ADVERSARIAL IMITATION VIA VARIATIONAL INVERSE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We consider a problem of learning the reward and policy from expert examples under unknown dynamics in high-dimensional scenarios. Our proposed method builds on the framework of generative adversarial networks and introduces the empowerment-regularized maximum-entropy inverse reinforcement learning to learn near-optimal rewards and policies. Empowerment-based regularization prevents the policy from overfitting expert demonstration, thus leads to a generalized behavior which results in learning near-optimal rewards. Our method simultaneously learns empowerment through variational information maximization along with the reward and policy under the adversarial learning formulation. We evaluate our approach on various high-dimensional complex control tasks. We also test our learned rewards in challenging transfer learning problems where training and testing environments are made to be different from each other in terms of dynamics or structure. Our experimentation shows that our proposed method not only learns near-optimal rewards and policies that are matching expert behavior but also performs significantly better than state-of-the-art inverse reinforcement learning algorithms.

1 INTRODUCTION

Reinforcement learning (RL) has emerged as a promising tool for solving complex decision-making and control tasks from predefined high-level reward functions (Silver et al., 2016; Qureshi et al., 2017). However, defining an optimizable reward function that inculcates the desired behavior can be challenging for many robotic applications, which include learning social-interaction skills (Qureshi et al., 2018), dexterous manipulation (Finn et al., 2016b), autonomous driving (Kuderer et al., 2015), and robotic surgery (Yip & Das, 2017).

Inverse reinforcement learning (IRL) (Ng et al., 2000) addresses the problem of learning reward functions from expert demonstrations, and it is often considered as a branch of imitation learning (Argall et al., 2009). The prior work in IRL includes maximum-margin (Abbeel & Ng, 2004; Ratliff et al., 2006) and maximum-entropy (Ziebart et al., 2008) formulations. Currently, maximum entropy (MaxEnt) IRL is a widely used approach towards IRL, and has been extended to use non-linear function approximators such as neural networks in scenarios with unknown dynamics by leveraging sampling-based techniques (Boularias et al., 2011; Finn et al., 2016b; Kalakrishnan et al., 2013). However, designing the IRL algorithm is usually complicated as it requires, to some extent, hand engineering such as deciding domain-specific regularizers (Finn et al., 2016b).

Rather than learning reward functions and solving the IRL problem, the imitation learning (IL) methods were proposed that learn a policy directly from expert demonstrations. Prior work addressed the IL problem through behavior cloning (BC) which learns a policy from expert trajectories using supervised learning (Pomerleau, 1991). Although BC methods are simple solutions to IL, these methods require a large amount of data because of compounding errors induced by covariate shift (Ross et al., 2011). To overcome BC limitations a generative adversarial imitation learning (GAIL) algorithm (Ho & Ermon, 2016) was proposed. GAIL uses the formulation of Generative Adversarial Networks (GANs)(Goodfellow et al., 2014), i.e., a generator-discriminator framework, where generator learns to generate expert-like trajectories and discriminator learns to distinguish between generated and expert trajectories. Although GAIL is highly effective and efficient framework, it does not recover transferable/portable reward functions along with the policies. Reward function

learning is ultimately preferable, if possible, over direct imitation learning as rewards are portable functions that represent the most basic and complete representation of agent intention, and can be re-optimized in new environments and new agents.

Reward learning is challenging as there can be many optimal policies explaining a set of demonstrations and many reward functions inducing an optimal policy (Ng et al., 2000; Ziebart et al., 2008). Recently, an adversarial inverse reinforcement learning (AIRL) framework (Fu et al., 2017), an extension of GAIL, was proposed that offers a solution to the former issue by exploiting the maximum entropy IRL method (Ziebart et al., 2008) whereas the latter issue is addressed through learning disentangled reward functions by modeling the reward as a function of state only instead of both state and action. The disentangled reward prevents actions-driven reward shaping (Fu et al., 2017) and is able to recover transferable reward functions, but has two main disadvantages. First, AIRL fails to recover the ground truth reward when the ground truth reward is a function of both state and action. For example, the reward function in any locomotion or ambulation tasks contains a penalty term that discourages actions with large magnitudes. This need for action regularization is well known in optimal control literature and limits the use cases of a state-only reward function in most practical real-life applications. Second, formulating reward as a function of state-actions induces action-driven reward shaping that often plays a vital role in quickly recovering policies (**?**) and thus for AIRL, it is usually not possible to learn optimal/near-optimal policies under state-only reward formulation.

In this paper, we propose the empowerment-regularized adversarial inverse reinforcement learning (EAIRL) algorithm[1]. Empowerment (Salge et al., 2014) is a mutual information-based theoretic measure, like state- or action-value functions, that assigns a value to a given state to quantify the extent to which an agent can influence its environment. Our method uses variational information maximization (Mohamed & Rezende, 2015) to learn empowerment in parallel to learning the reward and policy from expert data. The empowerment acts as a regularizer in policy update to prevent local behavior due to overfitting demonstrations which in practice leads to learning robust rewards. Our experimentation shows that the proposed method recovers not only near-optimal policies but also recovers robust, near-optimal, transferable, disentangled, state-action based reward functions. The results on reward learning also show that EAIRL outperforms several state-of-the-art IRL methods by recovering ground-truth reward functions. On policy learning, results demonstrate that policies learned through EAIRL perform comparably to GAIL and AIRL with non-disentangled (state-action) reward function but significantly outperform policies learned through AIRL with disentangled reward (state-only) and GAN interpretation of Guided Cost Learning (GAN-GCL) (Finn et al., 2016a).

2 BACKGROUND

We consider a Markov decision process (MDP) represented as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \rho_0, \gamma)$ where $\mathcal{S}$ denotes the state-space, $\mathcal{A}$ denotes the action-space, $\mathcal{P}$ represents the transition probability distribution, i.e., $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, $\mathcal{R}(s, a)$ corresponds to the reward function, ρ_0 is the initial state distribution $\rho_0 : \mathcal{S} \to \mathbb{R}$, and $\gamma \in (0, 1)$ is the discount factor. Let $q(a|s, s')$ be an inverse model that maps current state $s \in \mathcal{S}$ and next state $s' \in \mathcal{S}$ to a distribution over actions $\mathcal{A}$, i.e., $q : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$. Let π be a stochastic policy that takes a state and outputs a distribution over actions such that $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$. Let τ and τ_E denote a set of trajectories, a sequence of state-action pairs $(s_0, a_0, \cdots s_T, a_T)$, generated by a policy π and an expert policy π_E, respectively, where T denotes the terminal time. Finally, let $\Phi(s)$ be a potential function that quantifies a utility of a given state $s \in \mathcal{S}$, i.e., $\Phi : \mathcal{S} \to \mathbb{R}$. In our proposed work, we use an empowerment-based potential function $\Phi(\cdot)$ to regularize policy update under MaxEnt-IRL framework. Therefore, the following sections provide a brief background on MaxEnt-IRL, adversarial reward and policy learning, and variational information-maximization approach to learn the empowerment.

[1]Supplementary material is available at `https://sites.google.com/view/eairl`

2.1 MAXENT-IRL

MaxEnt-IRL (Ziebart et al., 2008) models expert demonstrations as Boltzmann distribution using parametrized reward $r_\xi(\tau)$ as an energy function, i.e.,

$$p_\xi(\tau) = \frac{1}{Z}\exp(r_\xi(\tau)) \tag{1}$$

where $r_\xi(\tau) = \sum_{t=0}^{T} r_\xi(s_t, a_t)$ is a commutative reward over given trajectory τ, parameterized by ξ, and Z is the partition function. In this framework, the demonstration trajectories are assumed to be sampled from an optimal policy π^*, therefore, they get the highest likelihood whereas the suboptimal trajectories are less rewarding and hence, are generated with exponentially decaying probability. The main computational challenge in MaxEnt-IRL is to determine Z. The initial work in MaxEnt-IRL computed Z using dynamic programming (Ziebart et al., 2008) whereas modern approaches (Finn et al., 2016b;a; Fu et al., 2017) present importance sampling technique to approximate Z under unknown dynamics.

2.2 ADVERSARIAL INVERSE REINFORCEMENT LEARNING

This section briefly describes Adversarial Inverse Reinforcement Learning (AIRL) (Fu et al., 2017) algorithm which forms a baseline of our proposed method. AIRL is state-of-the-art IRL method that builds on GAIL (Ho & Ermon, 2016), maximum entropy IRL framework (Ziebart et al., 2008) and GAN-GCL, a GAN interpretation of Guided Cost Learning (Finn et al., 2016b;a).

GAIL is a model-free adversarial learning framework, inspired from GANs (Goodfellow et al., 2014), where the policy π learns to imitate the expert policy behavior π_E by minimizing the Jensen-Shannon divergence between the state-action distributions generated by π and the expert state-action distribution by π_E through following objective

$$\min_{\pi} \max_{D\in(0,1)^{\mathcal{S}\times\mathcal{A}}} \mathbb{E}_\pi[\log D(s,a)] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))] - \lambda H(\pi) \tag{2}$$

where D is the discriminator that performs the binary classification to distinguish between samples generated by π and π_E, λ is a hyper-parameter, and $H(\pi)$ is an entropy regularization term $\mathbb{E}_\pi[\log \pi]$. Note that GAIL does not recover reward; however, Finn et al. (2016a) shows that the discriminator can be modeled as a reward function. Thus AIRL (Fu et al., 2017) presents a formal implementation of (Finn et al., 2016a) and extends GAIL to recover reward along with the policy by imposing a following structure on the discriminator:

$$D_{\xi,\varphi}(s,a,s') = \frac{\exp[f_{\xi,\varphi}(s,a,s')]}{\exp[f_{\xi,\varphi}(s,a,s')] + \pi(a|s)} \tag{3}$$

where $f_{\xi,\varphi}(s,a,s') = r_\xi(s) + \gamma h_\varphi(s') - h_\varphi(s)$ comprises disentangled reward term $r_\xi(s)$ with training parameters ξ, and shaping term $F = \gamma h_\varphi(s') - h_\varphi(s)$ with training parameters φ. The entire $D_{\xi,\varphi}(s,a,s')$ is trained as a binary classifier to distinguish between expert demonstrations τ_E and policy generated demonstrations τ. The policy is trained to maximize the discriminative reward $\hat{r}(s,a,s') = \log(D(s,a,s') - \log(1 - D(s,a,s')))$. Note that the function $F = \gamma h_\varphi(s') - h_\varphi(s)$ consists of free-parameters as no structure is imposed on $h_\varphi(\cdot)$, and as mentioned in (Fu et al., 2017), the reward function $r_\xi(\cdot)$ and function F are tied upto a constant $(\gamma - 1)c$, where $c \in \mathbb{R}$, thus the impact of F, the shaping term, on the recovered reward r is quite limited and therefore, the benefits of reward shaping are not fully realized.

2.3 EMPOWERMENT AS MAXIMAL MUTUAL INFORMATION

Mutual information (MI), an information-theoretic measure, quantifies the dependency between two random variables.

In intrinsically-motivated reinforcement learning, a maximal of mutual information between a sequence of K actions $\boldsymbol{a}$ and the final state $\boldsymbol{s'}$ reached after the execution of $\boldsymbol{a}$, conditioned on current state $\boldsymbol{s}$ is often used as a measure of internal reward (Mohamed & Rezende, 2015), known as Empowerment $\Phi(\boldsymbol{s})$, i.e.,

$$\Phi(\boldsymbol{s}) = \max I(\boldsymbol{a}, \boldsymbol{s'}|\boldsymbol{s}) = \max \mathbb{E}_{p(s'|a,s)w(a|s)}\left[\log\left(\frac{p(\boldsymbol{a}, \boldsymbol{s'}|\boldsymbol{s})}{w(\boldsymbol{a}|\boldsymbol{s})p(\boldsymbol{s'}|\boldsymbol{s})}\right)\right] \tag{4}$$

where $p(\boldsymbol{s'}|\boldsymbol{a},\boldsymbol{s})$ is a K-step transition probability, $w(\boldsymbol{a}|\boldsymbol{s})$ is a distribution over $\boldsymbol{a}$, and $p(\boldsymbol{a},\boldsymbol{s'}|\boldsymbol{s})$ is a joint-distribution of K actions $\boldsymbol{a}$ and final state $\boldsymbol{s'}$[2].

Intuitively, the empowerment $\Phi(s)$ of a state s quantifies an extent to which an agent can influence its future. Empowerment, like value functions, is a potential function that has been previously used in reinforcement learning but its applications were limited to small-scale cases due to computational intractability of MI maximization in higher-dimensional problems. However, recently a scalable method (Mohamed & Rezende, 2015) was proposed that learns the empowerment through the more-efficient maximization of variational lower bound, which has been shown to be equivalent to maximizing MI (Agakov, 2004). The lower bound was derived (for complete derivation see Appendix A.1) by representing MI in term of the difference in conditional entropies $H(\cdot)$ and utilizing the non-negativity property of KL-divergence, i.e.,

$$I^w(s) = H(a|s) - H(a|s',s) \geq H(a) + \mathbb{E}_{p(s'|a,s)w_\theta(a|s)}[\log q_\phi(a|s',s)] = I^{w,q}(s) \tag{5}$$

where $H(a|s) = -\mathbb{E}_{w(a|s)}[\log w(a|s)]$, $H(a|s',s) = -\mathbb{E}_{p(s'|a,s)w(a|s)}[\log p(a|s',s)]$, $q_\phi(\cdot)$ is a variational distribution with parameters ϕ and $w_\theta(\cdot)$ is a distribution over actions with parameters θ.

Finally, the lower bound in Eqn. 5 is maximized under the constraint $H(a|s) < \eta$ (to avoid divergence, see (Mohamed & Rezende, 2015)) to compute empowerment as follow:

$$\Phi(s) = \max_{w,q} \mathbb{E}_{p(s'|a,s)w(a|s)}[-\frac{1}{\beta}\log w_\theta(a|s) + \log q_\phi(a|s',s)] \tag{6}$$

where β is η dependent temperature term. Mohamed & Rezende (2015) also applied the principles of Expectation-Maximization (EM) (Agakov, 2004) to learn empowerment, i.e., alternatively maximizing Eqn. 6 with respect to $w_\theta(a|s)$ and $q_\phi(a|s',s)$. Given a set of training trajectories τ, the maximization of Eqn. 6 w.r.t $q_\phi(\cdot)$ is shown to be a supervised maximum log-likelihood problem whereas the maximization w.r.t $w_\theta(\cdot)$ is determined through the functional derivative $\partial I/\partial w = 0$ under the constraint $\sum_a w(a|s) = 1$. The optimal w^* that maximizes Eqn. 6 turns out to be $\frac{1}{Z(s)}\exp(\beta\mathbb{E}_{p(s'|s,a)}[\log q_\phi(a|s,s')])$, where $Z(s)$ is a normalization term. By substituting w^* in Eqn. 6 showed that the empowerment $\Phi(s) = \frac{1}{\beta}\log Z(s)$ (for full derivation, see Appendix A.2).

Note that $w^*(a|s)$ is implicitly unnormalized as there is no direct mechanism for sampling actions or computing $Z(s)$ (Mohamed & Rezende, 2015). Mohamed & Rezende (2015) introduced an approximation $w^*(a|s) \approx \log \pi(a|s) + \Phi(s)$ where $\pi(a|s)$ is a normalized distribution which leaves the scalar function $\Phi(s)$ to account for the normalization term $\log Z(s)$. Finally, the parameters of policy π and scalar function Φ are optimized by minimizing the discrepancy between the two approximations $(\log \pi(a|s) + \Phi(s))$ and $\beta \log q_\phi(a|s',s))$ through the mean-square or absolute error. In our work, we opt absolute error as follow:

$$l_I(s,a,s') = \left|\beta \log q_\phi(a|s',s) - (\log \pi_\theta(a|s) + \Phi_\varphi(s))\right| \tag{7}$$

3 Empowered Adversarial Inverse Reinforcement Learning

We present an inverse reinforcement learning algorithm that learns a robust, transferable reward function and policy from expert demonstrations. Our proposed method comprises (i) an inverse model $q_\phi(a|s',s)$ that takes the current state s and the next state s' to output a distribution over actions $\mathcal{A}$ that resulted in s to s' transition, (ii) a reward $r_\xi(s,a)$, with parameters ξ, that is a function of both state and action, (iii) an empowerment-based potential function $\Phi_\varphi(\cdot)$ with parameters φ that determines the reward-shaping function $F = \gamma\Phi_\varphi(s') - \Phi_\varphi(s)$ and also regularizes the policy update, and (iv) a policy model $\pi_\theta(a|s)$ that outputs a distribution over actions given the current state s. All these models are trained simultaneously based on the objective functions described in the following sections.

[2]In our proposed work, we consider only immediate step transitions i.e., $K = 1$, hence variables $\boldsymbol{s}, \boldsymbol{a}$ and $\boldsymbol{s'}$ will be represented in non-bold notations.

3.1 INVERSE MODEL $q_\phi(a|s, s')$ OPTIMIZATION

As mentioned in Section 2.3, learning the inverse model $q_\phi(a|s, s')$ is a maximum log-likelihood supervised learning problem. Therefore, given a set of trajectories $\tau \sim \pi$, where a single trajectory is a sequence states and actions, i.e., $\tau_i = \{s_0, a_0, \cdots, s_T, a_T\}_i$, the inverse model $q_\phi(a|s', s)$ is trained to minimize the mean-square error between its predicted action $q(a|s', s)$ and the action a taken according to the generated trajectory τ, i.e.,

$$l_q(s, a, s') = (q_\phi(\cdot|s, s') - a)^2 \tag{8}$$

3.2 EMPOWERMENT $\Phi_\varphi(s)$ OPTIMIZATION

Empowerment will be expressed in terms of normalization function $Z(s)$ of optimal $w^*(a|s)$, i.e., $\Phi_\varphi(s) = \frac{1}{\beta} \log Z(s)$. Therefore, the estimation of empowerment $\Phi_\varphi(s)$ is approximated by minimizing the loss function $l_I(s, a, s')$, presented in Eqn. 7, w.r.t parameters φ, and the inputs (s, a, s') are sampled from the policy-generated trajectories τ.

3.3 REWARD FUNCTION $r_\xi(s, a)$

To train the reward function, we first compute the discriminator as follow:

$$D_{\xi,\varphi}(s, a, s') = \frac{\exp[r_\xi(s, a) + \gamma\Phi_{\varphi'}(s') - \Phi_\varphi(s)]}{\exp[r_\xi(s, a) + \gamma\Phi_{\varphi'}(s') - \Phi_\varphi(s)] + \pi_\theta(a|s)} \tag{9}$$

where $r_\xi(s, a)$ is the reward function to be learned with parameters ξ. We also maintain the target φ' and learning φ parameters of the empowerment-based potential function. The target parameters φ' are a replica of φ except that the target parameters φ' are updated to learning parameters φ after every n training epochs. Note that keeping a stationary target $\Phi_{\varphi'}$ stabilizes the learning as also highlighted in (Mnih et al., 2015). Finally, the discriminator/reward function parameters ξ are trained via binary logistic regression to discriminate between expert τ_E and generated τ trajectories, i.e.,

$$\mathbb{E}_\tau[\log D_{\xi,\varphi}(s, a, s')] + \mathbb{E}_{\tau_E}[(1 - \log D_{\xi,\varphi}(s, a, s'))] \tag{10}$$

3.4 POLICY OPTIMIZATION POLICY $\pi_\theta(a|s)$

We train our policy $\pi_\theta(a|s)$ to maximize the discriminative reward $\hat{r}(s, a, s') = \log(D(s, a, s') - \log(1 - D(s, a, s')))$ and to minimize the loss function $l_I(s, a, s') = \big|\beta \log q_\phi(a|s, s') - (\log \pi_\theta(a|s) + \Phi_\varphi(s))\big|$ which accounts for empowerment regularization. Hence, the overall policy training objective is:

$$\mathbb{E}_\tau[\log \pi_\theta(a|s)\hat{r}(s, a, s')] + \lambda_I \mathbb{E}_\tau\big[l_I(s, a, s')\big] \tag{11}$$

where policy parameters θ are updated by taking KL-constrained natural gradient step using any policy optimization method such as TRPO (Schulman et al., 2015) or an approximated step such as PPO (Schulman et al., 2017).

Algorithm 1 outlines the overall training procedure to train all function approximators simultaneously. Note that the expert samples τ_E are seen by the discriminator only, whereas all other models are trained using the policy generated samples τ. Furthermore, as highlighted in (Fu et al., 2017), the discriminating reward $\hat{r}(s, a, s')$ boils down to the following expression:

$$\hat{r}(s, a, s') = f(s, a, s') - \log \pi(a|s)$$

where $f(s, a, s') = r_\xi(s, a) + \gamma\Phi_{\varphi'}(s') - \Phi_\varphi(s)$. Furthermore, an alternative way to express our policy training objective is $\mathbb{E}_\tau[\log \pi_\theta(a|s) r_\pi(s, a, s')]$ where $r_\pi(s, a, s') = \hat{r}(s, a, s') - \lambda_I l_I(s, a, s')$ which would undoubtedly yield the same results as Eqn. 11, i.e., maximize the discriminative reward and minimize the loss l_I. The analysis of this alternative expression is given in Appendix B to highlight that our policy update rule is equivalent to MaxEnt-IRL policy objective (Finn et al., 2016a) except that it also maximizes the empowerment, i.e.,

$$r_\pi(s, a, s') = r_\xi(s, a, s') + \gamma\Phi(s') + \lambda\hat{H}(\cdot)$$

Algorithm 1: Empowerment-based Adversarial Inverse Reinforcement Learning

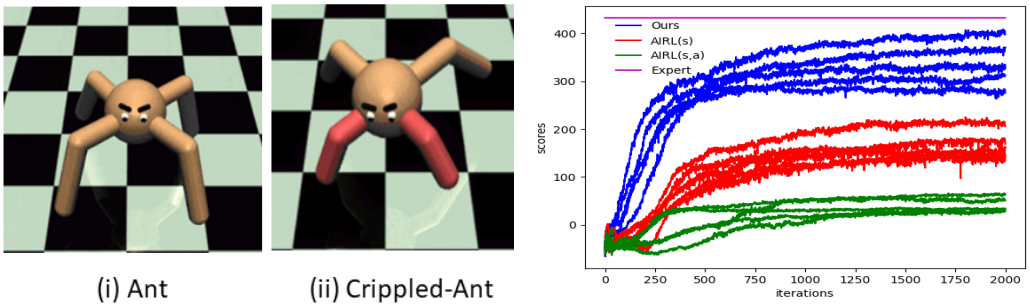
Initialize parameters of policy π_θ, and inverse model q_ϕ
Initialize parameters of target $\Phi_{\varphi'}$ and training Φ_φ empowerment, and reward r_ξ functions
Obtain expert demonstrations τ_E by running expert policy π_E
for $i \leftarrow 0$ ***to*** N **do**
  Collect trajectories τ by executing π_θ
  Update ϕ_i to ϕ_{i+1} with the gradient $\mathbb{E}_\tau[\bigtriangledown_{\phi_i} l_q(s,a,s')]$
  Update φ_i to φ_{i+1} with the gradient $\mathbb{E}_\tau[\bigtriangledown_{\varphi_i} l_I(s,a,s')]$
  Update ξ_i to ξ_{i+1} with the gradient:

$$\mathbb{E}_\tau[\bigtriangledown_{\xi_i} \log D_{\xi_i,\varphi_{i+1}}(s,a,s')] + \mathbb{E}_{\tau_E}[\bigtriangledown_{\xi_i}(1 - \log D_{\xi_i,\varphi_{i+1}}(s,a,s'))]$$

  Update θ_i to θ_{i+1} using TRPO/PPO update rule with the following objective:

$$\mathbb{E}_\tau\big[\bigtriangledown_{\theta_i} \log \pi_{\theta_i}(a|s)\hat{r}_{\xi_{i+1}}(s,a,s')\big] + \lambda_I \mathbb{E}_\tau\big[\bigtriangledown_{\theta_i} l_I(s,a,s')\big]$$

  After every n epochs sync φ' with φ



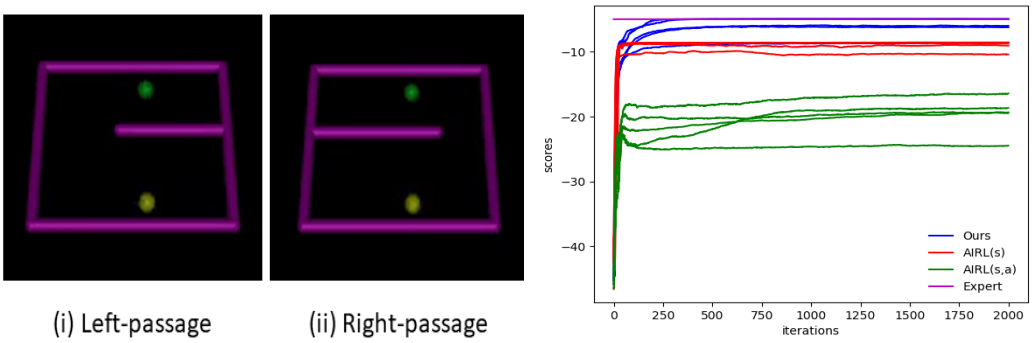
(a) Ant environments (b) Policy performance on learned rewards.

Figure 1: The policy performance in the crippled-ant environment based on the reward learned using expert demonstrations in the normal ant enviroment. It can be seen that our method performs significantly better than AIRL and exhibits expert-like performance in all five trials which implies that our method almost recovered ground-truth reward function.

where, λ and γ are hyperparameters, and $\hat{H}(\cdot)$ is the entropy-regularization term depending on $\pi(\cdot)$ and $q(\cdot)$. Hence, our policy is regularized by the empowerment which induces generalized behavior rather than locally overfitting to the limited expert demonstrations.

4 Results

Our proposed method, EAIRL, learns both reward and policy from expert demonstrations. We evaluate our method against state-of-the-art policy and reward learning techniques on several control tasks in OpenAI Gym. In case of policy learning, we compare our method against GAIL, GAN-GCL, AIRL with state-only reward, denoted as $\mathrm{AIRL}(s)$, and AIRL with state-action reward, denoted as $\mathrm{AIRL}(s,a)$. In reward learning, we only compare our method against $\mathrm{AIRL}(s)$ and $\mathrm{AIRL}(s,a)$ as GAIL does not recover reward, and GAN-GCL is shown to exhibit inferior performance than AIRL (Fu et al., 2017). Furthermore, in the comparisons, we also include the expert performances which represents a policy learned by optimizing a ground-truth reward using TRPO. The performance of different methods are evaluated in term of average total reward accumulated (denoted as score) by an agent during the trial, and for each experiment, we run five trials.



(a) Pointmass-maze environments

(b) Policy performance on transfered rewards.

Figure 2: The policy performance on a transfer learning task where the learned rewards are tested in a shifted maze. The task is to navigate the 2D agent (yellow) to the goal (green) and the transfer involves learning to take a completely an opposite route to the goal. It can be seen that our method recovers near-optimal reward functions and exhibit better performance than AIRL in all five trials.

Table 1: The evaluation of reward learning on transfer learning tasks. Mean scores (higher the better) with standard deviation are presented over 5 trials.

Algorithm	States-Only	Pointmass-Maze	Crippled-Ant
Expert	N/A	-4.98 ± 0.29	432.66 ± 14.38
EAIRL(Ours)	**No**	$\mathbf{-7.01 \pm 0.61}$	$\mathbf{348.43 \pm 43.17}$
AIRL	Yes	-8.07 ± 0.50	175.51 ± 27.31
AIRL	No	-19.28 ± 2.03	46.12 ± 14.37

4.1 Reward learning performance (Transfer learning experiments)

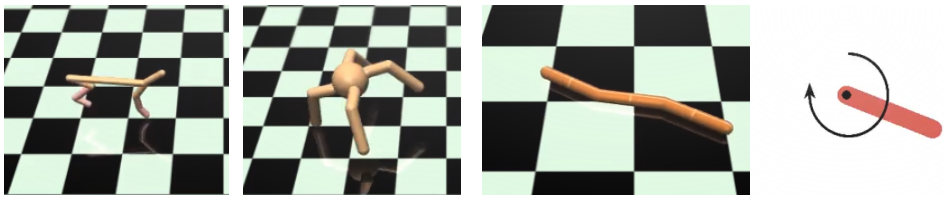
To evaluate the learned rewards, we consider a transfer learning problem in which the testing environments are made to be different from the training environments. More precisely, the rewards learned via IRL in the training environments are used to re-optimize a new policy in the testing environment using standard RL formulation. We consider two test cases, shown in the Fig. 1 and Fig. 2, in which the agent's dynamics and physical environment is modified, respectively.

In the first test case, as shown in Fig. 1(a), we modify the agent itself during testing. We trained a reward function to make a standard quadrupled ant to run forward. During testing, we disabled the front two legs (indicated in red) of the ant (crippled-ant), and the learned reward is used to re-optimize the policy to make a crippled-ant move forward. Note that the crippled-ant cannot move sideways (Appendix C.1). Therefore, the agent has to change the gait to run forward. In the second test case, shown in Fig 2(a), the agent learns to navigate a 2D point-mass to the goal region in a simple maze. We re-position the maze central-wall during testing so that the agent has to take a different path, compared to the training environment, to reach the target (Appendix C.2).

Fig. 1(b) and Fig. 2(b) compare the policy performance scores over five different trials of EAIRL, $\mathrm{AIRL}(s)$ and $\mathrm{AIRL}(s, a)$ in the aforementioned transfer learning tasks. The expert score is shown as a horizontal line to indicate the standard set by an expert policy. Table 1 summarizes the mean score of five trials with a standard deviation in above-mentioned transfer learning experiments. It can be seen that our method recovers near-optimal reward functions as the policy scores almost reach the expert scores in all five trials. Furthermore, our method performs significantly better than both $\mathrm{AIRL}(s)$ and $\mathrm{AIRL}(s, a)$ in matching an expert's performance.

4.2 Policy learning performance (Imitation learning)

Table 2 presents the means and standard deviations of policy learning performance scores, over the five different trials, in various control tasks. For each algorithm, we provided 20 expert demonstra-



(a) HalfCheetah (b) Ant (c) Swimmer (d) Pendulum

Figure 3: Benchmark control tasks for imitation learning

tions for imitation, generated by optimizing a policy on a ground-truth reward using TRPO. The tasks, shown in Fig. 3, include (i) making a 2D cheetah robot to run forward, (ii) making a 3D quadrupled robot (ant) to move forward, (iii) making a 2D robot to swim (swimmer), and (iv) keeping a friction less pendulum to stand vertically up. It can be seen that EAIRL, $\mathrm{AIRL}(s, a)$ and GAIL demonstrate similar performance and successfully learn to imitate expert policy whereas $\mathrm{AIRL}(s)$ and GAN-GCL fails to recover a policy.

Table 2: The evaluation of imitation learning on benchmark control tasks. Mean scores (higher the better) with standard deviation are presented over 5 trials for each method. The top two algorithms are highlighted in bold.

Methods	Environments			
	HalfCheetah	Ant	Swimmer	Pendulum
Expert	2139.83 ± 30.22	935.12 ± 10.94	76.21 ± 1.79	-100.11 ± 1.32
GAIL	$\mathbf{1880.05 \pm 15.72}$	$\mathbf{738.72 \pm 9.49}$	$\mathbf{50.21 \pm 0.26}$	$\mathbf{-116.01 \pm 5.45}$
GCL	-189.90 ± 44.42	16.74 ± 36.59	15.75 ± 7.32	-578.18 ± 72.84
AIRL(s,a)	1826.26 ± 19.64	$\mathbf{645.90 \pm 41.75}$	49.52 ± 0.48	-118.13 ± 11.33
AIRL(s)	121.10 ± 42.31	271.31 ± 9.35	33.21 ± 2.40	-134.82 ± 10.89
EAIRL	$\mathbf{1870.10 \pm 17.86}$	641.12 ± 25.92	$\mathbf{49.55 \pm 0.29}$	$\mathbf{-116.26 \pm 8.313}$

5 Discussion

This section highlights the importance of empowerment-regularized MaxEnt-IRL and modeling rewards as a function of both state and action rather than restricting to state-only formulation on learning rewards and policies from expert demonstrations.

In the scalable MaxEnt-IRL framework (Finn et al., 2016a; Fu et al., 2017), the normalization term is approximated by importance sampling where the importance-sampler/policy is trained to minimize the KL-divergence from the distribution over expert trajectories. In our proposed work, we regularize the policy update with empowerment. Hence, we update our policy to reduce the divergence from expert data distribution as well as to maximize the empowerment. The proposed regularization prevents premature convergence to local behavior thus leads to robust rewards learning without any restriction on modeling rewards as a function of states only. Furthermore, empowerment quantifies the extent to which an agent can control/influence its environment in the given state. Note that our method trains policy as to maximize the empowerment of next state that means the agent takes action a on observing state s such that it has maximum control/influence over the environment upon ending up in the future state s'.

Our experimentation also highlights the importance of modeling discriminator/reward functions as a function of both state and action in reward and policy learning under GANs framework. The reward learning results show that AIRL with state-only rewards (AIRL(s)) does not recover the action dependent terms of the ground-truth reward function that penalizes high torques. Therefore, the agent shows aggressive behavior and sometimes flips over after few steps (see the accompanying video), which is also the reason that crippled-ant trained with AIRL's disentangled reward function reaches only the half-way to expert scores as shown in Table 1. The reward formulation as a function of both states and actions is crucial in many real-life robotics applications for action regularization in any locomotion or ambulation tasks that discourage actions with large magnitudes. The policy

learning results further validate the importance of state-action reward formulation. Table 2 shows that AIRL with state-only rewards, AIRL(s), fails to learn a policy whereas EAIRL, GAIL, and AIRL that include state-action reward/discriminator formulation successfully recover the expert-like behavior policies. Hence, our empirical results show that it is crucial to model reward/discriminator as a function of state-action as otherwise, adversarial imitation learning fails to learn ground-truth rewards and expert-like policies from expert data.

6 Conclusions and Future Work

We present an approach to adversarial reward and policy learning from expert demonstrations by regularizing the maximum-entropy inverse reinforcement learning through empowerment. Our method learns the empowerment through variational information maximization in parallel to learning the reward and policy. We show that our policy is trained to imitate the expert behavior as well to maximize the empowerment over the environment. The proposed regularization prevents premature convergence to local behavior and leads to generalized policies that in turn guides the reward-learning process to recover near-optimal reward. We show that our method successfully learns near-optimal rewards, policies, and performs significantly better than state-of-the-art IRL methods in both imitation learning and challenging transfer learning problems. The learned rewards are shown to be transferable to environments that are dynamically or structurally different from training environments.

In our future work, we plan to extend our method to learn rewards and policies from diverse human/expert demonstrations as the proposed method assumes that a single expert generates the training data. Another exciting direction is to learn from sub-optimal demonstrations that also contains failures in addition to optimal behaviors.

References

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1. ACM, 2004.

David Barber Felix Agakov. The im algorithm: a variational approach to information maximization. *Advances in Neural Information Processing Systems*, 16:201, 2004.

Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 182–189, 2011.

Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852*, 2016a.

Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*, pp. 49–58, 2016b.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 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.

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

Mrinal Kalakrishnan, Peter Pastor, Ludovic Righetti, and Stefan Schaal. Learning objective functions for manipulation. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 1331–1336. IEEE, 2013.

Markus Kuderer, Shilpa Gulati, and Wolfram Burgard. Learning driving styles for autonomous vehicles from demonstration. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 2641–2646. IEEE, 2015.

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.

Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pp. 2125–2133, 2015.

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

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

Ahmed Hussain Qureshi, Yutaka Nakamura, Yuichiro Yoshikawa, and Hiroshi Ishiguro. Show, attend and interact: Perceivable human-robot social interaction through neural attention q-network. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 1639–1645. IEEE, 2017.

Ahmed Hussain Qureshi, Yutaka Nakamura, Yuichiro Yoshikawa, and Hiroshi Ishiguro. Intrinsically motivated reinforcement learning for human–robot interaction in the real-world. *Neural Networks*, 2018.

Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning*, pp. 729–736. ACM, 2006.

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.

Christoph Salge, Cornelius Glackin, and Daniel Polani. Empowerment–an introduction. In *Guided Self-Organization: Inception*, pp. 67–114. Springer, 2014.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897, 2015.

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

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

Michael Yip and Nikhil Das. Robot autonomy for surgery. *arXiv preprint arXiv:1707.03080*, 2017.

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.

APPENDICES

A VARIATIONAL EMPOWERMENT

For completeness, we present a derivation of presenting mutual information (MI) as variational lower bound and maximization of lower bound to learn empowerment.

A.1 VARIATIONAL INFORMATION LOWER BOUND

As mentioned in section 2.3, the variational lower bound representation of MI is computed by defining MI as a difference in conditional entropies, and the derivation is formalized as follow.

$$\begin{aligned}
I^{w,q}(s) &= H(a|s) - H(a|s',s) \\
&= H(a|s) + \mathbb{E}_{p(s'|a,s)w(a|s)}[\log p(a|s',s)] \\
&= H(a|s) + \mathbb{E}_{p(s'|a,s)w(a|s)}[\log \frac{p(a|s',s)q(a|s',s)}{q(a|s',s)}] \\
&= H(a|s) + \mathbb{E}_{p(s'|a,s)w(a|s)}[\log q(a|s',s)] + \mathbb{E}_{p(s'|a,s)w(a|s)}[\log \frac{p(a|s',s)}{q(a|s',s)}] \\
&= H(a|s) + \mathbb{E}_{p(s'|a,s)w(a|s)}[\log q(a|s',s)] + \mathrm{KL}[p(a|s',s)||q(a|s',s)] \\
&\geq H(a|s) + \mathbb{E}_{p(s'|a,s)w(a|s)}[\log q(a|s',s)] \\
&\geq -\mathbb{E}_{w(a|s)} \log w(a|s) + \mathbb{E}_{p(s'|a,s)w(a|s)}[\log q(a|s',s)]
\end{aligned}$$

A.2 VARIATIONAL INFORMATION MAXIMIZATION

The empowerment is a maximal of MI and it can be formalized as follow by exploiting the variational lower bound formulation (for details see (Mohamed & Rezende, 2015)).

$$\Phi(s) = \max_{w,q} \mathbb{E}_{p(s'|a,s)w(a|s)}[-\frac{1}{\beta}\log w(a|s) + \log q(a|s',s)] \tag{12}$$

As mentioned in section 2.3, given a training trajectories, the maximization of Eqn. 12 w.r.t inverse model $q(a|s',s)$ is a supervised maximum log-likelihood problem. The maximization of Eqn. 12 w.r.t $w(a|s)$ is derived through a functional derivative $\partial I^{w,q}/\partial w = 0$ under the constraint $\sum_a w(a|s) = 1$. For simplicity, we consider discrete state and action spaces, and the derivation is as follow:

$$\begin{aligned}
\hat{I}^w(s) &= \mathbb{E}_{p(s'|a,s)w(a|s)}[-\frac{1}{\beta}\log w(a|s) + \log q(a|s',s)] + \lambda\big(\sum_a w(a|s) - 1\big) \\
&= \sum_a \sum_{s'} p(s'|a,s)w(a|s)\{-\frac{1}{\beta}\log w(a|s) + \log q(a|s',s)\} + \lambda\big(\sum_a w(a|s) - 1\big)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \hat{I}^w(s)}{\partial w} &= \sum_a \{(\lambda - \beta) - \log w(a|s) + \beta \mathbb{E}_{p(s'|a,s)}[\log q(a|s',s)]\} = 0 \\
w(a|s) &= e^{\lambda-\beta} e^{\beta \mathbb{E}_{p(s'|a,s)}[\log q(a|s',s)]}
\end{aligned}$$

By using the constraint $\sum_a w(a|s) = 1$, it can be shown that the optimal solution $w^*(a|s) = \frac{1}{Z(s)}\exp(u(s,a))$, where $u(s,a) = \beta \mathbb{E}_{p(s'|a,s)}[\log q(a|s',s)]$ and $Z(s) = \sum_a u(s,a)$. This solution maximizes the lower bound since $\partial^2 I^w(s)/\partial w^2 = -\sum_a \frac{1}{w(a|s)} < 0$.

B EMPOWERMENT-REGULARIZED MAXENT-IRL FORMULATION.

In this section we derive the Empowerment-regularized formulation of maximum entropy IRL. Let τ be a trajectory sampled from expert demonstrations D and $p_\xi(\tau) \propto p(s_0)\Pi_{t=0}^{T-1} p(s_{t+1}|s_t, a_t) \exp^{r_\xi(s_t,a_t)}$ be a distribution over τ. As mentioned in Section 2, the IRL objective is to maximize the likelihood:

$$\max_\xi J(\xi) = \max_\xi \mathbb{E}_D[\log p_\xi(\tau)]$$

Furthermore, as derived in (Fu et al., 2017), the gradient of above equation w.r.t ξ can be written as:

$$\begin{aligned}\max_\xi J(\xi) &= \mathbb{E}_D[\sum_{t=0}^{T} \frac{\partial}{\partial \xi} r_\xi(s_t, a_t)] - \mathbb{E}_{p_\xi}[\sum_{t=0}^{T} \frac{\partial}{\partial \xi} r_\xi(s_t, a_t)] \\ &= \sum_{t=0}^{T} \mathbb{E}_D[\frac{\partial}{\partial \xi} r_\xi(s_t, a_t)] - \mathbb{E}_{p_{\xi,t}}[\frac{\partial}{\partial \xi} r_\xi(s_t, a_t)]\end{aligned}$$

where $r_\xi(\cdot)$ is a parametrized reward to be learned, and $p_{\xi,t} = \int_{s_{t'} \neq t, a_{t'} \neq t} p_\xi(\tau)$ denotes marginalization of state-action at time t. Since, it is unfeasible to draw samples from p_ξ, Finn et al. (2016a) proposed to train an importance sampling distribution $\mu(\tau)$ whose varience is reduced by defining $\mu(\tau)$ as a mixture of polices, i.e., $\mu(a|s) = \frac{1}{2}(\pi(a|s) + \hat{p}(a|s))$, where $\hat{p}$ is a rough density estimate over demonstrations. Thus the above gradient becomes:

$$\frac{\partial}{\partial \xi} J(\xi) = \sum_{t=0}^{T} \mathbb{E}_D[\frac{\partial}{\partial \xi} r_\xi(s_t, a_t)] - \mathbb{E}_{\mu_t}[\frac{p_{\xi,t}(s_t, a_t)}{\mu_t(s_t, a_t)} \frac{\partial}{\partial \xi} r_\xi(s_t, a_t)] \tag{13}$$

We train our importance-sampler/policy π to maximize the empowerment $\Phi(\cdot)$ for generalization and to reduce divergence from true distribution by minimizing $D_{\mathrm{KL}}(\pi(\tau)\|p_\xi(\tau))$. Since, $\pi(\tau) = p(s_0)\Pi_{t=0}^{T-1} p(s_{t+1}|s_t, a_t)\pi(s_t, a_t)$, the matching terms of $\pi(\tau)$ and $p_\xi(\tau)$ cancel out, resulting into entropy-regularized policy update. Furthermore, as we also include the empowerment $\Phi(\cdot)$ in the policy update to be maximized, hence the overall objective becomes:

$$\max_\pi \mathbb{E}_\pi[\sum_{t=0}^{T-1} r_\xi(s_t, a_t) + \Phi(s_{t+1}) - \log \pi(a_t|s_t)] \tag{14}$$

Our discriminator is trained to minimize cross entropy loss as mention in Eqn. 10, and for the proposed structure of our discriminator Eqn. 9, it can be shown that the discriminator's gradient w.r.t its parameters turns out to be equal to Equation 13. On the other hand, our policy training objective is

$$r_\pi(s, a, s') = \log(D(s, a, s')) - \log(1 - D(s, a, s')) - l_I(s, a, s') \tag{15}$$

In the next section, we show that the above policy training objective is equivalent to Equation 14.

B.1 POLICY OBJECTIVE

We train our policy to maximize the discriminative reward $\hat{r}(s, a, s') = \log(D(s, a, s') - \log(1 - D(s, a, s')))$ and minimize the information-theoretic loss function $l_I(s, a, s') = |\log q_\phi(a|s, s') - (\log \pi(a|s) + \Phi(s))|$. The discriminative reward $\hat{r}(s, a, s')$ simplifies to:

$$\begin{aligned}\hat{r}(s, a, s') &= \log(D(s, a, s')) - \log(1 - D(s, a, s')) \\ &= \log \frac{e^{f(s,a,s')}}{e^{f(s,a,s')} + \pi(a|s)} - \log \frac{\pi(a|s)}{e^{f(s,a,s')} + \pi(a|s)} \\ &= f(s, a, s') - \log \pi(a|s)\end{aligned}$$

where $f(s, a, s') = r(s, a) + \gamma\Phi(s') - \Phi(s)$. The entropy-regularization is usually scaled by the hyperparameter, let say $\lambda_h \in \mathbb{R}$, thus $\hat{r}(s, a, s') = f(s, a, s') - \lambda_h \log \pi(a|s)$. Hence, assuming single-sample (s, a, s') and $l_i > 0$, the policy is trained to maximize following:

$$\begin{aligned} r_\pi(s, a, s') &= f(s, a, s') - \lambda_h \log \pi(a|s) - l_I(s, a, s') \\ &= r(s, a) + \gamma\Phi(s') - \Phi(s) - \lambda_h \log \pi(a|s) - \log q(a|s, s') + \log \pi(a|s) + \Phi(s) \\ &= r(s, a) + \gamma\Phi(s') - \lambda_h \log \pi(a|s) - \log q(a|s, s') + \log \pi(a|s) \end{aligned}$$

Note that, the potential function $\Phi(s)$ cancels out and we scale the leftover terms of l_I with a hyperparameter λ_I. Hence, the above equation becomes:

$$r_\pi(s, a, s') = r(s, a, s') + \gamma\Phi(s') + (\lambda_I - \lambda_h) \log \pi(a|s) - \lambda_I \log q(a|s, s')$$

We combine the log terms together as:

$$r_\pi(s, a, s') = r(s, a) + \lambda_I \Phi(s') + \lambda \hat{H}(\cdot) \tag{16}$$

$$\tag{17}$$

where λ is a hyperparameter, and $\hat{H}(\cdot)$ is an entropy regularization term depending on $q(a|s, s')$ and $\pi(a|s)$. Therefore, it can be seen that the Eqn. 16 is equivalent/approximation to Eqn. 14.

C TRANSFER LEARNING PROBLEMS

C.1 ANT ENVIRONMENT

The following figures show the difference between the path profiles of standard and crippled Ant. It can be seen that the standard Ant can move sideways whereas the crippled ant has to rotate in order to move forward.

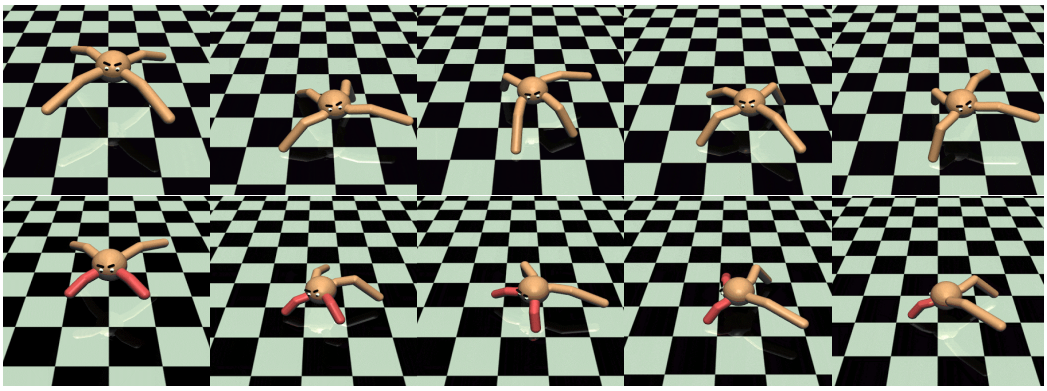


Figure 4: The top and bottom rows show the gait of standard and crippled ant, respectively.

C.2 MAZE ENVIRONMENT

The following figures show the path profiles of a 2D point-mass agent to reach the target in training and testing environment. It can be seen that in the testing environment the agent has to take the opposite route compared to the training environment to reach the target.

D IMPLEMENTATION DETAILS

D.1 NETWORK ARCHITECTURES

We use two-layer ReLU network with 32 units in each layer for the potential function $h_\varphi(\cdot)$ and $\Phi_\varphi(\cdot)$, reward function $r_\xi(\cdot)$, discriminators of GAIL and GAN-GCL. Furthermore, policy $\pi_\theta(\cdot)$ of

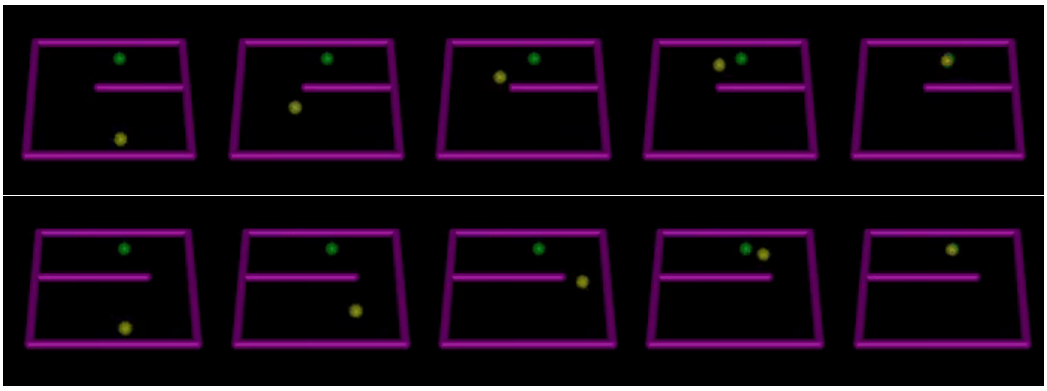


Figure 5: The top and bottom rows show the path followed by a 2D point-mass agent (yellow) to reach the target (green) in training and testing environment, respectively.

all presented models and the inverse model $q_\phi(\cdot)$ of EAIRL are presented by two-layer RELU network with 32 units in each layer, where the network's output parametrizes the Gaussian distribution, i.e., we assume a Gaussian policy.

D.2 HYPERPARAMETERS

For all experiments, we use the temperature term $\beta = 1$. We set entropy regularization weight to 0.1 and 0.001 for reward and policy learning, respectively. The hyperparameter λ_I was set to 1.0 for reward learning and 0.002 for policy learning. The target parameters of the empowerment-based potential function $\Phi_{\varphi'}(\cdot)$ were updated every 5 and 2 epochs during reward and policy learning respectively. Although reward learning hyperparameters are also applicable to policy learning, we decrease the magnitude of entropy and information regularizers during policy learning to speed up the policy convergence to optimal values. Furthermore, we set the batch size to 2000- and 20000-steps per TRPO update for the pendulum and remaining environments, respectively. For the methods (Fu et al., 2017; Ho & Ermon, 2016) presented for comparison, we use their suggested hyperparameters. We also use policy samples from previous 20 iterations as negative data to train the discriminator of all IRL methods presented in this paper to prevent the parametrized reward functions from overfitting the current policy samples.