2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

GAN-Based Interactive Reinforcement Learning from Demonstration and Human Evaluative Feedback

Jie Huang[1*], Jiangshan Hao[1*], Rongshun Juan[1], Randy Gomez[2], Keisuke Nakamura[2], Guangliang Li[1**]

Abstract— **Generative adversarial imitation learning (GAIL) — a general model-free imitation learning method, allows robots to directly learn policies from expert trajectories in large environments. However, GAIL shares the limitation of other imitation learning methods that they can seldom surpass the performance of demonstrations. In this paper, to address the limit of GAIL, we propose GAN-based interactive reinforcement learning (GAIRL) from demonstrations and human evaluative feedback, by combining the advantages of GAIL and interactive reinforcement learning. We test GAIRL in six physics-based control tasks, ranging from simple low-dimensional control tasks — Cart Pole, Mountain Car and Lunar Lander, to difficult high-dimensional tasks — Inverted Double Pendulum, Hopper and HalfCheetah. Our results suggest that, the GAIRL agent can generally surpass the performance of demonstrations in both low-dimensional and high-dimensional tasks and get an optimal or close to optimal policy.**

I. INTRODUCTION

Deep reinforcement learning (DRL) has achieved great success in fields ranging from games [1], [2], [3], [4] to complex locomotion behaviors [5], [6], robotic manipulators [7], intelligent transportation [8] etc. However, a DRL agent usually needs a specified reward function for a task before learning how to perform it. It is difficult or even unpractical to design an efficient reward function for complex and varied tasks, which makes applying traditional DRL methods to real-world robots a great challenge [9]

Since most robots will operate in human-inhabited environments, the ability to interact and learn from human users will be key to their success [10], [11], [12]. Many approaches for robot learning from interaction with a human user have been proposed [13], [14], [15], [16]. Among them, we are interested in this specific setting of learning to perform a task from demonstrations, where the learner is provided with demonstrations consisting of several sequences of state-action pairs.

The simplest approach in this setting is behavioral cloning (BC) [17], in which the goal is to learn the mapping from states to optimal actions. However, BC needs large amounts of data to learn and cannot generalize to unseen states [18], [19]. Another approach is inverse reinforcement learning (inverse RL), which learns a policy via RL, using a cost function extracted from expert trajectories [20]. Because of the assumption of expert optimality as prior on the space of policies, inverse RL can allow the learner to generalize expert behaviors to unseen states more effectively [21]. However, many of the proposed inverse RL algorithms need a model to solve a sequence of planning or reinforcement learning problems in an inner loop [22]. Moreover, their agents' performance might significantly degrade if the planning problems are not solved to optimality [21], [23]. Therefore, by drawing an analogy between imitation learning and generative adversarial networks (GANs) [24], Ho et al. [22] proposed generative adversarial imitation learning (GAIL) — a general model-free framework for directly learning policies from the expert trajectories, and extended inverse RL to large environments. However, in many tasks, the optimal demonstrations are hard to obtain practically, if the behavior of the demonstrator is suboptimal or far from optimal GAIL can seldom surpass the performance of demonstrations.

Fortunately, an agent via interactive reinforcement learning (interactive RL) from human evaluative feedback can generally surpass the trainer's performance in the task [25], [10], [26]. In this paper, we propose GAN-based interactive reinforcement learning (GAIRL) combining the advantages of GAIL and interactive RL from human evaluative feedback. Our results in six physics-based control tasks show that with much fewer time steps trained in total, the proposed GAIRL agent can get an optimal or close to optimal policy, and address the limitation of GAIL.

II. RELATED WORK

A. Imitation Learning via Inverse Reinforcement Learning

There are many inverse RL algorithms developed with linear approximation for the reward function, including apprenticeship learning [27], maximum entropy inverse RL [28], etc. While apprenticeship learning with linear programming (LP) can directly generate fixed policies [29], the performance of the LP solver can drop significantly if the planning problem is not optimally solved in the inner loop [23]. The game-theoretic apprenticeship learning is computationally faster, easier to implement [30]. However, recovering the agent's exact weights of linear approximation for the reward function is an ill-posed problem [20]. To resolve the ambiguity of choosing a decision distribution, Ziebart et al. [28] proposed maximum entropy inverse RL.

Most of above mentioned inverse RL methods are extremely expensive to run. Ho et al. [21] proposed to learn a class of cost functions by distinguishing the expert policy from all others. Taking further inspiration from the success of nonlinear cost function classes in inverse RL [31], Ho et al. [22] proposed GAIL. However, GAIL inherits the problems of GAN. For example, exploding gradients might stand out

[1]College of Information Science and Engineering, Ocean University of China, {guangliangli}@ouc.edu.cn
[2]Honda Research Institute Japan Co., Ltd, Wako, Japan. {r.gomez, keisuke}@jp.honda-ri.com
* Contributing equally. ** Corresponding author

when the given expert demonstrations are not optimal [32]. By introducing a new type of variational autoencoder on demonstration trajectories, Wang et al. [33] increased the robustness of GAIL and avoided its mode collapse. Instead of adversarial imitation learning, there are also many other methods integrating the preprocessed expert demonstrations into extensions of classic RL algorithms [34], [35], [36].

Above imitation learning methods are based on the premise that the provided demonstrations are optimal, which is difficult to obtain expert data due to several reasons, such as noisy demonstrations resulting from bounded rationality of demonstrating humans [37]. For that reason, Wu et al. [38] tried to learn a confidence weight to model the optimality of state-action pairs in demonstrations. Jing et al. [39] regarded the demonstrations as a soft constraint on regulating the policy exploration of the agent to learn from imperfect demonstrations. Our work differs by allowing an agent to learn from both demonstrations and human evaluative feedback, overcoming the limitation shared by imitation learning that it seldom surpasses the performance of demonstrations.

B. Interactive Reinforcement Learning

Inspired by potential-based reward shaping [40], interactive RL is proposed to allow an agent to learn from interaction with human and improve the learning efficiency of a RL and DRL agent at the same time [26]. For example, Thomaz and Breazeal [41] implemented a tabular Q-learning [42] agent learning from environmental and human rewards. The TAMER agent learns from only human reward signal by directly modeling it [43]. To facilitate an agent to learn in tasks with high-dimensional state space, Warnell et al. proposed deep TAMER [25]. The COnvergent Actor-Critic by Humans (COACH) algorithm learns by interpreting human reward signal as feedback to the current executing control policy of a robot [44]. COACH was also extended to deep COACH [45]. Ibarz et al. combined imitation learning and learning from trajectory preferences, in which humans compare pairs of short trajectory segments of an agent's behaviour and label those closer to the intended goal to train a reward model that acts as a preference predictor [46]. In addition, most related to our work, Li et al. proposed a method allowing an agent to learn from both human demonstrations and evaluative feedback [10]. Our work differs by allowing an agent to learn in complex environments with a GAN-based method.

III. BACKGROUND

A. Preliminaries

We consider an agent within the Markov decision process (MDP) framework. An MDP can be represented with a tuple $M = \{S, A, P, \mathbb{R}, \gamma\}$. $\pi \in \Pi$ is a policy that takes an action $a \in A$ given a state $s \in S$. Successor states are derived from the dynamics model $P(s' \mid s, a)$. During the process, the agent will get feedback $c(s,a)$ from a cost function $c : S \times A \to \mathbb{R}$. $\overline{\mathbb{R}}$ denotes the extended real numbers $\mathbb{R} \bigcup \{+\infty\}$. $\mathbb{E}_\pi[c(s,a)] \triangleq \mathbb{E}[\sum_{t=0}^{T} \gamma^t c(s_t, a_t)]$ denotes an expectation of the discounted return along the trajectory generated by policy π, where γ is a discounted factor and $\gamma \in (0,1]$. Similarly, $\mathbb{E}_\tau$ represents an empirical expectation with respect to trajectory samples τ. We use π_E to represent the expert policy and τ_E to represent the expert trajectory samples.

B. Generative Adversarial Imitation Learning

In GAIL, the cost function was set to be:

$$c(s,a) = log(D(s,a)), \tag{1}$$

where $D : S \times A \to (0,1)$ is a discriminative classifier. The cost function $c(s,a)$ will be used to provide reward signals to update the agent's policy. GAIL can be summarized as finding a saddle point (π, D) of the expression:

$$-\lambda H(\pi) + \mathbb{E}_\pi[log(D(s,a))] + \mathbb{E}_{\pi_E}[log(1 - D(s,a))], \tag{2}$$

where λ is the weight of entropy $H(\pi)$; a discriminator D is trained to distinguish expert transitions $(s,a) \sim \tau_E$ from agent transitions $(s,a) \sim \tau_{agent}$. The agent's trajectory samples τ_{agent} are obtained from its interaction with the environment using agent's current policy π. The agent is trained to "fool" the discriminator into thinking itself as the expert. Taking $-\lambda H(\pi)$ out of (2), the loss function is analogous to that of GAN, which draws an analogy between imitation learning and GAN. Specifically, the GAIL agent learns the policy directly by making its distribution of state-action pairs as close as possible to that of the demonstrator. The algorithm used in GAIL to update the agent's policy is trust region policy optimization (TRPO) [47].

IV. PROPOSED APPROACH

In this paper, we proposed GAN-based interactive reinforcement learning (GAIRL). GAIRL is expected to leverage demonstrations and human evaluative feedback to improve GAIL and outperform the demonstrator. In GAIRL, we introduce a new cost function which can be expressed as:

$$c_{gairl}(s,a) = log(D(s,a)) + \alpha \mathbb{H}(s,a), \tag{3}$$

where $\mathbb{H}(s,a)$ is the human reward function approximated with human reward network (HRN), and α is a weight vector for balancing the cost function $c_{gail}(s,a)$ out of the discriminator in GAIL and the learned HRN. GAIRL can be summarized as solving a GAIL-like step which can be expressed as:

$$\min_{\pi \in \Pi} \max_{D \in (0,1)^{S \times A}} \mathbb{E}_\pi[log(D(s,a))] + \alpha \mathbb{E}_\pi[\mathbb{H}(s,a)] + \mathbb{E}_{\pi_E}[log(1 - D(s,a))] - \lambda H(\pi) \tag{4}$$

and then performing a RL step expressed as:

$$RL(c_{gairl}) = \arg\min_{\pi \in \Pi} \mathbb{E}_\pi[c_{gairl}(s,a)]. \tag{5}$$

The pseudo code of GAIRL is summarized in Algorithm 1, which is mainly divided into two parts: GAIL alike step (solving (4)) and RL step (solving (5)). Specifically, in GAIL alike step, we use neural networks as functional approximators for the policy π and discriminator D. Similar to GAIL, in each new iteration, GAIRL starts by sampling demonstrated trajectories from demonstrations and agent trajectories via

Algorithm 1 Part 1 — GAIL alike step

Input: Demonstrations $\mathscr{E}$, hyperparameters: λ—weight of entropy; α—weight of HRN $\mathbb{H}$; lr—learning rate for both the Adam and Rmsprop optimizer; max_{kl}—the Kullback-Leibler loss threshold in TRPO

Output: Trained discriminator D and HRN $\mathbb{H}$, agent's policy π_i

1: Randomly initialize agent's policy π_0, discriminator D_0 and HRN $\mathbb{H}$
2: Assign a human reward of $R_h = +1$ to all the state-action pairs $[s_d, a_d]$ from $\mathscr{E}$
3: Store all the tuples $[s_d, a_d, +1]$ in replay buffer $\mathscr{B}_1$, $\mathscr{B}_2$
4: Number of iterations i = 0
5: **repeat**
6: Sample agent trajectories τ_i from π_i and demonstrated trajectories τ_E from $\mathscr{E}$
7: **if** Receive human feedback R_h with respect to $[s,a]$ from τ_i **then** store $[s,a,R_h]$ in $\mathscr{B}_1$
8: Randomly sample a batch of $[s,a,R_h]$ from $\mathscr{B}_1$
9: Update $\mathbb{H}$ using Rmsprop with loss $\mathscr{L}_{hrn}$
10: **end if**
11: Update D_i using Adam with loss
$-(\mathbb{E}_{\tau_i}[log(D_i(s,a))] + \mathbb{E}_{\tau_E}[log(1 - D_i(s,a))])$
12: Update π_i using TRPO with loss
$-\lambda H(\pi) + \mathbb{E}_{\tau_i}[c_{gairl}(s,a)]$
$= -\lambda H(\pi) + \mathbb{E}_{\tau_i}[log(D_i(s,a))] + \alpha \mathbb{E}_{\tau_i}[\mathbb{H}(s,a)]$
13: Get D_i's expert accuracy AC_e and agent accuracy AC_a
14: $i = i + 1$
15: **until** Both AC_e and $AC_a \geq 0.99$
16: $D = D_i$

interaction with the environment using the latest policy π_i (line 6). However, different from GAIL, in GAIRL, a human trainer can provide evaluative feedback R_h by evaluating the agent's behavior according to her knowledge in the task (line 7). The human reward R_h is defined as below:

$$R_h = \begin{cases} +N, & \textit{agent reaches the goal} \quad (6a) \\ +1, & \textit{good action} \quad (6b) \\ -1, & \textit{bad action} \quad (6c) \\ -N, & \textit{agent fails} \quad (6d) \end{cases}$$

where N is set to be different values in different tasks. We use human evaluative feedback as labels of corresponding samples to train HRN for predicting it (line 8-9). Specifically, we store the sample $[s,a,R_h]$ in the replay buffer $\mathscr{B}_1$ for HRN and randomly sample a minibatch of samples from $\mathscr{B}_1$ to update HRN by minimizing the loss:

$$\mathscr{L}_{hrn} = \frac{1}{n}\sum_{i=0}^{n}(R_h - \mathbb{H}(s,a))^2, \quad (7)$$

where $\mathbb{H}(s,a)$ is the estimated HRN, n is the number of samples. In addition, we assign a human reward of $R_h = +1$ to all the state-action pairs in the demonstrations, and store $[s_d, a_d, +1]$ in the replay buffer to train HRN (line 2-3). Note that R_h is given by humans only to train HRN, while the human evaluative feedback used to update the policy π is output from HRN rather than given directly by humans.

Then, GAIRL will perform an Adam gradient step on the current discriminator D_i using the latest sampled τ_i and τ_E (line 11). The cost function $c_{gail}(s,a)$ will be generated from the latest discriminator D_i as in GAIL. The learned human reward function $\mathbb{H}$ and cost function $c_{gail}(s,a)$ will be used in the new local cost function $c_{gairl}(s,a)$ of our method to provide reward signals to update the agent's current policy π_i with TRPO (line 12). However, the human reward network $\mathbb{H}$ might be not good enough at this step. Therefore, we set α in $c_{gairl}(s,a)$ to be 0.001. Then we can update the discriminator network as in GAIL while learning the human reward network $\mathbb{H}$ simultaneously with little effect on the cost function $c_{gairl}(s,a)$.

Algorithm 1 Part 2 — RL step

Input: Trained Discriminator D, trained HRN $\mathbb{H}$, hyperpatameters: α—weight of HRN $\mathbb{H}$; lr—learning rate for the optimizer in both DQN and TD3; su—the soft update coefficient in TD3

17: Initialize agent's policy with π_i from Part 1
18: **while** Policy Improves **do**
19: Sample agent trajectories τ_i from π_i
20: **if** Receive human feedback R_h with respect to $[s,a]$ from τ_i **then** store $[s,a,R_h]$ in $\mathscr{B}_2$
21: Randomly sample a batch of $[s,a,R_h]$ from $\mathscr{B}_2$
22: Update $\mathbb{H}$ using Rmsprop with loss $\mathscr{L}_{hrn}$
23: **end if**
24: Update π_i using DQN/TD3 with reward
$c_{gairl}(s,a) = log(D(s,a)) + \alpha \mathbb{H}(s,a)$
25: **end while**

This step is to obtain a "good" discriminator D to distinguish demonstrated transitions from agent transitions. As in GAIL, a desired discriminator's output $D(s,a)$ should be greater than 0.5 when taking the demonstrated transitions $(s,a)_e$ as input and less than 0.5 when taking the agent transitions $(s,a)_a$ as input. Therefore, we take $AC_e = \frac{n_e}{N_e}$ as the discriminator's expert accuracy and $AC_a = \frac{n_a}{N_a}$ as a discriminator's agent accuracy, where N_e and N_a are the total number of demonstrated transitions and agent transitions respectively; n_e and n_a are the number of times when $D((s,a)_e) > 0.5$ and $D((s,a)_a) < 0.5$. In each iteration, AC_e and AC_a will be calculated (line 13). Both the discriminator and agent's policy (i.e., generator) will be updated by repeating the above steps, until both AC_e and AC_a are greater than 0.99 (line 15). With Algorithm 1 part 1, we obtained discriminator D and HRN $\mathbb{H}$ with high accuracy for predicting rewards and good generator π.

In the RL step, the agent's policy will be initialized with π_i from the Part 1 in GAIRL (line 17). In addition, the human trainer can further provide evaluative feedback to train the human reward function $\mathbb{H}$ (line 20-22). By setting α to be a larger value, the new cost function $c_{gairl}(s,a)$ consisting of both the cost function $c_{gail}(s,a)$ from the discriminator D and learned human reward function $\mathbb{H}$ from Part 1 in GAIRL will serve as a reward function for the agent to perform updates

on the policy (line 24). In our work, we choose deep Q network (DQN) [48] in tasks with discrete action spaces and twin delayed deep deterministic policy gradient (TD3) [49] in tasks with continuous action spaces.

V. EXPERIMENTS

A. Experimental Tasks

We test GAIRL in six physics-based control tasks: Inverted Double Pendulum (MuJoCo); Hopper and HalfCheetah (Pybullet); Cart Pole, Mountain Car and Lunar Lander (OpenAI Gym). Note that, a custom reward function $|s[0]-(-0.06)|$ was used in Mountain Car, while the original defined reward functions were used in other tasks. The dimensions of state and action spaces are shown in Table I.

TABLE I: Dimensions of state, action spaces of each task.

Task	State space	Action space
Cart Pole	4, continuous	2, discrete
Mountain Car	2, continuous	3, discrete
Lunar Lander	8, continuous	4, discrete
Inverted Double Pendulum	11, continuous	1, continuous
Hopper	15, continuous	3, continuous
HalfCheetah	26, continuous	6, continuous

B. Experimental Setup

In each task, we train five agents which can be divided into three groups:

- Learning from demonstrations: a GAIL agent as baseline; a BC agent; a DQND or TD3D agent was trained as GAIRL, but in the RL step, it learns from rewards provided by the cost function extracted from demonstrations in the GAIL-like step;
- Learning from human reward: a DQNH or TD3H agent was trained as GAIRL, but in the RL step, it learns from human reward function estimated in the GAIL-like step;
- Learning from both demonstrations and human reward: a GAIRL agent learning with the new cost function $c_{gairl}(s,a)$.

Note that, a fixed cost function $c_{gail}(s,a)$, human reward function $\mathbb{H}$ and the agent's policy will be learned in the GAIL-like step of GAIRL. In the RL step of GAIRL, all agents' policies will be initialized with the learned policy from the the GAIL-like step. The DQND/TD3D agent and DQNH/TD3H agent are used in the ablation studies to investigate the contribution of demonstrations and human evaluative feedback in GAIRL. For each task, a true reward function from OpenAI Gym [50] is used for evaluating the agent performance, but never for learning.

During experiments, we create demonstrated policies of different qualities – optimal or suboptimal, by running DQN/TD3. Then datasets of trajectories are sampled from the demonstrated policies, each consisting of about 200 state-action pairs. For all experiments, we use 10 demonstrations as suggested by Ho et al. [22]. The first authors trained the human reward network HRN for all tasks. The total number of human evaluative feedback given to train HRN is about 150 in Cart Pole and Mountain Car, 200 in Lunar Lander and Inverted Double Pendulum, 300 in Hopper and HalfCheetah. The hyperparameters are set as below: $\lambda = 0.001$; $lr = 0.001$ for all optimizers; $max_{kl} = 0.01$; $su = 0.005$.

We train all agents' policies of the same feedforward neural network architecture for all tasks: two hidden layers of 100 units each, with *tanh* nonlinearities in between. The architecture of discriminator network is two hidden layers of 100 units each, with *tanh* nonlinearities in between and *sigmoid* nonlinearities in the output layer. The human reward network has two hidden layers of 100 units each, with *Relu* nonlinearities in between and *tanh* nonlinearities multiplying *N* in the output layer. All networks are always initialized randomly at the start of each trial. In each task, three random seeds are used for the environment simulator and random initialization of the network. Table II shows the total number of time steps used to train the five agents in different tasks.

TABLE II: Total number of time steps used to train the GAIL, GAIRL, DQND/TD3D and DQNH/TD3H agents.

	GAIL	GAIRL	DQND/TD3D	DQNH/TD3H
		Total steps	GAIL alike step	RL step
Cart Pole	1500K	1006K ± 20K	1000K ± 20K	6K
Mountain Car	1500K	1006K ± 50K	1000K ± 50K	6K
Lunar Lander	2000K	1810K ± 100K	1800K ± 100K	10K
Inverted Double Pendulum	20000K	17000K ± 200K	15000K ± 200K	2000K
Hopper	20000K	19000K ± 500K	17000K ± 500K	2000K
HalfCheetah	20000K	19000K ± 1000K	17000K ± 1000K	2000K

VI. RESULTS AND DISCUSSION

In this section, we present and analyze the experimental results by comparing the performances of GAIL, GAIRL, DQND/TD3D, DQNH/TD3H, and BC agents in the six tasks. The performance metric, if not specified, used throughout the experiments is the episode reward in terms of the true reward function from Gym. The shaded area is the 0.95 confidence interval and the bold line is the mean performance. Both optimal and suboptimal demonstrations are offered in the Cart Pole task; only suboptimal demonstrations are offered in other tasks.

A. Optimal Integration of Demonstration and Human Evaluative Feedback

To investigate the optimal way of integrating demonstrations and human evaluative feedback, we compare the five agents' final performance with varied αs for GAIRL. The final policies of all agents are frozen and tested for 100 episodes with the true reward function from Gym in all tasks, then the accumulated reward per episode is averaged over the tested 100 episodes as final performance. Fig. 1 shows the normalized final performance of five agents in the six tasks.

From Fig. 1 we can see that, in relatively simple tasks (e.g., Cart Pole, Mountain Car, Lunar Lander), as α increases, the final performance of the GAIRL agent goes up to optimal when learning with optimal demonstrations and close to optimal when learning with only suboptimal demonstrations. In Inverted Double Pendulum, Hopper and HalfCheetah, the final performance of the GAIRL agent only improves and reaches close to optimal when α increases from 0.1 to 0.5, then stagnates in Inverted Double Pendulum and decreases in Hopper and HalfCheetah.

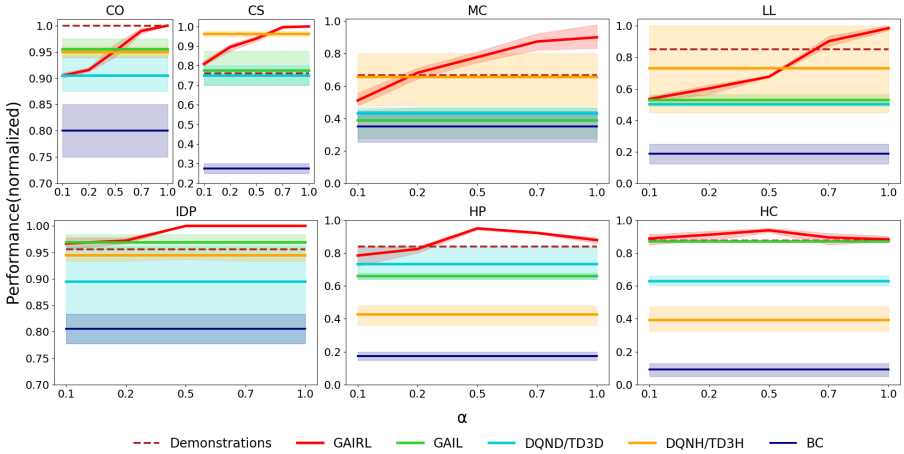
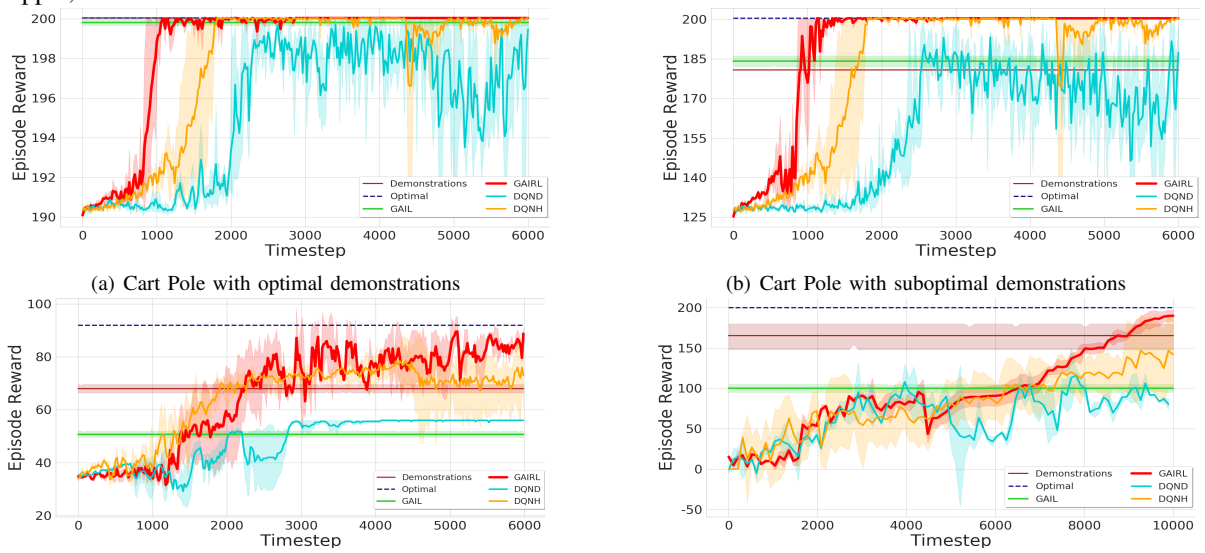


Fig. 1: Final performance of the five agents with different integrations of demonstrations and human evaluative feedback. y-axis is normalized episode reward with optimal performance as 1. Note: CO—Cart Pole with Optimal Demonstrations, CS—Cart Pole with Suboptimal Demonstrations, MC— Mountain Car, LL—Lunar Lander, IDP—Inverted Double Pendulum, HP—Hopper, HC—HalfCheetah.



(a) Cart Pole with optimal demonstrations

(b) Cart Pole with suboptimal demonstrations

(c) Mountain Car with suboptimal demonstrations

(d) Lunar Lander with suboptimal demonstrations

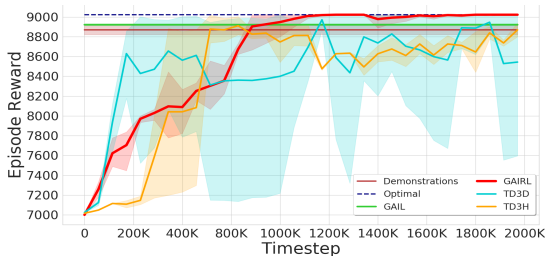
Fig. 2: The learning curves of DQNH, DQND and GAIRL agents in the RL step of low-dimensional tasks.

In addition, results in Fig. 1 show that in relatively simple tasks (e.g., Cart Pole, Mountain Car, Lunar Lander), the DQNH/TD3H agent can get better performance than the GAIL and DQND/TD3D agent especially when the demonstrations are suboptimal. In contrast, in complex tasks (e.g., Inverted Double Pendulum, Hopper and HalfCheetah), the GAIL and DQND/TD3D agent can get better performance than the DQNH/TD3H agent, except the performance of the DQND/TD3D agent in Inverted Double Pendulum is a bit worse than that of the DQNH/TD3H agent. This might be the reason why the value of α for a GAIRL agent to achieve the best performance in complex tasks is smaller than in relatively simple tasks. However, the BC agent obtains the worst performance in all tasks, since the learned policy cannot generalize. Our results show that with a proper value of α in all tasks, the GAIRL agent can obtain much better performance than learning from demonstrations and human evaluative feedback alone separately, achieving optimal or close to optimal performance.

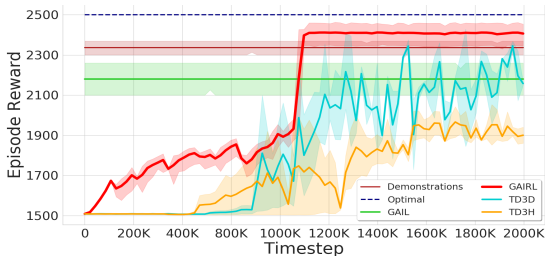
B. Learning Curve and Ablation Study

We further analyze the effect of demonstrations and human evaluative feedback on GAIRL agent's performance by comparing the learning curves of the GAIRL, DQND/TD3D and DQNH/TD3H agents, with ablation studies. All learning curves in the RL step (Part 2 of our algorithm) are plotted. The RL step builds on the GAIL alike step (Part 1 of our algorithm), which initializes the agent's policy using the policy learned in the GAIL alike step, and further learns from the generated fixed cost function $c_{gail}(s,a)$ and human reward function $\mathbb{H}$. Therefore, we only plotted the final performance for the GAIL agent only for comparison rather than the learning curve. Since the BC agent does not learn interactively, its performance is not shown. Throughout the RL step, each policy of the three agents is frozen every a fixed interval and tested for 100 episodes with the true reward function from Gym. The mean performance was used for comparison. Note that α is set to be able to achieve the best performance in all tasks for the GAIRL agent (1 for

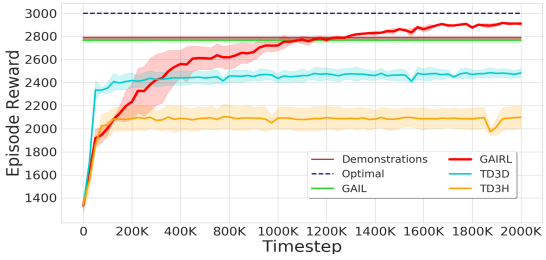
Cart Pole, Mountain Car and Lunar Lander, 0.5 for Inverted Double Pendulum, Hopper and HalfCheetah).



(a) Inverted Double Pendulum with suboptimal demo



(b) Hopper with suboptimal demo



(c) HalfCheetah with suboptimal demo

Fig. 3: The learning curves of TD3H, TD3D and GAIRL agents in the RL step of high-dimensional tasks.

1) Low-dimensional Tasks: Fig. 2(a), (b), (c) and (d) show the learning curves of DQNH, DQND and GAIRL agents in the RL step in low-dimensional tasks. From Fig. 2 we can see that, generally, the DQNH agent learns much faster and better than the DQND agent. The DQND agent's performance is worse than or similar to the GAIL agent, both of which are worse than demonstrations, while the performance of the DQNH agent can be better than or similar to demonstrations. In contrast, the GAIRL agent can learn a much better policy that is optimal or close to optimal faster than the DQND agent and DQNH agent. Moreover, it takes the GAIRL agent only a few thousand time steps in the RL step (1k-2k in Cart Pole and Mountain Car, and 7k-9k in Lunar Lander, shown in Fig. 2) to surpass the GAIL agent and demonstrations.

2) High-dimensional Tasks: Fig. 3 shows the learning curves of the GAIRL agent, TD3D agent and TD3H agent in three high-dimensional tasks. From Fig. 3 we can see that, in high-dimensional tasks, the TD3D agent learns faster and better than the TD3H agent, and the performance of both agents are generally much worse than that of the GAIL agent and demonstrations. In contrast, the GAIRL agent can learn a much better policy that is optimal or close to optimal than the TD3D agent and TD3H agent. Moreover, it takes the GAIRL agent only about 1000k time steps (1000k in Inverted Double Pendulum and Hopper, and 1200k in HalfCheetah, shown in Fig. 3) to surpass the GAIL agent and demonstrations.

C. Stability Analysis

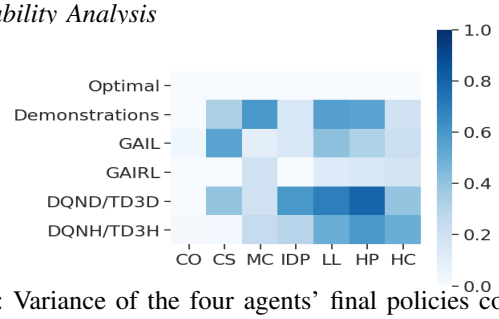


Fig. 4: Variance of the four agents' final policies compared to the demonstrated and optimal policy in the six tasks. Note: labels for horizontal axis is the same as in Fig.1.

We also study the stability of the final learned policy of the four agents: GAIL, GAIRL, DQND/TD3 and DQNH/TD3H, by testing them for 10 times with 10 random seeds in each task and computing the normalized variance over the 10 performances for each policy. The variances of all policies, including the demonstrated policy and the optimal policy, are shown in Fig. 4. The optimal policy is always stable while the stability of the demonstrated policy is different in each task. Fig. 4 shows that the GAIRL agent always learns a much more stable policy than the demonstrated policy and most of the time almost as stable as the optimal one, while the GAIL agent's policy is only slightly more stable than the demonstrated policy and even a bit less stable in Cart Pole with both optimal and suboptimal demonstrations. In addition, the policy of DQND/TD3D and DQNH/TD3H agents become generally less stable as the task's difficulty increases (from left to right in Fig. 4), while the stability of the GAIRL agent keeps at a similar and much higher level than both of them. This suggests the demonstrations and human evaluative feedback might have a complementary effect for further learning at the RL step. For example, demonstrations might provide a high-level initialization of the human's overall reward function, while human evaluative feedback like preferences can explore specific, fine-grained aspects of it [15].

VII. CONCLUSION

In this paper, to address the limit of GAIL that it seldom surpasses the performance of demonstrations, we propose GAN-based interactive reinforcement learning (GAIRL) by combining the advantages of GAIL and interactive RL. We tested GAIRL in six physics-based control tasks from the classic RL literature. Our results suggest that, human evaluative feedback can result in more effective learning in low-dimensional tasks, while demonstrations result in efficient and better learning in high-dimensional tasks. Our proposed GAIRL agent can obtain a more stable policy with better performance than learning separately from demonstrations or human evaluative feedback alone in both low-dimensional and high-dimensional tasks. The proposed GAIRL agent can get an optimal or close to optimal policy with higher sample efficiency compared to GAIL.

VIII. Acknowledgement

This work was partially supported by Natural Science Foundation of China (under grant No. 51809246) and Honda Research Institute Japan Co., Ltd.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[2] G. A. DeepMind, "Mastering the real-time strategy game starcraft ii," 2019.

[3] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[4] Y. Gu, Y. Cheng, C. P. Chen, and X. Wang, "Proximal policy optimization with policy feedback," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[5] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. Eslami, *et al.*, "Emergence of locomotion behaviours in rich environments," *arXiv preprint arXiv:1707.02286*, 2017.

[6] C. Florensa, Y. Duan, and P. Abbeel, "Stochastic neural networks for hierarchical reinforcement learning," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2017.

[7] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[8] T. Tan, F. Bao, Y. Deng, A. Jin, Q. Dai, and J. Wang, "Cooperative deep reinforcement learning for large-scale traffic grid signal control," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2687–2700, 2019.

[9] M. Riedmiller, R. Hafner, T. Lampe, M. Neunert, J. Degrave, T. Wiele, V. Mnih, N. Heess, and J. T. Springenberg, "Learning by playing solving sparse reward tasks from scratch," in *Proceedings of International Conference on Machine Learning (ICML)*, 2018, pp. 4344–4353.

[10] G. Li, B. He, R. Gomez, and K. Nakamura, "Interactive reinforcement learning from demonstration and human evaluative feedback," in *Proceedings of the 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2018, pp. 1156–1162.

[11] B. Gebru, L. Zeleke, D. Blankson, M. Nabil, S. Nateghi, A. Homaifar, and E. Tunstel, "A review on human–machine trust evaluation: Human-centric and machine-centric perspectives," *IEEE Transactions on Human-Machine Systems*, 2022.

[12] T. B. Sheridan, "Human–robot interaction: Status and challenges," *Human factors*, vol. 58, no. 4, pp. 525–532, 2016.

[13] K. Akash, G. McMahon, T. Reid, and N. Jain, "Human trust-based feedback control: Dynamically varying automation transparency to optimize human-machine interactions," *IEEE Control Systems Magazine*, vol. 40, no. 6, pp. 98–116, 2020.

[14] Z. Huang, F. Ren, M. Hu, and S. Chen, "Facial expression imitation method for humanoid robot based on smooth-constraint reversed mechanical model (srmm)," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 6, pp. 538–549, 2020.

[15] E. Bıyık, D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, "Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences," *arXiv preprint arXiv:2006.14091*, 2020.

[16] M. Matarese, A. Sciutti, F. Rea, and S. Rossi, "Toward robots' behavioral transparency of temporal difference reinforcement learning with a human teacher," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 6, pp. 578–589, 2021.

[17] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the 30th International Conference on Artificial Intelligence and Statistics*, 2010, pp. 661–668.

[18] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," in *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

[19] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7559–7566.

[20] A. Y. Ng, S. J. Russell, *et al.*, "Algorithms for inverse reinforcement learning." in *Proceedings of International Conference on Machine Learning (ICML)*, vol. 1, 2000, p. 2.

[21] J. Ho, J. Gupta, and S. Ermon, "Model-free imitation learning with policy optimization," in *Proceedings of International Conference on Machine Learning (ICML)*, 2016, pp. 2760–2769.

[22] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 4565–4573.

[23] S. Ermon, Y. Xue, R. Toth, B. Dilkina, R. Bernstein, T. Damoulas, P. E. Clark, S. DeGloria, A. Mude, C. Barrett, *et al.*, "Learning large-scale dynamic discrete choice models of spatio-temporal preferences with application to migratory pastoralism in east africa." in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015, pp. 644–650.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 2672–2680.

[25] G. Warnell, N. Waytowich, V. Lawhern, and P. Stone, "Deep tamer: Interactive agent shaping in high-dimensional state spaces," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018.

[26] G. Li, R. Gomez, K. Nakamura, and B. He, "Human-centered reinforcement learning: a survey," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 4, pp. 337–349, 2019.

[27] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the 21st International Conference on Machine Learning (ICML)*, 2004, p. 1.

[28] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[29] U. Syed, M. Bowling, and R. E. Schapire, "Apprenticeship learning using linear programming," in *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 2008, pp. 1032–1039.

[30] U. Syed and R. E. Schapire, "A game-theoretic approach to apprenticeship learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2008, pp. 1449–1456.

[31] N. D. Ratliff, D. Silver, and J. A. Bagnell, "Learning to search: Functional gradient techniques for imitation learning," *Autonomous Robots*, vol. 27, no. 1, pp. 25–53, 2009.

[32] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2017.

[33] Z. Wang, J. Merel, S. Reed, G. Wayne, N. de Freitas, and N. Heess, "Robust imitation of diverse behaviors," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, 2017, pp. 5326–5335.

[34] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, *et al.*, "Deep q-learning from demonstrations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[35] Y. Gao, H. Xu, J. Lin, F. Yu, S. Levine, and T. Darrell, "Reinforcement learning from imperfect demonstrations," in *Proceedings of International Conference on Learning Representations (ICLR) Workshop*, 2018.

[36] S. Reddy, A. D. Dragan, and S. Levine, "Sqil: Imitation learning via reinforcement learning with sparse rewards," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.

[37] M. Kwon, E. Biyik, A. Talati, K. Bhasin, D. P. Losey, and D. Sadigh, "When humans aren't optimal: Robots that collaborate with risk-aware humans," in *2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2020, pp. 43–52.

[38] Y.-H. Wu, N. Charoenphakdee, H. Bao, V. Tangkaratt, and M. Sugiyama, "Imitation learning from imperfect demonstration," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6818–6827.

[39] M. Jing, X. Ma, W. Huang, F. Sun, C. Yang, B. Fang, and H. Liu, "Reinforcement learning from imperfect demonstrations under soft expert guidance," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5109–5116.

[40] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proceedings of International Conference on Machine Learning (ICML)*, vol. 99, 1999, pp. 278–287.
[41] A. L. Thomaz and C. Breazeal, "Teachable robots: Understanding human teaching behavior to build more effective robot learners," *Artificial Intelligence*, vol. 172, no. 6-7, pp. 716–737, 2008.
[42] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.
[43] W. B. Knox and P. Stone, "Interactively shaping agents via human reinforcement: The TAMER framework," in *Proceedings of the 5th International Conference on Knowledge Capture*, 2009, pp. 9–16.
[44] J. MacGlashan, M. K. Ho, R. Loftin, B. Peng, G. Wang, D. L. Roberts, M. E. Taylor, and M. L. Littman, "Interactive learning from policy-dependent human feedback," in *Proceedings of International Conference on Machine Learning (ICML)*. PMLR, 2017, pp. 2285–2294.
[45] D. Arumugam, J. K. Lee, S. Saskin, and M. L. Littman, "Deep reinforcement learning from policy-dependent human feedback," *arXiv preprint arXiv:1902.04257*, 2019.
[46] B. Ibarz, J. Leike, T. Pohlen, G. Irving, S. Legg, and D. Amodei, "Reward learning from human preferences and demonstrations in atari," in *Advances in Neural Information Processing Systems*, 2018, pp. 8011–8023.
[47] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of International Conference on Machine Learning (ICML)*, 2015, pp. 1889–1897.
[48] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," in *Proceedings of Deep Learning Workshop at International Conference on Neural Information Processing Systems (NIPS)*, 2013.
[49] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of International Conference on Machine Learning (ICML)*, 2018, pp. 1587–1596.
[50] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.