Imitating Agents in A Complex Environment by Generative Adversarial Imitation Learning

1st Wanxiang Li
School of Information Science
JAIST
Nomi, Ishikawa, Japan
wanxaing.li@jaist.ac.jp

2nd Chu-Hsuan Hsueh
School of Information Science
JAIST
Nomi, Ishikawa, Japan
hsuehch@jaist.ac.jp

3rd Kokolo Ikeda
School of Information Science
JAIST
Nomi, Ishikawa, Japan
kokolo@jaist.ac.jp

***Abstract*—The generative adversarial imitation learning (GAIL) shows the ability to find reward functions to explain expert players' behaviors in some low-dimensional environments using hand-crafted features as inputs. In this research, we aim to extend GAIL to complex environments and using raw images as inputs. We propose to (1) use convolutional neural networks to deal with image inputs, (2) adopt a structure called global-local discriminator to GAIL, and (3) represent trajectories as state-state pairs instead of state-action pairs. Our approach successfully imitates given players in Super Mario Bros. To our knowledge, the results are the first to have successful imitations in complex environments based on image inputs.**

***Index Terms*—Reinforcement Learning, Generative Adversarial Imitation Learning, Super Mario Bros, Global-Local Discriminator**

I. Introduction

In recent years, deep learning (DL) methods have been proven to be powerful tools in various fields such as natural language processing, robotics, and pattern recognition [9]. A branch of machine learning is reinforcement learning (RL), which aims to find the best policy for the given tasks by maximizing *rewards* [12]. By combining with DL, RL reached superhuman levels in some video games with using only images as input [10]. RL methods need well-defined reward functions, which tell the agents how well they are doing. However, in most real-world situations, the reward functions are usually hard to define.

To solve this problem, inverse reinforcement learning (IRL) [2] is introduced to help RL learning experts' policy and getting reward functions to explain the experts' behaviors from the given experts' trajectories. For most classical IRL methods, a large number of expert trajectories should be provided, but in many cases, the trajectories are not easy to get.

On the other hand, Goodfellow et al. [4] proposed a framework called generative adversarial networks (GAN), which aims to generate data similar to given ones. Briefly speaking, GAN consists of two parts, the generator and the discriminator. Taking image generation as an example, the generator is usually inputted by noises [4], images [6], or texts [14], and outputs images. The discriminator is inputted by images and outputs the probabilities that the images are from the generator.

978-1-7281-4533-4/20/$31.00 ©2020 IEEE

By combining the ideas of IRL and GAN, generative adversarial imitation learning (GAIL) [5] is proposed to learn from a small number of expert trajectories. More specifically, the goal of GAIL is to train generators, also called actors, that have similar behaviors to the given experts. Meanwhile, the discriminators can serve as the reward functions for RL, which judge whether the behaviors look like the experts. Currently, most GAIL research and its variants are applied to some relatively simple environments [3], [5], [13], where fully-connected neural networks with hand-crafted features already worked well. Torabi et al. [13] tried to use raw visual data as inputs, but their method was still hard to reach the levels of the given experts.

In this research, we target on successfully imitating specific players (1) in relatively complex games (2) without hand-crafted features. We employ the famous real-time action game Super Mario Bros as our environment. The state observation from the environment is the screen captures (i.e., images) of the game, and the actions are represented by a one-hot vector indicating the operations on the controller. Some problems need to be solved when images are used as input.

In our approach, we combine three existing methods to solve the problems. First, under GAIL's mechanism, the big difference in the dimension between states and actions makes the neural networks hard to learn. To solve this problem, we apply state-state pairs [13] instead of the original state-action pairs. Second, a new structure called global-local discriminator [7], [8] is adopted. In the original GAN, a single discriminator is used, which makes the generator not good at fine-tuning details in the generated images. Global-local discriminator was proposed to solve this problem in generating images, and to our knowledge, this is the first time to combine into GAIL. Third, since the inputs are images, we applied convolutional neural networks (CNN) [9], which have obtained impressive results in image processing. In our experiments, the proposed method successfully imitates given players in Super Mario Bros. To our knowledge, this paper is the first to have successful imitations of given players in complex environments based on image inputs.

II. Background

A. Generative Adversarial Networks (GAN)

GAN is a framework proposed by Goodfellow et al. [4] aiming to generate new data that fit the distribution of a given dataset. The framework consists of two neural networks, which are the generator and the discriminator, respectively. In their work, GAN was applied to generate images from some famous datasets such as MNIST. The generator took noises as inputs and output images. The discriminator took images as inputs and output a probability indicating whether the image was from the given dataset. The training process can be imagined as a two-player game between the generator and the discriminator. The generator tries to generate images similar to the given dataset to confuse the discriminator, while the goal of the discriminator is to perfectly distinguish images from the dataset and the generator. They successfully generated images that looked like given datasets.

B. Generative Adversarial Imitation Learning (GAIL)

Harnessing the ideas of GAN, Ho and Ermon [5] proposed GAIL, which aimed to learn the behaviors of given experts from a small number of trajectories in a given environment. The structure, as shown in Fig. 1, is similar to GAN in that it also contains a generator and a discriminator. Datasets in GAN are analogous to expert trajectories in GAIL. The generator, also called the actor, took states from the environment as inputs and output actions to generate trajectories. The discriminator worked similarly to that in GAN, which took trajectories as inputs and tired to distinguish between the generator and the experts. The discriminator can be regarded as a reward function telling the actors how similar they looked like experts. In their experiments, GAIL successfully achieved expert performance in several control tasks.

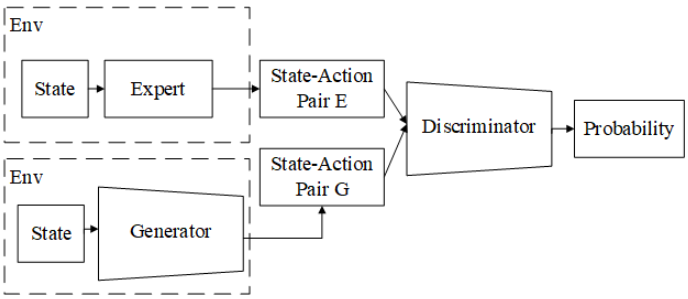

Fig. 1. Structure of GAIL.

In most GAIL research, the environments were relatively simple, where the states were usually described by some hand-crafted features. The inputs to the discriminators were trajectories, more specifically, state-action pairs. However, when more complex state representations such as raw images are employed, the difference of dimension between states and actions becomes extremely large, which causes the discriminators to hardly extracting information about actions. To solve this problem, Torabi et al. [13] proposed generative adversarial imitation from observation (GAIfo). In GAIfo, instead of actions, the next states, i.e., the states after actions, were inputted along with the states before actions. In other words, they input state-state pairs instead of state-action pairs. With image inputs, their generator performed better than several methods that also learned from expert demonstrations, though it still did not reach the expert levels.

C. Global-Local Discriminator

Global-local discriminator [7] is a variant of GAN, which employs multiple discriminators instead of one. The structure was successfully applied to enhance low-light images, which means to make low-light images into normal-light ones. The original GAN could obtain reasonable results in general looking but failed to fine-tune some details. For example, a small bright region in an overall dark background should be enhanced differently from other parts. To solve this problem, they added another discriminator, called local discriminator, apart from the original discriminator, called global discriminator. The inputs for the local discriminator were randomly cropped patches from both the generated images and those in the given dataset. With this additional discriminator, the generator needed to focus on not only the overall looking but also the fine-tuning of details in the images.

III. Approaches

Our goal is to create agents that can successfully imitate given experts with raw images as inputs. Our approach is based on GAIL [5] but with three differences. First, to deal with image inputs, CNNs are employed instead of fully-connected neural networks. Second, as GAIfo [13], we provide state-state pairs for discriminators as inputs instead of state-action pairs. Third, we introduce the idea of global-local discriminator, where the structure is shown in Fig. 2.

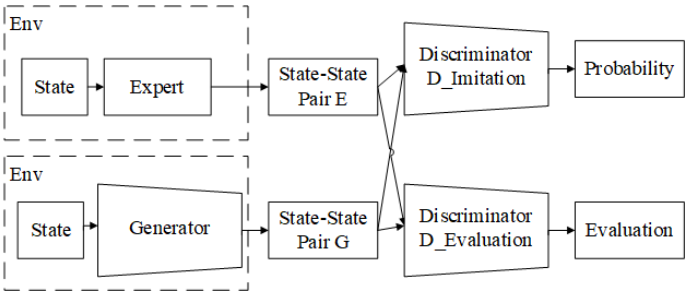

Fig. 2. Structure of Proposed System.

The details of combining global-local discriminator into GAIL are presented as follows. The global discriminator, D_Imitation, aims to distinguish the behaviors between the generator and the expert, as the original GAIL. As for the local discriminator, patches of images [7] do not work in this case. The reason is that for many environments such as video games, an action usually does not greatly change the screen. Random patches are highly likely to fail to locate the critical parts. Besides, we observed that with D_Imitation only, the generator focused on imitating the experts and ignored the real goals behind. For example, in Super Mario Bros, the goals of

the experts are usually to clear the stages while obtaining high scores. If a less experienced player learns to play the game only by imitating a limited number of expert demonstrations, the player cannot learn how to deal with different situations to clear stages.

To solve the two problems, our local discriminator, D_Evaluation, aims to figure out whether the generator's behaviors can accomplish the real goal of the given environment, e.g., clearing stages in Super Mario Bros. Although using additional evaluations seems to against the goal of GAIL or IRL, our approach is still general in the sense that we do not need complex reward functions. For example, in Super Mario Bros, to evaluate how players play well, usually, indicators such as defeated enemies, collected coins, and playing time are considered. Different players may have different playstyles, where some try to collect as many coins as possible while some try to clear stages as soon as possible, all for the final goal to clear stages. We expect general behaviors for clearing stages to be learned from D_Evaluation and behaviors for imitating playstyles to be learned from D_Imitation.

IV. Experiment and Evaluation

A. Environment Setting

We chose the world-famous game Super Mario Bros as our environment. Besides its popularity among human players, it is also a challenging test-bench for AI research. Our experiments were based on an open-sourced library called Gym Super Mario Bros [1]. Our goal was to create agents that behaved similarly to given experts while also playing the game well. Each state, or frame, was represented by an $84 * 84$ image. We employed the frameskip technique, which means the agent took the same actions in continuous n frames. Hence, more precisely, a state consisted of a current frame and the past $n - 1$ frames, which was an $84 * 84 * n$ matrix.

All agents were tested on the stage of Super Mario Bros world 4-1. In Gym Super Mario Bros, three action modes are available, which are "Complex," "Simple," and "Right-only." The difference between these modes lies in the valid actions, where "Complex" contains the most combinations and "Right-only" the least. In the preliminary study, we chose the middle one "Simple," where all valid actions are "noop," "right," "right + jump," "right + dash," "right + jump + dash," "jump," and "left."

B. Experiment Setting

For expert agents, we employed proximal policy optimization (PPO) [11], an algorithm succeeded in many RL problems. We also considered human players, but it was hard to restrict them from playing in the "Simple" mode. We leave the application to the "Complex" mode and imitation of human players as future work.

All PPO agents and the GAIL agents were based on CNNs. The network structures are shown in Fig. 3. For PPO agents and the generators of GAIL, the inputs were $84 * 84 * 4$ matrices since we applied the frameskip technique with $n = 4$. The outputs represented the probabilities of taking the seven actions in the "Simple" mode. For discriminators of GAIL, since state-state pairs were used, the inputs were $84 * 84 * 8$ matrices. The outputs for both D_Imitation and D_Evaluation were probabilities, for behaving like the given experts and being able to clear stages, respectively. The structure inside the networks was the same, which consisted of three convolutional layers followed by a fully-connected layer with 512 nodes and tanh as the activation function. The convolutional layers, using ReLU as the activation functions, had kernel sizes of 32*8*8, 64*4*4, and 32*3*3 and stride sizes of 4, 2, and 1.

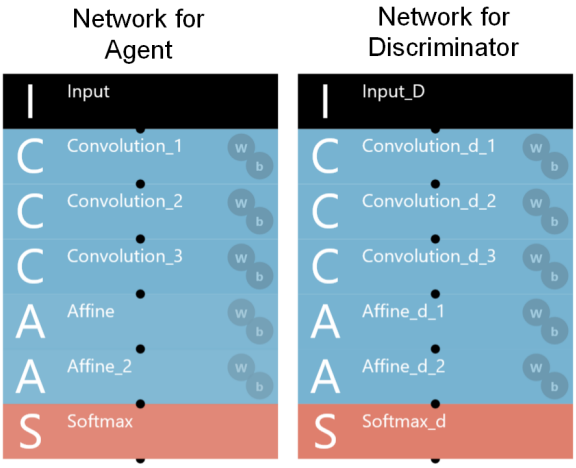

Fig. 3. Structure of Neural Network.

During training, both the PPO agents and the generators of GAIL applied ϵ-greedy to select actions. With a probability of ϵ, a random action was selected; otherwise, the agents selected the actions with the highest probabilities. The ϵ in our experiment was 0.05 for training and testing.

To evaluate the similarity between agents, we used similarity and cosine similarity. The similarity was the proportion that two agents performed the same actions in given states, i.e, how much the two agents act the same. However, even when two players look similar, not the same actions are taken for all states. Thus, we employed another evaluation for playstyles. We collected each agent's action frequency, which was represented by a $1 * 7$ vector. Each element in the vector stands for the frequency of each action. We calculated the cosine similarity between two action frequency vectors A and B by $\sum_{i=1}^{n} A_i B_i / \sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}$. The higher the cosine similarity is, the more the playstyles are considered close.

For comparing two agents, our experiments were conducted in the following manner. We let the two agents play 1000 games, respectively, so that we would collect 2000 games. Then, we asked the two agents to select actions in the states of the games played by the other agent. As a result, all states in the 2000 games were paired by the actions from the two agents so that we could calculate the similarity and the cosine similarity accordingly.

For comparison, we trained two independent PPO agents as the experts, notated by PPO_1 and PPO_2, respectively.

The two agents were trained by the reward functions provided by Gym Super Mario Bros, which is more complex than only clear or not. We then trained two GAIL agents to imitate the two experts from 100 games of demonstrations. The two imitating agents are notated by GAIL_PPO_1 and GAIL_PPO_2, respectively.

C. Result and Evaluation

The similarity between each pair of the four agents is listed in Table. I. From the results, the similarity between GAIL_PPO_1 and PPO_1 and that between GAIL_PPO_2 and PPO_2 were apparently higher than other pairs. However, as mentioned earlier, even for behaviors look similar in general, the players may not take the same actions given the same states. Thus, although the similarity values were relatively low, we still concluded that the GAIL agents learned to imitate the given experts to some degree.

TABLE I
SIMILARITY BETWEEN AGENTS

Compared agents	Similarity
PPO_1 to PPO_2	0.2452
PPO_1 to GAIL_PPO_1	0.3385
PPO_1 to GAIL_PPO_2	0.2194
PPO_2 to GAIL_PPO_1	0.2673
PPO_2 to GAIL_PPO_2	0.3592
GAIL_PPO_1 to GAIL_PPO_2	0.2064

To confirm whether playstyles look similar, we calculated the cosine similarity of action frequency. The results for each pair of the four agents are listed in Table. II. The cosine similarity between PPO_1 and GAIL_PPO_1 and that between PPO_2 and GAIL_PPO_2 were the highest two. The results concluded that our GAIL approach successfully learned the playstyles of the given experts. For example, if a player prefers to perform the jump action even if not needed, we expect our approach to learn such a playstyle from few demonstrations.

TABLE II
COSINE SIMILARITIES OF ACTION FREQUENCY

Compared agents	Cosine Similarity
PPO_1 to PPO_2	0.4656
PPO_1 to GAIL_PPO_1	0.7236
PPO_1 to GAIL_PPO_2	0.4032
PPO_2 to GAIL_PPO_1	0.5281
PPO_2 to GAIL_PPO_2	0.9427
GAIL_PPO_1 to GAIL_PPO_2	0.6275

V. CONCLUSION AND FUTURE WORK

In this research, we adopt GAIL to a complex environment Super Mario Bros. We use CNNs to deal with the image inputs. Also, we present the trajectories as state-state pairs instead of state-action pairs. Moreover, the global-local discriminator is first introduced into the GAIL system. Combining these methods, the similarity between agents from the modified GAIL and their experts is apparently higher than other unrelated agents. Besides, we analyze the action frequency of each agent for accessing playstyles. The cosine similarity of the action frequency between the modified GAIL and their experts is higher than others. From the results, we conclude that our modified GAIL successfully imitates given experts in a complex environment.

There are still some points to improve. First, the trajectory of the proposed GAIL consists of the current state and the past one state. We expect longer sequences to have better performance on imitation since more information is provided. Second, the proposed GAIL only imitates RL agents. We will try to imitate human players under the action mode of "Complex" instead of "Simple." Third, although we adopt the global-local discriminator, only two discriminators work in the system. It is interesting to include more discriminators. Finally, except for Super Mario Bros, we will try to apply our approach to imitate agents or human players in other games.

REFERENCES

[1] Gym super mario bros. https://github.com/Kautenja/gym-super-mario-bros. Accessed: 2020-05-20.

[2] Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *arXiv preprint arXiv:1806.06877*, 2018.

[3] Nir Baram, Oron Anschel, Itai Caspi, and Shie Mannor. End-to-end differentiable adversarial imitation learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 390–399. JMLR. org, 2017.

[4] 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*, pages 2672–2680, 2014.

[5] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573, 2016.

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[7] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. Enlightengan: Deep light enhancement without paired supervision. *arXiv preprint arXiv:1906.06972*, 2019.

[8] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8878–8887, 2019.

[9] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[10] 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):529–533, 2015.

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

[12] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[13] Faraz Torabi, Garrett Warnell, and Peter Stone. Generative adversarial imitation from observation. *arXiv preprint arXiv:1807.06158*, 2018.

[14] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017.