Learning Belief Representations for Imitation Learning in POMDPs

Tanmay Gangwani
Dept. of Computer Science
UIUC
gangwan2@uiuc.edu

Joel Lehman
Uber AI Labs
San Francisco, CA 94103
joel.lehman@uber.com

Qiang Liu
Dept. of Computer Science
UT Austin
lqiang@cs.utexas.edu

Jian Peng
Dept. of Computer Science
UIUC
jianpeng@uiuc.edu

Abstract

We consider the problem of imitation learning from expert demonstrations in partially observable Markov decision processes (POMDPs). Belief representations, which characterize the distribution over the latent states in a POMDP, have been modeled using recurrent neural networks and probabilistic latent variable models, and shown to be effective for reinforcement learning in POMDPs. In this work, we investigate the belief representation learning problem for generative adversarial imitation learning in POMDPs. Instead of training the belief module and the policy separately as suggested in prior work, we learn the belief module jointly with the policy, using a task-aware imitation loss to ensure that the representation is more aligned with the policy's objective. To improve robustness of representation, we introduce several informative belief regularization techniques, including multi-step prediction of dynamics and action-sequences. Evaluated on various partially observable continuous-control locomotion tasks, our belief-module imitation learning approach (BMIL) substantially outperforms several baselines, including the original GAIL algorithm and the task-agnostic belief learning algorithm. Extensive ablation analysis indicates the effectiveness of task-aware belief learning and belief regularization. Code for the project is available online[1].

---

[1]https://github.com/tgangwani/BMIL. Part of this work was done while Tanmay was an intern at Uber AI Labs.

1 INTRODUCTION

Recent advances in reinforcement learning (RL) have found successful applications in solving complex problems, including robotics, games, dialogue systems, and recommendation systems, among others. Despite such notable success, the application of RL is still quite limited to problems where the observation-space is rich in information and data generation is inexpensive. On the other hand, the environments in real-world problems, such as autonomous driving and robotics, are typically stochastic, complex and partially observable. To achieve robust and practical performance, RL algorithms should adapt to situations where the agent is being fed noisy and incomplete sensory information. To model these types of environments, partially observable Markov decision processes (POMDPs; Aström (1965)) have been proposed and widely studied. In a POMDP, since the current observation alone is insufficient for choosing optimal actions, the agent's history (its past observations and actions) is encoded into a *belief state*, which is defined as the distribution (representing the agent's beliefs) over the current latent state. Although belief states can be used to derive optimal policies (Kaelbling et al., 1998; Hauskrecht, 2000), maintaining and updating them requires knowledge of the transition and observation models of the POMDP, and is prohibitively expensive for high-dimensional spaces. To overcome this difficulty, several algorithms have been proposed that perform approximate inference of the belief state representation from raw observations, using recurrent neural networks (Guo et al., 2018), variational autoencoders (Igl et al., 2018; Gregor & Besse, 2018), and Predictive State Representations (Venkatraman et al., 2017). After the belief model has been learned, a policy optimization algorithm is then applied to the belief representation to optimize a predefined reward signal.

As an alternative to RL from predefined rewards, imitation learning often provides a fast and efficient way for

training an agent to complete tasks. Expert demonstrations are provided to guide a learner agent to mimic the actions of the expert without the need to specify a reward function. A large volume of work has been done over the past decades on imitation learning for fully observable MDPs, including the seminal work on generative adversarial imitation learning (GAIL, Ho & Ermon (2016)), but there has been little focus on applying these ideas to partially observable environments.

In this paper, we study the problem of imitation learning for POMDPs. Specifically, we introduce a new belief representation learning approach for generative adversarial imitation learning in POMDPs. Different from previous approaches, where the belief state representation and the policy are trained in a decoupled manner, we learn the belief module jointly with the policy, using a task-aware imitation loss which helps to align the belief representation with the policy's objective. To avoid potential belief degeneration, we introduce several informative belief regularization techniques, including auxiliary losses of predicting multi-step past/future observations and action-sequences, which improve the robustness of the belief representation. Evaluated on various partially observable continuous-control locomotion tasks built from MuJoCo, our belief-module imitation learning approach (BMIL) substantially outperforms several baselines, including the original GAIL algorithm and the task-agnostic belief learning algorithm. Extensive ablation analysis indicates the effectiveness of task-aware belief learning and belief regularization.

2 BACKGROUND AND NOTATION

Reinforcement Learning. We consider the RL setting where the environment is modeled as a partially-observable Markov decision process (POMDP). A POMDP is characterized by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{R}, \mathcal{T}, \mathcal{U}, , p(s_0), \gamma)$, where $\mathcal{S}$ is the state-space, $\mathcal{A}$ is the action-space, and $\mathcal{O}$ is the observation-space. The true environment states $s_t \in \mathcal{S}$ are latent or unobserved to the agent. Given an action $a_t \in \mathcal{A}$, the next state is governed by the transition dynamics $s_{t+1} \sim \mathcal{T}(s_{t+1}|s_t, a_t)$, an observation is generated as $o_{t+1} \sim \mathcal{U}(o_{t+1}|s_{t+1})$, and reward is computed as $r_t = \mathcal{R}(r_t|s_t, a_t)$. The RL objective involves maximization of the expected discounted sum of rewards, $\eta(\pi_\theta) = \mathbb{E}_{p_0, \mathcal{T}, \pi}\big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\big]$, where $\gamma \in (0, 1]$ is the discount factor, and $p(s_0)$ is the initial state distribution. The action-value function is $Q^\pi(s_t, a_t) = \mathbb{E}_{p_0, \mathcal{T}, \pi}\big[\sum_{t'=t}^{\infty} \gamma^{t'-t} r(s_{t'}, a_{t'})\big]$. We define the unnormalized γ-discounted state-visitation distribution for a policy π by $\rho_\pi(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t{=}s|\pi)$, where $P(s_t{=}s|\pi)$ is the probability of being in state s at time t, when following policy π and starting state $s_0 \sim$

p_0. The expected policy return $\eta(\pi_\theta)$ can then be written as $\mathbb{E}_{\rho_\pi(s,a)}[r(s, a)]$, where $\rho_\pi(s, a) = \rho_\pi(s)\pi(a|s)$ is the state-action visitation distribution (also referred to as the occupancy measure). For any policy π, there is a one-to-one correspondence between π and its occupancy measure (Puterman, 1994). Using the policy gradient theorem (Sutton et al., 2000), the gradient of the RL objective can be obtained as $\nabla_\theta \eta(\pi_\theta) = \mathbb{E}_{\rho_\pi(s,a)}\big[\nabla_\theta \log \pi_\theta(a|s) Q^\pi(s, a)\big]$.

Imitation Learning. Learning in popular RL algorithms (such as policy-gradients and Q-learning) is sensitive to the quality of the reward function. In many practical scenarios, the rewards are either unavailable or extremely sparse, leading to difficulty in temporal credit assignment (Sutton, 1984). In the absence of explicit environmental rewards, a promising approach is to leverage *demonstrations* of the completed task by experts, and learn to imitate their behavior. Behavioral cloning (BC; Pomerleau (1991)) poses imitation as a supervised-learning problem, and learns a policy by maximizing the likelihood of expert-actions in the states visited by the expert. The policies produced with BC are generally not very robust due to the issue of compounding errors; several approaches have been proposed to remedy this (Ross et al., 2011; Ross & Bagnell, 2014). Inverse Reinforcement Learning (IRL) presents a more principled approach to imitation by attempting to recover the cost function under which the expert demonstrations are optimal (Ng et al., 2000; Ziebart et al., 2008). Most IRL algorithms, however, are difficult to scale up computationally because they require solving an RL problem in their inner loop. Recently, Ho & Ermon (2016) proposed framing imitation learning as an occupancy-measure matching (or divergence minimization) problem. Their architecture (GAIL) forgoes learning the optimal cost function in order to achieve computational tractability and sample-efficiency (in terms of the number of expert demonstrations needed). In detail, if $\rho_\pi(s, a)$ and $\rho_E(s, a)$ represent the state-action visitation distributions of the policy and the expert, respectively, then minimizing the Jenson-Shanon divergence $\min_\pi D_{JS}[\rho_\pi(s, a) \,||\, \rho_E(s, a)]$ helps to recover a policy with a similar trajectory distribution as the expert. GAIL iteratively trains a policy (π_θ) and a discriminator (D_ω) to optimize the mini-max objective:

$$\min_\theta \max_\omega \mathbb{E}_{(s,a)\sim\pi,\mathcal{T}}\big[\log(1 - D_\omega(s, a))\big] \\ + \mathbb{E}_{(s,a)\sim\mathcal{M}_E}\big[\log D_\omega(s, a)\big] \tag{1}$$

where $D_\omega : \mathcal{S} \times \mathcal{A} \to (0, 1)$, $\mathcal{M}_E$ is the buffer with expert demonstrations, and $\mathcal{T}$ is the transition dynamics.

3 METHODS

3.1 BELIEF REPRESENTATION IN POMDP

In a POMDP, the observations are by definition non-Markovian. A policy $\pi(a_t|o_t)$ that chooses actions based on current observations performs sub-optimally, since o_t does not contain sufficient information about the true state of the world. It is useful to infer a distribution on the true states based on the experiences thus far. This is referred to as the *belief state*, and is formally defined as the filtering distribution: $p(s_t|o_{\leq t}, a_{<t})$. It combines the memory of past experiences with uncertainty about unobserved aspects of the world. Let $h_t := (o_{\leq t}, a_{<t})$ denote the observation-action history, and $b_t := \phi(h_t)$ be a function of h_t. If b_t is learned such that it forms the sufficient statistics of the filtering posterior over states, i.e., $p(s_t|o_{\leq t}, a_{<t}) \approx p(s_t|b_t)$, then b_t could be used as a surrogate code (or representation) for the belief state, and be used to train agents in POMDPs. Henceforth, with slight abuse of notation, we would refer to b_t as the *belief*, although it is a high-dimensional representation rather than an explicit distribution over states.

An intuitive way to obtain this belief is by combining the observation-action history using aggregator functions such as recurrent or convolution networks. For instance, the intermediate hidden states in a recurrent network could represent b_t. In the RL setting with environmental rewards, the representation could be trained by conditioning the policy on it, and back-propagating the RL (e.g. policy gradient) loss. However, the RL signal is generally too weak to learn a rich representation b_t that provides sufficient statistics for the filtering posterior over states. Moreno et al. (2018) provide empirical evidence of this claim by training oracle models where representation learning is supervised with privileged information in form of the (unknown) environment states, and comparing them with learning solely using the RL loss. The problem is only exacerbated when the environmental rewards are extremely sparse. In our imitation learning setup, the belief update is incorporated into the mini-max objective for adversarial imitation of expert trajectories, and hence the representation is learned with a potentially stronger signal (Section 3.3). Prior work has shown that representations can be improved by using auxiliary losses such as reward-prediction (Jaderberg et al., 2016), depth-prediction (Mirowski et al., 2016), and prediction of future sensory data (Dosovitskiy & Koltun, 2016; Oh et al., 2015). Inspired by this, in Section 3.4, we regularize the representation with various prediction losses.

Recently, Ha & Schmidhuber (2018) proposed an architecture (World-Models) that decouples model-learning from policy-optimization. In the model-learning phase,

a variational auto-encoder compresses the raw observations to latent-space vectors, which are then temporally integrated using an RNN, combined with a mixture density network. In the policy-optimization phase, a policy conditioned on the RNN hidden-states is learned to maximize the rewards. We follow a similar *separation-of-concerns* principle, and divide the architecture into two modules: 1) a **policy module** $\pi_\theta(a_t|b_t)$ which learns a distribution over actions, conditioned on the belief; and 2) a **belief module** B_ϕ which learns a good representation of the belief $b_t := B_\phi(h_t)$, from the history of observations and actions, $h_t := (o_{\leq t}, a_{<t})$. While the policy module is trained with imitation learning, the belief module can be trained in a task-agnostic manner (like in World-Models), or in a task-aware manner. We describe these approaches in following sections.

3.2 POLICY MODULE

The goal of our agent is to learn a policy by imitating a few expert demonstration trajectories of the form $\{o_i, a_i\}_{i=0}^{|\tau|}$. Similar to the objective in GAIL, we hope to minimize the Jenson-Shanon divergence between the state-action visitation distributions of the policy and the expert: $\min_\pi D_{JS}[\rho_\pi(s, a) \,||\, \rho_E(s, a)]$. However, since the true environment state s_t is unobserved in POMDPs, we modify the objective to involve the belief representation b_t instead, since it characterizes the posterior over s_t via the generative process $p(s_t|b_t)$. Defining the *belief-visitation* distribution $\rho_\pi(b)$ for a policy analogously to the state-visitation distribution, the data processing inequality for f-divergences provides that: $D_{JS}[\rho_\pi(s) \,||\, \rho_E(s)] \leq D_{JS}[\rho_\pi(b) \,||\, \rho_E(b)]$. Please see Appendix 7.1 for the proof. The objective $\min_\pi D_{JS}[\rho_\pi(b) \,||\, \rho_E(b)]$ thus minimizes an upper bound on the D_{JS} between the state-visitation distributions of the expert and the policy. A further relaxation of this objective allows us to explicitly include the belief-conditioned policy $\pi(a|b)$ into the divergence minimization objective: $D_{JS}[\rho_\pi(b) \,||\, \rho_E(b)] \leq D_{JS}[\rho_\pi(b, a) \,||\, \rho_E(b, a)]$, where $\rho_\pi(b, a) = \rho_\pi(b)\pi(a|b)$ is the belief-action visitation (proof in Appendix 7.1).

Minimizing $\mathbf{D_{JS}}[\rho_\pi(\mathbf{b}, \mathbf{a}) \,||\, \rho_\mathbf{E}(\mathbf{b}, \mathbf{a})]$. Although explicitly formulating these visitation distributions is difficult, it is possible to obtain an empirical distribution of $\rho_\pi(b, a)$ by rolling out trajectories $(o_1, a_1, \dots)$ from π, and using our belief module to produce samples of belief-action tuples (b_t, a_t), where $b_t := B_\phi(h_t), h_t := (o_{\leq t}, a_{<t})$. Similarly, the expert demonstrations buffer $\mathcal{M}_E$ contains observation-actions sequences, and can be used as an estimate of $\rho_E(b, a)$. Therefore, D_{JS} [2] can

---

[2]To reduce clutter, we shorthand $D_{JS}[\rho_\pi(b, a) \,||\, \rho_E(b, a)]$ with just D_{JS} for the remainder of this section.

be approximated (up to a constant shift) with a binary classification problem as exploited in GANs (Goodfellow et al., 2014):

$$D_{JS}(\theta; \phi) \approx \max_\omega \widetilde{\mathbb{E}}_{(b,a) \sim \mathcal{M}_E} \big[\log D_\omega(b,a) \big]$$
$$+ \widetilde{\mathbb{E}}_{(b,a) \sim \pi, \mathcal{T}} \big[\log(1 - D_\omega(b,a)) \big] \quad (2)$$

where θ are the parameters for the policy $\pi_\theta(a|b)$, D_ω is the discriminator, and $\mathcal{T}$ is the transition dynamics. It should be noted that D_{JS} is a function of the belief module parameters ϕ through its dependence on the belief states. The imitation learning objective for optimizing the policy is then obtained as:

$$\min_\theta D_{JS}(\theta; \phi) \approx \min_\theta \max_\omega \widetilde{\mathbb{E}}_{(b,a) \sim \mathcal{M}_E} \big[\log D_\omega(b,a) \big]$$
$$+ \widetilde{\mathbb{E}}_{(b,a) \sim \pi, \mathcal{T}} \big[\log(1 - D_\omega(b,a)) \big] \quad (3)$$

In Equation 2, denoting the functional maximum over D_ω by D^*, the gradient for policy optimization is: $\nabla_\theta D_{JS}(\theta; \phi) \approx \nabla_\theta \widetilde{\mathbb{E}}_{(b,a) \sim \pi, \mathcal{T}} \big[\log(1 - D^*(b,a)) \big]$. Figure 1 shows the stochastic computation graph (Schulman et al., 2015a) for this expectation term, where the stochastic nodes are represented by circles, deterministic nodes by rectangles, and we have written belief as a function of the history. Given fixed belief module parameters (ϕ), the required gradient w.r.t θ is obtained using the policy gradient theorem (Sutton et al., 2000) as:

$$\nabla_\theta D_{JS}(\theta; \phi) \approx \nabla_\theta \widetilde{\mathbb{E}}_{(b,a) \sim \pi, \mathcal{T}} \big[\log(1 - D^*(b,a)) \big]$$
$$= \widetilde{\mathbb{E}}_{(b,a) \sim \pi, \mathcal{T}} \big[\nabla_\theta \log \pi_\theta(a|b) \hat{Q}^\pi(b,a) \big], \text{ where}$$
$$\hat{Q}^\pi(b_t, a_t) = \widetilde{\mathbb{E}}_{(b,a) \sim \pi, \mathcal{T}} \Big[\sum_{t'=t}^\infty \gamma^{t'-t} \log(1 - D^*(b_{t'}, a_{t'})) \Big]$$
$$(4)$$

Therefore, updating the policy to minimize D_{JS} is approximately the same as applying the standard policy-gradient using the rewards obtained from a learned discriminator, $r(b,a) = -\log(1 - D^*(b,a))$. As is standard practice, we do not train the discriminator to optimality, but rather jointly train the policy and discriminator using iterative gradient updates. The discriminator is updated using the gradient from Equation 2, while the policy is updated with gradient from Equation 4. We now detail the update rule for ϕ.

3.3 BELIEF MODULE

This module transforms the history $(o_{\leq t}, a_{<t})$ into a belief representation. Various approaches could be used to aggregate historical context, such as RNNs, masked convolutions (Gehring et al., 2017) and attention-based methods (Vaswani et al., 2017). In our implementation, we model the belief module B_ϕ with an RNN, such that
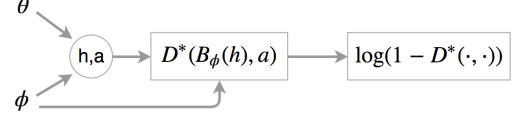

Figure 1: Stochastic computation graph for the expectation: $\widetilde{\mathbb{E}}_{(b,a) \sim \pi, \mathcal{T}} \big[\log(1 - D^*(b,a)) \big]$. Both the policy ($\theta$) and belief module ($\phi$) parameters influence the generation of observation-action sequences, $(h_t, a_t) := \{o_{\leq t}, a_{\leq t}\}$ through environment interaction. $b = B_\phi(h)$ is the belief. Circles represent stochastic nodes; rectangles are deterministic nodes.

$b_t = B_\phi(b_{t-1}, o_t, a_{t-1})$. We use GRUs (Cho et al., 2014) as they have been demonstrated to have good empirical performance. We denote by $\mathcal{R}$, a replay-buffer which stores observation-action sequences (current and past) from the agent. As stated before, the belief module could be learnt in a task-agnostic manner (similar to Ha & Schmidhuber (2018)), or with task-awareness.

Task-agnostic learning (separately from policy). An unsupervised approach to learning ϕ without accounting for the agent's objective, is to maximize the joint likelihood of the observation sequence, conditioned on the actions, $\log p(o_{\leq T}|a_{<T})$. This decomposes autoregressively as $\sum_t \log p(o_t|o_{<t}, a_{<t})$. The objective can be optimized by conditioning a generative model for o_t on the RNN hidden state b_{t-1}^ϕ and action a_{t-1}, and using MLE. Using a unimodal Gaussian generative model (learned function g for the mean, and fixed variance), the autoregressive loss to minimize is:

$$\mathcal{L}^{AR}(\phi) = \mathbb{E}_\mathcal{R} ||o_t - g(b_{t-1}^\phi, a_{t-1})||_2^2 \quad (5)$$

Task-aware learning (jointly with policy). Since the policy is conditioned on the belief, an intuitive way to improve the agent's performance is to learn the belief with an objective more aligned with policy-learning. Since the agent minimizes $D_{JS}(\theta, \phi)$, as defined in Equation 2, the same imitation learning objective naturally can also be used for learning ϕ:

$$\mathcal{L}^{IM}(\phi) := D_{JS}(\theta, \phi) \approx$$
$$\widetilde{\mathbb{E}}_{(h,a) \sim \mathcal{M}_E} \big[\log D^*(B_\phi(h), a) \big]$$
$$+ \widetilde{\mathbb{E}}_{(h,a) \sim \pi_\theta(a|B_\phi(h)), \mathcal{T}} \big[\log(1 - D^*(B_\phi(h), a)) \big]$$
$$(6)$$

which is the same as Equation 2 except for the use of the optimal discriminator (D^*), and that we have written the belief in terms of history $b := B_\phi(h)$ to explicitly bring out the dependence on ϕ. The gradient of the first expectation term w.r.t ϕ is straightforward; the gradient of the second expectation term w.r.t ϕ (for given fixed parameters θ) comprises of a policy-gradient term and a pathwise-derivative term (Figure 1). Therefore,

$\nabla_\phi D_{JS}(\theta; \phi)$ can be approximated with:

$$\widetilde{\mathbb{E}}_{(h,a)\sim\mathcal{M}_E}\left[\nabla_\phi \log D^*(B_\phi(h), a)\right]$$

$$+ \underbrace{\widetilde{\mathbb{E}}_{(h,a)\sim\pi_\theta(a|B_\phi(h)),\mathcal{T}}\left[\nabla_\phi \log \pi_\theta(a|B_\phi(h))\hat{Q}^\pi\right]}_{\text{policy-gradient term}}$$

$$+ \underbrace{\widetilde{\mathbb{E}}_{(h,a)\sim\pi_\theta(a|B_\phi(h)),\mathcal{T}}\left[\nabla_\phi \log(1 - D^*(B_\phi(h), a))\right]}_{\text{pathwise-derivate term}}$$

$$(7)$$

where $\hat{Q}$ is as defined in Equation 4. The overall min-max objective for jointly training the policy, belief and discriminator is:

$$\min_{\phi,\theta} \max_\omega \widetilde{\mathbb{E}}_{(b,a)\sim\mathcal{M}_E}\left[\log D_\omega(b, a)\right]$$

$$+ \widetilde{\mathbb{E}}_{(b,a)\sim\pi,\mathcal{T}}\left[\log(1 - D_\omega(b, a))\right] \quad (8)$$

3.4 BELIEF REGULARIZATION

With the min-max objective (Equation 8), it may be possible that the belief parameters (ϕ) are driven towards a degenerate solution that ignores the history ($o_{\leq t}, a_{<t}$), thereby producing constant (or similar) beliefs for policy and expert trajectories. Indeed, if we omit the actions (a) in the discriminator $D_\omega(b, a)$, a constant belief output is an optimal solution for Equation 8. To learn a belief representation that captures relevant historical context and is useful for deriving optimal policies, we add forward-, inverse- and action-regularization to the belief module. We define and motivate them from the perspective of mutual information maximization.

Notation. For two continuous random variables X, Y, mutual information is defined as $I(X; Y) := H(X) - H(X|Y)$, where H denotes the differential entropy [3]. Intuitively, $I(X; Y)$ measures the dependence between X and Y. Conditional mutual information is defined as $I(X; Y|Z) := \mathbb{E}_z[(I(X; Y)|z]$. Given Y, if X and Z are independent ($X \perp Z|Y$), then X, Y, Z form a Markov Chain ($X \to Y \to Z$), and the data processing inequality for mutual information states that $I(X; Z) \leq I(X; Y)$.

Forward regularization. As discussed in Section 3.1, an ideal belief representation completely characterizes the posterior over the true environment states $p(s_t|b_t)$. Therefore, it ought to be correlated with future true states (s_{t+k}), conditioned on the intervening future actions ($a_{t:t+k-1}$). We frame this objective as maximization of the following conditional mutual information: $I(b_t; s_{t+k}|a_{t:t+k-1})$. Since $o_{t+k} \perp b_t|s_{t+k}$ because of the observation generation process in a POMDP, we get

---

[3]The usual notation for differential entropy (h) is not used to avoid confusion with the history h_t used in previous sections.

---

the following after using the data processing inequality for mutual information:

$$I(b_t; s_{t+k}|a_{t:t+k-1}) \geq I(b_t; o_{t+k}|a_{t:t+k-1})$$
$$= \mathbb{E}_{a_{t:t+k-1}}\left[H(o_{t+k}|a_{t:t+k-1}) \right.$$
$$\left. - H(o_{t+k}|b_t; a_{t:t+k-1})\right]$$
$$\geq \mathbb{E}_{a_{t:t+k-1}}\left[H(o_{t+k}|a_{t:t+k-1}) \right. \quad (9)$$
$$\left. + \mathbb{E}_{o_{t+k},b_t}\left[\log q(o_{t+k}|b_t; a_{t:t+k-1})\right]\right]$$

where the final inequality follows because we can lower bound the mutual information using a variational approximation q, similar to the variational information maximization algorithm (Agakov & Barber, 2004). Therefore, we maximize a lower bound to the mutual information $I(b_t; s_{t+k}|a_{t:t+k-1})$ with the surrogate objective:

$$\max_{\phi,q} \mathbb{E}_{o_{t+k},b_t, \atop a_{t:t+k-1}}\left[\log q(o_{t+k}|b_t^\phi; a_{t:t+k-1})\right]$$

With the choice of a unimodal Gaussian (learned function g for the mean, and fixed variance) for the variational distribution q, the loss function for forward regularization of the belief module is:

$$\mathcal{L}^f(\phi) = \mathbb{E}_\mathcal{R}||o_{t+k} - g(b_t^\phi, a_{t:t+k-1})||_2^2$$

where the expectation is over trajectories ($o_1, a_1, \dots$) sampled from the replay buffer $\mathcal{R}$.

Inverse regularization. It is desirable that the belief at time t is correlated with the past true states (s_{t-k}), conditioned on the intervening past actions ($a_{t-k:t-1}$). This should improve the belief representation by helping to capture long-range dependencies. Proceeding in a manner similar to above, the conditional mutual information between these signals, $I(s_{t-k}; b_t|a_{t-k:t-1})$, can be lower bounded by $I(o_{t-k}; b_t|a_{t-k:t-1})$ using the data processing inequality. Again, as before, this can be further lower bounded using a variational distribution q for generating past observation o_{t-k}. A unimodal Gaussian (mean function g) for q yields the following loss for inverse regularization, which is optimized using trajectories from the replay $\mathcal{R}$:

$$\mathcal{L}^i(\phi) = \mathbb{E}_\mathcal{R}||o_{t-k} - g(b_t^\phi, a_{t-k:t-1})||_2^2$$

Action regularization. We also wish to maximize $I(a_{t:t+k-1}; s_{t+k} \mid b_t)$ for the reason that, conditioned on the current belief b_t, a sequence of k subsequent actions ($a_{t:t+k-1}$) should provide information about the resulting true future state (s_{t+k}). Similar lower bounding and use of a variational distribution with mean function g for generating action-sequences gives us the loss:

$$\mathcal{L}^a(\phi) = \mathbb{E}_\mathcal{R}||(a_{t:t+k-1}) - g(b_t^\phi, o_{t+k})||_2^2$$
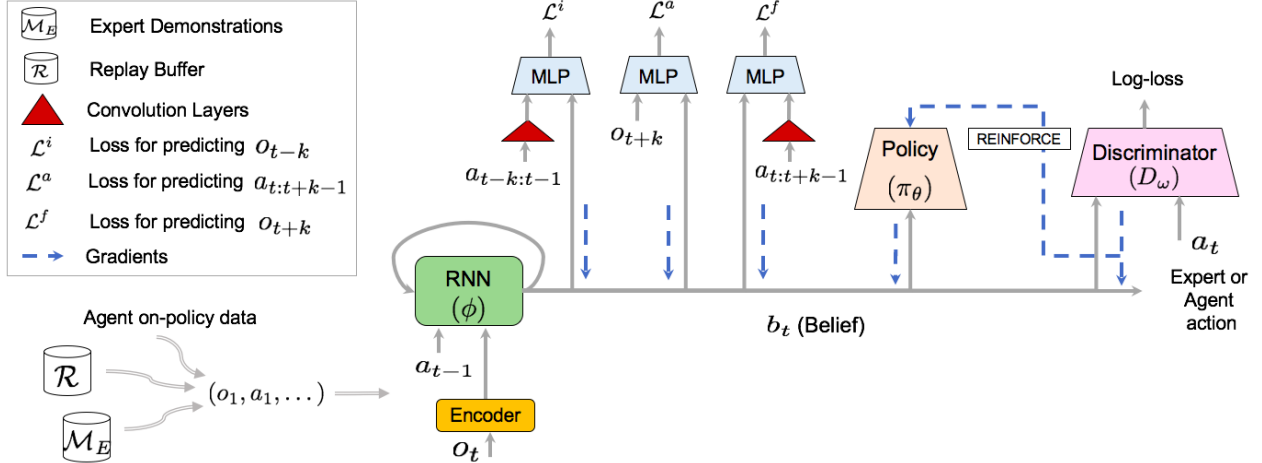
Figure 2: Schematic diagram of our complete architecture. The belief module B_ϕ is a recurrent network with GRU cells, and encodes trajectories $(o_1, a_1, \dots)$ from agent (on-policy data), replay buffer $\mathcal{R}$ (off-policy data) and expert demonstrations memory $\mathcal{M}_E$ into belief representations (b_t). B_ϕ is updated with imitation-loss (Equation 6) computed from the current policy and discriminator networks. It is further regularized with forward-, inverse- and action-regularization using MLPs (colored in blue in the figure). Convolution layers (colored in red) encode the past actions ($a_{t-k:t-1}$) and future actions ($a_{t:t+k-1}$) into compact representations, which are then fed into the MLPs. The policy $\pi_\theta(a_t|b_t^\phi)$ is conditioned on the belief, and updated using imitation learning (Equation 4). The discriminator $D_\omega(b_t^\phi, a_t)$ is a binary classifier trained on tuples from the agent and expert demonstrations (Equation 2).

The complete loss function for training the belief module results from a weighted combination of the imitation-loss and regularization terms. Imitation-loss uses on-policy data and expert demonstrations ($\mathcal{M}_E$), while the regularization losses are computed with on-policy and off-policy data, as well as $\mathcal{M}_E$.

$$\mathcal{L}(\phi) = \mathcal{L}^{IM} + \lambda_1 \mathcal{L}^f + \lambda_2 \mathcal{L}^i + \lambda_3 \mathcal{L}^a \qquad (10)$$

We derive our final expressions for $(\mathcal{L}^f, \mathcal{L}^i, \mathcal{L}^a)$ by modeling the respective variational distributions (q) as fixed-variance, unimodal Gaussians. We later show that using this simple model results in appreciable performance benefits for imitation learning. Other expressive model classes, such as mixture density networks and flow-based models (Rezende & Mohamed, 2015), can be readily used as well, to learn complex and multi-modal distributions over the future observations (o_{t+k}), past observations (o_{t-k}) and action-sequences ($a_{t:t+k-1}$).

3.5 LEARNING ALGORITHM

Figure 2 shows the schematic diagram of our complete architecture, including an overview of implemented neural networks. In Algorithm 1, we outline the major steps of the training procedure. In each iteration, we run the policy for a few steps and obtain shaped rewards from the current discriminator (Line 6). The policy parameters are then updated using A2C, which is the syn-

chronous adaptation of asynchronous advantage actor-critic (A3C; Mnih et al. (2016)), as the policy-gradient algorithm (Line 10). Other RL algorithms, such as those based on trust-regions methods (Schulman et al., 2015b) could also be readily used. Similar to the policy (actor), the baseline (critic) used for reducing variance of the stochastic gradient-estimation is also conditioned on the belief. To further reduce variance, Generalized Advantage Estimation (GAE; Schulman et al. (2015c)) is used to compute the advantage. Apart from the policy-gradient, on-policy data also enables computing the gradient for the discriminator network (Line 13) and the belief module (Line 14). The belief is further refined by minimizing the regularization losses on off-policy data from the replay buffer $\mathcal{R}$ (Line 15).

The regularization losses ($\mathcal{L}^f, \mathcal{L}^i, \mathcal{L}^a$) described in Section 3.4 include a hyperparameter k that controls the temporal offset of the predictions. For instance, for $\mathcal{L}^i(\phi; k)$, the larger the k, the farther back in time the observation predictions are made, conditioned on the current belief and past actions. The temporal abstractions provided by multi-step predictions ($k>1$) should help to extract more global information from the input stream into the belief representation. Our ablations (Section 5) show the performance benefit of including multi-step losses. Various strategies for selecting k are possible, such as uniform sampling from a range (Guo et al., 2018) and adaptive se-

lection based on a curriculum (Oh et al., 2015). For simplicity, we choose fixed values, and leave the exploration of the more sophisticated approaches to future work. Hence, our total regularization loss comprises of single-step (k=1) and multi-step (k=5) forward-, inverse-, and action-prediction losses. For encoding a sequence of past or future actions into a compact representation, we use multi-layer convolution networks (Figure 2).

Algorithm 1: Belief-module Imitation Learning (BMIL)

---

1 **for** *each iteration* **do**
2     $d_\pi = \{\}, d_E = \{\}$
    /* Rollout c steps from policy */
3   **repeat**
4   Get observation o_t from environment
5        $a_t \sim \pi_\theta(a_t|b_t)$, where $b_t = B_\phi(o_{\leq t}, a_{<t})$
6        $r_t = -\log(1 - D_\omega(b_t, a_t))$
7        $d_\pi \leftarrow d_\pi \cup (b_t, a_t, r_t)$
8   If o_t is terminal, add rollout $\{o_i, a_i\}_{i=0}^{|\tau|}$ to $\mathcal{R}$
9   **until** $|d_\pi| == c$;
    /* Update Policy */
10   Update θ with policy-gradient (Eq. 4)
    /* Update discriminator ω */
11   Fetch $(o_t, a_t, \dots)$ of length c from $\mathcal{M}_E$
12   Generate belief-action tuples $d_E = \{(b_i, a_i)\}_{i=t}^{t+c-1}$
13   Update ω with log-loss objective using d_π and d_E
    /* Update Belief Module ϕ */
14   Update ϕ with $\nabla_\phi \mathcal{L}(\phi)$ using d_π and d_E (Eq. 10)
    /* Off-policy Updates */
15   **for** *few update steps* **do**
16   Fetch $(o_t, a_t, \dots)$ of length c from $\mathcal{R}$
17   Update ϕ with $\nabla_\phi(\lambda_1 \mathcal{L}^f + \lambda_2 \mathcal{L}^i + \lambda_3 \mathcal{L}^a)$
18   **end**
19 **end**

---

4 RELATED WORK

While we cannot do full justice to the extensive literature on algorithms for training agents in POMDPs, we here mention some recent related work. Most prior algorithms for POMDPs assume access to a predefined reward function. These include approaches based on Q-learning (DRQN; Hausknecht & Stone (2015)), policy-gradients (Igl et al., 2018), partially observed guided policy search (Zhang et al., 2016), and planning methods (Silver & Veness, 2010; Ross et al., 2008; Pineau et al., 2003). In contrast, we propose to adapt ideas from generative adversarial imitation learning to learn policies

in POMDPs without environmental rewards.

Learning belief states from history $(o_{\leq t}, a_{<t})$ was recently explored in Guo et al. (2018). The authors show that training the belief representation with a Contrastive Predictive Coding (CPC, Oord et al. (2018)) loss on future observations, conditioned on future actions, helps to infer knowledge about the underlying state of the environment. Predictive State Representations (PSRs) offer another approach to modeling the belief state in terms of observable data (Littman & Sutton, 2002). The assumption in PSRs is that the filtering distribution can be approximated with a distribution over the k future observations, conditioned on future actions, $p(s_t|o_{\leq t}, a_{<t}) \approx p(o_{t+1:t+k}|o_{\leq t}, a_{<t+k})$. PSRs combined with RNNs have been shown to improve representations by predicting future observations (Venkatraman et al., 2017; Hefny et al., 2018). While we also make future predictions, a key difference compared to aforementioned methods is that our belief representation is additionally regularized by predictions in the past, and in action-space, which we later show benefits our approach.

State-space models (SSMs; Fraccaro et al. (2016); Goyal et al. (2017); Buesing et al. (2018)), which represent the unobserved environment states with latent variables, have also been used to obtain belief states. Igl et al. (2018) use a particle-filtering method to train a VAE, and represent the belief state with a weighted collection of particles. The model is also updated with the RL-loss using a belief-conditioned policy. Gregor & Besse (2018) proposed TD-VAE, which explicitly connects belief distributions at two distant timesteps, and enforces consistency between them using a transition distribution and smoothing posterior. Although we use a deterministic model for our belief module (B_ϕ), our methods apply straightforwardly to SSMs as well.

5 EXPERIMENTS

The goal in this section is to evaluate and analyze the performance of our proposed architecture for imitation learning in partially-observable environments, given some expert demonstrations. Herein, we describe our environments, provide comparisons with GAIL, and perform ablations to study the importance of the design decisions that motivate our architecture.

Partially-observable locomotion tasks. We benchmark high-dimensional, continuous-control locomotion environments based on the MuJoCo physics simulator, available in OpenAI Gym (Brockman et al., 2016). The standard Gym MuJoCo library of tasks, however, consists of MDPs (and not POMDPs), since observations in such tasks contain sufficient state-information to learn an op-

timal policy conditioned on only the current observation. As such, it has been extensively used to evaluate performance of reinforcement-learning and imitation-learning algorithms in the MDP setting (Schulman et al., 2017; Ho & Ermon, 2016). To transform these tasks into POMDPs, we follow an approach similar to Duan et al. (2016), and redact some sensory data from the observations. Specifically, from the default observations, we remove measurements for the translation and angular velocities of the torso, and also the velocities for all the link joints. We denote the original (MDP) observations by $s \in \mathcal{S}$, and the curtailed (POMDP) observations by $o \in \mathcal{O}$. The details for MDP and POMDP sensors for our MuJoCo tasks are in Appendix 7.2.

For all experiments, we assume access to 50 expert demonstrations of the type $\{o_i, a_i\}_{i=0}^{|\tau|}$, for each of the tasks. The policy and discriminator networks are feed-forward MLPs with two 64-unit layers. The policy network outputs include the action mean and per-action variances (i.e. actions are assumed to have an independent Gaussian distribution). In the belief module, the dimension of the GRU cell is 256, while the MLPs used for regularization have two 64-unit feed-forward layers. More details and hyperparameters are in Appendix 7.3.

5.1 COMPARISON TO GAIL

Our first baseline is modeled after the architecture used in the original GAIL approach (Ho & Ermon, 2016). It consists of feed-forward policy and discriminator networks, without the recurrent belief module. The policy is conditioned on o_t, and the discriminator performs binary classification on (o_t, a_t) tuples. The update rules for the policy and discriminator are obtained in similar way as Equation 4 and Equation 2, respectively, by replacing the belief b_t with observation o_t. The next baseline, referred to as GAIL+Obs. stack, augments GAIL by concatenating 3 previous observations to each o_t, and feeding the entire stack as input to the policy and discriminator networks. This approach has been found to extract useful historical context for a better state-representation (Mnih et al., 2015). We abbreviate our complete proposed architecture (Figure 2) by BMIL, short for Belief-Module Imitation Learning. BMIL jointly trains the policy, belief and discriminator networks using a mini-max objective (Equation 8), and additionally regularizes the belief with multi-step predictions. Table 1 compares the performance of different designs on POMDP MuJoCo. We shows the mean episode-returns, averaged over 5 runs with random seeds, after 10M timesteps of interaction with the environment. We observe that GAIL—both with and without observation stacking—is unable to successfully imitate the expert behavior. Since the observation

o_t alone does not contain adequate information, the policy conditioned on it performs sub-optimally. Also, the discriminator trained on (o_t, a_t) tuples does not provide robust shaped rewards. Using the full state s_t instead of o_t in our experiments leads to successful imitation with GAIL, suggesting that the performance drop in Table 1 is due to partial observability, rather than other artifacts such as insufficient network capacity, or lack of algorithmic or hyperparameter tuning. Further, we see that techniques such as stacking past observations provide only a marginal improvement in some of the tasks. In contrast, in BMIL, the belief module curates a belief representation from the history $(o_{\leq t}, a_{<t})$, which is used both for discriminator training, and to condition the action-distribution (policy). BMIL achieves scores very close to those of the expert.

	GAIL	GAIL + Obs. stack	BMIL (Ours)	Expert ($\approx$ Avg.)
Inv.DoublePend.	109	1351	**9104**	9300
Hopper	157	517	**2665**	3200
Ant	895	1056	**1832**	2400
Walker	357	562	**4038**	4500
Humanoid	1686	1284	**4382**	4800
Half-cheetah	205	-948	**5860**	6500

Table 1: Mean episode-returns, averaged over 5 runs with random seeds, after 10M timesteps in POMDP MuJoCo.

5.2 COMPARISON TO GAIL WITH RECURRENT NETWORKS

For our next two baselines, we replace the feed-forward networks in GAIL with GRUs. GAIL-RF uses a recurrent policy and a feed-forward discriminator, while in GAIL-RR, both the policy and the discriminator are recurrent. In both these baselines, the belief is created internally in the recurrent policy module. Importantly, unlike BMIL, the belief is not shared between the policy and the discriminator. The average final performance of GAIL-RF and GAIL-RR in our POMDP environments is shown in Table 2. We observe that GAIL-RR does not perform well on most of the tasks. A plausible explanation for this is that using the adversarial binary classification loss for training the discriminator parameters does not induce a sufficient representation of the belief state in its recurrent network. The other baseline, GAIL-RF, avoids this issue with a feed-forward discriminator trained on (o_t, a_t) tuples from the expert and the policy. This leads to much better performance. However, BMIL consistently outperforms GAIL-RF, most significantly in *Humanoid* ($1.6\times$ higher score), indicating the effectiveness of the decoupled architecture and other design decisions that motivate BMIL. Figure 4 (in Appendix) plots the learning curves for these experiments. Appendix 7.6 further shows that BMIL outperforms the strongest base-

line (GAIL-RF) on additional environments with accentuated partial observability.

	GAIL-RR	GAIL-RF	BMIL (Ours)
Inv.DoublePend.	8965	9103	**9104**
Hopper	955	2164	**2665**
Ant	-533	1612	**1832**
Walker	400	3188	**4038**
Humanoid	3829	2761	**4382**
Half-cheetah	-922	5011	**5860**

Table 2: Mean episode-returns, averaged over 5 runs with random seeds, after 10M timesteps in POMDP MuJoCo.

5.3 ABLATION STUDIES

How crucial is belief regularization? To quantify this, we compare the performance of our architecture with and without belief regularization (BMIL vs. BMIL w/o Reg.). Figure 3 (in Appendix) plots the mean episode-returns vs. timesteps of environment interaction, over the course of learning. We observe that including regularization leads to better episode-returns and sample-complexity for most of the tasks considered, indicating that it is useful for shaping the belief representations. The single- and multi-step predictions for $(\mathcal{L}^f, \mathcal{L}^i)$ are made in the observation space. Although we see good improvements for tasks with high-dimensional spaces, such as Humanoid ($|\mathcal{O}|=269$) and Ant ($|\mathcal{O}|=97$), predicting an entire raw observation may be sub-optimal and computationally wasteful for some tasks, since it requires modeling the complex structures within an observation. To avoid this, predictions can be made in a compact encoding space (output of Encoder in Figure 2). In Appendix 7.5, we show the performance of BMIL with predictions in encoding-space rather than observation-space, and note that the scores are quite similar.

Task-aware vs. Task-agnostic belief learning. Next, we compare with a design in which the belief module is trained separately from the policy, using a task-agnostic loss ($\mathcal{L}^{AR}$, Section 3.3). This echos the philosophy used in World-Models (Ha & Schmidhuber, 2018). As Figure 3 (in Appendix) shows, this results in mixed success for imitation in POMDPs. While the agent achieves good scores in tasks such as Ant and HalfCheetah, the performance in Walker and Hopper is unsatisfactory. In contrast, BMIL, which uses a task-aware imitation-loss for the belief module, is consistently better.

Are all of $\mathcal{L}^f, \mathcal{L}^i, \mathcal{L}^a$ useful? We introduced 3 different regularization terms for the belief module – forward ($\mathcal{L}^f$), inverse ($\mathcal{L}^i$) and action ($\mathcal{L}^a$). To assess their benefit individually, in Figure 6 (Appendix 7.4), we plot learning curves for two tasks, with each of the regularizations applied in isolation. We compare them with BMIL,

which includes all of them, and BMIL w/o Reg., which excludes all of them (no regularization). For the Ant task, we notice that each of $\{\mathcal{L}^f, \mathcal{L}^i, \mathcal{L}^a\}$ provides performance improvement over the no-regularization baseline, and combining them (BMIL) performs the best. With the Walker task, we see better mean episode-returns at the end of training with each of $\{\mathcal{L}^f, \mathcal{L}^i, \mathcal{L}^a\}$, compared to no-regularization; BMIL attains the best sample-complexity.

Are multi-step predictions useful? As argued before, making coarse-scale, multi-step ($k>1$) predictions for forward, inverse observations and action-sequences could improve representations by providing temporal abstractions. In Figure 7 (Appendix 7.4), we plot BMIL, which uses single- and multi-step losses $k=\{1, 5\}$, and compare it with two versions: first that uses a different temporal offset $k=\{1, 10\}$, and second that predicts only at the single-step granularity $k=\{1\}$. For both tasks, we get better sample-complexity and higher final mean episode-returns with multi-step, suggesting its positive contribution to representation learning.

6 CONCLUSION AND FUTURE WORK

In this paper, we study imitation learning for POMDPs, which has been considerably less explored compared to imitation learning for MDPs, and learning in POMDPs with predefined reward functions. We introduce a framework comprised of a belief module, and policy and discriminator networks conditioned on the generated belief. Crucially, all networks are trained jointly with a min-max objective for adversarial imitation of expert trajectories.

Within this flexible setup, many instantiations are possible, depending on the choice of networks. Both feed-forward and recurrent networks can be used for the policy and the discriminator, while for the belief module there is an expansive set of options based on the rich literature on representation learning, such as CPC (Guo et al., 2018) and using auxiliary tasks (Jaderberg et al., 2016). Many more methods based on state-space latent-variable models are also applicable (c.f. Section 4). In our instantiation of the belief module, we use the task-based imitation loss (Equation 6), and improve robustness of representations by regularizing with multi-step prediction of past/future observations and action-sequences. One benefit of our proposed framework is that in future work, it would be straightforward to substitute other methods for learning belief representations for the one we arrived at in our paper. Similarly, recent advancements in GAN and RL literature could guide the development of better discriminator and policy networks for imitation learning in POMDPs. Exploring these ranges, as well as their interplay, are important directions.

References

Felix Agakov and David Barber. Variational information maximization for neural coding. In *International Conference on Neural Information Processing*, pp. 543–548. Springer, 2004.

Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1):131–142, 1966.

K. J. Aström. Optimal control of Markov decision processes with incomplete state estimation. *J. Math. Anal. Appl.*, 10:174–205, 1965.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Lars Buesing, Theophane Weber, Sebastien Racaniere, SM Eslami, Danilo Rezende, David P Reichert, Fabio Viola, Frederic Besse, Karol Gregor, Demis Hassabis, et al. Learning and querying fast generative models for reinforcement learning. *arXiv preprint arXiv:1802.03006*, 2018.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

Giuseppe Cuccu, Julian Togelius, and Philippe Cudré-Mauroux. Playing atari with six neurons. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 998–1006. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

Alexey Dosovitskiy and Vladlen Koltun. Learning to act by predicting the future. *arXiv preprint arXiv:1611.01779*, 2016.

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, pp. 1329–1338, 2016.

Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *Advances in neural information processing systems*, pp. 2199–2207, 2016.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1243–1252. JMLR. org, 2017.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Anirudh Goyal ALIAS PARTH Goyal, Alessandro Sordoni, Marc-Alexandre Côté, Nan Rosemary Ke, and Yoshua Bengio. Z-forcing: Training stochastic recurrent networks. In *Advances in neural information processing systems*, pp. 6713–6723, 2017.

Karol Gregor and Frederic Besse. Temporal difference variational auto-encoder. *arXiv preprint arXiv:1806.03107*, 2018.

Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Bernardo A Pires, Toby Pohlen, and Rémi Munos. Neural predictive belief representations. *arXiv preprint arXiv:1811.06407*, 2018.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, pp. 2455–2467, 2018.

Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *2015 AAAI Fall Symposium Series*, 2015.

Milos Hauskrecht. Value-function approximations for partially observable markov decision processes. *Journal of artificial intelligence research*, 13:33–94, 2000.

Ahmed Hefny, Zita Marinho, Wen Sun, Siddhartha Srinivasa, and Geoffrey Gordon. Recurrent predictive state policy networks. *arXiv preprint arXiv:1803.01489*, 2018.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pp. 4565–4573, 2016.

Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. Deep variational reinforcement learning for pomdps. *arXiv preprint arXiv:1806.02426*, 2018.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Michael L Littman and Richard S Sutton. Predictive representations of state. In *Advances in neural information processing systems*, pp. 1555–1561, 2002.

Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J Ballard, Andrea Banino,

Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673*, 2016.

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

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937, 2016.

Pol Moreno, Jan Humplik, George Papamakarios, Bernardo Avila Pires, Lars Buesing, Nicolas Heess, and Théophane Weber. Neural belief states for partially observed domains. In *NeurIPS 2018 workshop on Reinforcement Learning under Partial Observability*, 2018.

Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, pp. 663–670, 2000.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. In *Advances in neural information processing systems*, pp. 2863–2871, 2015.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Joelle Pineau, Geoff Gordon, Sebastian Thrun, et al. Point-based value iteration: An anytime algorithm for pomdps. In *IJCAI*, volume 3, pp. 1025–1032, 2003.

Dean A Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, 1991.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994. ISBN 0471619779.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

Stéphane Ross, Joelle Pineau, Sébastien Paquet, and Brahim Chaib-Draa. Online planning algorithms for pomdps. *Journal of Artificial Intelligence Research*, 32:663–704, 2008.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635, 2011.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, pp. 3528–3536, 2015a.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I Jordan, and Philipp Moritz. Trust region policy optimization. In *Icml*, volume 37, pp. 1889–1897, 2015b.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015c.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

David Silver and Joel Veness. Monte-carlo planning in large pomdps. In *Advances in neural information processing systems*, pp. 2164–2172, 2010.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

Richard Stuart Sutton. Temporal credit assignment in reinforcement learning. 1984.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Arun Venkatraman, Nicholas Rhinehart, Wen Sun, Lerrel Pinto, Martial Hebert, Byron Boots, Kris Kitani, and J Bagnell. Predictive-state decoders: Encoding the future into recurrent networks. In *Advances in Neural Information Processing Systems*, pp. 1172–1183, 2017.

Marvin Zhang, Zoe McCarthy, Chelsea Finn, Sergey Levine, and Pieter Abbeel. Learning deep neural network policies with continuous memory states. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 520–527. IEEE, 2016.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.