sensors

MDPI

Article

Domain Adaptation for Imitation Learning Using Generative Adversarial Network

Tho Nguyen Duc [1], Chanh Minh Tran [1], Phan Xuan Tan [2,*] and Eiji Kamioka [1]

[1] Graduate School of Engineering and Science, Shibaura Institute of Technology, Tokyo 135-8548, Japan; nb20501@shibaura-it.ac.jp (T.N.D.); nb20502@shibaura-it.ac.jp (C.M.T.); kamioka@shibaura-it.ac.jp (E.K.)
[2] Department of Information and Communications Engineering, Shibaura Institute of Technology, Tokyo 135-8548, Japan
* Correspondences: tanpx@shibaura-it.ac.jp

Abstract: Imitation learning is an effective approach for an autonomous agent to learn control policies when an explicit reward function is unavailable, using demonstrations provided from an expert. However, standard imitation learning methods assume that the agents and the demonstrations provided by the expert are in the same domain configuration. Such an assumption has made the learned policies difficult to apply in another distinct domain. The problem is formalized as domain adaptive imitation learning, which is the process of learning how to perform a task optimally in a learner domain, given demonstrations of the task in a distinct expert domain. We address the problem by proposing a model based on Generative Adversarial Network. The model aims to learn both domain-shared and domain-specific features and utilizes it to find an optimal policy across domains. The experimental results show the effectiveness of our model in a number of tasks ranging from low to complex high-dimensional.

Keywords: imitation learning; domain adaptive imitation learning; generative adversarial network

Citation: Nguyen Duc , T.; Tran, C.M.; Tan, P.X.; Kamioka, E. Domain Adaptation for Imitation Learning Using Generative Adversarial Network. *Sensors* **2021**, *21*, 4718. https://doi.org/10.3390/s21144718

Academic Editors: Anastasios Doulamis, Nikolaos Doulamis and Athanasios Voulodimos

Received: 11 June 2021
Accepted: 7 July 2021
Published: 9 July 2021

1. Introduction

The demand for autonomous agents capable of mimicking human behaviors has grown significantly in recent years. For example, self-driving vehicles, assistive robots, and human–computer interaction fields rely on the ability of agents that can not only make optimal decisions but also behave like humans [1], which can enable the agents' actions to be believable and appear natural. In order for autonomous agents to acquire such human complex behaviors, they are supplied with reward functions indicating the goals of the desired behaviors. However, reward functions can be difficult to be defined manually. In fact, humans can learn complex behaviors from imitation: we observe other experts performing the tasks, infer the tasks, then attempt to accomplish the same tasks ourselves. Inspired by this learning procedure, imitation learning has been widely used for training autonomous agents using expert-provided demonstrations [1–4].

Imitation learning works by extracting information about the behavior of the expert and learning a mapping between the observation state and demonstrated behavior [1,5]. Unfortunately, the traditional imitation learning algorithms are still far from being comparable with the human imitation due to the lack of the following abilities:

1. Humans tend to imitate the goal of a task rather than a particular behavior of the expert [6,7].
2. Humans can recognize structural differences (i.e., domain shift) and similarities between the expert and themselves in order to adapt their behaviors accordingly [8].

The first aspect of human imitation can be modeled using Inverse Reinforcement Learning (IRL) [9,10]. IRL seeks to estimate a reward function to explain an expert behavior from demonstrations and subsequently train an agent on it [9–12]. Recent studies [13–20] utilize Generative Adversarial Network (GAN) [21], which has a discriminator to judge

whether a given behavior is from an expert or agent, and then a policy is trained using the discriminator as a reward. However, these approaches do not take into account the second aspect of human learning: imitation with the presence of domain shift between the expert and the agent. Such domain shift can mislead the feature learning, resulting in poor task performance.

The problem is formalized as domain adaptive imitation learning, which is a process of learning how to perform a task optimally in a learner domain, given demonstrations of the task in a distinct expert domain [14]. In order to solve this problem, the authors in [14] proposed a two-step approach: alignment followed by adaptation. Firstly, the Generative Adversarial MDP Alignment (GAMA) was introduced to learn the state—action maps from demonstrations. Then, in the adaptation step, an optimal policy for the learner domain was obtained using the learned alignment from the first step. Despite showing a promising result, their model was evaluated only in low-dimensional tasks. In addition, they updated the learned policy by using behavioral cloning, which was vulnerable to cascading errors. This could lead to poor adaptation performance in more complex high-dimensional tasks.

Unlike most previous studies in domain adaptive imitation learning, this work proposes a model that aims to learn both domain-shared and domain-specific features. Such features enable the agents to learn optimal policies without being affected by the shift between two domains. The learning procedure can be achieved within one training process by utilizing adversarial training [21]. In summary, the main contributions of this paper are as follows:

- A features extractor, which is capable of deriving domain-shared and domain-specific features, is proposed.
- The DAIL-GAN model is proposed. The model leverages adversarial training [21] to learn the extracted features, while at the same time, seeking for an optimal learner domain policy.
- A comprehensive experiment on both low and high-dimensional tasks is conducted to evaluate the performance of the proposed model.

The rest of this paper is organized as follows. In Section 2, the related works of the proposed model is introduced. Section 3 formulates the domain adaptive imitation learning problem. The details of the proposed DAIL-GAN model is presented in Section 4 and evaluated in Section 5. Section 6 discusses and analyzes the evaluation results. Finally, Section 7 concludes this paper.

2. Related Work

Imitation learning has been a popular method for training autonomous agents from expert demonstrations [1]. A simple approach to imitation learning is Behavioral Cloning (BC) [22], which mimics such demonstrations by learning the policy through supervised learning. Despite being successfully applied in many control problems [2,22,23], BC was found to be vulnerable to cascading errors [24]. On the other hand, Inverse Reinforcement Learning (IRL) [9] methods try to recover a reward function from the expert demonstrations [9–12]. This reward function is then used to optimize an imitation policy by running a standard reinforcement learning [25,26]. Accordingly, IRL has succeeded in a wide range of tasks [27–30]. However, in order to train an IRL model, it requires iterations of reinforcement learning, which can be extremely computationally expensive for high-dimensional tasks. Recently, Generative Adversarial Network [21] has been introduced and successfully employed to tackle complex challenges in image generation, translation, and enhancement [31–34]. Inspired by the great ability of GAN, recent studies [13,15–20] have applied it in imitation learning to define expert behaviors by fitting the distributions of states and actions. These models outperform competing methods when applying to complex high-dimensional tasks over various amounts of expert data.

Unfortunately, the common major weakness of the above-mentioned models is that they require the experts to provide demonstrations in the same configuration and domain as the learners. Thus, the presence of a shift between the expert and learner domains may

lead to a significant performance deterioration of those models. A popular approach is to employ a domain adaptation, which attempts to recover the learned policy from one domain and to adapt it to a different domain. The work in [35] proposed a model to recover domain-agnostic features and utilized it to find optimal policies in the setting of third person imitation, in which the expert and learner observations come from different views. Furthermore, the authors in [14] introduced a two-step approach that could be applied to imitate demonstrations observed from a distinct domain. They proposed to find a state–action mapping between the expert and learner domains. After that, the learned mapping was utilized to adapt the learned policy to the learner domain. Although achieving high performance on low-dimensional tasks, the effectiveness of their methods on more complex high-dimensional tasks was not fully inspected yet.

Our method is different from previous methods [14,35], which aims to learn both domain-shared and domain-specific features in expert and learner domains. These features enable our proposed model to find an optimal learner domain policy that can achieve high performance without being affected by the shift between two domains.

3. Problem Formulation

In this section, we formalize the domain adaptive imitation learning as a Markov decision problem. A Markov Decision Process (MDP) $\mathcal{M}$ with finite time horizon [36] is represented as the following equation:

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}) \tag{1}$$

where $\mathcal{S}$ and $\mathcal{A}$ represent the state and action space, respectively; $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ denotes the transition function, and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function—whereas, a policy $\pi : \mathcal{S} \to \mathcal{A}$ for $\mathcal{M}$ describes a mapping from states $\mathcal{S}$ to actions $\mathcal{A}$. In general reinforcement learning setting, the goal is to find an optimal policy π^* that achieves the highest expected discounted sum of rewards J:

$$\pi^* = \underset{\pi}{argmax} J(\pi), \tag{2}$$

$$\text{subject to } J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t r^t \right] \tag{3}$$

where $\gamma \in (0, 1]$ is the discount factor and $r^t = \mathcal{R}(s^t, a^t)$ is the reward at timestep t.

However, in the domain adaptive imitation learning setting, the reward function is not given beforehand. Therefore, the MDP for a domain x without reward is defined as $\mathcal{M}_x = (\mathcal{S}_x, \mathcal{A}_x, \mathcal{P}_x)$. In this paper, all examined domains are assumed to be alignable. That is, if considering two domains x and y, $\mathcal{M}_x$ can be reduced to $\mathcal{M}_y$, denoted as $\mathcal{M}_x \geq \mathcal{M}_y$, or vice versa [14]. An example is illustrated in Figure 1. Based on this expression, let $\mathcal{E}$ and $\mathcal{L}$ be the expert and the learner domain, respectively, $\mathcal{M}_\mathcal{E}$ and $\mathcal{M}_\mathcal{L}$ are said to be alignable if and only if $\mathcal{M}_\mathcal{E} \geq \mathcal{M}_\mathcal{L}$ or $\mathcal{M}_\mathcal{L} \geq \mathcal{M}_\mathcal{E}$ [14].

Furthermore, $\tau_x = \{(s_x^t, a_x^t) : t \in [0, T]\}$ denotes a demonstration in the domain x, which is a sequence of state–action pairs. Then, a set of demonstrations $\mathcal{D}_\mathcal{E} = \{\tau_\mathcal{E}^i : i \in [1, N]\}$ from $\mathcal{E}$ is assumed to be available at the training time. With those assumptions, our main objective is being able to learn an optimal learner domain policy $\pi_\mathcal{L}^*$ against unknown reward function $\mathcal{R}_\mathcal{L}$, given the expert demonstrations $\mathcal{D}_\mathcal{E}$.

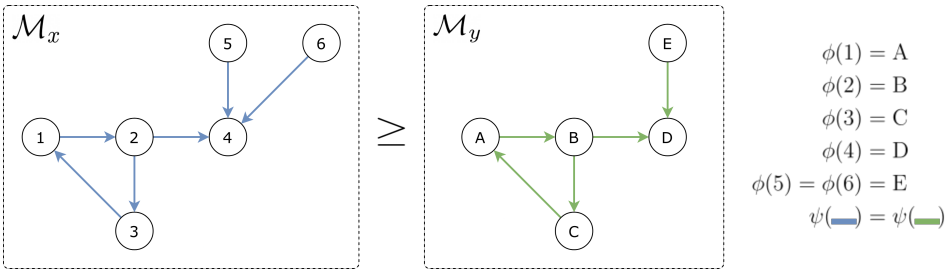


Figure 1. An example of two MDPs for x and y domains where $\mathcal{M}_x \geq \mathcal{M}_y$. $\phi : \mathcal{S}_x \rightarrow \mathcal{S}_y$ and $\psi : \mathcal{A}_x \rightarrow \mathcal{A}_y$ are state and actions maps, respectively. States correspond to nodes and actions to colors. States 5, 6 in $\mathcal{S}_x$ are merged to state e in $\mathcal{S}_y$ and blue actions in $\mathcal{A}_x$ are mapped to green actions in $\mathcal{A}_y$.

4. The Proposed DAIL-GAN Model

In this section, we introduce our proposed DAIL-GAN model. The model relies on learning the domain-shared and domain-specific features in order to recover expert behaviors and adapt them to the learner domain. The architecture of our proposed model is illustrated in Figure 2. The model includes three deep feed-forward networks F, G, and D that holds different responsibilities.

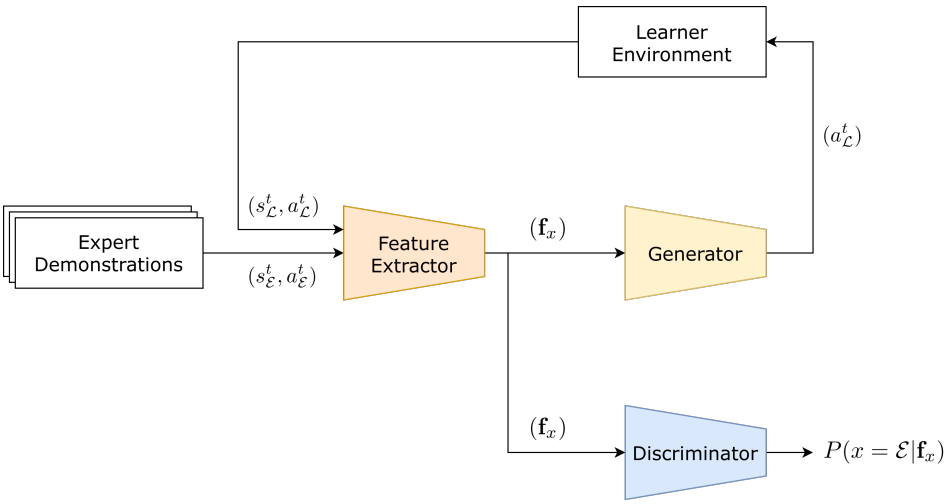


Figure 2. The architecture of the proposed DAIL-GAN model.

4.1. Feature Extractor Network F

A state–action pair (s^t_x, a^t_x) in domain x is input into the feature extractor F to produce a feature vector $\mathbf{f}_x = F(s^t_x, a^t_x)$. F is trained to capture the structural similarities or the shared features between $\mathcal{E}$ and $\mathcal{L}$ domains by minimizing the distance between two features $\mathbf{f}_\mathcal{E}$ and $\mathbf{f}_\mathcal{L}$. Therefore, the loss function of F is defined as:

$$\mathfrak{L}_F(F, G) = \mathbb{E}\left[\left\|F(s^t_\mathcal{E}, a^t_\mathcal{E}) - F(s^t_\mathcal{L}, a^t_\mathcal{L})\right\|\right] \tag{4}$$

$$= \mathbb{E}\left[\left\|F(s^t_\mathcal{E}, a^t_\mathcal{E}) - F(s^t_\mathcal{L}, G(F(s^t_\mathcal{E}, a^t_\mathcal{E})))\right\|\right] \tag{5}$$

4.2. Discriminator Network D and Generator Network G

The discriminator D is designed to distinguish between expert feature vector $\mathbf{f}_\mathcal{E}$ and learner feature vector $\mathbf{f}_\mathcal{L}$. Specifically, D receives a feature vector $\mathbf{f}_x$ and outputs a probability $P(x = \mathcal{E}|\mathbf{f}_x)$ to classify whether $\mathbf{f}_x$ is from $\mathcal{E}$ or $\mathcal{L}$. Meanwhile, the generator G aims to generate an action a^t_L so that $\mathbf{f}_\mathcal{L} = F(s^t_\mathcal{L}, a^t_\mathcal{L})$ looks as similar as possible to $\mathbf{f}_\mathcal{E}$. In the proposed DAIL-GAN model, we apply adversarial loss [21] for both networks:

$$\mathfrak{L}_{GAN}(G, D) = \mathbb{E}[\log D(F(s_{\mathcal{E}}^t, a_{\mathcal{E}}^t))] + \mathbb{E}[\log(1D(F(s_{\mathcal{L}}^t, a_{\mathcal{L}}^t)))] \quad (6)$$

$$= \mathbb{E}[\log D(F(s_{\mathcal{E}}^t, a_{\mathcal{E}}^t))] + \mathbb{E}[\log(1D(F(s_{\mathcal{L}}^t, G(F(s_{\mathcal{E}}^t, a_{\mathcal{E}}^t)))))] \quad (7)$$

The optimal policy is achieved using a RL-based policy gradient, which relies on reward signal $r = -\log D(F(s_{\mathcal{E}}^t, a_{\mathcal{E}}^t))$ provided by the learned discriminator.

4.3. Full Objective

During the learning phase, we aim to learn domain-shared features between $\mathcal{E}$ and $\mathcal{L}$ domains. Thus, the feature extractor F and the generator G are optimized to minimize the feature extractor loss $\mathfrak{L}_F$. At the same time, given a feature vector $\mathbf{f}_x$ of domain x, we want to judge whether $\mathbf{f}_x$ is from $\mathcal{E}$ or $\mathcal{L}$ by minimizing the domain classification loss $\mathfrak{L}_{GAN}$. This encourages domain-specific features to be captured by F. Overall, our full objective function is:

$$\underset{F,G}{max}\underset{D}{min}\mathfrak{L}(F, G, D) \quad (8)$$

$$\text{subject to } \mathfrak{L}(F, G, D) = \mathfrak{L}_{GAN}(G, D) - \lambda\mathfrak{L}_F \quad (9)$$

We wish to find a saddle point, where:

$$(\hat{F}, \hat{G}) = \underset{F,G}{argmax}\mathfrak{L}(F, G, \hat{D}) \quad (10)$$

$$\hat{D} = \underset{D}{argmin}\mathfrak{L}(\hat{F}, \hat{G}, D) \quad (11)$$

At the saddle point, the $\hat{D}$ minimizes the domain classification loss. The feature extractor $\hat{F}$ and the generator $\hat{G}$ minimize the distance between both domains (i.e., the features are shared between domains), while maximizing the domain classification loss (i.e., the features are specific to each domain). The parameter λ controls the trade-off between domain-shared features and domain-specific features should be learned by F.

The algorithm of the proposed model is outlined in Algorithm 1.

Algorithm 1 DAIL-GAN

Input
  $\mathcal{D}_{\mathcal{E}}$ A set of expert demonstrations
Randomly initialize feature extractor network F, generator G and discriminator D
for i = 0, 1, 2, ... **do**
  Sample an expert demonstration $\tau_{\mathcal{E}}^i \sim \mathcal{D}_{\mathcal{E}}$
  Update the parameters of feature extractor network F with the gradient

$$\mathbf{E}[\nabla_F log(D(\mathbf{f}_{\mathcal{E}}))] + \mathbf{E}[\nabla_F log(1 - D(\mathbf{f}_{\mathcal{L}}))] - \lambda\mathbf{E}[\nabla_F \|\mathbf{f}_{\mathcal{E}} - \mathbf{f}_{\mathcal{L}}\|]$$

  Update the discriminator parameters with the gradient

$$\mathbf{E}[\nabla_D log(D(\mathbf{f}_{\mathcal{E}}))] + \mathbf{E}[\nabla_D log(1 - D(\mathbf{f}_{\mathcal{L}}))]$$

  Update policy π_L with the reward signal $r = -logD(\mathbf{f}_{\mathcal{E}})$
end for
Output
  π_L Learned policy for learner domain

5. Performance Evaluation

In this section, the performance of the proposed DAIL-GAN model is evaluated by comparing with various baseline models on a number of tasks ranging from low to complex high-dimensional. The details of the experimental settings and evaluation results are presented in the following subsections.

5.1. Experimental Settings

5.1.1. Environments

In this experiment, five simulated environments were considered: Pendulum [37], Acrobot [37–39], CartPole [37,40], Door [41], and Hammer [41]. The detailed descriptions and visualizations of these environment are shown in Table 1 and Figure 3, respectively. From such environments, five domain adaptive tasks were decided, each of which included two different environments—an expert domain and a learner domain. These tasks can be divided into two categories as follows:

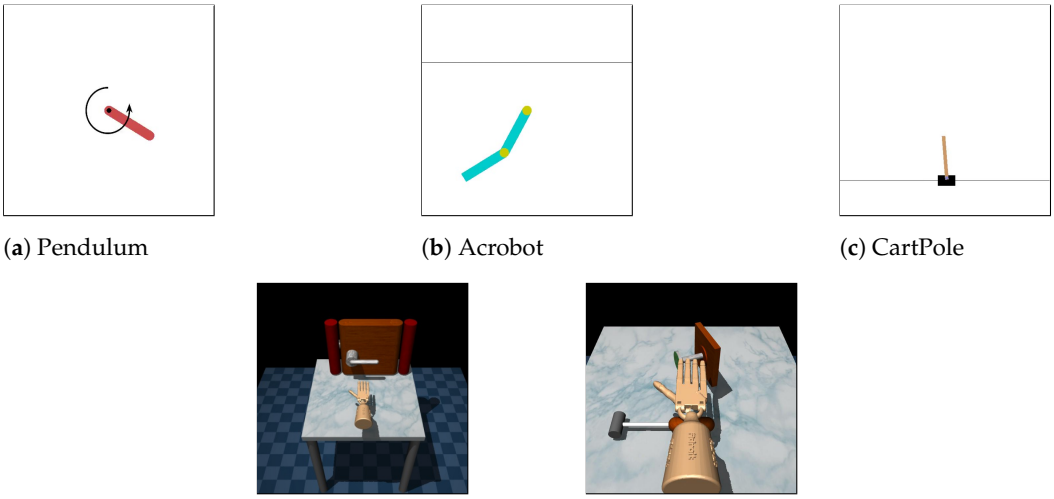
- Low-dimensional tasks:
  - Pendulum–Acrobot: Expert domain is Pendulum and learner domain is Acrobot.
  - Pendulum–CartPole: Expert domain is Pendulum and learner domain is CartPole.
  - Acrobot–CartPole: Expert domain is Acrobot and learner domain is CartPole.

  To provide expert demonstrations, for each task, the Trust Region Policy Optimization method [42] is first trained on the expert domain using the shaped reward signal. Then, 20 expert demonstrations are collected by executing the learned policies in the expert domain simulator. Each demonstration includes a sequence of state–action pairs. It should be noted that we only collect successful demonstrations where the learned policies can accomplish the task. The impacts of demonstrations on the performance of the proposed model will be analyzed in our future work.
- High-dimensional tasks:
  - Door–Door: The expert and learner domains have different friction parameters. The friction parameter in expert domain is $[1, 1, 1]$, while, in the learner domains, it is $[4.0, 4.0, 4.0]$.
  - Hammer–Hammer: The expert and learner domains have different mass of the hammer. The mass of the hammer in expert domain is 0.253442, while, in the learner domain, it is 1.0.

  We also use 20 expert demonstrations for each task. The demonstrations are collected from humans using the Mujoco HAPTIX system [43] and publicly available [41].

Table 1. Description of five simulated environments used in our experiment.

Task	State Space	Action Space	Description
Pendulum [37]	3 (continuous)	1 (continuous)	Swinging up a pendulum.
Acrobot [37–39]	6 (continuous)	3 (discrete)	Swinging the end of the lower link up to a given height
CartPole [37,40]	4 (continuous)	2 (discrete)	Preventing the pendulum from falling over by applying a force to the cart.
Door [41]	39 (continuous)	28 (continuous)	A 24-DoF hand attempts to undo the latch and swing the door open.
Hammer [41]	46 (continuous)	26 (continuous)	A 24-DoF hand attempts to use a hammer to drive the nail into the board.



(**a**) Pendulum (**b**) Acrobot (**c**) CartPole

(**d**) Door (**e**) Hammer

Figure 3. Visual rendering of five simulated environments used in our experiment.

5.1.2. Baselines

The performance of the proposed DAIL-GAN model was evaluated in comparison with the following baseline methods:

- Trust Region Policy Optimization (TRPO) [42] is a Reinforcement learning-based model. The model was trained directly on the learner domain and had access to the shaped reward function. This baseline set an upper bound for the performance of domain adaptation algorithms.
- GAMA-PA [14]: The model introduced a two-step approach for domain adaptation in imitation learning. It first learns the state–action maps between expert and learner domains, and then utilizes it to learn an optimal policy. The model parameters are employed as reported in [14] in order to ensure a fair comparison.

5.1.3. Network Structure and Hyperparameters

Deep feed-forward networks with two hidden layers are used for three F, G, D networks of the proposed model. The network hyperparameters are shown in Table 2. In this experiment, the learning rate was 0.0003. Adam was used as an optimizer.

Table 2. DAIL-GAN hyperparameters used in the experiment. Each number corresponds to the number of nodes in a network layer.

	Feature Extractor F	**Generator G**	**Discriminator D**
Low-dimensional Tasks	(s_x^t, a_x^t) - 32 - 32 - 16	$(\mathbf{f}_x)$ - 32 - 32 - $(a_{\mathcal{L}}^t)$	$(\mathbf{f}_x)$ - 32 - 32 - 1
High-dimensional Tasks	(s_x^t, a_x^t) - 128 - 128 - 64	$(\mathbf{f}_x)$ - 128 - 64 - $(a_{\mathcal{L}}^t)$	$(\mathbf{f}_x)$ - 128 - 64 - 1

5.2. Results

In this subsection, the evaluation results of the proposed DAIL-GAN model on low- and high-dimensional tasks are presented to highlight its superior capability in domain adaptive imitation learning.

5.2.1. Low-Dimensional Tasks

Table 3 reports the quantitative evaluations of the proposed DAIL-GAN model on low-dimensional tasks, in terms of average cumulative rewards. The numerical results clearly indicate that, for all evaluated tasks, TRPO [42] provided the best performance as its average cumulative rewards were at the highest. This was actually predictable because TRPO [42] had direct access to states and shaped rewards of the learner domain. On the other hand, inputs of GAMA-PA [14] and DAIL-GAN were limited to expert demonstrations only. As a result, their performances deteriorated compared to TRPO [42]. However, Table 3 also determines that the proposed DAIL-GAN outperformed GAMA-PA [14] across all three tasks. Additionally, for the Pendulum–Acrobot task, the proposed model almost achieved as high performance as TRPO [42]. In order to understand the observed results more deeply, Figures 4–6 visualize the behaviors of learned policies provided by the evaluated models when performing the Pendulum–Acrobot, Pendulum–CartPole, and Acrobot–CartPole tasks, respectively.

Table 3. The performance of the proposed models on low-dimensional tasks. These scores represent the cumulative rewards obtained from executing a learned policy in the simulator, averaged over 100 episodes.

Task	TRPO [42]	GAMA-PA [14]	DAIL-GAN
Pendulum–Acrobot	-63.18 ± 7.05	-386.31 ± 49.20	-83.31 ± 32.61
Pendulum–CartPole	497.13 ± 28.56	144.03 ± 89.09	289.74 ± 171.21
Acrobot–CartPole	497.13 ± 28.56	93.05 ± 88.97	153.86 ± 81.79

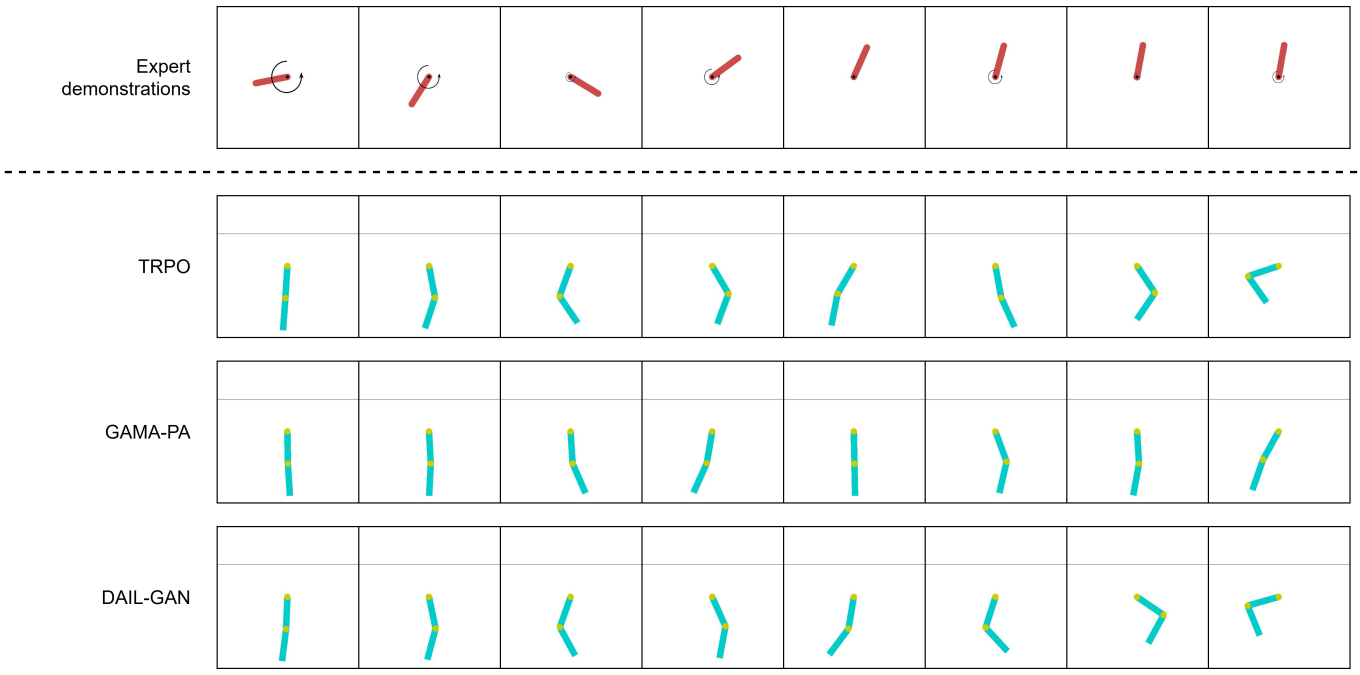


Figure 4. Pendulum–Acrobot.

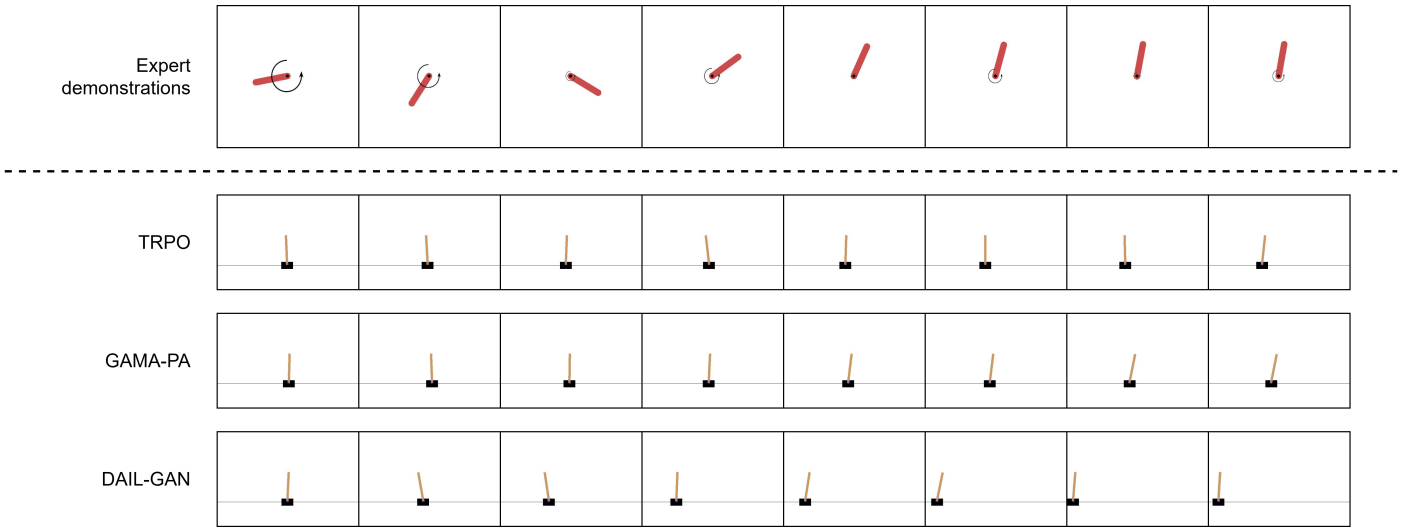


Figure 5. Pendulum–CartPole.

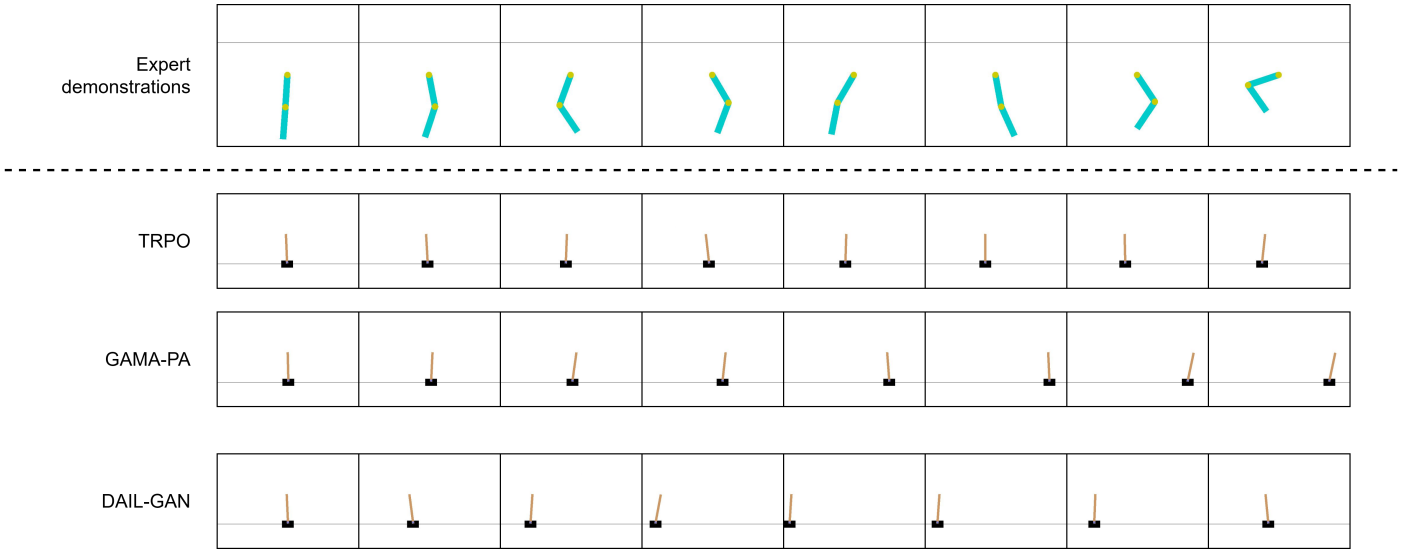


Figure 6. Acrobot–CartPole.

In the expert demonstration of the Pendulum–Acrobot task in Figure 4 and the Pendulum-CartPole task in Figure 5, expert behaviors were to apply a strong force, expressed by a rotation velocity, at first to make the pendulum swing upright. After that, a few light forces were applied to maintain it vertically. Observing from Figure 4, the policies trained with GAMA-PA [14] failed to apply strong enough forces to swing the lower link as high as the proposed DAIL-GAN. In addition, Figure 5 expresses that the GAMA-PA [14] could not move the cart at an appropriate velocity to keep the pole vertical. We speculate that it was because the expert demonstration also did not show much movement after successfully swinging the pendulum upright as it only applied light forces. Meanwhile, the policies learned by our DAIL-GAN model could accomplish the task. Interestingly, we observed that the learned policies are able to produce behaviors that are relatively similar to the expert: the cart was first pushed to the left by a strong force; then, small forces are applied to prevent the pole from falling over.

For the Acrobot–CartPole task in Figure 6, the behaviors of the expert were that the link was swung back and forth to gain enough velocity to reach a higher height. Similarly, the GAMA-PA's learned policy could move the cart faster compared to the Pendulum–CartPole task. However, it still failed to maintain appropriate velocity to keep the pole standing. On the contrary, our DAIL-GAN was able to remain the pole vertical.

It is important to note that the learned policy of our model could move the cart in both directions, which is also similar to the expert behaviors.

The above observations show that the proposed DAIL-GAN model not only succeeded in imitating the expert behaviors but also adapted the learned policies well to a distinct learner domain. Meanwhile, although the GAMA-PA [14] could learn the state–action maps from expert to learner domain, its adaptation algorithm was inefficient to help it accomplish the tasks.

5.2.2. High-Dimensional Tasks

In this subsection, the performance of the proposed DAIL-GAN versus the referenced models on the high-dimensional task is assessed. The average cumulative rewards of the evaluated models are shown in Table 4. As expected, the TRPO model achieved the highest average cumulative reward since it was trained directly on the learner domain. It is also revealed that DAIL-GAN outperformed GAMA-PA, although they were both unable to accomplish the Door–Door task. In addition, Figures 7 and 8 depict the policies learned by TRPO [42], GAMA-PA [14], and our DAIL-GAN model, from which we observed some interesting behaviors.

Table 4. The performance of the proposed models on high-dimensional tasks. These scores represent the cumulative rewards obtained from executing a learned policy in the simulator, averaged over 100 episodes.

Task	TRPO [42]	GAMA-PA [14]	DAIL-GAN
Door–Door	2449.06 ± 1175.25	−65.19 ± 0.77	−33.51 ± 8.87
Hammer–Hammer	17,030.25 ± 4357.23	−252.52 ± 4.91	−78.84 ± 19.28

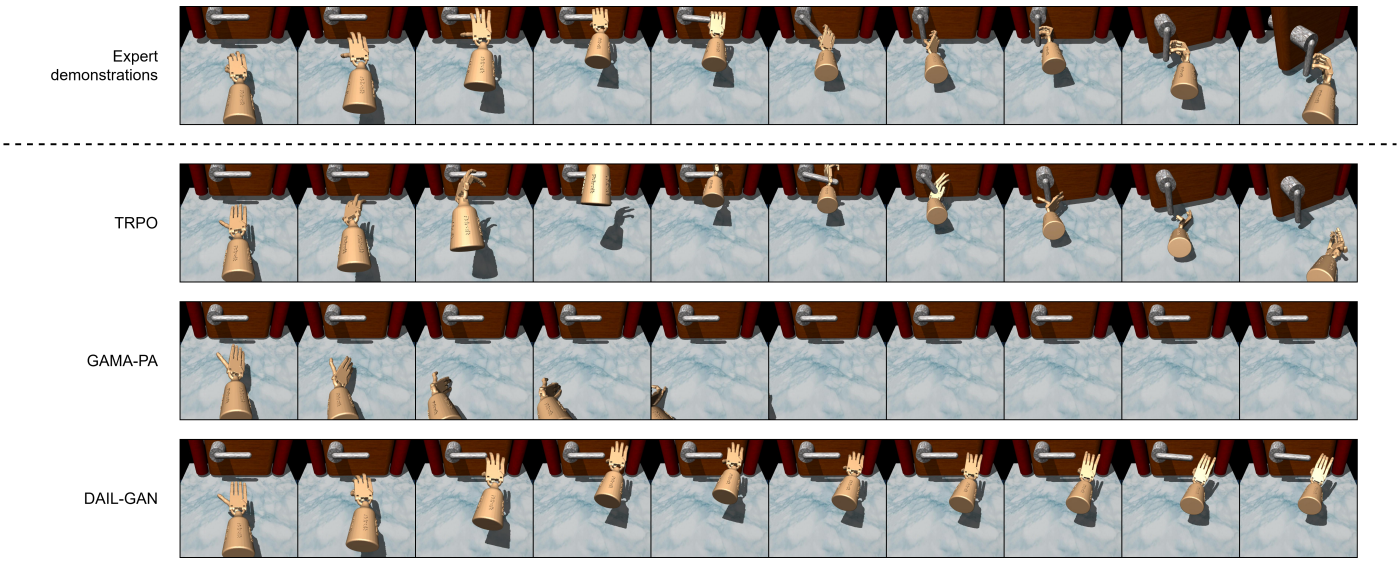


Figure 7. Door–Door.

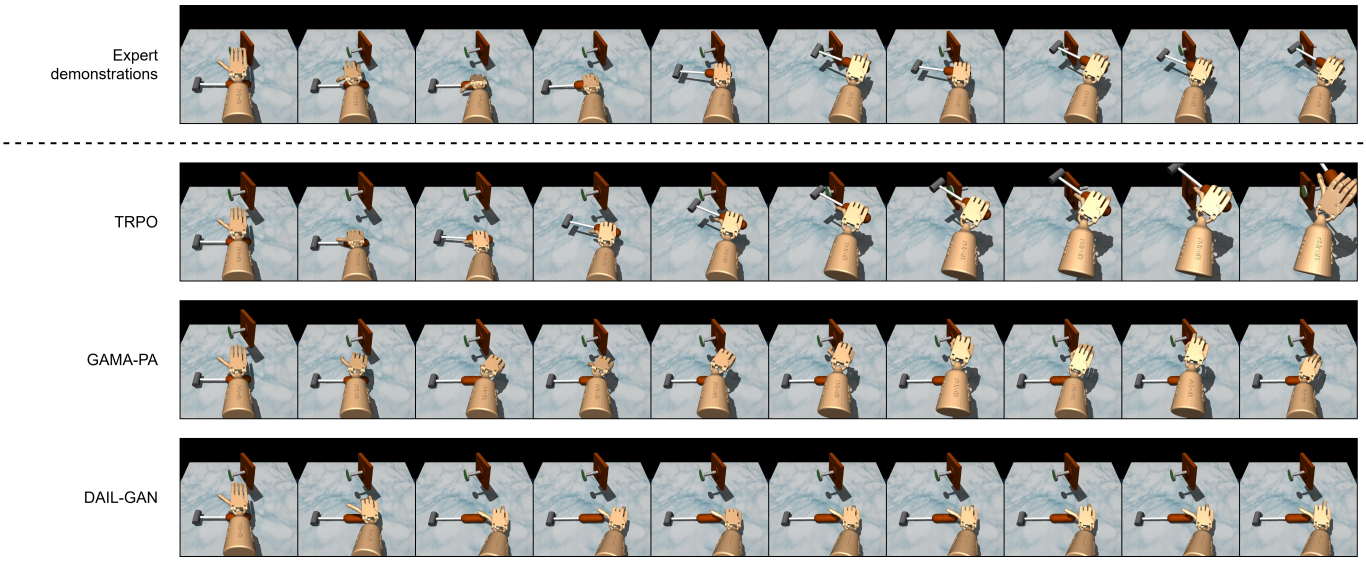


Figure 8. Hammer–Hammer.

As illustrated in Figure 7, the expert behaviors were understandable since their demonstrations were collected from humans: grab the handle, rotate it, then open the door. In Figure 8, the expert behaviors were to pick up and hammer multiple times in order to drive the nail into the board. While the policy trained with the TRPO could accomplish the task, it produced behaviors that were not human-like, i.e., unnatural use of the wrist to rotate the handle. The main reason behind these unnatural behaviors was that the TRPO depended on a careful reward shaping, and it was challenging to formalize human-like behaviors into a mathematical reward function. On the other hand, with the use of expert demonstrations, the GAMA-PA and the proposed DAIL-GAN were expected to generate human-like behaviors. However, the policy learned by GAMA-PA failed to control the hand properly, as shown in Figures 7 and 8, due to the failure of the adaptation step in a high-dimensional task. Meanwhile, it can be observed from Figures 7 and 8 that the policy trained with DAIL-GAN could produce more natural and human-like behaviors to move the robot hand closer to the door handle or the hammer. Unfortunately, our DAIL-GAN model could not rotate the handle or pick up the hammer in order to accomplish the task. Nevertheless, the human-like behaviors of the trained policies proved that our model could effectively extract and imitate expert behaviors from their demonstrations.

6. Discussion

This section discusses the overall performance of the proposed DAIL-GAN model, followed by the importance of the feature extractor.

The quantitative and qualitative results assessed from the previous section have shown the potential of the proposed DAIL-GAN model in tackling the domain adaptation problem in imitation learning. On both low- and high-dimensional tasks, DAIL-GAN could imitate expert behaviors from their demonstrations. In particular, the policies acquired by DAIL-GAN could even generate natural and human-like behaviors despite the high complexity of the Door–Door and Hammer–Hammer tasks. This indicates that the proposed DAIL-GAN could scale up to a complex manipulation task with a high-dimensional state and action space. Furthermore, the proposed model could adapt the learned policies to a distinct learner domain and accomplish low-dimensional tasks without being affected by the presence of domain shift between expert and learner domains. Although the success rate remained limited and depended on the complexity of the tasks, the proposed model can be improved to provide a better performance toward practical real-world imitation learning tasks.

The promising performance of the proposed DAIL-GAN also praises the effectiveness of the proposed feature extractor F. The feature extractor aims to learn both domain-shared

and domain-specific features between expert and learner domains. In Figure 5, the learned policy tended to move the cart to the left by a strong force initially, then followed by small forces; this behavior was similar to that of the expert demonstration. Such a similarity indicated that the feature extractor could extract the structural similarities or domain-shared features between expert and learner domain, resulting in comparable behaviors between them. Furthermore, it can also be observed in Figure 5 that, although strong forces were applied, the learned policies still managed to keep the pole stay upright. This showed that the feature extractor was able to learn the differences between the expert and learner domains so that it could adapt the learned policies to the learner domain and accomplish the task. In summary, the feature extractor has proven its important role in our model. It could acquire shareable behaviors in both domains by learning the domain-shared features and adapting those behaviors to the learner domain regardless of the domain shift by learning the domain-specific features.

7. Conclusions

In this paper, we proposed a novel model for domain adaptive imitation learning, in which a feature extractor was introduced to learn the domain-shared and domain-specific features. The comprehensive evaluation on both low and high-dimensional tasks demonstrates that the policies learned by the proposed model can imitate expert behaviors and adapt them to a distinct learner domain. Thus, the potential of our proposed model and the effectiveness of the feature extractor were verified. In future work, we intend to extend the proposed model to improve its performance on more complex real-world imitation tasks.

Author Contributions: Conceptualization, T.N.D., C.M.T., and P.X.T.; methodology, T.N.D., C.M.T., and P.X.T.; software, T.N.D.; validation, T.N.D., P.X.T., and E.K.; writing—original draft preparation, T.N.D. and C.M.T.; writing—review and editing, P.X.T. and E.K.; visualization, T.N.D.; supervision, P.X.T. and E.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [https://sites.google.com/view/d4rl/home].

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RL	Reinforcement Learning
IRL	Inverse Reinforcement Learning
GAN	Generative Adversarial Network
BC	Behavior Cloning
MDP	Markov Decision Process

References

1. Argall, B.D.; Chernova, S.; Veloso, M.; Browning, B. A survey of robot learning from demonstration. *Robot. Auton. Syst.* **2009**, *57*, 469–483, doi:10.1016/j.robot.2008.10.024.
2. Duan, Y.; Andrychowicz, M.; Stadie, B.C.; Ho, J.; Schneider, J.; Sutskever, I.; Abbeel, P.; Zaremba, W. One-shot imitation learning. *arXiv* **2017**, arXiv:1703.07326.
3. Sermanet, P.; Lynch, C.; Chebotar, Y.; Hsu, J.; Jang, E.; Schaal, S.; Levine, S.; Brain, G. Time-Contrastive Networks: Self-Supervised Learning from Video. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; Institute of Electrical and Electronics Engineers Inc.: Brisbane, Australia, 2018; pp. 1134–1141, doi:10.1109/ICRA.2018.8462891.

4. Liu, Y.; Gupta, A.; Abbeel, P.; Levine, S. Imitation from Observation: Learning to Imitate Behaviors from Raw Video via Context Translation. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; Institute of Electrical and Electronics Engineers Inc.: Brisbane, Australia, 2018; pp. 1118–1125, doi:10.1109/ICRA.2018.8462901.
5. Schaal, S. Is imitation learning the route to humanoid robots? *Trends Cogn. Sci.* **1999**, *3*, 233–242.
6. Baker, C.L.; Tenenbaum, J.B.; Saxe, R.R. Goal inference as inverse planning. In Proceedings of the Annual Meeting of the Cognitive Science Society, Nashville, USA, 1-4 August 2007; Volume 29.
7. Bao, Y.; Cuijpers, R.H. On the imitation of goal directed movements of a humanoid robot. *Int. J. Soc. Robot.* **2017**, *9*, 691–703.
8. Tomov, M.S.; Schulz, E.; Gershman, S.J. Multi-task reinforcement learning in humans. *Nat. Hum. Behav.* **2021**, *5*, 764–773.
9. Ng, A.Y.; Russell, S.J.; Algorithms for inverse reinforcement learning. In Proceedings of the ICML, Stanford, USA, 29 June–02 July 2000; Volume 1, p. 2.
10. Abbeel, P.; Ng, A.Y. Apprenticeship learning via inverse reinforcement learning. In Proceedings of the Twenty-First International Conference on MACHINE Learning, Alberta, Canada, 4-8 July 2004; p. 1.
11. Levine, S.; Popovic, Z.; Koltun, V. Nonlinear inverse reinforcement learning with gaussian processes. *Adv. Neural Inf. Process. Syst.* **2011**, *24*, 19–27.
12. Ziebart, B.D.; Maas, A.L.; Bagnell, J.A.; Dey, A.K. Maximum entropy inverse reinforcement learning. In Proceedings of the AAAI, Chicago, IL, USA, 13-17 July 2008; Volume 8, pp. 1433–1438.
13. Ho, J.; Ermon, S. Generative Adversarial Imitation Learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 4565–4573.
14. Kim, K.; Gu, Y.; Song, J.; Zhao, S.; Ermon, S. Domain Adaptive Imitation Learning. In Proceedings of the 37th International Conference on Machine Learning, PMLR, 13-18 July 2020; Volume 119, pp. 5286–5295.
15. Baram, N.; Anschel, O.; Caspi, I.; Mannor, S. End-to-End Differentiable Adversarial Imitation Learning. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6-11 August 2017; Precup, D., Teh, Y.W., Eds.; PMLR: Sydney, Australia, 2017; Volume 70, pp. 390–399.
16. Behbahani, F.; Shiarlis, K.; Chen, X.; Kurin, V.; Kasewa, S.; Stirbu, C.; Gomes, J.; Paul, S.; Oliehoek, F.A.; Messias, J.; others. Learning from demonstration in the wild. In Proceedings of the IEEE 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 775–781.
17. Li, Y.; Song, J.; Ermon, S. InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Long Beach, CA, USA, 4-9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 3815–3825.
18. Zhang, X.; Li, Y.; Zhou, X.; Luo, J. cGAIL: Conditional Generative Adversarial Imitation Learning—An Application in Taxi Drivers' Strategy Learning. *IEEE Trans. Big Data* **2020**, doi:10.1109/TBDATA.2020.3039810.
19. Chi, W.; Dagnino, G.; Kwok, T.M.; Nguyen, A.; Kundrat, D.; Abdelaziz, M.E.; Riga, C.; Bicknell, C.; Yang, G.Z. Collaborative robot-assisted endovascular catheterization with generative adversarial imitation learning. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 2414–2420.
20. Zhou, Y.; Fu, R.; Wang, C.; Zhang, R. Modeling Car-Following Behaviors and Driving Styles with Generative Adversarial Imitation Learning. *Sensors* **2020**, *20*, 5034.
21. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *arXiv* **2014**, arXiv:1406.2661.
22. Pomerleau, D.A. *Alvinn: An Autonomous Land Vehicle in a Neural Network*; Technical Report; Carnegie-Mellon Univ Pittsburgh pa Artificial Intelligence and Psychology: Pittsburgh, PA, USA 1989.
23. Rahmatizadeh, R.; Abolghasemi, P.; Bölöni, L.; Levine, S. Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 3758–3765.
24. Ross, S.; Gordon, G.; Bagnell, D. 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, JMLR Workshop and Conference Proceedings, Fort Lauderdale, FL, USA, 11-13 April 2011; pp. 627–635.
25. Finn, C.; Levine, S.; Abbeel, P. Guided cost learning: Deep inverse optimal control via policy optimization. In Proceedings of the International Conference on Machine Learning, PMLR, NY, USA, 20-22 June 2016; pp. 49–58.
26. Kalakrishnan, M.; Pastor, P.; Righetti, L.; Schaal, S. Learning objective functions for manipulation. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 1331–1336.
27. Arora, S.; Doshi, P. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artif. Intell.* **2021**, *297*, 103500.
28. Naumann, M.; Sun, L.; Zhan, W.; Tomizuka, M. Analyzing the Suitability of Cost Functions for Explaining and Imitating Human Driving Behavior based on Inverse Reinforcement Learning. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 5481–5487.
29. Bing, Z.; Lemke, C.; Cheng, L.; Huang, K.; Knoll, A. Energy-efficient and damage-recovery slithering gait design for a snake-like robot based on reinforcement learning and inverse reinforcement learning. *Neural Netw.* **2020**, *129*, 323–333.
30. Zelinsky, G.J.; Chen, Y.; Ahn, S.; Adeli, H.; Yang, Z.; Huang, L.; Samaras, D.; Hoai, M. Predicting Goal-directed Attention Control Using Inverse-Reinforcement Learning. *arXiv* **2020**, arXiv:2001.11921.

31. KWON, H.; KIM, Y.; YOON, H.; CHOI, D. CAPTCHA Image Generation Systems Using Generative Adversarial Networks. *IEICE Trans. Inf. Syst.* **2018**, 543–546, doi:10.1587/transinf.2017EDL8175.
32. Qi, M.; Li, Y.; Wu, A.; Jia, Q.; Li, B.; Sun, W.; Dai, Z.; Lu, X.; Zhou, L.; Deng, X.; others. Multi-sequence MR image-based synthetic CT generation using a generative adversarial network for head and neck MRI-only radiotherapy. *Med. Phys.* **2020**, *47*, 1880–1894.
33. Song, J.; He, T.; Gao, L.; Xu, X.; Hanjalic, A.; Shen, H.T. Unified binary generative adversarial network for image retrieval and compression. *Int. J. Comput. Vis.* **2020**, *128*, 2243–2264.
34. Chen, H.; Wang, Y.; Shu, H.; Wen, C.; Xu, C.; Shi, B.; Xu, C.; Xu, C. Distilling portable generative adversarial networks for image translation. In Proceedings of the AAAI Conference on Artificial Intelligence, NY, USA, 7-12 February 2020; Volume 34, pp. 3585–3592.
35. Stadie, B.C.; Abbeel, P.; Sutskever, I. Third-Person Imitation Learning. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24-26 April 2017.
36. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
37. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. Openai gym. *arXiv* **2016**, arXiv:1606.01540.
38. Sutton, R.S. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 1996; pp. 1038–1044.
39. Geramifard, A.; Dann, C.; Klein, R.H.; Dabney, W.; How, J.P. RLPy: A value-function-based reinforcement learning framework for education and research. *J. Mach. Learn. Res.* **2015**, *16*, 1573–1578.
40. Barto, A.G.; Sutton, R.S.; Anderson, C.W. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Trans. Syst. Man Cybern.* **1983**, *5*, 834–846.
41. Rajeswaran, A.; Kumar, V.; Gupta, A.; Vezzani, G.; Schulman, J.; Todorov, E.; Levine, S. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In Proceedings of the Robotics: Science and Systems (RSS), Pittsburgh, PA, USA, 26-30 June 2018.
42. Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; Moritz, P. Trust Region Policy Optimization. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 7-9 July 2015; pp. 1889–1897.
43. Kumar, V.; Todorov, E. Mujoco haptix: A virtual reality system for hand manipulation. In Proceedings of the 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), Seoul, South Korea, 3-5 November 2015; pp. 657–663.