Initialization Study of Generative Adversarial Imitation Learning

Yue Yang
College of Computing
Atlanta, United States of America
yyang941@gatech.edu

Tongzhou Yu
College of Computing
Atlanta, United States of America
tyu310@gatech.edu

Julia Zhu
College of Computing
Atlanta, United States of America
jzhu407@gatech.edu

***Abstract*—Generative Adversarial Imitation Learning (GAIL) is an approach of imitation learning that uses demonstration data by experts and learns the unknown environment's policy directly from data. One of its drawbacks is that it is unstable and difficult to train, so we propose a flexible framework for studying GAIL's initialization. We design 3 experiments and show that certain instantiations of our framework yields significant improvement gains. We have also provided several conclusions that are valuable for practical GAIL training.**

I. Introduction

Reinforcement learning (RL) is a popular algorithm that could be helpful to solving complicated control problems, where manually designing reward function is oftentimes necessary. However, this could be intractable and time consuming. A feasible way to solving this problem is Imitation learning (IL). IL is an agent's acquisition of skills or behaviors by observing an expert demonstrating a given task, and extracts information about the expert behavior and surrounding environments, then learning a mapping between the scenario and given behavior. A standard imitation learning process starts with acquiring demonstrations from an expert which are then used to train a policy. The agent then acts out the policy and refines it depending on performance[Hussein and Jayne, 2017].

Inverse reinforcement learning (IRL) is one popular algorithm in IL that implicitly learns the reward function. Most IRL algorithms require reinforcement learning in an inner loop and thus is expensive to run especially when scaled to large environments. GAIL was designed to bypass those intermediate RL steps and directly learn policy from data. It runs reinforcement learning on a cost function learned by maximum causal entropy IRL.

However, GAIL is not sample efficient in regards to environment interaction during training, and struggles with learning a policy from multi-modal demonstrations as it assumes all demonstrations come from a single expert. Adversarial imitation learning is quite successful in various environments, but "adversarial methods are shown to be unstable, and in the presence of low amounts of data, can take a long time to converge" [Jena and Sycara, 2020b]. Ho and Ermon [2016] argue that the learning speed of GAIL could be significantly improved once behavioral learning is used for initialization, which inspires us to study the best way for initialization. We propose a framework that can flexibly test pre-trained policies generated by different methods with different dynamics as initialization for the initialization of GAIL. Contributions of this work are shown as follows:

1) Propose a framework for flexibly studying the initialization of GAIL.
2) Design different and meaningful settings for different initialization methods, and provide 3 detailed experiments to study them.
3) Draw several valuable conclusions that could be guidance for GAIL training.

II. Related Works

The two main approaches to imitation learning are behavioral cloning [Torabi et al., 2018], which learns a policy as a supervised learning problem over state-action pairs from expert trajectories, and inverse reinforcement learning (IRL) [Ng et al., 2000], which finds a cost function under which the expert behavior is uniquely optimal and based on cost rather than policies. GAIL [Ho and Ermon, 2016] was designed with a framework that directly learns policies from data, which bypasses any intermediate IRL steps. The algorithm also uses generative adversarial training to fit distributions of states and actions defining expert behavior, and has been proven to outperform competing methods (e.g. behavioral cloning, feature expectation matching, game-theoretic apprenticeship learning) in training policies for high-dimensional physics-based control tasks over expert data.

Transfer learning is used to improve performance of one learner in one domain by transferring knowledge learned from learner in the other domain [Weiss et al., 2016], which has achieved great success in computer vision area. For control tasks, skills learned by one agent could be useful for learning other skills via training an invariant feature space [Gupta et al., 2017]. Eysenbach et al. [2018] has shown that a pretrained policy generated by maximizing the entropy of states and minimizing the conditional entropy of states with respect to actions could provide a good initialization for downstream tasks, which is similar to a vanilla transfer learning setting.

Initializing parameters can have a drastic impact on training, as any errors can be propagated during training. In deep neural networks, incorrect initialization can lead to issues such as the vanishing or exploding gradient problem. For instance, initializing all weights with zeros leads the neurons to learn

the same features during training[Katanforoosh and Kunin]. Traditional initialization methods include gaussian distribution and uniform initialization. Gaussian distribution initializes weights with a distribution that has a mean of zero and a standard deviation of one. In uniform initialization, weights belong to a uniform distribution. Our proposed method could bring a much more powerful initialization because a pre-training process is needed, but we argue it's worthwhile.

III. Methodology

In this section, we present a simple framework of studying the initialization of generative adversarial imitation learning (GAIL). The problem formulation will be shown and we'll also give a brief introduction to core methods that are parts of the proposed framework.

A. Preliminaries

1) Problem Formulation: Let us consider a Markov Decision Process (MDP) task that is defined as $M = (S, A, T, r, \rho_0, \gamma)$, where S represents the state space, A represents the action space, $T : S \times A \times S \to \mathbb{R}$ represents the transition matrix, r denotes the reward function, ρ_0 represents the initial state distribution, and γ denotes the discount factor. Let π_θ denote the learned policy and π_E denote the expert policy.

2) Generative Adversarial Imitation Learning: Generative Adversarial Imitation Learning (GAIL) [Ho and Ermon, 2016] originated from inverse reinforcement learning where you need to estimate some unknown reward and learn a policy based on the recovered reward. This could be extremely expensive because you have to solve a reinforcement learning problem within a learning loop. GAIL solved this issue by introducing a discriminator to implicitly learn the reward function automatically. In the GAIL framework, the expert policy's behavior is imitated by the learned policy via minimizing the distance between two distributions generated by each other. The distance is measured by introducing the Jensen-Shannon divergence, where the GAIL objective could be derived:

$$\min_{\pi_\theta} \max_{D} \quad \mathbb{E}_{\pi_\theta}[logD(s,a)] + \mathbb{E}_{\pi_E}[log1 - D(s,a)] - \lambda H(\pi_\theta) \tag{1}$$

where D is the discriminator classifier which tries to tell apart the distribution generated by the expert policy π_E and learned policy π_θ. $H(\pi_\theta) \triangleq \mathbb{E}_{\pi_\theta}[-log\pi_\theta(a|s)]$ is the causal entropy used as a policy regularizer controlled by $\lambda \geq 0$ [Bloem and Bambos, 2014].

3) Behavioral Cloning: Behavioral Cloning (BC) is one line work of imitation learning, which tries to cast the task of learning from demonstrations to a supervised learning method. Although there exists some concerns about its shortcomings [Ross et al., 2011, Tu et al., 2021], it's still practical in some real scenario tasks [Zhang et al., 2018]. Many methods are devoted to solve the existing issues of BC, but we here will adopt the simplest setting. As most supervised learning method, the BC takes a form of $\mathbf{a} = \mathbb{F}_\theta(\mathbf{s})$, where $\mathbb{F}_\theta$ denotes a feed-forward model (e.g., a deep neural network), $\mathbf{s}$ represents the current state, and $\mathbf{a}$ is the predicted action. The predicted action will be compared with those in demonstrations to force the $\mathbb{F}_\theta$ generate more similar actions at the current state $\mathbf{s}$.

B. Framework for GAIL's initialization

Despite the success adversarial learning has achieved in many fields, instability and difficulty for converging given low amounts of data have always been a big issue [Jena and Sycara, 2020a]. This kind of issues also hold in GAIL considering its adversarial essence. In addition, some researchers [Li et al., 2017] point out that GAIL tends to fail generalizing environments with different dynamics. Actually, GAIL seems to assume the single-modality of demonstrations [Fu et al., 2017], so it's quite struggle for GAIL to learn from heterogeneous demonstrations. Therefore, it's vital to find a good initialization that could oftentimes provide a proper start point for the following optimization. We propose to use pre-trained method's policy as initial policy for GAIL as our initialization method, which have the following benefits: 1) pre-trained method could bring start point to a better position compared to traditional methods. 2) pre-trained method has the potential to distill knowledge from demonstrations with different modality, which is beneficial for GAIL that can only learn from demonstrations of single modality.

In this section, we present a framework for the initialization of GAIL, where the general idea is using pre-trained policy as the initial policy for GAIL before the formal training. The proposed framework could flexibly take two important features that may affect the initialization performance into consideration, including the pre-trained methods and dynamic parameters. As shown in Figure 1, the framework is composed of two parts: Pre-train Module and GAIL Module. As for the GAIL module, it's a vanilla GAIL algorithm. What makes it different is the initial policy π_θ that will be replaced by the output of Pre-train Module. In the Pre-train Module, different combinations of pre-train method and dynamics will be used to generate the initial policy for GAIL Module. We have designed four different pre-train methods, including randomization, Behavioral Cloning (BC), RL with simple reward function and another GAIL. We modify the dynamics by changing the length of agent's parts. The selected method will be trained in the selected dynamics and give the initial policy for downstream GAIL Module.

Note that the Pre-train Module is repeatable, which means that it can be stacked one by one. For instance, to stack two Pre-train Modules, the generated policy by the first Pre-train Module will be used as input policy for the second Pre-train Module. And the second Pre-train Module will then provide initial policy for the downstream GAIL module.

IV. Experiments and Results

In this section, we will show design of experiments and the correspond results.

A. Experimental Setup

1) Demonstrations: The expert demonstrations for GAIL and BC to imitate are collected using Soft Actor-Critic

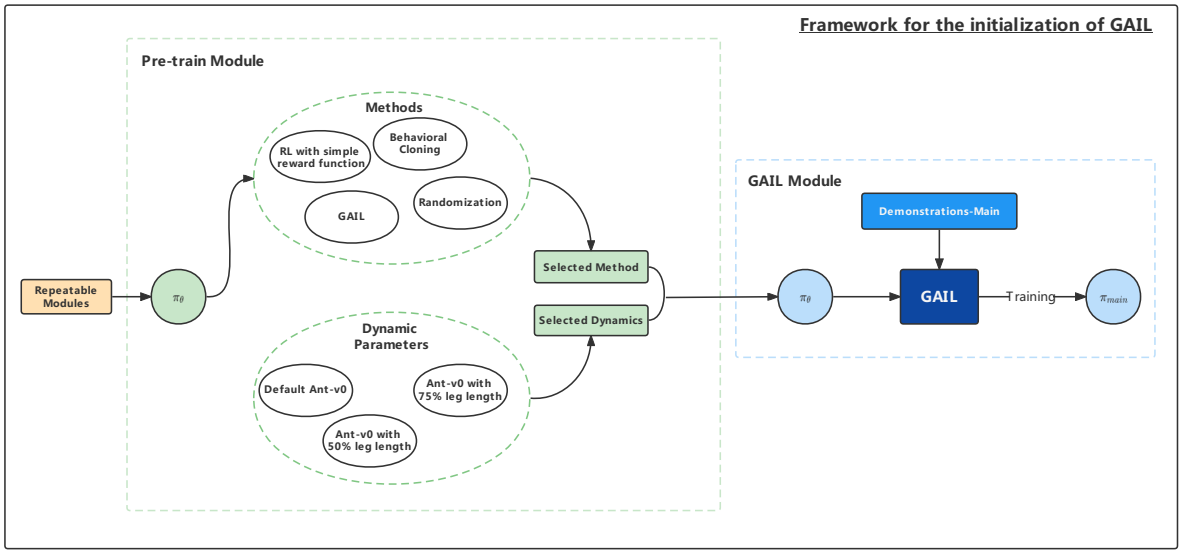

Fig. 1. Framework for the initialization of GAIL. The GAIL method will be used as a downstream module for a repeatable pre-train module, where different combinations of method and dynamics will be studied in this report.

(SAC) [Haarnoja et al., 2018]. The algorithm is run on the selected dynamics and 50 trajectories of rollouts will later be collected as our expert demonstrations.

2) Training details: For all experiments, the epoch number is set to 70. We have also run each experiment for 3 times by using different random seeds in order to avoid variance. The best performance in these 3 iterations will be selected.

3) Dynamic parameters setting: We choose Mujoco [Todorov et al., 2012] as our simulation environments and Ant-v0 as the agent. We make dynamic parameters variable by setting length of two ant's legs to different float numbers. In our setting, two lengths are used: 50% original length and 75% original length (As shown in Figure 2). We also try repeating the pre-train method on the former first and then on the later so as to study the stack feature of Pre-train Module. This is called "Ant-v0 with repeated dynamics".

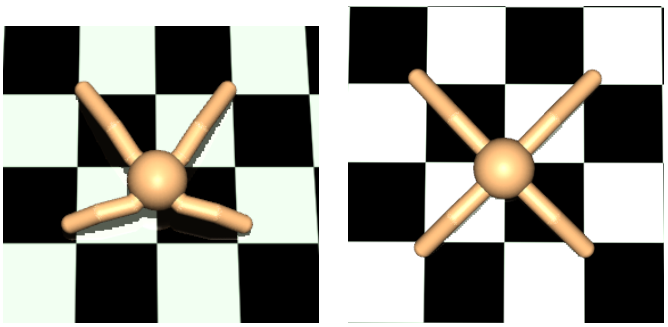

Fig. 2. Ant-v0 with 50% original length(left) and Ant-v0 with 75% original length(right)

4) Experiments design: GAIL with randomization initialization in default Ant-v0 is used as our baseline method. We totally design 3 experiments to study three different aspects of initialization for GAIL:

(a) **Experiment-1**: Study the best pre-trained method. BC and RL with simple reward function are compared here. The dynamics will be fixed to default Ant-v0 for both of them.

(b) **Experiment-2**: Study the best dynamics setting in Pre-train module. default Ant-v0, Ant-v0 with 50% length, Ant-v0 with 75% length and Ant-v0 with repeated dynamics will be compared. The pre-train method is fixed to BC.

(c) **Experiment-3**: Study whether GAIL itself can learn a good initialization by pre-training GAIL in different dynamic parameters. GAIL pre-trained on Ant-v0 with 50% length, Ant-v0 with 75% length and Ant-v0 with repeated dynamics are compared.

B. Results

In this section, we show the results generated by following the experimental design.

1) Experiment-1: Experiment-1 tries to study the best pre-trained method via comparing BC and RL with simple reward function. As shown in Figure 3, performance of GAIL pre-trained with behavioral cloning is much better than by RL with a simple reward function.

The possible reason why rl with simple reward function gets a bad result is as follows: we simulate the scenario of designing simple reward function as just adding noise to observation and action space because we suppose they all get not so good results, but this simulation process may bring some variance that negatively affect the performance of RL. And the BC is better because it also learn from demonstrations, which is the same as GAIL, that could provide a similar policy.

2) Experiment-2: Experiment-2 tries to study the best dynamics setting in Pre-train module. As shown in Figure 4, We

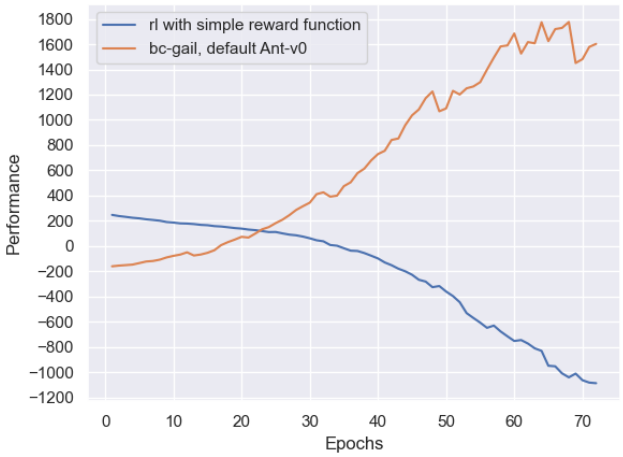

Fig. 3. Study the best pre-trained method.

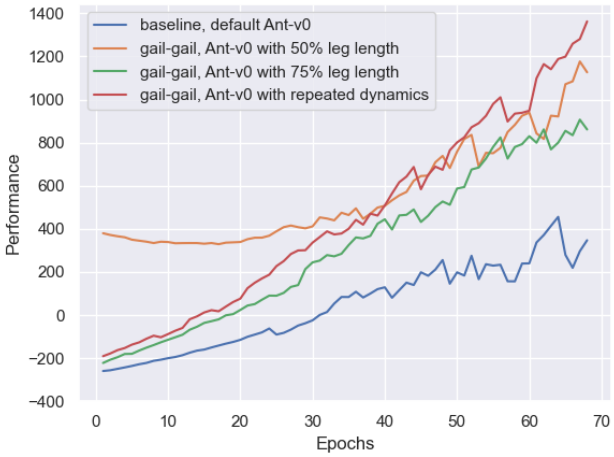

Fig. 5. Study whether GAIL itself can learn a good initialization by pre-training GAIL in different dynamic parameters.

have the observation that the best performance will be achieved when same dynamics are shared between the Pre-train Module and GAIL Module.

A possible explanation is that GAIL is sensitive to change of dynamic parameters, so BC trained in the same dynamic parameters as the GAIL Module could bring more benefits.

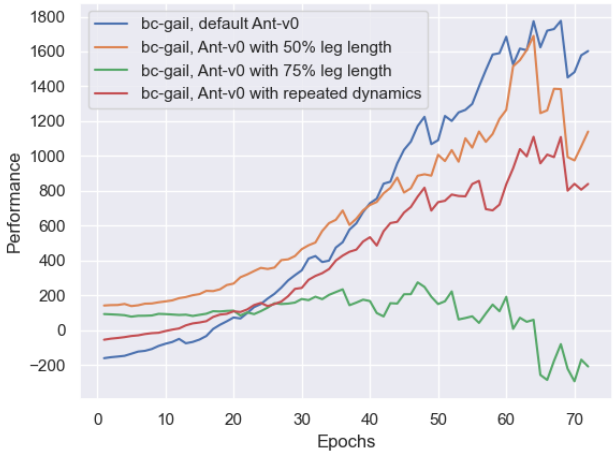

Fig. 4. Study the best dynamics setting in Pre-train module.

3) Experiment-3: Experiment-3 tries to study whether GAIL itself can learn a good initialization by pre-training GAIL in different dynamic parameters. As shown in Figure 5, we have two observations: 1) GAIL pre-trained on different dynamics are better than random initialization (i.e., baseline method), which might result from that GAIL can learn better on simple tasks when pre-trained on harder tasks if we assume ant with two legs shorter are harder to run. 2) Pre-train GAIL in Ant-v0 with repeated dynamics could gain the best performance. We suppose the possible reason is that policy experiencing different dynamic parameters might average them and thus give a more meaningful initial policy.

4) Summary of all experiments: All methods with different dynamic parameters are presented in Figure 6. We have also calculated statistical data by evaluating the last ten epochs, which are shown in Table I. We could see that the best setting is fix pre-train method as BC and for Pre-train Module use the same dynamics as GAIL module.

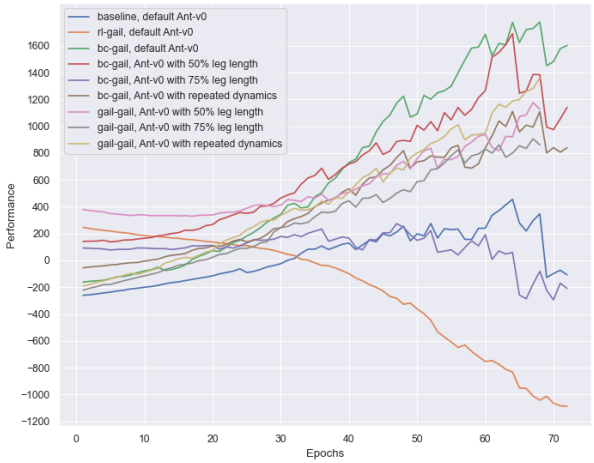

Fig. 6. Summary of all experiments' results

V. Discussion

As for RL with simple reward function, we see a very poor performance. We argue that it results from the wrong way for us to implement it as stated above. Another issue emerged in our results is that some settings' curve don't show a plateau and are still increasing, that's because the training epoch is not large enough owing to time limitation.

VI. Conclusion and Future Work

In this work, we proposed a framework for the initialization of GAIL, using which we see a huge improvements for several settings. By designing three experiments, we could gain the following valuable conclusions from the generated results:

1) Training GAIL using BC as pre-trained method is a good idea.

TABLE I
STATISTICAL DATA FOR ALL SETTINGS

	Mean Value	Std Value
baseline, default Ant-v0	160.44	223.17
rl-gail, default Ant-v0	-984.87	93.23
BC-gail, default Ant-v0	1634.30	108.57
BC-gail, Ant-v0 with 50% leg length	1273.91	33.33
BC-gail, Ant-v0 with 70% leg length	-159.08	121.48
BC-gail, Ant-v0 with repeated dynamics	946.20	112.06
gail-gail, Ant-v0 with 50% leg length	1090.50	90.81
gail-gail, Ant-v0 with 70% leg length	911.82	110.13
gail-gail, Ant-v0 with repeated dynamics	1157.92	128.72

2) When pre-train the BC, same dynamic parameters as the GAIL module could bring the most benefits. However, we argue that demonstrations from the same dynamic parameters are not always available. So BC pre-trained in different dynamics should be considered as it also brings improvements according to our results.
3) GAIL could learn a good initialization by itself via pre-training GAIL in different dynamic parameters.

We believe these conclusions are meaningful and instructive for getting good results from GAIL. In the future, RL with simple reward function should be implemented in another way and more epochs should be set to gain more solid results.

REFERENCES

Michael Bloem and Nicholas Bambos. Infinite time horizon maximum causal entropy inverse reinforcement learning. In *53rd IEEE conference on decision and control*, pages 4911–4916. IEEE, 2014.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

Abhishek Gupta, Coline Devin, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Learning invariant feature spaces to transfer skills with reinforcement learning. *arXiv preprint arXiv:1703.02949*, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Gaber Mohamed Elyan Eyad Hussein, Ahmed and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys*, 50, 2017.

Rohit Jena and Katia Sycara. Loss-annealed gail for sample efficient and stable imitation learning. *arXiv preprint arXiv:2001.07798*, 5, 2020a.

Rohit Jena and Katia Sycara. Loss-annealed gail for sample efficient and stable imitation learning. 2020b.

Katanforoosh and Kunin. Initializing neural networks.

Yunzhu Li, Jiaming Song, and Stefano Ermon. Infogail: Interpretable imitation learning from visual demonstrations. *Advances in Neural Information Processing Systems*, 30, 2017.

Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

Stephen Tu, Alexander Robey, Tingnan Zhang, and Nikolai Matni. On the sample complexity of stability constrained imitation learning. *arXiv preprint arXiv:2102.09161*, 2021.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635. IEEE, 2018.