Model Based Adversarial Inverse Reinforcement Learning

Lalit Lal
University of Toronto
lallalit@cs.toronto.edu

Saad Saleem
University of Toronto
saad@cs.toronto.edu

Yip Sang Leung
University of Toronto
yipsang@cs.toronto.edu

Abstract: In policy imitation, generative adversarial learning has been successfully applied in a model free approach as well as a model-based approach. The main benefits of the model-based approach include sample efficiency and parameter count reduction, as well as tractability in the computation graph for end-to-end training using backpropagation. This approach is called MAIL.

On another view, reinforcement learning is a framework for decision making and control, but often needs explicit reward and/or feature engineering. Inverse reinforcement learning (IRL) methods show promise to learn rewards from unknown dynamics. A method applying adversarial inverse reinforcement learning (AIRL) is able to recover reward functions that are robust to changes in dynamics that were not seen during training. However, IRL has proven to be difficult in large, high-dimensional problems with unknown dynamics, and there is potential for a model-based setup to help alleviate this.

Since model-based methods have shown to allow for end-to-end training using backpropagation, but can suffer if the environment dynamics change during testing, an intuitive approach would be to attempt to learn a robust reward that can adapt to such changes. In this paper we introduce Model-based Adversarial Inverse Reinforcement Learning (MAIRL), a novel algorithm that jointly learns a model and reward function in an adversarial setting (i.e., from a discriminator). The MAIL and MAIRL algorithms are benchmarked against an open source variant of OpenAI MuJoCo environments, called Pybullet-Gym. We show that MAIRL is not only able to significantly outperform the given expert agent as well as MAIL, but also learn a meaningful reward that can be applied to transfer settings, and is the only tested method that can successfully traverse an agent in an environment with new dynamics. A link to our code is provided in Appendix B.

1 Introduction

Imitation learning is an approach that aims to train an agent performing a task based on expert demonstrations. This alleviates the difficulty of training an agent from scratch. Imitation learning is useful in a variety of applications in including automation, policy improvement (using expert as starting point, granted it is known), and potentially even simplification (finding reduced set of parameters to make the policy more efficient).

In our setting, we assume that sample trajectories of an expert policy π_E for an environment are given $\{s_0, a_0, s_1, ...\}_{i=0}^{N}$, and we attempt to learn a forward model, discriminator, and policy.

Imitation problems can either be approached with matching the behaviour of an expert, or attempting to generalize and improve upon the expert by learning underlying signals. The first approach is known as behavioural cloning (BC) which learns the the conditional distribution of action over states $p(a|s)$ from demonstration using supervised learning, using the reward signal as the supervision [16]. However, BC methods suffer from large sample complexity and compounding errors known as covariate shifts because they don't take into account the dynamics of the environment (BC methods are trained on single state-action pairs) [13].

4th Conference on Robot Learning (CoRL 2020), Cambridge MA, USA.

The second approach to imitation problems include recovering a reward function that helps explain the expert behaviour to be uniquely optimal, an approach known as Inverse Reinforcement Learning (IRL) [1], and then applying reinforcement learning (RL) techniques to maximize the discounted cumulative expected return, $\mathbb{E}_\pi R = \mathbb{E}_\pi[\Sigma_{t=0}^{T} \gamma^t \widehat{r}_t]$. The issue with IRL is that it does not scale to high-dimensional environments and requires a lot of domain knowledge to define a reward [11].

On another note, Generative adversarial networks (GANs) have recently been proven successful in many generative tasks [6]. GANs use a discriminator D neural network (NN) to provide a supervision signal in the form of a reward. The discriminator is trained to distinguish between the generative model G output and the expert data.

An innovative idea was raised to combine the best of both worlds - imitation learning and GANs. Generative Adversarial Imitation learning (GAIL) was proposed to train an imitation policy under the GAN setting and the result was promising [8]. This work showed that adversarial learning can help address covariate shift and and potentially sample complexity typically found in BC approaches.

Following the success of GAIL, Model-based Adversarial Imitation Learning (MAIL) [12] and Adversarial Inverse Reinforcement Learning (AIRL) [9] are two successors that aim to resolve some of the issues of GAIL. MAIL improves GAIL by attempting to make the entire system differentiable and avoiding the need of high variance gradient estimation [12]. MAIL successfully extracts an analytical policy gradient from the gradient of the discriminator.

AIRL instead follows the IRL setting trying to recover a disentangled reward function from the expert via a GAN to better generalize [9] in test environments with modified dynamics. The discriminator is defined by simultaneously learning a reward function and value function, operating on single state-action pairs, to recover a robust reward function that is invariant to changing dynamics.

In our project, we propose to compare the performance and generalizability of MAIL under a modified test environment in order to evaluate whether the model is able to perform competitively in high dimensional, continuous action tasks. In addition, we propose a novel algorithm that is model-based adversarial inverse reinforcement learning (MAIRL), to investigate an approach that can combine the advantages of both MAIL and AIRL by incorporating MAIL's end-to-end differentiable model with AIRL's invariant reward learning. We show in the results section that MAIRL significantly outperforms MAIL and the expert on imitation learning tasks in an open-source version of OpenAI's MuJoCo, called Pybullet Gymperium [2]. We also show that MAIRL is the only method that is able to learn a policy that can traverse an environment where the agent's dynamics are drastically changed at test time.

2 Problem & Related Work

Branching from the problem definition, the pioneering work that utilized adversarial learning methods in imitation learning is Generative Adversarial Imitation Learning (GAIL) [8]. This work is done in a model free setup.

In the model-free case (such as GAIL), a limitation is that generative models cannot be trained by backpropagating the gradients of losses from the discriminator. Building on GAIL, MAIL [12] proposes a model-based version of adversarial imitation learning, that uses a model to create a framework that is trained end-to-end with simple backpropagation. As the authors mention, direct policy optimization can be derived from the gradient of the discriminator rather than high-variance gradient estimation in model-free approaches.

Furthermore, inverse reinforcement learning (IRL) methods have proven to work well with adversarial learning. Specifically, authors in adversarial inverse reinforcement learning [9] uses a trick called reward disentanglement to learn a reward from the discriminator. Their unique method is used and further explained in the Background and Methodology section.

Some issues in MAIL and imitation learning in general are that it it fails to generalize well to changing environments and it does not recover a reward function that may be used to train policies on new tasks [9].

Given some expert demonstration, the discriminator in the MAIL model will drive the learned policy to mimic the demonstration data until it cannot distinguish between the two. This does not allow for learning policies in environments that vary from the training data. Other works propose a discriminator model structured specifically to recover a reward function that can be used to optimize policies on various environments[9].

Further, the accuracy of the dynamics model learned in MAIL is crucial to training the optimal policy. An inaccurate model will lead to noisy gradients when optimizing the policy and slow down convergence. In environments with highly varying dynamics, our policy may explore states with different transition distributions than the demonstration data. This may cause inaccurate predictions by the dynamics model. To overcome this, the model can also benefit from using state transition observations made during policy optimization to improve its accuracy.

In the next sections we see how we can expand on two inspiring methods, MAIL and AIRL as mentioned earlier, to learn robust rewards using IRL for model-based methods that perform well not only in imitation settings, but also environments with changing dynamics.

3 Methodology

3.1 Background

Our work builds on an entropy-regularized Markov Decision Process (MDP) defined by a tuple $(S, A, \mathcal{T}, r, \gamma, \rho_0)$. In this setting, S and A are state action spaces, $\mathcal{T}(s'|a, s)$ is the system dynamics, ρ_0 is the initial state distribution, and $r(s, a)$ is the reward function that is unknown and queried by interacting with the MDP under the standard RL framework.

In standard RL, the purpose is to maximize the expected entropy-regularized discounted reward, under π, τ, and ρ_0.

$$\pi^* = argmax_{\pi} \mathbb{E}_{\tau \sim \pi}[\Sigma_{t=0}^{T} \gamma^t (r(s_t, a_t) + H(\pi(\cdot|s_t)))]$$

In the above definition, $\tau = (s_0, a_0, ...s_T, a_T)$ is a trajectory of states and actions under a particular MDP. Work [14] has shown that the trajectory distribution induced by the optimal policy takes the form $\pi^*(a|s) \propto exp\{Q^*_{soft}(s_t, a_t)\}$, where $Q^*_{soft}(s_t, a_t) = r_t(s, a) + \mathbb{E}_{(s_{t+1},...) \sim \pi}[\Sigma_{t'=t}^{T} \gamma^{t'} r(s_{t'}, a_{t'}) + H(\pi(\cdot|s_{t'}))]$ is the soft Q-function

We also build on the maximum causal entropy IRL framework which aims to learn a reward function $r(s, a)$ given demonstration data, $\mathcal{D} = \{\tau_1, ..., \tau_N\}$. An assumption made in this setting is that demonstration data is drawn from an optimal policy, $\pi^*(a|s)$. As seen in inspiring works [9], we can formulate the task or IRL as solving the maximum likelihood problem:

$$\max_{\theta} \mathbb{E}_{\tau \sim \mathcal{D}[log p_{\theta}(\tau)]} \tag{1}$$

Where $p_\theta(\tau) \propto p(s_0)\Pi_{t=0}^{T} p(s_{t+1}|s_t, a_t) e^{\gamma^t r_\theta(s_t, a_t)}$ parameterizes the reward function $r_\theta(s, a)$ for a fixed dynamics and initial state distribution.

Building on the work of GANs [6], equation 1 can be viewed as an optimization problem based on a trajectory-based or single state-action based problem. For example, for the trajectory-based format, discriminator is constrained to take the form

$$D_\theta(\tau) = \frac{exp\{f_\theta(\tau)\}}{exp\{f_\theta(\tau)\} + \pi(\tau)} \tag{2}$$

In equation 2, $f_\theta(\tau)$ is a learned function and $\pi(\tau)$ is pre-computed. During training, the policy is trained to maximize the reward $R = log(1 - D(\tau) - log(D(\tau))$. Therefore by training a discriminator using standard deep learning methods, the reward function is also being updated. It is important to note here that this work is not novel in this portion, but merely an inspiration from AIRL [9]. AIRL farther expands the constraint on equation 2 for the single state-action case

$$D_\theta(s, a) = \frac{exp\{f_\theta(s, a)\}}{exp\{f_\theta(s, a)\} + \pi(a|s)} \tag{3}$$

Interestingly, when trained at optimality it can be shown that $f^*_\theta(s,a)\} = log\pi^*(a|s) = A^*(s,a)$, which is the advantage function of the optimal policy. This is justified and proven in Appendix A of [9], also showing how it solves the IRL problem.

Unfortunately the reward learned in this setting is only satisfactory in imiation settings, but not robust enough for invariant reward learning that can be applied to testing enviornments with modified dynamics. The reason the reward is not invariant is because it is learned in a supervised fashion based on an optimal policy in a training MDP.

The inspiring work in AIRL shows that by modifying the discriminator in equation 3 to the following form can decouple the reward from the advantage function mentioned above.

$$D_{\theta,\phi}(s,a,s') = \frac{exp\{f_{\theta,\phi}(s,a,s')\}}{exp\{f_{\theta,\phi}(s,a,s')\} + \pi(a|s)} \tag{4}$$

Here $f_{\theta,\phi}$ is restricted to a reward approximation term g_θ and a shaping term as h_ϕ as the following:

$$f_{\theta,\phi}(s,a,s') = g_\theta + \gamma h_\phi(s') - h_\phi(s) \tag{5}$$

The key thing to note here is that this decoupling helps to avoid unwanted shaping and take it into account separately.

In Appendix C of AIRL, it can be shown that in deterministic environments with state-only rewards,

$$g^*(s) = r^*(s) + const$$

$$h^*(s) = V^*(s) + const$$

where $r^*(s)$ is the true reward function and $V^*(s)$ is the optimal value function. By combining this notation into 4,

$$f^*(s,a,s') = r^*(s) + \gamma V^*(s') - V^*(s) = A^*(s,a) \tag{6}$$

This work forms the foundation and justification for defining a constrained discriminator to learn disentangled rewards to be applied in unseen test environments.

3.2 Model-Based Justification

MAIL provides a solid justification in introducing a forward model to learn the environment dynamics. Adapted from their paper, MAIL authors visually demonstrate the benefit of using a forward model by showing the block diagram change noted in the difference between figures 1 and 2. The key difference is that the error from the state input is successfully propagated back when adding a forward model, otherwise it is unused and leaves the computation graph incomplete.

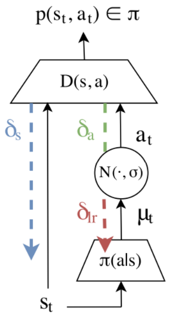

Figure 1: Model-Free adversarial imitation learning block diagram, adapted from [12]

Furthermore, the authors of MAIL mathematically justify a fully tractable discriminator learning algorithm by taking advantage of its partial derivatives. Specifically, given a discriminator that predicts

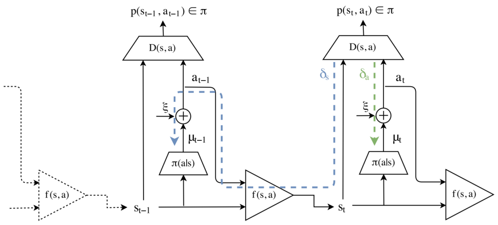

Figure 2: Model-Based adversarial imitation learning block diagram, adapted from [12]

a conditional distribution, $D(s,a) = p(y|s,a) = p(\pi|s,a) = \frac{p(s,a|\pi)p(\pi)}{p(s,a)} = \frac{p(s,a|\pi)}{p(s,a|\pi)+p(s,a|\pi_E)}$, (using uniform prior distributions for expert and learned policy). With rearranging and factoring, they work with a discriminator of the following form:

$$D(s,a) = \frac{1}{1 + \varphi(s,a) \cdot \psi(s,a)} \tag{7}$$

In equation 7 $\varphi(s,a)$ is $\frac{p(a|s,\pi_E)}{p(a|s,\pi)}$, also known as the policy likelihood ratio, which addresses how likely an action given a state is under the expert versus generated policy.

In equation 7 $\psi(s,a)$ is $\frac{p(s|\pi_E)}{p(s|\pi)}$ is the state distribution likelihood ratio which addresses if a state is more likely to be under the distribution induced by the expert policy versus learned policy.

In the model based approach, MAIL is able to use 7 to define partial derivatives with respect to both the state and action, allowing them to define a multi-step gradient algorithm that can fully account for the error in future state distributions. Specifically, the MAIL algorithm is a multi-step gradient formulation that is outlined in algorithm 1.

Algorithm 1: Model-Based Adversarial Imitation Learning

```
Given empty experience buffer $\beta$ ;
Obtain expert trajectories $\tau_i^E$ ;
for trajectory = 0 to $\infty$ do
    for $t = 0$ to $T$ do
        Act on environment: $a = \pi(s, \zeta; \theta)$;
        Push (s,a,s') into $\beta$;
    end
    Train forward model $f$ using $\beta$;
    Train discriminator model $D$ using $\beta$;
    set $j'_s = 0$ $j'_\theta = 0$ for $t = T$ down to $0$ do
        $j_\theta = [D_a \pi_\theta + \gamma(j'_{s'} f_a \pi_\theta + j'_\theta)]|_\zeta$;
        $j_s = [D_s + D_a \pi_s + \gamma j'_{s'}(f_s + f_a \pi_\theta)]|_\zeta$;
    end
    Apply gradient update using $j_\theta^0$
end
```

In the MAIL algorithm j_s and j_θ are defined as

$$j_s = \mathbb{E}_{p(a|s)} \mathbb{E}_{p(s',|s,a)} \mathbb{E}_{p(\zeta|s,a,s')} [D_s + D_a \pi_s + \gamma j'_{s'}(f_s + f_a \pi_s)]$$

$$j_\theta = \mathbb{E}_{p(a|s)} \mathbb{E}_{p(s',|s,a)} \mathbb{E}_{p(\zeta|s,a,s')} [D_a \pi_\theta + \gamma(j'_{s'} f_a \pi_\theta + j'_\theta)]$$

.

3.3 MAIRL Algorithm

This section builds on the background information mentioned earlier; at a high level, the format of a constrained discriminator can be useful for learning a reward in the adversarial, model-based setting.

It is suggested that by looking at the model-based benefits of MAIL and the robust reward learning benefits of AIRL, a hybrid algorithm can be developed, called ”MAIRL” - model based adversarial inverse reinforcement learning.

Now, instead of minimizing total discriminator beliefs along a trajectory such as in MAIL, we focus on learning a robust, decoupled discriminator definition from AIRL mentioned in the background section. Using the discriminator definition in equation 4, we define MAIRL in algorithm 2. It is key to note here that we implement the multi-step policy update step identically to MAIL, which is valid due to the computation graph established using a forward model, as seen in figure 2.

Algorithm 2: Model-Based Adversarial Inverse Reinforcement Learning

```
Given empty experience buffer $\beta$ ;
Obtain expert trajectories $\tau_i^E$ ;
for trajectory = 0 to $\infty$ do
    Collect trajectories $\tau_i = (s_0, a_0, ..., s_T, a_T)$ by executing $\pi$ with forward model;
    Add trajectory to $\beta$;
    Train forward model $f$ using $\beta$;
    Train $D_{\theta,\phi}$ via binary logistic regression to classify expert data $\tau_i^E$ from samples $\tau_i$;
    Update reward $r_{\theta,\phi}(s, a, s^{'}) \leftarrow logD_{\theta,\phi}(s, a, s^{'}) - log(1 - D_{\theta,\phi}(s, a, s^{'}))$;
    Update $\pi$ using $-r_{\theta,\phi}$ using $f$;
end
```

4 Results & Evaluation

We performed experiments in environments provided by PyBullet, an open source alternative to GYM Mujoco environments [2]. The evaluated environments are shown below in figure 3.

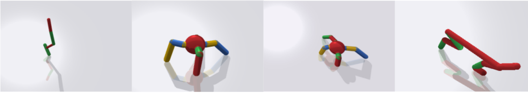

Figure 3: Pybullet-Gym environments from left to right: Hopper, Ant, Disabled-Ant (custom), Half-Cheetah

After performing tests on the environment with the same dynamics as the training environment, we applied changes to the dynamics in order to test the generalization of the model with unseen environments.

First, we compare results of training a policy using TRPO as a baseline and an expert for MAIL and MAIRL. This was done on three PyBullet environments; hopper, ant, half-cheetah. The average rewards of these tests are reported in Table 1.

Model	TRPO	MAIL	MAIRL (state-only reward)	MAIRL (state-action reward)
Hopper (10k)	382.63	1504.32	1356.02	**1696.69**
Ant (100k)	1038.94	1153.94	633.12	**2118.51**
Half-Cheetah (100k)	924.174	1326.36	400.70	**1681.86**

Table 1: Average reward over 5 episodes up to 500 steps at the end of 10k or 100k policy training iterations

Next, to test the effectiveness of MAIRL at learning a more robust reward, we test how well it does compared to MAIL in a transfer learning setting. The discriminator trained on the regular Ant environment is used without updates to train a new policy and forward model in the Disabled Ant environment. Two of the Ant's legs are shortened in the new environment which will undoubtedly change the environment dynamics. MAIRL using state-action rewards is the only model able to learn to walk in the new environment. The results of those best models in the experiment are reported in Table 2

Model	TRPO	MAIL	MAIRL (state-only reward)	MAIRL (state-action reward)
Ant (100k)	1038.94	1153.94	633.12	**2118.51**
Disabled Ant (10k)	954.23	878.95	393.77	**1131.13**

Table 2: Transfer learning results for MAIL and MAIRL compared to baseline TRPO

Using MAIRL to learn a state-action reward function we are able to outperform TRPO and MAIL in all environments and beat the expert policy. Interestingly, the state-only version of MAIRL did not perform as well. In AIRL, the state-only version performed better in transfer learning settings since it learns a reward that is disentangled from the unknown environment dynamics and depends only on the state. With the introduction of a forward model in the training loop, the results suggest the state-action reward is more effective in environments where the dynamics can be modelled as well.

However, the learned policies for transfer learning for all models declined in performance over time. The average rewards were very low by the end of training, which suggests there is still some instability in the transfer learning setting. One possible reason may be that the forward model was not sufficiently trained before learning a policy in the new environment.

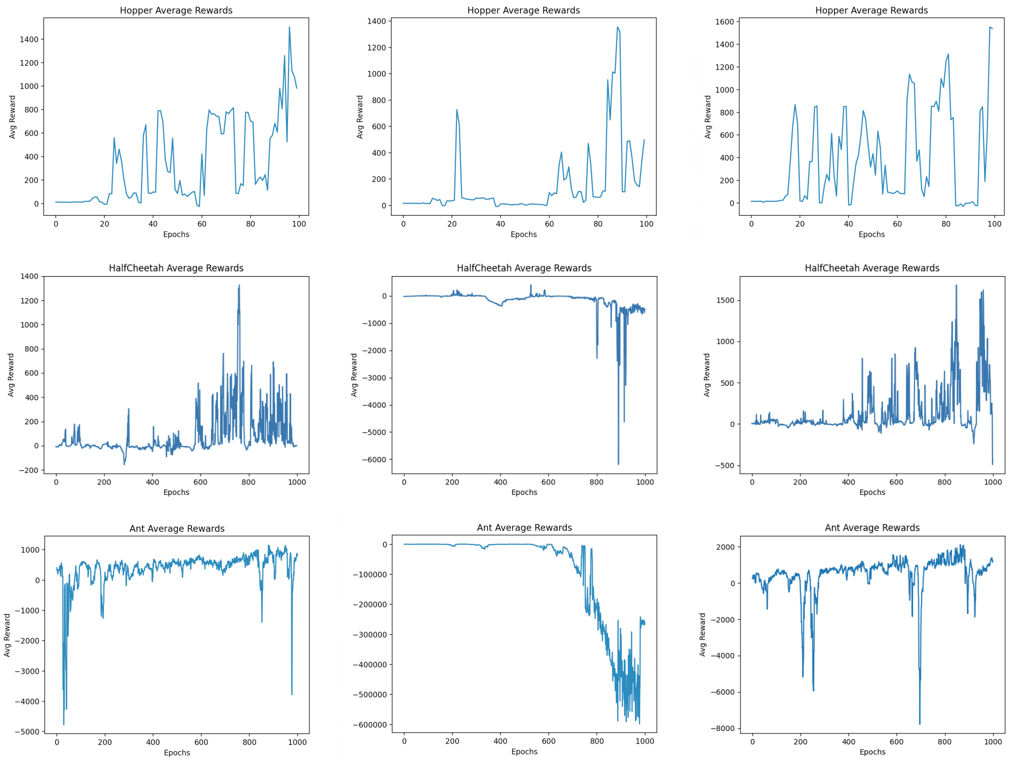

Figure 4: Rewards against epochs graphs. Row: Hopper, Half Cheeta and Ant. Column: MAIL, MAIRL(s) and MAIRL(sa).

5 Limitations & Future Work

5.1 Limitations

Although we were able to train MAIRL agents that can outperform other baseline models, there were still some limitations we observed along the way.

Different reward scale. Since we did not have access to OpenAI GYM Mujoco which is used widely in different prior RL works, we used the open source variant PyBullet to benchmark our models. However, it seems their scales of reward are not exactly the same as observed from the different values of expert and MAIL agents in Table 1 to those mentioned in MAIL's paper. Therefore, the benchmarks we obtained in the project cannot be directly compared to those in MAIL and AIRL. If we would like to make a rigorous statement that our MAIRL is outperforming MAIL and AIRL, it would be more accurate to use MuJoCo in the future.

Expertise level of demonstration data. During the training process of the expert agent, we only tried several hyperparameter settings and trained the policies until convergence instead of a thorough grid search of hyperparameters. We then picked the best performing agent quantitatively which showed reasonable behaviour as our expert, and generated demonstration data with it. Since we used PyBullet, as mentioned in the previous paragraph that the reward was not directly comparable to previous works, we did not know whether our experts were as good as those in previous papers [9][12]. Also, we did not know to what extent the expertise level would affect the performance of the imitating agent.

Stochastic and high dimensional next state. At the beginning of the project, we tried to train MAIL in a seemingly simpler environment - Four Room Maze in which the agent needs to search for the goal with partial observation. However, we were not able to train an agent with learned behaviours after a lot of trials. We later figured out that this is likely because the next state inferred from a single partial observation is uncertain, which means given the current partial observation there can be multiple possible next states. In addition, the partially observed state is a relatively high-dimensional 2D matrix (7x7x3) which is much larger than that of Hopper (state size of 15) used in MAIL. As a result, the forward model cannot correctly predict the next state based on the partial observation and the gradient through it is not accurate enough to optimize the policy, thereby failing to train an agent either in MAIL or MAIRL.

5.2 Future work

More transfer learning. Since we just tried transfer learning scenario on the ant environment, we can possibly do more transfer learning testing using more pairs of original vs modified environments to further verify the effectiveness of the disentangled reward function.

Verification of disentangled reward function. In addition to transfer learning testing, in the AIRL paper, randomly generated MDPs are used to verify whether AIRL can recover the ground truth reward function so as to claim that AIRL is able to learn a disentangled reward function which can more efficiently adapt to changed dynamics [9]. Unfortunately, we did not have the sufficient time to carry out such experiments. This can be a future work to further verify whether MAIRL can also recover a disentangled reward function completely.

GAIL and AIRL. As both papers that motivated our project are based on GAIL, it may be worth trying to test GAIL in those environments as well especially in the transfer learning setting, so as to understand the contribution of the forward model in the transfer learning robustness. We may as well benchmark on AIRL using the same environments so that we can compare MAIRL with AIRL directly.

State history in forward model. As mentioned in the limitations, the forward model is inaccurate when there can be multiple possible next states given the current state. One possible solution is to include the history of states in the forward model, so that there is less ambiguity about the next state. It can be achieved by either using a recurrent neural network or input a stack of frame of states into the neural network.

Hindsight Experience Replay. In the proposal stage of this project, we planned to use the relabelling technique of HER to see whether it can possibly enhance the model performance as a stretch goal [10]. We did not have sufficient time to implement this, but it can be a potential future work.

This will require converting the problem into a goal-conditioned one. Similar approach has been investigated before [16]. Combining it with the forward model and disentangled reward function is a promising direction.

6 Conclusions

We proposed MAIRL, a novel model-based approach for inverse reinforcement learning in an adversarial setting, that combines a differentiable forward model and disentangled rewards learning. We showed that not only can MAIRL outperform other adversarial IRL methods in a typical learning setting but also in the transfer learning setting empirically. More surprisingly, it even beats its expert in all of our experiments. This implies that the more accurate gradient estimation end-to-end from the discriminator together with the disentangled reward function give rise to a robust policy that not just performs well in the training environment but is also able to adapt to changing dynamics easily.

References

[1] A. Ng, S. J Russell, et al. Algorithms for inverse reinforcement learning. In Icml, pages 663–670, 2000.

[2] B. Ellenberger. Pybullet gymperium. https://github.com/benelot/pybullet-gym, 2018.

[3] B. Ziebart. Modeling purposeful adaptive behavior with the principle of maximum causal entropy. PhD thesis, Carnegie Mellon University, 2010.

[4] D. Pomerleau. Efficient training of artificial neural networks for autonomous navigation. Neural Computation, 3(1):88–97, 1991

[5] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba. OpenAI Gym. arXiv:1606.01540. 2016.

[6] I. Goodfellow, J.Pouget-Abadie, M.Mirza, B.Xu, D.Warde-Farley, S.Ozair, A.Courville, and Y.Bengio. Generative adversarial nets. In NIPS, pages 2672–2680, 2014.

[7] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. Trust Region Policy Optimization. In International Conference on Machine Learning (ICML), 2015.

[8] J. Ho and S. Ermon. Generative adversarial imitation learning. In NIPS, pp. 4565–4573, 2016.

[9] J. Fu, K. Luo, S. Levine. Learning Robust Rewards with adversarial Inverse Reinforcement Learning.

[10] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. Hindsight Experience Replay. Advances in Neural Information Processing Systems, 2017.

[11] M. Dorigo and M. Colombetti. Robot shaping: an experiment in behavior engineering. MIT press, 1998.

[12] N. Baram, O. Anschel, S. Mannor. Model-based Adversarial Imitation Learning.

[13] S. Ross and Drew Bagnell. Efficient reductions for imitation learning. In ´ AISTATS, pages 661–668, 2010.

[14] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine. Reinforcement learning with deep energy-based policies. In International Conference on Machine Learning (ICML), 2017.

[15] T. Schaul, D. Horgan, K. Gregor, and D. Silver. Universal Value Function Approximators. Internation Conference in Machine Learning, 2015.

[16] Y. Ding, C. Florensa, M. Phielipp, P. Abbeel. Goal-Conditioned Imitation Learning. In 33rd Conference on Neural Information Processing Systems (NeurIPS), 2019

Appendix A Contributions

Lalit's main contributions involved the following: prototyping and integrating RL libraries for training TRPO experts and generating expert trajectories, scoping and integrating valid Gym-based environments to validate MAIRL, developing and integrating a replica of a disabled Ant environment as mentioned in AIRL, and helping build the MAIRL algorithm from the base of MAIL algorithm for later validation tasks.

Yip Sang (Anson)'s main contributions involved the following: training TRPO experts with RL library in several environments (four room maze, half-cheetah etc.), running MAIL and MAIRL experiments in several environments and completing the implementation of MAIRL algorithm.

Saad contributed to configuring MAIL to use new environments for training, researched combining AIRL into MAIL by taking advantage of their similar structures, as well as helping generate model performance graphs, analyzing model performance for MAIL and MAIRL over various environments.

Appendix B Code

All code used to produce our results can be found here `https://github.com/yipsang/mgail`