL one, and we draw theoretical connections to offline RL and conservative fine-tuning algorithms. We provide a conservative optimization bound, as well as a practical algorithm and evaluate it on challenging manipulation tasks. The proposed algorithm achieves faster and more stable performance as compared to previous imitation learning approaches. In future work we plan to evaluate our method on further domains.

Acknowledgments

References

Pieter Abbeel and Andrew Y. Ng. Exploration and apprenticeship learning in reinforcement learning. In *ICML*, 2005.

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

Nir Baram, Oron Anschel, and Shie Mannor. Model-based adversarial imitation learning. *Conference on Neural Information Processing Systems*, 2016.

Lionel Blondé and Alexandros Kalousis. Sample-efficient imitation learning via generative adversarial nets. *AISTATS*, 2019.

Eli Bronstein, Mark Palatucci, Dominik Notz, Brandyn White, Alex Kuefler, Yiren Lu, Supratik Paul, Payam Nikdel, Paul Mougin, Hongge Chen, Justin Fu, Austin Abrams, Punit Shah, Evan Racah, Benjamin Frenkel, Shimon Whiteson, and Dragomir Anguelov. Hierarchical model-based imitation learning for planning in autonomous driving, 2022.

Catherine Cang, Aravind Rajeswaran, Pieter Abbeel, and Michael Laskin. Behavioral priors and dynamics models: Improving performance and domain transfer in offline rl. *arXiv preprint arXiv:2106.09119*, 2021.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Branton DeMoss, Paul Duckworth, Nick Hawes, and Ingmar Posner. Ditto: Offline imitation learning with world models. 2023.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

Yunhai Feng, Nicklas Hansen, Ziyan Xiong, Chandramouli Rajagopalan, and Xiaolong Wang. Fine-tuning offline world models in the real world. *Conference on Robot Learning*, 2023.

Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *ArXiv Preprint*, 2016a.

Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*, pages 49–58. PMLR, 2016b.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *International Conference on Learning Representations*, 2018.

Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems*, 34: 4028–4039, 2021.

Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. *Conference on Robot Learning*, 2019.

Ian J. Goodfellow, Jean Pouget-Abadie, M. Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 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 Machine Learning*, 2019.

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

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020b.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Nicklas Hansen, Yixin Lin, Hao Su, Xiaolong Wang, Vikash Kumar, and Aravind Rajeswaran. Modem: Accelerating visual model-based reinforcement learning with demonstrations. *International Conference on Learning Representations*, 2022a.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. *International Conference on Machine Learning*, 2022b.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control, 2023.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Conference on Neural Information Processing Systems*, 2016.

Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15273–15282, 2021.

Anthony Hu, Gianluca Corrado, Nicolas Griffiths, Zak Murez, Corina Gurau, Hudson Yeo, Alex Kendall, Roberto Cipolla, and Jamie Shotton. Model-based imitation learning for urban driving. *arXiv preprint arXiv:2210.07729*, 2022.

Liyiming Ke, Matt Barnes, W. Sun, Gilwoo Lee, Sanjiban Choudhury, and S. Srinivasa. Imitation learning as f-divergence minimization. *ArXiv*, abs/1905.12888, 2019.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel : Model-based offline reinforcement learning. In *NeurIPS*, 2020a.

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

Ilya Kostrikov, Kumar Krishna Agrawal, Debidatta Dwibedi, Sergey Levine, and Jonathan Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. *International Conference on Learning Representations*, 2019.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Conference on Neural Information Processing Systems*, 2020.

Patrick Lancaster, Nicklas Hansen, Aravind Rajeswaran, and Vikash Kumar. Modem-v2: Visuo-motor world models for real-world robot manipulation, 2023.

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.

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.

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

Rafael Rafailov, Tianhe Yu, A. Rajeswaran, and Chelsea Finn. Offline reinforcement learning from images with latent space models. *ArXiv*, abs/2012.11547, 2020a.

Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. Offline reinforcement learning from images with latent space models. *arXiv preprint arXiv:2012.11547*, 2020b.

Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. Visual adversarial imitation learning using variational models, 2021.

Rafael Rafailov, Kyle Hatch, Victor Kolev, John D. Martin, Mariano Phielipp, and Chelsea Finn:. Moto: Offline pre-training to online fine-tuning for model-based robot learning. *Conference on Robot Learning*, 2023.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. *AISTATS*, 2011.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. *International Conference on Machine Learning*, 2020.

Jonathan Spencer, Sanjiban Choudhury, Arun Venkatraman, Brian Ziebart, and J. Andrew Bagnell. Feedback in imitation learning: The three regimes of covariate shift. *ArXiv Preprint*, 2021.

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

Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *Conference on Neural Information Processing Systems*, 2015.

Zhao-Heng Yin, Weirui Ye, Qifeng Chen, and Yang Gao. Planning for sample efficient imitation learning. *Conference on Neural Information Processing Systems*, 2022.

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

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*, 2020b.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

Sheng Yue, Guanbo Wang, Wei Shao, Zhaofeng Zhang, Sen Lin, Ju Ren, and Junshan Zhang. Clare: Conservative model-based reward learning for offline inverse reinforcement learning. *International Conference on Learning Representations*, 2023.

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

Wenjia Zhang, Haoran Xu, Haoyi Niu, Peng Cheng, Ming Li, Heming Zhang, Guyue Zhou, and Xianyuan Zhan. Discriminator-guided model-based offline imitation learning. *Conference on Robot Learning*, 2023.