DITTO: OFFLINE IMITATION LEARNING WITH WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose DITTO, a fully offline approach to imitation learning which addresses the problem of covariate shift without access to an oracle or any additional online interactions. By unrolling agent policies in the latent space of a learned world model and penalizing drift from expert demonstrations, we can use online reinforcement learning algorithms to learn policies which solve the imitation objective, without access to the underlying environment or reward function. Decoupling policy and world model learning lets us leverage datasets of any quality to learn latent representations which provide a natural reward signal for imitation learning, avoiding the need for complex adversarial or sparse imitation-inducing rewards. Compared to competitive baselines, our method achieves state-of-the-art performance in a variety of challenging environments from pixel observations alone.

1 INTRODUCTION

Generating agents which can capably act in complex environments is challenging. In the most difficult environments, hand-designed controllers are often insufficient and learning-based methods must be used to achieve good performance. When we can exactly specify the goals and constraints of the problem using a reward function, reinforcement learning (RL) offers an approach which has been extremely successful at solving a range of complex tasks, such as the strategy games of Go (Silver et al., 2016), Starcraft (Vinyals et al., 2019), and poker (Brown et al., 2020), and difficult real world control problems like quadrupedal locomotion (Lee et al., 2020), datacenter cooling (Lazic et al., 2018), and chip placement (Mirhoseini et al., 2020). RL lifts the human designer's work from explicitly designing a good policy, to designing a good reward function. However, this optimization often results in policies that maximize reward in undesirable ways that their designers did not intend (Lehman et al., 2018) - the so-called *reward hacking* phenomenon (Krakovna et al., 2020). To combat this, practitioners spend substantial effort observing agent failure modes and tuning reward functions with extra regularization terms and weighting hyperparameters to counteract undesirable behaviors (Wang et al., 2022) (Peng et al., 2017).

Imitation learning (IL) offers an alternative approach to policy learning which bypasses reward specification by directly mimicking the behavior of an expert demonstrator. The simplest kind of IL, behavior cloning (BC), trains an agent to predict an expert's actions from observations, then acts on these predictions at test time. This approach fails to account for the sequential nature of decision problems, since decisions at the current step affect which states are seen later. The distribution of states seen at test time will differ from those seen during training unless the expert training data covers the entire state space, and the agent makes no mistakes. This distribution mismatch, or covariate shift, leads to a compounding error problem: initially small prediction errors lead to small changes in state distribution, which lead to larger errors, and eventual departure from the training distribution altogether (Pomerleau, 1989). Intuitively, the agent has not learned how to act under its own induced distribution. This was formalized in the seminal work of Ross & Bagnell (2010), who gave a tight regret bound on the difference in return achieved by expert and learner, which is quadratic in the episode length for BC. Follow-up work in Ross et al. (2011) showed that a linear bound on regret can be achieved if the agent learns online in an interactive setting with the expert.

Online learning typically refers to learning while acting *in the real environment*, while on-policy learning refers to the strong assumption of some RL algorithms that the data used in the current update step is generated by the current policy. Online and on-policy learning are not usually differ-

entiated because RL is typically studied in safe virtual environments like video games and physical simulations - but this differentiation is significant when training embodied robotic agents, where learning in the real world can be unsafe, time-consuming, or otherwise infeasible.

Ha & Schmidhuber (2018) propose a two-stage approach to policy learning, where agents first learn to predict the environment dynamics with a recurrent neural network called a "world model" (WM), and then learn the policy inside the WM alone. This approach is desirable since it enables on-policy learning *offline*, given the existence of the world model. Similar model-based learning methods have recently achieved success in standard online RL settings (Hafner et al., 2021), and impressive zero-shot transfer of policies trained solely in the WM to physical robots (Wu et al., 2022).

In this paper, we propose an imitation learning algorithm called **Dream Imitation (DITTO)**, which addresses the problem of compounding error due to covariate shift by training an agent using on-policy RL inside a learned world model. Specifically, we define a reward which measures divergence between the agent and expert demonstrations in the latent space of the world model, and show that optimizing this reward with RL induces imitation on the expert without any online interaction. We discuss the relationship between our method and the imitation learning as divergence minimization framework (Ghasemipour et al., 2019), and show that our method optimizes a similar bound without requiring adversarial training. We compare our method against BC and GAIL (Ho & Ermon, 2016) (adapted for the offline setting), and show that we achieve better performance and sample efficiency in challenging Atari environments from pixels alone.

Our main contributions are summarized as follows:

- We demonstrate the first fully offline model-based imitation learning method that achieves strong performance on Atari from pixels, and show that our agent outperforms competitive baselines adapted to the offline setting.
- We find that the latent space of learned world models provides a natural reward signal for imitation learning, and propose a latent-matching intrinsic reward which compares favorably against commonly used adversarial and sparse formulations.
- We advocate for the use of world models as offline simulators when demonstration data is abundant but online learning is infeasible.

2 Related Work

2.1 Behavior Cloning

Imitation learning algorithms can be classified according to the set of resources needed to produce a good policy. Ross et al. (2011) give strong theoretical and empirical results in the online interactive setting, which assumes that we can both learn while acting online in the real environment, and that we can interactively query an expert policy to e.g. provide the learner with the optimal action in the current state. Follow-up works have progressively relaxed the resource assumptions needed to produce good policies. Sasaki & Yamashina (2021) show that the optimal policy can be recovered with a modified form of BC when learning from imperfect demonstrations, given a constraint on the expert sub-optimality bound. Brantley et al. (2020) study covariate shift in the online, non-interactive setting, and demonstrate an approximately linear regret bound by jointly optimizing the BC objective with a novel policy ensemble uncertainty cost, which encourages the learner to return to and stay in the distribution of expert support. They achieve this by augmenting the BC objective with the following uncertainty cost term:

$$\text{Var}_{\pi \sim \Pi_E}\left(\pi(a|s)\right) = \frac{1}{E}\sum_{i=1}^{E}(\pi_i(a|s) - \frac{1}{E}\sum_{j=1}^{E}\pi_j(a|s))^2 \tag{1}$$

This term measures the total variance of a policy ensemble $\Pi_E = \{\pi_1, ..., \pi_E\}$ trained on disjoint subsets of the expert data. They optimize the combined BC plus uncertainty objective using standard online RL algorithms, and show that this mitigates covariate shift.

2.2 INVERSE REINFORCEMENT LEARNING

Inverse reinforcement learning (IRL) can achieve improved performance over BC by learning a reward signal for which the expert demonstrations are optimal, then optimizing that reward using RL. While IRL methods have demonstrated strong performance in a number of domains, they come with their own set of challenges: the reward is typically under-specified by the expert demonstrations and requires additional ad-hoc constraints (Ziebart et al., 2008), and the most common frameworks use an adversarial setting to estimate the reward (Ho & Ermon, 2016) (Fu et al., 2018) (Finn et al., 2016), which can lead to instability and difficult training in practice.

A recent line of work treats IRL as *divergence minimization*: instead of directly copying the expert actions, they minimize a divergence measure between expert and learner state distributions

$$\min_{\pi} \mathbb{D}\left(\rho^{\pi}, \rho^{E}\right) \tag{2}$$

where $\rho^{\pi}(s,a) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^{t}P(s_t = s, a_t = a)$ is the discounted state-action distribution induced by π, and $\mathbb{D}$ is a divergence measure between probability distributions. The popular GAIL algorithm (Ho & Ermon, 2016) constructs a minimax game in the style of GANs (Goodfellow et al., 2014) between the learner policy π, and a discriminator D_{ψ} which learns to distinguish between expert and learner state distributions

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

This formulation minimizes the Jensen-Shannon divergence between the expert and learner policies, and bounds the expected return difference between agent and expert. However, Wang et al. (2019) point out that adversarial reward learning is inherently unstable since the discriminator is always trained to penalize the learner state-action distribution, even if the learner has converged to the expert policy. This finding is consistent with earlier work (Brock et al., 2019) which observed discriminator overfitting, necessitating early stopping to prevent training collapse. Multiple works have reported difficulty getting GAIL to work reliably in pixel-based observation environments (Brantley et al., 2020) (Reddy et al., 2020).

To combat problems with adversarial training, Wang et al. (2019) and Reddy et al. (2020) consider reducing IL to RL on an intrinsic reward

$$r(s,a) = \begin{cases} 1 & \text{if } (s,a) \in \mathcal{D}^{E} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $\mathcal{D}^{E}$ is the expert dataset. While this sparse formulation is impractical e.g. in continuous action settings, they show that a generalization of the intrinsic reward using support estimation by random network distillation (Burda et al., 2019) results in stable learning that matches the performance of GAIL without the need for adversarial training. Ciosek (2022) showed that this formulation is equivalent to divergence minimization under the total variation distance, and produced a bound on the difference in extrinsic reward achieved between the expert and a learner trained with this approach.

2.3 OFFLINE LEARNING

Kumar et al. (2019) showed how naïve application of Bellman backups in off-policy RL results in incorrect, overly optimistic value estimation. These off-policy bootstrapping errors are further compounded in the model-based setting, since states where either the learned dynamics or reward functions generalize poorly will be found and incorrectly backed out by a naïve Q-learner. This is acceptable in the online setting, since areas of state space which the agent is overly-optimistic about will tend to be visited, resulting in natural corrections to inaccuracies; but the offline setting does not share this property. To counteract these problems, offline RL methods either constrain the learner policy to stay on the expert distribution directly (Wu et al., 2019), or use pessimistic methods to discourage value estimates from becoming too large out of distribution (Kumar et al., 2020) (Rigter et al., 2022).

2.4 WORLD MODELS

World models have recently emerged as a promising approach to model-based learning. Ha & Schmidhuber (2018) defined the prototypical two-part model: a variational autoencoder (VAE) is

trained to reconstruct observations from individual frames, while a recurrent state-space model (RSSM) is trained to predict the VAE encoding of the next observation, given the current latent state and action. World models can be used to train agents entirely inside the learned latent space, without the need for expensive decoding back to the observation space. Hafner et al. (2020) introduced Dreamer, an RL agent which is trained purely in the latent space of the WM, and successfully transfers to the true environment at test-time. Wu et al. (2022) showed that the same approach can be used to simultaneously learn a model and agent policy to control a physical quadrupedal robot online, with none of the usual control errors usually associated with transferring policies trained only in simulation to a physical system (Hwangbo et al., 2019).

In this work, we propose the use of world models to address a number of common problems in imitation learning. Intrinsic rewards which induce imitation learning, like those introduced in Reddy et al. (2020) and Wang et al. (2019), can pose challenging online learning problems, since the rewards are sparse or require tricky additional training procedures to work in high-dimensional observation spaces. Similarly, approaches like GAIL (Ho & Ermon, 2016) and AIRL (Fu et al., 2018) require adversarial on-policy training that is difficult to make work in practice. In contrast, our approach remedies both the online learning and reward specification problems by performing safe offline policy learning solely inside the compact latent space of the world model, and uses a natural divergence measure as reward: distance between learner and expert in the world model latent space. This provides a conceptually simple and dense reward signal for imitation by reinforcement learning, which we find outperforms competitive approaches in data efficiency and asymptotic performance.

3 DREAM IMITATION

We study imitation learning in a partially observable Markov decision process (POMDP) with discrete time-steps and actions, and high dimensional observations generated by an unknown environment. The POMDP $\mathcal{M}$ is composed of 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, γ is the discount factor, and $r = \mathcal{R}(s, a)$ is the reward function. The transition dynamics are Markovian, and given by $s_{t+1} \sim \mathcal{T}(\cdot \mid s_t, a_t)$. The agent does not have access to the underlying states, and only receives observations represented by $x_t \sim \mathcal{U}(\cdot \mid s)$. The goal is to maximize the discounted sum of extrinsic (environment) rewards $\mathbb{E}[\Sigma_t \gamma^t r_t]$, which the agent does not have access to.

Training proceeds in two parts: we first learn a world model from recorded sequences of observations, then train an actor-critic agent to imitate the expert in the world model. The latent dynamics of the world model define a fully observable Markov decision process (MDP), since the model states $\hat{s}_t$ are Markovian. Model-based rollouts always begin from an observation drawn from the expert demonstrations, and continue for a fixed set of time steps H, the agent training horizon. The agent is rewarded for matching the latent trajectory of the expert.

3.1 PRELIMINARIES

We show that bounding the learner-expert state distribution divergence in the world model also bounds their return difference in the actual environment, and connect our method to the *IL as divergence minimization* framework (Ghasemipour et al., 2019). Rafailov et al. (2021) showed that for a learned dynamics model $\widehat{\mathcal{T}}$ whose total variation from the true transitions is bounded such that $\mathbb{D}_{\text{TV}}(\mathcal{T}(s,a), \widehat{\mathcal{T}}(s,a)) \leq \alpha \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}$ and $R_{\text{max}} = \max_{(s,a)} \mathcal{R}(s,a)$ then

$$\left|\mathcal{J}(\pi^E, \mathcal{M}) - \mathcal{J}(\pi, \mathcal{M})\right| \leq \underbrace{\alpha \frac{R_{\text{max}}}{(1-\gamma)^2}}_{\text{learning error}} + \underbrace{\frac{R_{\text{max}}}{1-\gamma} \mathbb{D}_{\text{TV}}\left(\rho^E_{\mathcal{M}}, \rho^\pi_{\widehat{\mathcal{M}}}\right)}_{\text{adaptation error}} \tag{5}$$

where $\mathcal{J}(\pi, \mathcal{M})$ is the expected return of policy π in MDP $\mathcal{M}$. This implies the difference between the expert return and the learner return in the true environment is bounded by two terms, 1) a term proportional to the model approximation error α, which could in principle be reduced with more data, and 2) a model domain adaptation error term, which captures the generalization error of a model trained under data from one policy, and deployed under another. Rafailov et al. (2021) also show that bounding the divergence between *latent* distributions upper bounds the true state distribution divergence. Formally, given a latent representation of the transition history $z_t = q(x_{\leq t}, a_{<t})$

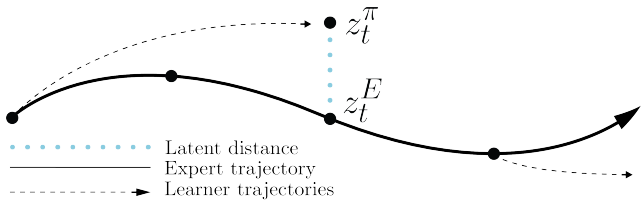

Figure 1: The learner begins from random expert latent states during training, and generates on-policy latent trajectories in the world model. The intrinsic reward 7 encourages the learner to recover from its mistakes over multiple time steps to match the expert trajectory.

and a belief distribution $P(s_t \mid x_{\leq t}, a_{<t}) = P(s_t \mid z_t)$, then if the policy conditions only on the latent representation z_t such that the belief distribution is independent of the current action $P(s_t \mid z_t, a_t) = P(s_t \mid z_t)$, then the divergence between the latent state distribution of the expert and learner upper bounds the divergence between their true state distribution:

$$\mathbb{D}_f(\rho^\pi_{\mathcal{M}}(x,a) \parallel \rho^E_{\mathcal{M}}(x,a)) \leq \mathbb{D}_f(\rho^\pi_{\mathcal{M}}(s,a) \parallel \rho^E_{\mathcal{M}}(s,a)) \leq \mathbb{D}_f(\rho^\pi_{\mathcal{M}}(z,a) \parallel \rho^E_{\mathcal{M}}(z,a)) \tag{6}$$

Where $\mathbb{D}_f$ is a generic f-divergence , e.g. KL or TV. This result, along with equation 5, suggests that minimizing divergence in the model latent space is sufficient to bound the expected expert-learner return difference.

Reward To bound expert-learner state distribution divergences, prior approaches have focused on sparse indicator function rewards (Ciosek, 2022), or adversarial reward learning (Ghasemipour et al., 2019). We propose a dense intrinsic reward function r^{int} which penalizes the divergence between the expert and learner latent state densities *at the same time-step*, respectively z_t^E and z_t^π as shown in Figure 1. To measure divergence in latent space, natural choices include negative L_2 distance or cosine-similarity, however we found the max-cosine objective proposed by Hafner et al. (2022) performed best:

$$r_t^{int}(z_t^E, z_t^\pi) = \frac{z_t^E \cdot z_t^\pi}{\max(\|z_t^E\|, \|z_t^\pi\|)^2} \tag{7}$$

Unlike L_2, this formulation is independent of the scale of the underlying feature space, and improves upon cosine-similarity by penalizing magnitude mismatch, in addition to angular mismatch. Empirically we compute this reward over both a continuous and discrete latent channel, but restrict our analysis to the discrete case. The discrete latent densities produced by our world model directly parameterize a categorical distribution, so minimizing the L_2 distance between expert and learner latent state densities is equivalent to minimizing the KL divergence between their corresponding distributions. Since maximizing the max-cosine objective is equivalent to minimizing the L_2 norm between two vectors, a return lower bound on the max-cosine objective corresponds to an upper bound on the expert-learner latent distribution divergence. As above, this upper bounds the state divergence which in turn bounds the return difference between the expert and learner.

3.2 World Model

Dataset World model training can be performed using datasets generated by policies of any quality, since the model only predicts transition dynamics. The transition dataset is composed of N episodes e_n of sequences of observations x_t, actions a_t: $\mathcal{D} = \{(x_t, a_t)_{t=0}^{\|e_n\|} \mid n \in N\}$.

Model architecture We adapt the architecture proposed by Hafner et al. (2021), which is composed of an image encoder, a recurrent state-space model (RSSM) which learns the transition dynamics, and a decoder which reconstructs observations from the compact latent states. The encoder uses a convolutional neural network (CNN) to produce representations, while the decoder is a transposed CNN. The RSSM predicts a sequence of length T deterministic recurrent states $(h_t)_{t=0}^T$, each of which are used to parameterize two distributions over stochastic hidden states. The stochastic posterior state z_t is a function of the current observation x_t and recurrent state h_t, while the stochastic prior state $\hat{z}_t$ is trained to match the posterior without access to the current observation. The current observation is reconstructed from the full model state, which is the concatenation of the

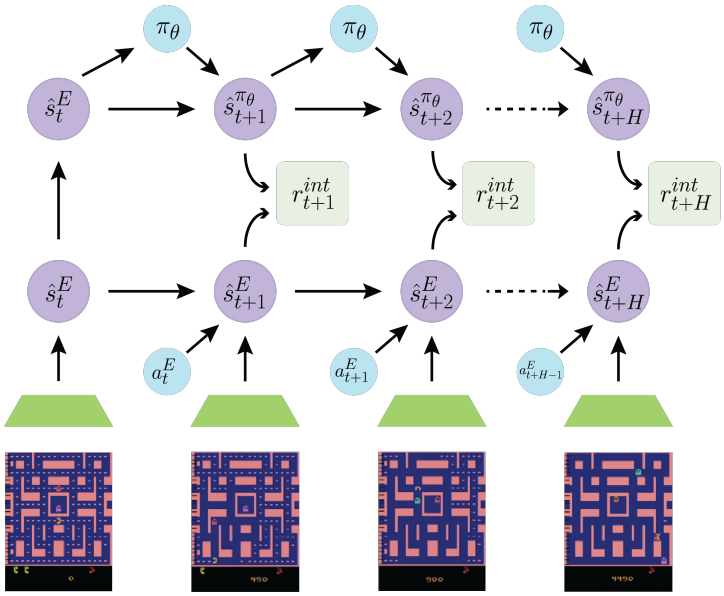

Figure 2: Policy training episodes begin from arbitrary expert latent states $\hat{s}_t^E$, and continue for H steps. The learner policy π_θ predicts an action from the current model state $\hat{s}_t$. The current model state and policy action are used to predict the next model state. The intrinsic reward r^{int} 7 measures the agreement between the current expert and policy model states. At test-time, the policy is composed with the world model encoder and RSSM to act from pixel observations alone.

deterministic and stochastic states $\hat{s}_t = (h_t, z_t)$. The RSSM components are:

$$\begin{aligned}
&\text{Model state:} && \hat{s}_t = (h_t, z_t) \\
&\text{Recurrent state:} && h_t = f_\phi(\hat{s}_{t-1}, a_{t-1}) \\
&\text{Prior predictor:} && \hat{z}_t \sim p_\phi(\hat{z}_t \mid h_t) \\
&\text{Posterior predictor:} && z_t \sim q_\phi(z_t \mid h_t, x_t) \\
&\text{Image reconstruction:} && \hat{x}_t \sim p_\phi(\hat{x}_t \mid \hat{s}_t)
\end{aligned} \tag{8}$$

All components are implemented as neural networks, with a combined parameter vector ϕ. Since the prior model predicts the current model state using only the previous action and recurrent state, without using the current observation, we can use it to learn behaviors without access to observations or decoding back into observation space. The prior and posterior models predict categorical distributions which are optimized with straight-through gradient estimation (Bengio et al., 2013). All components of the model are trained jointly with a modified ELBO objective:

$$\min_\phi \mathbb{E}_{q_\phi(z_{1:T} \mid a_{1:T}, x_{1:T})} \left[\sum_{t=1}^{T} -\log p_\phi(x_t \mid \hat{s}_t) + \beta D_{\text{KL-B}}(q_\phi(z_t \mid \hat{s}_t) \parallel p_\phi(\hat{z}_t \mid h_t)) \right] \tag{9}$$

where $D_{\text{KL-B}}(q \parallel p)$ denotes KL balancing (Hafner et al., 2021), which is used to control the regularization of prior and posterior towards each other with a parameter δ,

$$D_{\text{KL-B}}(q \parallel p) = \delta \underbrace{D_{\text{KL}}(q \parallel sg(p))}_{\text{posterior regularizer}} + (1-\delta) \underbrace{D_{\text{KL}}(sg(q) \parallel p)}_{\text{prior regularizer}} \tag{10}$$

and $sg(\cdot)$ is the stop gradient operator. The idea behind KL balancing is that the prior and posterior should not be regularized at the same rate: the prior should update more quickly towards the posterior, which encodes strictly more information.

3.3 Agent

Agent architecture The agent is composed of a stochastic actor which samples actions from a learned policy with parameter vector θ, and a deterministic critic which predicts the expected

discounted sum of future rewards the actor will achieve from the current state with parameter vector ψ. Both the actor and critic condition only on the current model state $\hat{s}_t$, which is Markovian:

$$\begin{aligned} &\text{Actor:} \quad a_t \sim \pi_\theta(a_t \mid \hat{s}_t) \\ &\text{Critic:} \quad v_\psi(\hat{s}_t) \approx \mathbb{E}_{\pi_\theta, p_\phi}[\Sigma_{t=0}^{H} \gamma^t r_t] \end{aligned} \tag{11}$$

We train the critic to regress the λ-target (Sutton & Barto, 2005)

$$V_t^\lambda = r_t + \gamma \left((1-\lambda) v_\psi(\hat{s}_{t+1}) + \lambda V_{t+1}^\lambda \right), \quad V_{t+H}^\lambda = v_\psi(\hat{s}_{t+H}) \tag{12}$$

which lets us control the temporal-difference (TD) learning horizon with the hyperparameter λ. Setting $\lambda = 0$ recovers 1-step TD learning, while $\lambda = 1$ recovers unbiased Monte Carlo returns, and intermediate values represent an exponentially weighted sum of n-step returns. In practice we use $\lambda = 0.95$. To train the critic, we regress the λ-target directly with the objective:

$$\min_\psi \mathbb{E}_{\pi_\theta, p_\phi} \left[\sum_{t=1}^{H-1} \tfrac{1}{2} (v_\psi(\hat{s}_t) - sg(V_t^\lambda))^2 \right] \tag{13}$$

There is no loss on the last time step since the target equals the critic there. We follow Mnih et al. (2015), who suggest using a copy of the critic which updates its weights slowly, called the target network, to provide the value bootstrap targets.

The actor is trained to maximize the discounted sum of rewards predicted by the critic. We train the actor to maximize the same λ-target as the critic, and add an entropy regularization term to encourage exploration and prevent policy collapse. We optimize the actor using REINFORCE gradients (Williams, 2004) and subtract the critic value predictions from the λ-targets for variance reduction. The full actor loss function is:

$$\mathcal{L}(\theta) = \mathbb{E}_{\pi_\theta, p_\phi} \left[\sum_{t=1}^{H-1} \underbrace{-\log \pi_\theta(a_t \mid \hat{s}_t) sg(V_t^\lambda - v_\psi(\hat{s}_t))}_{\text{reinforce}} - \underbrace{\eta H(\pi_\theta(\hat{s}_t))}_{\text{entropy regularizer}} \right] \tag{14}$$

Algorithm Learning proceeds in two phases: First, we train the WM on all available demonstration data using the ELBO objective 9. Next, we encode expert demonstrations into the world model latent space, and use the on-policy actor critic algorithm described above to optimize the intrinsic reward 7, which measures the divergence between agent and expert over time in latent space. In principle, any on-policy RL algorithm could be used in place of actor-critic. We describe the full procedure in Algorithm 1.

Algorithm 1 Dream Imitation (DITTO)

Require demonstration data $\mathcal{D} = \left\{ (x_t, a_t, x_{t+1})_{t=0}^{\|e_n\|} \mid n \in N \right\}$
Initialize world model parameters ϕ
while *not converged* **do** ▷ World model learning
    Draw B_{wm} transition sequences $\{(x_t, a_t, x_{t+1})_{t=k}^{k+L}\} \sim \mathcal{D}$
    Compute all sequential RSSM components according to eqn 8
    Update ϕ with ELBO loss 9
end while
Initialize actor and critic parameters θ, ψ
while *not converged* **do** ▷ Agent training
    Draw B_{ac} expert latent state sequences $(\hat{s}_\tau^E) \sim \hat{\mathcal{D}}^E$
    Generate trajectories $(\hat{s}_\tau^\pi, a_\tau)_{\tau=t}^{t+H}$ with $a_\tau \sim \pi_\theta(\cdot \mid \hat{s}_\tau)$
    Compute rewards $r_\tau^{\text{int}}(\hat{s}_\tau, \hat{s}_\tau^E)$ and values $v_\psi(\hat{s}_\tau)$
    Compute λ-returns $V_t^\lambda = r_t + \gamma \left((1-\lambda) v(\hat{s}_{t+1}) + \lambda V_{t+1}^\lambda \right), \quad V_{t+H}^\lambda = v(\hat{s}_{t+H})$
    Update critic on λ-targets: $\sum_{t=1}^{H-1} \frac{1}{2} (v_\psi(\hat{s}_t) - sg(V_t^\lambda))^2$
    Update actor with eqn 14
end while

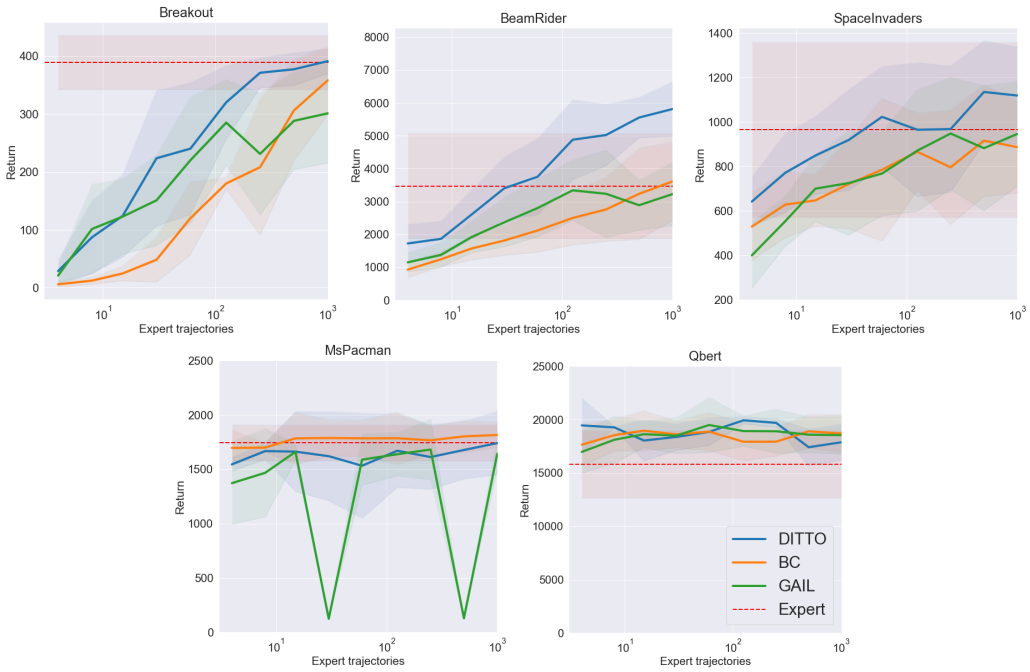

Figure 3: Results on five Atari environments from pixels, with fixed horizon $H = 15$. The plots show mean final performance across 5 runs with 20 validation simulations, and shaded regions show $\pm$1SD. The experts are strong pre-trained PPO agents from the RL Baselines3 Zoo.

4 Experiments

To the best of our knowledge, we are the first to study completely offline imitation learning without behavior cloning in high-dimensional observation environments. Prior works generally focus on improving behavior cloning (Sasaki & Yamashina, 2021), or study a mixed setting with limited online interactions allowed (Rafailov et al., 2021) (Kidambi et al., 2021).

To test the performance of our algorithm, we compare DITTO against two strong baselines which we modify to use the same learned world model: 1) We train a BC agent to predict expert actions from world model latent states, and 2) we adapt GAIL to the offline setting by training both the generator policy and discriminator in the world model. The world model adapted baselines are denoted D-BC and D-GAIL respectively, since the world model is based on the Dreamer (Hafner et al., 2020) architecture. At test-time, agent policies are composed with the world model encoder and RSSM to convert high-dimensional observations into latent representations. All policies in our experiments use an identical multi-layer perceptron (MLP) architecture for fair comparison in terms of the policies' representation capacity.

We evaluate our algorithm and baselines on 5 Atari environments, using strong PPO agents (Schulman et al., 2017) from the RL Baselines3 Zoo (Raffin, 2020) as expert demonstrators. To train the world models, we generate 1000 episodes from a pre-trained policy, either PPO or advantage actor-critic (A2C) (Mnih et al., 2016). We found that the quality of the policy used to generate world model training episodes did not affect final performance, so we report results from world models trained with 1000 A2C demonstrations, which achieve substantially lower reward compared to PPO. For imitation learning, we generate datasets of $N^E = \{4, 8, 15, 30, 60, 125, 250, 500, 1000\}$ expert episodes to train the agent policies in the world model.

The D-BC agent is trained with maximum-likelihood estimation on the expert demonstrations in latent space, with an additional entropy regularization term which we found stabilized learning:

$$L_{BC} = \mathbb{E}_{(\hat{s},a)\sim\hat{\rho}^E}\left[-\log\left(\pi(a|\hat{s})\right) - \eta_{BC} H(\pi(\hat{s}_t))\right] \tag{15}$$

The D-GAIL agent is trained using the adversarial objective from Equation 3, while the DITTO agent optimizes the intrinsic reward defined in Equation 7. DITTO and D-GAIL are both optimized with the actor-critic formulation described in Section 3.3.

In all tested environments, DITTO recovers or exceeds average expert performance, and matches or outperforms the baselines in terms of sample efficiency and asymptotic performance. We train all agents with a fixed horizon of $H = 15$. In contrast to prior works which reported negative results with online GAIL on Atari environments from images (Brantley et al., 2020) (Reddy et al., 2020), we found that D-GAIL was able to consistently learn effective policies, and only exhibited adversarial collapse in the *MsPacman* environment.

5 Conclusion

Addressing covariate shift in imitation learning is a long-standing problem. In this work we proposed DITTO, a method which addresses this problem in the challenging offline setting where no environment interactions are allowed while learning. The offline setting exacerbates covariate shift, since the agent cannot typically estimate its own induced state distribution. Recent strong IL methods such as DRIL (Brantley et al., 2020) and V-MAIL (Rafailov et al., 2021) achieve excellent sample efficiency in terms of expert demonstrations, but still require environment interaction. Our agent consistently recovers expert performance without online interaction in the tested Atari environments from pixel observations, and matches or exceeds the performance of strong baselines which we implement in the world model setting.

DITTO learns by decomposing imitation learning into two parts: First, a world model is learned to approximate the underlying environment dynamics from all available demonstration data, regardless of the quality of the policy which generated it. Next, expert demonstrations are encoded into trajectories in the world model latent space. Finally, DITTO produces on-policy rollouts in the world model, and optimizes an intrinsic reward which measures the agent's drift from expert trajectories. By optimizing this reward with standard online reinforcement learning algorithms, DITTO learns to recover from its own mistakes across multiple time-steps, and match the expert state distribution, which theoretically bounds the agent-expert return difference in the true environment.

Decoupling dynamics and policy learning lets us learn the world model from any historical demonstration data, and provides us with a surprisingly rich signal for imitation learning: the latent distance between demonstrations. This contrasts with recent IL methods which need to employ adversarial (Fu et al., 2018) or sparse rewards Wang et al. (2019) to induce imitation, which can be difficult to train. Furthermore, other methods from reinforcement and imitation learning are orthogonal to and compatible with ours, and could be combined with DITTO to achieve greater performance and sample efficiency, e.g. by adding behavioral cloning rewards or uncertainty costs to DITTO's objective.

6 Reproducibility Statement

In order to ensure reproducibility of the results, all hyperparameters used in our experiments are included in Appendix A. The code for the experiments will be made available for the reviewing process and we intend to release the code for the camera-ready version of the paper.

References

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *ArXiv*, abs/1308.3432, 2013. 6

Kianté Brantley, Wen Sun, and Mikael Henaff. Disagreement-regularized imitation learning. In *ICLR*, 2020. 2, 3, 9

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm. 3

Noam Brown, Anton Bakhtin, Adam Lerer, and Qucheng Gong. Combining deep reinforcement learning and search for imperfect-information games. *ArXiv*, abs/2007.13544, 2020. 1

Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. *ArXiv*, abs/1810.12894, 2019. 3

Kamil Ciosek. Imitation learning by reinforcement learning. In *ICLR*, 2022. URL https://openreview.net/forum?id=1zwleytEpYx. 3, 5

Chelsea Finn, Paul Francis Christiano, P. Abbeel, and Sergey Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *ArXiv*, abs/1611.03852, 2016. 3

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *ArXiv*, abs/1710.11248, 2018. 3, 4, 9

Seyed Kamyar Seyed Ghasemipour, Richard S. Zemel, and Shixiang Shane Gu. A divergence minimization perspective on imitation learning methods. In *CoRL*, 2019. 2, 4, 5

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 3

David R Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *ArXiv*, abs/1809.01999, 2018. 2, 3

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *ArXiv*, abs/1912.01603, 2020. 4, 8

Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *ArXiv*, abs/2010.02193, 2021. 2, 5, 6

Danijar Hafner, Kuang-Huei Lee, Ian S. Fischer, and P. Abbeel. Deep hierarchical planning from pixels. *ArXiv*, abs/2206.04114, 2022. 5

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *NIPS*, 2016. 2, 3, 4

Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019. doi: 10.1126/scirobotics.aau5872. URL https://www.science.org/doi/abs/10.1126/scirobotics.aau5872. 4

Rahul Kidambi, Jonathan Chang, and Wen Sun. Mobile: Model-based imitation learning from observation alone. In *NeurIPS*, 2021. 8

Victoria Krakovna, Jonathan Uesato, Vladimir Mikulik, Matthew Rahtz, Tom Everitt, Ramana Kumar, Zac Kenton, Jan Leike, and Shane Legg. Specification gaming: the flip side of ai ingenuity, 2020. https://www.deepmind.com/blog/specification-gaming-the-flip-side-of-ai-ingenuity, Last accessed on 2022-9-12. 1

Aviral Kumar, Justin Fu, G. Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *NeurIPS*, 2019. 3

Aviral Kumar, Aurick Zhou, G. Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *ArXiv*, abs/2006.04779, 2020. 3

Nevena Lazic, Craig Boutilier, Tyler Lu, Eehern Wong, Binz Roy, M. K. Ryu, and Gregory P. Imwalle. Data center cooling using model-predictive control. In *NeurIPS*, 2018. 1

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5, 2020. 1

Joel Lehman, Jeff Clune, Dusan Misevic, Christoph Adami, Lee Altenberg, Julie Beaulieu, Peter John Bentley, Samuel Bernard, Guillaume Beslon, David M. Bryson, Patryk Chrabaszcz, Nick Cheney, Antoine Cully, Stéphane Doncieux, Fred C. Dyer, Kai Olav Ellefsen, Robert Feldt, Stephan Fischer, Stephanie Forrest, Antoine Frénoy, Christian Gagné, Léni K. Le Goff, Laura M. Grabowski, Babak Hodjat, Frank Hutter, Laurent Keller, Carole Knibbe, Peter Krcah, Richard E.

Lenski, Hod Lipson, Robert MacCurdy, Carlos Maestre, Risto Miikkulainen, Sara Mitri, David E. Moriarty, Jean-Baptiste Mouret, Anh M Nguyen, Charles Ofria, Marc Parizeau, David P. Parsons, Robert T. Pennock, William F. Punch, Thomas S. Ray, Marc Schoenauer, Eric Shulte, Karl Sims, Kenneth O. Stanley, François Taddei, Danesh Tarapore, Simon Thibault, Westley Weimer, Richard A. Watson, and Jason Yosinksi. The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. *Artificial Life*, 26:274–306, 2018. 1

Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim M. Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Sungmin Bae, Azade Nazi, Jiwoo Pak, Andy Tong, Kavya Srinivasa, Will Hang, Emre Tuncer, Ananda Babu, Quoc V. Le, James Laudon, Richard Ho, Roger Carpenter, and Jeff Dean. Chip placement with deep reinforcement learning. *ArXiv*, abs/2004.10746, 2020. 1

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charlie Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015. 7

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016. 8

Xue Bin Peng, Glen Berseth, KangKang Yin, and Michiel van de Panne. Deeploco: dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Trans. Graph.*, 36:41:1–41:13, 2017. 1

Dean A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *NeurIPS*, 1989. 1

Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. Visual adversarial imitation learning using variational models. *ArXiv*, abs/2107.08829, 2021. 4, 8, 9

Antonin Raffin. Rl baselines3 zoo. `https://github.com/DLR-RM/rl-baselines3-zoo`, 2020. 8

Siddharth Reddy, Anca D. Dragan, and Sergey Levine. Sqil: Imitation learning via reinforcement learning with sparse rewards. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=S1xKd24twB`. 3, 4, 9

Marc Rigter, Bruno Lacerda, and Nick Hawes. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *ArXiv*, abs/2204.12581, 2022. 3

Stéphane Ross and J. Andrew Bagnell. Efficient reductions for imitation learning. In *AISTATS*, 2010. 1

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011. 1, 2

Fumihiro Sasaki and Ryota Yamashina. Behavioral cloning from noisy demonstrations. In *ICLR*, 2021. 2, 8

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017. 8

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, L. Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 2016. 1

Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction. *IEEE Transactions on Neural Networks*, 16:285–286, 2005. 7

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, L. Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander Sasha Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom Le Paine, Caglar Gulcehre, Ziyun Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy P. Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, pp. 1–5, 2019. 1

Ruohan Wang, Carlo Ciliberto, Pierluigi Vito Amadori, and Y. Demiris. Random expert distillation: Imitation learning via expert policy support estimation. In *ICML*, 2019. 3, 4, 9

Tsun-Hsuan Wang, Alexander Amini, Wilko Schwarting, Igor Gilitschenski, Sertaç Karaman, and Daniela Rus. Learning interactive driving policies via data-driven simulation. *2022 International Conference on Robotics and Automation (ICRA)*, pp. 7745–7752, 2022. 1

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 2004. 7

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and P. Abbeel. Daydreamer: World models for physical robot learning. *ArXiv*, abs/2206.14176, 2022. 2, 4

Yifan Wu, G. Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *ArXiv*, abs/1911.11361, 2019. 3

Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, 2008. 3

A APPENDIX

Table 1: Hyperparameters

Description	**Symbol**	**Value**
Number of world model training episodes	N	1000
Number of expert training episodes	N^E	$\{4, 8, 15, 30, 60, 125, 250, 500, 1000\}$
World model training batch size	B_{wm}	50
World model training sequence length	L	50
Agent training batch size	B_{ac}	512
Agent training horizon	H	15
Discount factor	γ	0.95
$TD(\lambda)$ parameter	λ	0.95
KL-Balancing weight	β	0.1
KL-Balancing trade-off parameter	δ	0.8
Actor-critic entropy weight	η	5×10^{-2}
Behavior cloning entropy weight	η_{BC}	0.1
Optimizer	-	Adam
All learning rates	-	3×10^{-4}
Actor-critic target network update rate	-	100 steps