Auto-Encoding Adversarial Imitation Learning

Extended Abstract

Kaifeng Zhang
Shanghai Qi Zhi Institute
Shanghai, China
zhangkf@sqz.ac.cn

Rui Zhao
Tencent AI Lab
Shenzhen, China
rui.zhao.ml@gmail.com

Ziming Zhang
Worcester Polytechnic Institute
Worcester, United States
zzhang15@wpi.edu

Yang Gao
Tsinghua University, Shanghai Qi Zhi Institute
Shanghai Artificial Intelligence Laboratory
Beijing, China
gaoyangiiis@mail.tsinghua.edu.cn

ABSTRACT

Reinforcement learning (RL) provides a powerful framework for decision-making, but its application in practice often requires a carefully designed reward function. Adversarial Imitation Learning (AIL) sheds light on automatic policy acquisition without access to the reward signal from the environment. In this work, we propose Auto-Encoding Adversarial Imitation Learning (AEAIL), a robust and scalable AIL framework. To induce expert policies from demonstrations, AEAIL utilizes the reconstruction error of an auto-encoder as a reward signal, which provides more information for optimizing policies than the prior discriminator-based ones. Subsequently, we use the derived objective functions to train the auto-encoder and the agent policy. Experiments show that our AEAIL performs superior compared to state-of-the-art methods on both state and image based environments. More importantly, AEAIL shows much better robustness when the expert demonstrations are noisy.

KEYWORDS

Reinforcement Learning, Adversarial Imitation Learning, Auto-Encoding

ACM Reference Format:
Kaifeng Zhang, Rui Zhao, Ziming Zhang, and Yang Gao. 2024. Auto-Encoding Adversarial Imitation Learning: Extended Abstract. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION

Reinforcement learning (RL) provides a powerful framework for automated decision-making. However, RL still requires significantly engineered reward functions for good practical performance. Imitation learning offers the instruments to learn policies directly from the demonstrations, without an explicit reward function. It enables the agents to learn to solve tasks from expert demonstrations, such as helicopter control [1–3, 5, 7, 10], robot navigation [4, 11, 13, 14], and building controls [6].

The goal of imitation learning is to induce the expert policies from expert demonstrations without access to the reward signal from the environment. We divide these methods into three broad categories: Behavioral Cloning (BC), Inverse Reinforcement Learning (IRL), and Adversarial Imitation Learning (AIL). AIL induces expert policies by minimizing the distribution distance between expert samples and agent policy rollouts. Prior AIL methods model the reward function as a discriminator to learn the mapping from the state-action pair to a scalar value, i.e., reward [8, 9, 12]. However, the discriminator in the AIL framework would easily find the differences between expert samples and agent-generated ones, even though some differences are minor. Therefore, the discriminator-based reward function would yield a sparse reward signal to the agent. Consequently, how to make AIL robust and efficient to use is still subject to research.

Our AEAIL is an instance of AIL by formulating the reward function as an auto-encoder. Since auto-encoder reconstruct the full state-action pairs, unlike traditional discriminator based AIL, our method will not overfit to the minor differences between expert samples and generated samples. In many cases, our reward signal provides richer feedback to the policy training process. Thus, our new method achieves better performance on a wide range of tasks.

2 METHOD

Our approach is to minimize the distance between the state action distribution of the policy π_θ and that of the expert demonstrations.

The objective formulation we used in our method is Wasserstein divergence:

$$d(\pi_E, \pi_\theta) = \sup_{|r_w|_L \leq K} E_{\pi_E}[r_w(s,a)] - E_{\pi_\theta}[r_w(s,a)], \qquad (1)$$

where the reward function network's parameters are denoted as w and the policy network's parameters are represented as θ. Minimizing this distance will induce the expert policy from expert demonstrations. Therefore, the optimization of the policy π_θ and the reward function $r_w(s,a)$ forms a bi-level optimization problem, which can be formally defined as:

$$\min_{\pi_\theta} \max_{r_w} \left(E_{(s,a)\sim D_E}[r_w(s,a)] - E_{(s,a)\sim \pi_\theta}[r_w(s,a)] \right). \qquad (2)$$

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

Here, we clip the weights of the reward function to keep it K-Lipschitz. Optimizing Equation 2 leads to an adversarial formulation for imitation learning. The outer level minimization with respect to the policy leads to a learned policy that is close to the expert. The inner level maximization recovers a reward function that attributes higher values to regions close to the expert data and penalizes all other areas.

In our method, we use an auto-encoder based surrogate pseudo-reward function instead, which is defined as:

$$r_w(s,a) = 1/(1 + \mathrm{AE}_w(s,a)), \tag{3}$$

where AE is the reconstruction error of an auto-encoder:

$$\mathrm{AE}(x) = \|\mathrm{Dec} \circ \mathrm{Enc}(x) - x\|_2^2 \tag{4}$$

Here, x represents the state-action pairs. Equation 4 is the mean square error between the sampled state-action pairs and the reconstructed samples. This form of the reward signal uses the reconstruction error of an auto-encoder to score the state-action pairs in the trajectories. Equation 3 is a monotonically decreasing function over the reconstruction error of the auto-encoder. We give high rewards to the state-action pairs with low reconstruction errors of the auto-encoding process and vice versa. Section "Overview" **??** motivates that this form of reward signal focuses more on the full-scale differences between the expert and generated samples, and it won't easily overfit to the noise of the expert data.

Training the auto-encoder is an adversarial process considering the objective 2, which is minimizing the reconstruction error for the expert samples while maximizing this error for generated samples. When combining Equation 2 and Equation 3, we can obtain the training objective for the auto-encoder as:

$$\begin{aligned}\mathcal{L} =& E_{(s,a)\sim\pi_\theta}[r_w(s,a)] - E_{(s,a)\sim D_E}[r_w(s,a)] && (5)\\ =& E_{(s,a)\sim\pi_\theta}[1/(1 + \mathrm{AE}_w(s,a))] && (6)\\ & - E_{(s,a)\sim D_E}[1/(1 + \mathrm{AE}_w(s,a))]. && (7)\end{aligned}$$

With this adversarial objective, the auto-encoder learns to maximize the full-scale differences between the expert and the generated samples. As a result, it gives the agent a denser reward signal. Furthermore, the agent can also be more robust when facing noisy expert data due to the robust auto-encoding objective.

3 EXPERIMENTS

Question 1. *Does our AEAIL achieve best performance compared to these four ablation methods?*

Our method's overall scaled reward is about 0.921, whereas the best ablation method is 0.83 for JSD. There is an about 11% relative improvement. Our method outperforms other ablations on all locomotion tasks except for Hopper and HalfCheetah. Here we would like to point out that our AEAIL has already achieved 97.8% of the expert performance on Hopper while 91.7% on HalfCheetah, which is very close to completely solving the tasks.

Question 2. *Is our AEAIL robust to the noisy expert data?*

To show the robustness of our proposed AEAIL, we further conduct experiments on noisy expert data. We add a Gaussian noise distribution $(0, 0.3)$ to the expert data for Walker2d, Hopper, Swimmer and HalfCheetah. Since Ant and Humanoid are much more sensitive to noise, we add $(0, 0.03)$ Gaussian noise to these two tasks.

Table 1: Relative improvements for different variants of our AEAIL compared to the best baseline JSD and GOT on clean and noisy data, respectively.

Improvements	Ours	Ours-JS	Ours-VAE
Clean Data	11.0%	10.5%	7.6%
Noisy Data	50.7%	44.9%	42.1%

The results show that our method outperforms other ablations on all tasks except for Swimmer, on which F-KLD wins. Our AEAIL offers an excellent capability in learning from noisy expert data on these tasks. The overall scaled rewards for our AEAIL is 0.813, whereas the best ablation is 0.539 for GOT on the noisy expert setting. There is an about 50.7% relative improvement. Other discriminator based ablations, are very sensitive to the noisy expert.

Question 3. *What is the major contributing factor of our AEAIL? Could it be the specific W-distance metric?*

To analyze the major contributing factor of our AEAIL, we conduct an ablation study that replaces the distance to other distribution divergences. Comparable performances indicate that the major contributing factor is the encoding-decoding process rather than the specific distribution divergence.

Table 1 shows that our JS-based variant achieves 10.5% and 44.9% relative improvement compared to the best baseline JSD method and GOT method on clean and noisy expert data, respectively. Similar to the original AEAIL, our JS-based variant also greatly improves the imitation performance on these benchmarks. The relative improvements are comparable between the two distance metrics. It indicates that the major contributing factor of our AEAIL is the encoding-decoding process. This also shows that AEAIL works not limited to a specific distance metric.

Question 4. *Is AEAIL limited to the specific type of auto-encoders? How about utilizing variational auto-encoders?* Table 1 shows our VAE-based variant gets 7.6% and 42.1% relative improvement compared to the best baseline JSD and GOT on clean and noisy expert data, respectively. This means that our VAE-based variant improves the imitation performance considerably compared to other baselines. It justifies that our AEAIL is flexible with different auto-encoders.

4 CONCLUSIONS

This paper presents a straightforward and robust adversarial imitation learning method based on auto-encoding (AEAIL). We utilize the reconstruction error of an auto-encoder as the surrogate pseudo-reward function for reinforcement learning. The advantage is that the auto-encoder based reward function focused on the full-scale differences between the expert and generated samples, which provides a denser reward signal to the agent. As a result, it enables the agents to learn better policies. Experimental results show that our methods achieve strong competitive performances on both clean and noisy expert data. In the future, we want to further investigate our approach in more realistic scenarios, such as autonomous driving and robotics.

REFERENCES

[1] Pieter Abbeel, Adam Coates, Timothy Hunter, and Andrew Y. Ng. 2008. Autonomous Autorotation of an RC Helicopter. In *International Symposium on Robotics*.

[2] Pieter Abbeel, Adam Coates, and Andrew Y. Ng. 2010. Autonomous Helicopter Aerobatics through Apprenticeship Learning. *The International Journal of Robotics Research* (2010).

[3] Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y. Ng. 2007. An Application of Reinforcement Learning to Aerobatic Helicopter Flight. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[4] Pieter Abbeel, Dmitri Dolov, Andrew Y. Ng, and Sebastian Thrun. 2008. Apprenticeship learning for motion planning with application to parking lot navigation. In *IEEE/RSj International Conference on Intelligent Robots and Systems*.

[5] Pieter Abbeel, Varun Ganapathi, and Andrew Y. Ng. 2006. A Learning vehicular dynamics, with application to modeling helicopters. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[6] Enda Barrett and Stephen Linder. 2015. Autonomous hva control, a reinforcement learning approach. *Machine Learning and Knowledge Discovery in Databases* (2015).

[7] Adam Coates, Pieter Abbeel, and Andrew Y. Ng. 2008. Learning for Control from Multiple Demonstrations. In *International Conference on Machine Learning (ICML)*.

[8] Justin Fu, Katie Luo, and Sergey Levine. 2017. Learning robust rewards with adversarial inverse reinforcement learning. *International Conference on Learning Representations (ICLR)* (2017).

[9] Jonathan Ho and Stefano Ermon. 2016. Generative Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[10] Andrew Y. Ng, Adam Coates, Mark Diel, Varun Ganapathi, Jamie Schulte, Ben Tse, Eric Berger, and Eric Liang. 2004. Inverted autonomous helicopter flight via reinforcement learning. In *International Symposium on Experimental Robotics*.

[11] Nathan D. Ratliff, J. Andrew Bagnell, and Martin A. Zinkevich. 2006. Maximum Margin Planning. In *International Conference on Machine Learning (ICML)*.

[12] Xin Zhang, Yanhua Li, Ziming Zhang, and Zhi-Li Zhang. 2020. f-GAIL: Learning f-Divergence for Generative Adversarial Imitation Learning. *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020).

[13] Brian D. Ziebart, J. Andrew Bagnell, and Anind K. Dey. 2010. Modeling Interaction via the Principle of Maximum Causal Entropy. In *International Conference on Machine Learning (ICML)*.

[14] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. 2008. Maximum Entropy Inverse Reinforcement Learning. In *AAAI Conference on Artificial Intelligence (AAAI)*.