Improved Generative Adversarial Imitation Learning Method for Stable Learning in Image Sequence Input based Game

Wonsup Shin, Hyolim Kang*, Sunghoon Hong*, Sung-Bae Cho

Dept. of Computer Science, Yonsei University, Seoul 03722, Korea
{wsshin2013, simcity429, shhong01, sbcho}@yonsei.ac.kr

Abstract

Recently, generative adversarial imitation learning (GAIL) have shown remarkable possibilities for solving practical Markov decision process problems. However, they lack the capability to manage low-level, and high-dimensional state input, such as image sequences. Furthermore, the reward function learned in the traditional GAIL only lies in a positive range, acting as a non-penalized reward and making the agent difficult to learn the optimal policy. In this paper, we propose a new algorithm based on the GAIL that can stably learn from image sequence input. Our method proposes a new component called global encoder to solve two issues that arise when applying GAIL to high-dimensional image state. Also, it has the penalization mechanism which provides more adequate reward to the agent, resulting in stable performance improvement. The potential of our approach can be backed up by the fact that it is generally applicable to variants of GAIL. We conducted in-depth experiments by applying our methods to various variants of the GAIL. The results proved that our method significantly improves the performances when it comes to image sequence state input game. For a given game, only the proposed method reached the optimal solution.

Introduction

Many games can be represented as Markov decision process (MDP). In addition, advances in storage device have made it possible to store expert trajectory data on games. In this context, imitation learning (IL), a method that can directly imitate expert behavior, has attracted much attention as a method for efficient reinforcement learning (RL) agent. Among them, generative adversarial imitation learning (GAIL) approach is showing tremendous performance over traditional IL approaches (Ho and Ermon, 2016). It was also verified that GAIL works well on high-dimensional tasks consisting of 376 sensor information. Moreover, various follow-up studies have been proposed to construct a hierarchical policy and enhance the balance of learning (Li et al., 2017; Sharma et al, 2018; Peng et al. 2018).

However, real-world problems and games often provide only low-level and high-dimensional state inputs such as image sequences. For example, an autonomous driving car task (Sallab et al., 2017) and a video game such as Atari (Mnih et al. 2015, Hessel et al. 2018) or Minecraft (Oh et al., 2016) comes with a raw image sequence an input. Unlike the sensor input where one element contains one information, it is more difficult to extract features from the image sequence input because several pixels are gathered to form one information. And, the image sequence input itself consists of over the thousands of dimensions.

When dealing with image sequence input in the GAIL, the issue can arise where state dimensions dominate action dimensions. The discriminator of the GAIL is a multi-modal model that receives state-action pairs as input. According to multi-modal studies, differences in input dimensions lead to imbalance of importance (Atrey et al. 2010). In order to solve this issue, a method of configuring an additional encoder for state input in the discriminator is mainly used (Atrey et al. 2010). But, in the GAIL, there is also a balancing issue between RL agent and discriminator, since learning in the adversarial learning setting is inherently unstable (Goodfellow et al., 2015). In the above case, the learning instability can be aggravated because each network looks at the state from different perspectives by encoding the state with respective encoder.

In this paper, we propose a novel extension model of GAIL that can solve image sequence input issues. The proposed model has a global encoder structure in which the RL agent and discriminator share the encoder of state. This improves robustness by inducing two components to give the same perspective on the state. At the same time, the problem of dimension imbalance between state-action pairs in the discriminator is also alleviated. We also propose a simple but powerful reward shaping mechanism in GAIL.

*These authors contributed equally to this work.

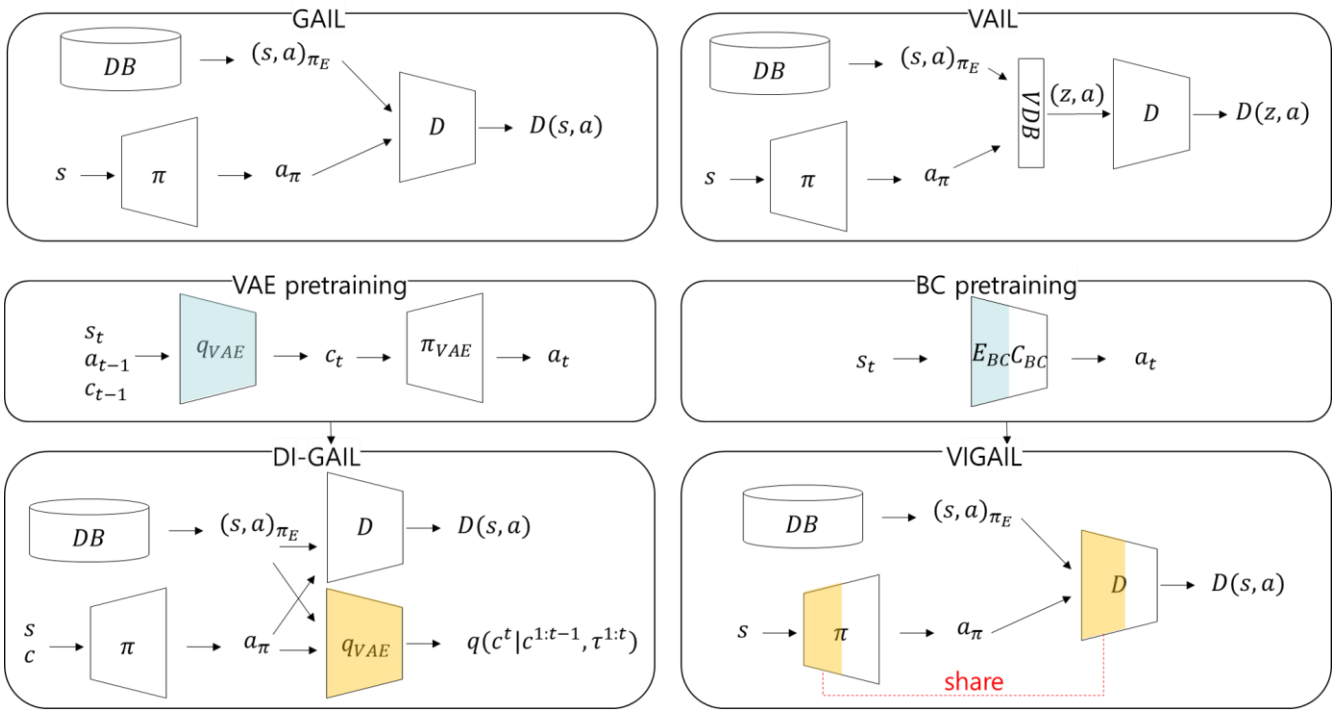

Figure 1. Schematic diagrams of GAIL, VAIL, DI-GAIL and VIGAIL

All rewards earned through the existing GAIL are positive. According to (Sutton and Barto, 1998), this reward function makes it difficult for the agent to reach optimal policy. The proposed reward shaping mechanism, reward penalization, adjusts the range of rewards to include negative numbers, providing agents with more useful rewards. The proposed approach can be used generally for the GAIL framework because it maintains adversarial learning between RL agent and discriminator, which is the basic principle of GAIL. We call this extension model as VIGAIL, video input generative adversarial imitation learning. Finally, we prove the usefulness of the proposed method in comparison with the variants of GAIL in game that use RGB image sequences as input.

Background

Imitation Learning

Although reinforcement learning can solve MDP problems, there are lots of cases that the reinforcement signal r, which is necessary to run reinforcement learning, is not provided. For this cases, imitation learning tries to yield best policy for the task by using provided expert trajectories. There are two main approaches of IL. The first is behavioral cloning (BC), which tries to yield a best policy by adopting supervised learning over the expert state-action pairs (Pomerleau, 1991). The second is inverse reinforcement learning (IRL). It tries to find optimal cost function c which derives best reward schemes that can explain the given expert trajectories (Andrew and Russell, 2000; Ziebart et al, 2008; Ziebart et al, 2010). Equation 1 shows typical object function of IRL for state s and action a.

$$maximize_{c\in C}(min_{\pi\in\Pi} - H(\pi) + \boldsymbol{E}_\pi[c(s,a)]) - \mathbb{E}_{\pi_E}[c(s,a)] \quad (1)$$

Where $H(\pi) \equiv E_\pi[-\log\pi(a|s)]$ denotes the γ-discounted entropy of the policy π, and π_E denotes the expert policy that is given as sampled trajectories in practice.

Generative Adversarial Imitation Learning

Conventional IRL approaches require additional RL step over the reward scheme derived from the IRL to get the best policy for the given task. However, inspired by GAN, GAIL derive the best policy directly from given expert trajectories (Ho et al. 2016). The formal GAIL objective is following:

$$min_\pi max_{D_{s\sim S,a\sim A}\in(0,1)}\mathbb{E}_\pi[\log D(s,a)] + \mathbb{E}_{\pi_E}[\log(1-D(s,a))] \quad (2)$$

Where D denotes the discriminator, which tries to distinguish state-action pairs from the trajectories generated by π or π_E. Theoretically, it is proved that optimizing equation 1 includes both IRL and RL step.

Variants of Generative Adversarial Imitation Learning

Variational adversarial imitation learning (VAIL) is a method of adjusting the balance between generator and discriminator by giving a constraint to the discriminator using a variational encoder called a variational discriminative bottleneck (VDB). This method adds a term to the object function that maximizes the mutual information

between $E(z|x)$ and $r(z)$ so that the discriminator can produce a significant reward.

Directed-info GAIL (DI-GAIL) is a model that agent can learn hierarchical policy without knowledge of option, meaning macro action. In order to learn option, they use directed information (Kramer, 1998) as a measure to map option to latent variable c. Therefore, they add a term that maximizes the directed information between c and trajectory $\tau = (s_1, a_1, s_2, a_2, \dots s_{t-1})$ In order to approximate the distribution of c necessary for the use of the objective function, the approximate function q is trained by the pre-training phase and then transferred.

While the above two variants improve the stability based on information theory, the proposed model improves robustness through structural modification, so it can be easily combined with the variants without friction. In experiment section, we have described the performance of the combined model and discuss the results.

Video Input Generative Adversarial Imitation Learning

In this section, we present overall structure and detailed description of our approach. Figure 1 shows the schematic diagram of GAIL, VAIL, DI-GAIL, and VIGAIL. The light blue part of the component means that it is transferred to the next phase and the orange part means the part that has been transferred.

Global Encoder

Our idea is simple. The idea is that the RL agent and discriminator share an encoder for state encoding. Because the global encoder learns a reduced representation of the state, it can solve the problem of the state dimension dominating on the action dimension in the discriminator. At the same time, the RL agent and discriminator learn using the same state representation, which solves the instability problem that can occur when using the state encoder. Since global encoders are used for both RL agents and discriminators, stable learning of global encoders is a key element in the learning process. We separated the entire training process into global encoder pre-training phase and main phase for stable learning of global encoder. In the pre-training phase, we train the actor $\pi_{BC} = \{E_{BC}, C_{BC}\}$ using BC algorithm. Where E_{BC} is encoder part of actor and C_{BC} is classifier part of actor. After that, the trained E_{BC} is transferred to the global encoder and fixed. After that the RL agent is learned in the same manner as the GAIL. Because BC does supervised learning, it can model the expert trajectory most robustly. The object function of Video-Input-GAIL is defined as equation 3.

$$min_\pi max_{D_{s\sim S, a\sim A}\in(0,1)} \mathbb{E}_\pi[\log D(E_{BC}(s), a)] + \mathbb{E}_{\pi_E}\left[\log\left(1 - D(E_{BC}(s), a)\right)\right] \quad (3)$$

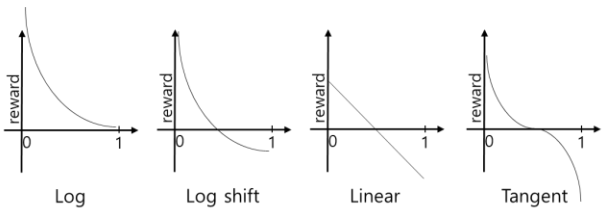

Figure 2. Visualization of reward schemes for reward function.

Reward Penalization

From the RL agent's point of view, $min_\pi E_\pi[\log D(s,a)]$ part of equation 2 can be reinterpreted as a reward function:

$$R(s,a) = -\log D(s,a) \quad (4)$$

Note that the domain of $D(s,a)$ is [0,1] and the equilibrium is formed when $D(s,a) = 0.5$ for all the (s,a) pairs. If $R(s,a) = -\log D(s,a)$, our agent will get positive reward function for every step, even though the agent did not learn at all. To solve it, we suggest new reward function:

$$R(s,a) = -\log(D(s,a) + 0.5) \quad (5)$$

that satisfies $R(s,a) = 0$ when $D(s,a) = 0.5$. Not only this reward transformation results in stable performance near the equilibrium, it also provides a more strict reward for the RL agent. This is because the transformed reward function now has a negative range that can lead to suppression of bad policy.

At the same time, we also consider the reward scheme. Figure 2 shows the reward schemes which considered in this work. The 'Log shift' scheme used in equation 5 shows that the reward increases rapidly as the RL agent follows the expert trajectory, and the reward decreases slowly as it does not follow. This can be interpreted as having the effect of inducing exploitation in good policy and exploration in bad policy. However, exploitation and exploration are very task dependent. For example, a task with a lot of serious local optimum needs to avoid exploitation in good policy, and a task with a very large search space needs to avoid exploration. Therefore, we finally propose a reward function where the reward scheme is generalized as δ.

$$R(s,a) = -\delta(D(s,a) + 0.5) \quad (6)$$

Equation 6 allows us to modify the exploitation and exploration strategies by changing δ according to a given task. We compared and analyzed the performance of the 'Linear' or 'Tangent' based reward scheme in the experimental section.

We summarize the learning algorithms of VIGAIL in algorithm 1. As mentioned above, our method is generally applicable to the variants of GAIL. The application to VAIL is very similar to algorithm 1. Since there is already a VAE pre-training step in DI-GAIL, we summarized in algorithm 2 how to design DI-VIGAIL, a variant of DI-GAIL.

Algorithm 1. Video Input GAIL (VIGAIL)

Phase 1: Pre-training encoder step

Input: expert trajectories $\tau_E \sim \pi_E$, initial global encoder, actor network parameters η_0, α_0

for i $= 0, 1, 2, \cdots,$ n:

1. Sample τ from τ_E
2. Update the $\eta_i \rightarrow \eta_{i+1}, \alpha_i \rightarrow \alpha_{i+1}$, with minimize$\{\mathrm{L} = -\log \pi_{BC}(s)_i\}$

Output: global encoder parameter η_n

Phase 2: Main step

Input: expert trajectories $\tau_E \sim \pi_E$, initial actor, critic and discriminator network parameters $\alpha_0, \beta_0, \delta_0$, and trained global encoder parameter η_n from phase 1.

1. Load η_n to the global encoder and fix
2. Learn under GAIL

Algorithm 2. Directed Info VIGAIL (DI-VIGAIL)

Phase 1: Pre-training encoder step

Input: expert trajectories $\tau_E \sim \pi_E$, initial global encoder, actor network parameters η_0, α_0

for i $= 0, 1, 2, \cdots,$ n:

1. Sample τ from τ_E
2. Update the $\eta_i \rightarrow \eta_{i+1}, \alpha_i \rightarrow \alpha_{i+1}$, with minimize$\{\mathrm{L} = -\log \pi_{BC}(s)_i\}$

Output: global encoder parameter η_n

Phase 2: Pre-training posterior step

Input: expert trajectories $\tau_E \sim \pi_E$, initial actor and posterior network parameters α_0, φ_0, and trained global encoder parameter η_n.

for i $= 0, 1, 2, \cdots,$ n:

1. Sample τ from τ_E
2. Sample c_i from posterior network
3. Update the $\varphi_i \rightarrow \varphi_{i+1}, \alpha_i \rightarrow \alpha_{i+1}$, with minimize$\{ \mathrm{L}_{VAE}\ loss\ on$ (Sharma et al., 2018)$\}$

Output: posterior parameter η_n

Phase 3: Main step

Input: expert trajectories $\tau_E \sim \pi_E$, initial actor, critic and discriminator network parameters $\alpha_0, \beta_0, \delta_0$, and trained global encoder and posterior network parameter η_n, φ_m from phase 1 and 2.

1. Load η_n, φ_m to the global encoder and posterior and fix
2. Learn under DI-GAIL

Experiments

We demonstrate the effectiveness of our method on a hierarchical navigation game. In addition, we also investigate various methods for reward penalization on the LunarLander-v2 environment (Brockman et al., 2016). We will show that (1) global encoder is able to learn meaningful representation of raw image sequence input, (2) reward penalization has remarkable effect to performance and (3) our method can be applied to variants of GAIL.

The source codes of our experiments can be seen at https://github.com/sunghoonhong/VI-GAIL

Environments

To validate proposed approach, we choose hierarchical navigation task in grid world environment, which consists of a 7 x 7 grid with four rooms connected via bottleneck passage. Each grid is represented by 4 x 4 pixel with RGB formulation. So, we got 32 x 32 x 4 size state input. The agent spawns at a random grid and its goal is to reach a key, then reach a car. Both key and car spawn at a random grid in top left room and bottom right room each. In the figures, the agent is represented by the green rectangle, the key by the blue rectangle, and the car by the red rectangle. The reward is given as much as shortest distance when the agent achieves the goal, otherwise -1 for each timestep. We utilize about 1M state-action pairs generated by shortest path algorithm as expert trajectory.

We also experiment in LunarLander-v2 environment, provided in OpenAI Gym (Brockman et al., 2016). The agent spawns at the top of the screen and its goal is to land on the landing pad. The action can be firing main, left or right engine or doing nothing. The state consists of position, velocity, angle, angle velocity and contact of legs. The reward is given for leg ground contact or landing on landing pad. On the other hand, the penalty is given for firing engine or crashing. We use about 10K state-action pairs generated by the agent trained using PPO (Schulman et al., 2017) algorithm as expert demonstration.

On the hierarchical navigation task, we conduct three experiments: demonstrating that our method is able to learn in raw image state, analyzing encoded states, and applying our method to variants of GAIL on various settings, including DI-GAIL for hierarchical learning. Furthermore, we analyze several ways of reward penalization on LunarLander-v2 environment.

Variants of VIGAIL

We combine our proposed approach, Video-Input-GAIL, to VAIL and DI-GAIL. In addition, we also apply it to GAIL without global encoder for demonstrating that reward penalization is effective even in another experiment. For implementation detail, we use PPO algorithm for training agents rather than TRPO (Schulman et al., 2015).

Table 1. Results on the navigation task.

Model	Best score	score	Meets -10	After meets -10
GAIL	-97.491	-99.43±0.80	-	-
VAIL	-99.949	-100.00±0.01	-	-
GAIL_LS	-1.516	-31.18±29.54	28K	-9.10±9.26
VAIL_LS	-1.237	-25.71±28.28	23K	-10.89±13.02
GAIL_GE	-99.939	-99.97±0.03	-	-
VIGAIL	**1**	-6.63±15.56	13K	-3.37±9.26
VIGAIL + VDB	**0.996**	-3.45±11.97	**3K**	-2.39±8.59
DI-GAIL_GE	-92.824	-96.91±2.05	-	-
DI-VIGAIL-	**1**	-6.00±17.79	**3K**	-4.12±13.32
DI-VIGAIL + VDB	**0.995**	-4.84±14.28	**3K**	-3.18±9.78

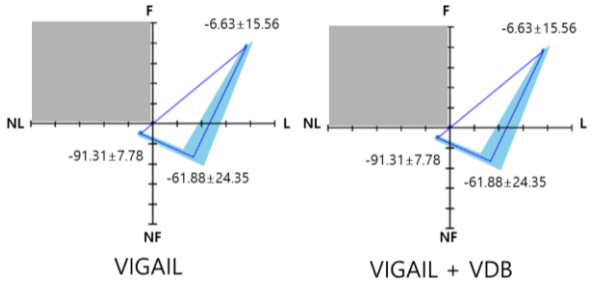

Figure 3. Comparison of the training strategy for global encoder.

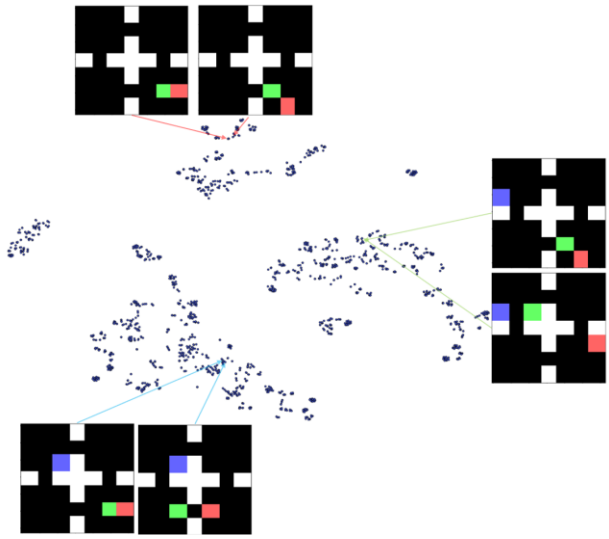

Figure 4. Visualization of encoded state using t-SNE.

Performance results

We apply each component step by step to various variations on the navigation task and show the quantitative evaluation in terms of performance and learning stability. The result is calculated by the score which is a mean return over 1000 episodes. We assume that the agent has learned enough if the score meets -10. As can be seen in Table 1, variants of GAIL applied our method show superior performance rather than naïve methods without ours. Firstly, GAIL and VAIL cannot solve the task at all. And it seems that the agents only with global encoder improve very little bit, but still cannot solve either. On the other hand, the agents with only with reward penalization show much better performance but still cannot completely solve either. As a result, the agents with our method meet score 1 which is the upper bound and show stable performance

Analysis on global encoder

In figure 3, we demonstrate that (1) loading the weights of global encoder from BC and (2) fixing the weight of global encoder during learning are necessary. To show these we experiment in other strategies. We experiment for both VIGAIL and VIGAIL + VAIL. 'F' means fixing the weights of global encoder, 'L' means loading the weights from BC pre-training. One does not load the weights from BC pre-training but randomly initialize and train. The other loads the weights from the pre-training and does not fix the weights during training. We skip the case which randomly initializes and fix the weights. Only with our proposed strategy the agent is trained properly. On the other hand, agents with other strategies fail to learn to get enough score, and most of them collapse as learning progresses. For the reason, we suspect that loading and fixing the weights reduce instability in GAIL framework which is inherently fluctuating.

Furthermore, we analyze how the encoded states are distributed in figure 4. For the states at the bottom and right side, in case of the agent hasn't taken key yet, the encoder gives a focus on the position of key rather than the position of car. For the states at the top, in case of the agent has already taken the key, it seems that encoder gives a focus on the distance between the agent and the car. Additionally, while the left-side state of the top and the left-side state of the bottom has the same distance between the agent and car, the distance of encoded states

Table 2. Comparison of reward penalization schemes on navigation task.

Scheme	Score
Log	-99.97±0.03
Log scaled	-97.43±2.48
Log shift	-6.63±15.56
Linear	-5.02±14.18
Tan	-6.50±17.88

Table 3. Comparison of reward penalization schemes on LunarLander-v2 environment.

Scheme	Score	Solved (%)	Collapsed (%)
Log	91.84±86.53	0	0
Log scaled	92.46±87.77	0	10
Log shift	135.91±99.24	20	10
Linear	146.40±97.11	30	10
Tan	187.90±91.62	40	0

is huge. The only difference between two states is whether the agent has taken the key or not. These show that the encoder works in terms of informative encoding and give a focus on which is important to the agent.

Analysis on reward penalization

For comparison of several schemes, we experiment using VIGAIL with 5 reward schemes on two environments. For LunarLander-v2 environment, we assume that it is solved if the score is over 200 and collapsed if the score is under 0 after learning.

We investigate several reward penalization schemes on two environments. For a baseline, we use the original log reward which is always positive, and scaled log reward which is divided by 10 so that the reward is bounded in smaller range. Then we compare shifted log reward, linear reward and tangent reward which is positive in [0, 0.5) and negative in (0.5, 1]. We used $\tan\big(0.5 - D(s,a)\big)$, $0.5 - \mathrm{D(s,a)}$ for each tangent reward and linear reward.

As can be seen in Table 2,3, it is obvious that the agents trained under reward penalization show remarkably high performance rather than non-penalization. On the other hand, there is no superior scheme among three penalization schemes. On the navigation task, linear scheme shows the best result, but on LunarLander-v2 environment, tangent scheme seems that the most effective scheme. As a result, we demonstrate that reward penalization significantly improves the performance, and the choice of scheme can be a hyperparameter.

Analysis on latent code

We also apply our method to DI-GAIL which can learn hierarchical policy based on option framework. We

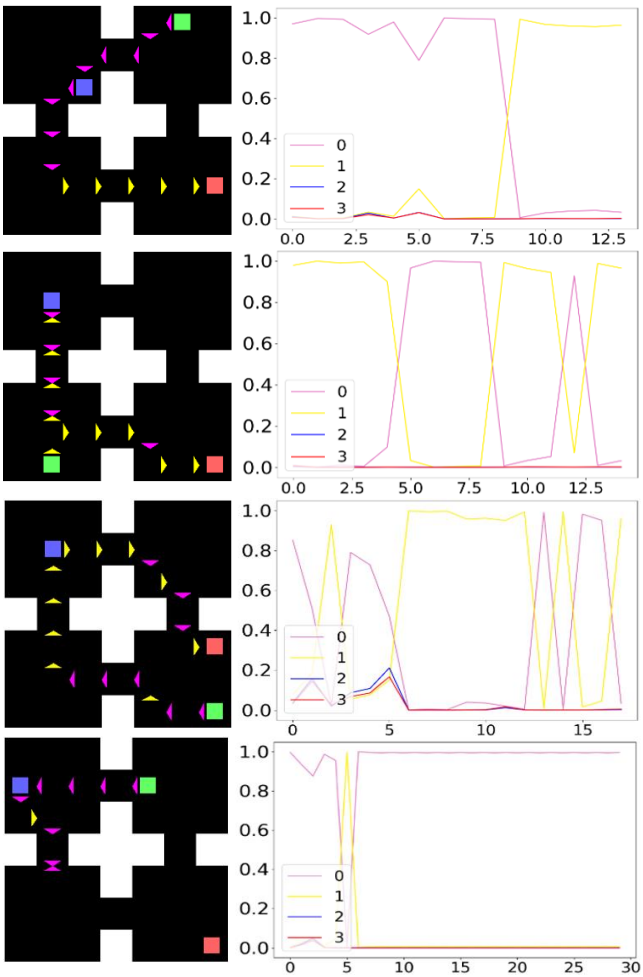

Figure 5. Trajectories and predictions of each code for DI-VIGAIL on navigation task.

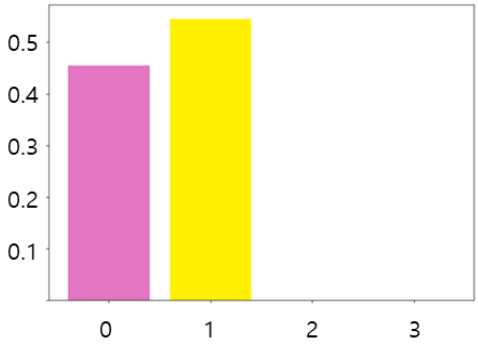

Figure 6. Proportion of codes during total episodes.

set the latent code as four different categorical variables. In Figure 5, the arrow means the action by its direction and the code by its color. According to Figure 6, only two code variables are used which are un-supervisedly learned from pre-training. The agent appropriately uses two codes as episode proceeds. It seems that each code corresponds to different traversal strategy. The code 0 denoted as pink color tends to direct the agent to traverse

along left and downward, while the code 1 denoted as yellow color tends to direct the agent to traverse along right and upward. The last trajectory in Figure 5 shows that the agent which is trained using naïve method fails to learn. While the code in 8th timestep tells the agent to move downward, it chooses to move upward. It means that pre-trained distribution of code properly provides the agent to choose correct actions, even there is some possibilities of learning failure of the agent. From the above results, we can say it is possible that the DI-VIGAIL agent is able to learn consistent and meaningful latent code variables in unsupervised method and solve the problem which has hierarchy.

Discussion

Adopting the pretrained encoder showed meaningful performance improvement. It seems that using pre-trained global encoder through BC mitigates inherent instability of GAIL framework. However, reconstruction pre-trained used in the World Model (Ha et al, 2018) doesn't work in our experiment. Further study to find better structure or pre-training method for global encoder is needed. For instance, adopting transfer learning to the global encoder can be a feasible attempt.

Moreover, it was revealed that penalizing reward played significant role when it comes to performance improvement of imitation learning task. Due to its ease of implementation and potential of general application, it can be considered as meaningful contribution.

Furthermore, since there are many real-world problems which have complicated hierarchical structure and high-dimensional state space, especially raw image sequence, we expect high potential in our method in that it is able to learn hierarchical policy from raw image sequence inputs.

Conclusion

We have proposed VIGAIL, a novel GAIL based method that is adaptable to games that use image sequence inputs. The key ideas are the global encoder and reward penalization mechanism. Also, the proposed method is generally available for GAIL framework. As a result of in-depth experiments, the proposed method outperforms the existing methods, and further experiments demonstrate the usefulness of the proposed method.

Acknowledgements

This research was supported by Korea Electric Power Corporation. (Grant number:R18XA05)

References

Andrew Y. Ng and S. Russell. 2000. Algorithms for inverse reinforcement learning. *In ICML*.

Atrey, P. K.; Hossain, M. A.; El Saddik, A.; and Kankanhalli, M. S. 2010. Multimodal fusion for multimedia analysis: a survey. *Multimedia systems*. 16(6):345-379.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym. arXiv preprint arXiv:1606.01540.

Ha, D.; Schmidhuber, J. 2018. World Models. arXiv preprint arXiv:1803.10122.

Hessel, M.; Modayil, J.; Van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; ... and Silver, D. 2018. Rainbow: Combining improvements in deep reinforcement learning. *In AAAI*.

Ho J. and Ermon S. 2016. Generative adversarial imitation learning. *In NIPS*, 4565-4573.

Kramer, G. 1998, Directed information for channels with feedback. *Hartung-Gorre*.

Li, Y.; Song, J.; and Ermon, S. 2017. Infogail: Interpretable imitation learning from visual demonstrations. *In NIPS*. pp. 3812-3822.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. Nature 518(7540):529–533.

Oh J.; Chockalingam V.; Singh S.; and Lee H. Control of memory, active perception, and action in minecraft. *In ICML*. 2016.

Peng, X. B.; Kanazawa, A.; Toyer, S.; Abbeel, P.; and Levine, S. 2018. Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow. *arXiv preprint arXiv:1810.00821*.

Pomerleau. D.A. 1991. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 88-97.

Sallab, A. E.; Abdou, M.; Perot, E.; and Yogamani, S. 2017. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 19:70-76.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; Moritz, P. 2015. Trust region policy optimization. In *International conference on machine learning* 1889-1897.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv preprint arXiv:1707.06347.

Sharma, A.; Sharma, M.; Rhinehart, N.; and Kitani, K. M. 2018. Directed-info gail: Learning hierarchical policies from unsegmented demonstrations using directed information. *arXiv preprint arXiv:1810.01266*.

Sutton, R. S.; and Barto, A. G. 1998. Introduction to reinforcement learning. *Cambridge: MIT press*. 2(4).

Ziebart B. D.; Maas A.; Bagnell J. A.; and Dey A. K. 2008. Maximum entropy inverse reinforcement learning. *In AAAI*.

Ziebart B. D.; Bagnell J. A.; and A. K. Dey. 2010. Modeling interaction via the principle of maximum causal entropy. *In ICML*. 1255-1262.