Imitation Learning: Progress, Taxonomies and Challenges

BOYUAN ZHENG, SUNNY VERMA, JIANLONG ZHOU, IVOR TSANG, and FANG CHEN, University of Technology Sydney, Autralia

Imitation learning aims to extract knowledge from human experts' demonstrations or artificially created agents in order to replicate their behaviours. Its success has been demonstrated in areas such as video games, autonomous driving, robotic simulations and object manipulation. However, this replicating process could be problematic, such as the performance is highly dependent on the demonstration quality, and most trained agents are limited to perform well in task-specific environments. In this survey, we provide a systematic review on imitation learning. We first introduce the background knowledge from development history and preliminaries, followed by presenting different taxonomies within Imitation Learning and key milestones of the field. We then detail challenges in learning strategies and present research opportunities with learning policy from suboptimal demonstration, voice instructions and other associated optimization schemes.

Additional Key Words and Phrases: datasets, neural networks, gaze detection, text tagging

Reference Format:
Boyuan Zheng, Sunny Verma, Jianlong Zhou, Ivor Tsang, and Fang Chen. 2022. Imitation Learning: Progress, Taxonomies and Challenges. (October 2022), 21 pages.

1 INTRODUCTION

Imitation learning (IL), also known as learning from demonstration, makes responses by mimicking behavior in a relatively simple approach. It extracts useful knowledge to reproduce the behavior in the environment which is similar to the demonstrations'. The presence of IL facilitates the research on autonomous control system and designing artificially intelligent agents, as it demonstrates good promise in real-world scenario and efficiency to train a policy. Recent developments in machine learning field like deep learning, online learning and Generative Adversarial Network (GAN) [23] make further improvement on IL, not only alleviating existing problems like dynamic environment, frequent inquiries and high-dimensional computation, but also achieving faster convergence, more robust to the noise and more sample-efficient learning process. These improvements of IL promote the applications in both continuous and discrete control domains. For example, in the continuous control domain, imitation learning could be applied to autonomous vehicle manipulation to reproduce appropriate driving behavior in a dynamic environment[11, 13, 14, 22, 31, 52, 53, 80]. In addition, imitation learning is also applied to robotic, ranging from basic grabbing and placing to surgical assistance[21, 37, 43, 46, 48, 49, 67, 79]. In the discrete control domain, imitation learning makes contribution to fields like game theory[5, 19, 24, 55], navigation tasks[28, 62, 76], cache management[38] and so on.

It is worth noting that the demonstrations could be gathered either from human experts or artificial agents. In most cases, the demonstration is collected from human experts, but there are also some studies that obtain the demonstration through another artificial agent. For example, Chen et al.[13] proposed a teacher-student training structure, they train a teacher agent with additional information and use this trained agent to teach a student agent without additional information. This process is not redundant, using the demonstration from other agent benefits the training process as student agents can rollout their own policy by frequently querying trained agents and learn policies from similar configurations while classic IL needs to overcome the kinematic shifting problem.

Authors' address: Boyuan Zheng, 14055661@student.uts.edu.au; Sunny Verma; Jianlong Zhou; Ivor Tsang; Fang Chen, University of Technology Sydney, PO Box 123, Sydney, New South Wales, Autralia, 2007.

Published in The *IEEE Transactions on Neural Networks and Learning Systems*.

IL has a close relationship with Reinforcement Learning (RL). Both IL and RL commonly solve the problem under Markov Decision Process, and improvements like TRPO[60] in RL could benefit IL as well, but they reproduce the behavior in a different manner. In comparing to RL, IL is more *efficient*, *accessible*, and *human-interactive*. In terms of *efficiency*, comparing with trial and error, the IL agents usually spend less time to produce the desired behavior by using the demonstrations as guidance. In terms of *accessibility*, achieving autonomous behavior in the RL approach requires human experts who are familiar with the problem setting, together with hard-coded reward functions which could be impractical and non-intuitive in some settings. For example, people learn to swim and walk almost from demonstration instead of math functions, and it is hard to formulate these behavior mathematically. IL also prompts interdisciplinary integration, experts who are novice to programming can contribute to the design and evaluating paradigms. In terms of *human-interaction*, IL highlights human's influence through providing demonstration or preference to accelerate the learning process, which efficiently leverages and transfers the experts' knowledge. Although IL presents the above merits, it also faces challenges and opportunities, and this content will be detailed in the following sections.

This survey is organized as follows:

- **Systematic review** This survey presents research in imitation learning under categories *behavioural cloning vs. inverse reinforcement learning* and *model-free vs. model-based*. It then summarizes IL research into two new categories namely *low-level tasks vs. high-level tasks* and *BC vs. IRL vs. Adversarial Structured IL*, which are more adapted to the development of IL.
- **Background knowledge** A comprehensive description of IL's evolution is presented in Section 2, followed by fundamental knowledge in Section 3 and the most common learning framework in Sections 5.
- **Future direction** This survey presents the remaining challenges of IL, like learning diverse behavior, leveraging various demonstration and better representation. Then we discuss the future directions with respect to methods like transfer learning and importance sampling.

2 BACKGROUND

One of the earliest well-known research on IL is the Autonomous Land Vehicle In a Neural Network (ALVINN) project at Carnegie Mellon University proposed by Pomerleau[52]. In 1998, a formal definition of Inverse Reinforcement Learning (IRL) was proposed by Russell[58]. Inverse reinforcement learning aims to recover reward function from demonstrations. A year after, a formal definition of another important category – Behavioural Cloning (BC) was proposed in[6]. BC works in a supervised learning fashion and seeks to learn a policy that builds a direct mapping between states and actions, then output a control strategy for control tasks. Although BC demonstrates significant advantage in efficiency, it also suffers from various problems. In 2010, SMiLe[55] was proposed, it mixed a new policy $\hat{\pi}^{n+1}$ with a fixed probability α as next policy, this method promotes the development of IL and set up the foundation for the later proposed DAgger[57]. DAgger was proposed by Ross et al. and it updates the dataset in each iteration and trains a new policy in the subsequent iteration based on the updated dataset. Compared with previous methods like SMILe [55] and SEARN [16], DAgger alleviates the problem on the unseen scenario and achieve data-efficiency. Later research like[38, 56, 67] were proposed to make improvements on DAgger. Besides DAgger and its derivatives, other BC methods also make contribution to the development of IL like MMD-IL[32], LOLS[12]. As for applications, one of the notable applications of BC was proposed by Abbeel et al.[1], a model-free BC method on autonomous helicopter project, developed an open-loop iterative learning control. Another famous BC application was an autonomous surgical knot-tying robotic proposed by Osa et al.[49], which achieved online trajectory planning

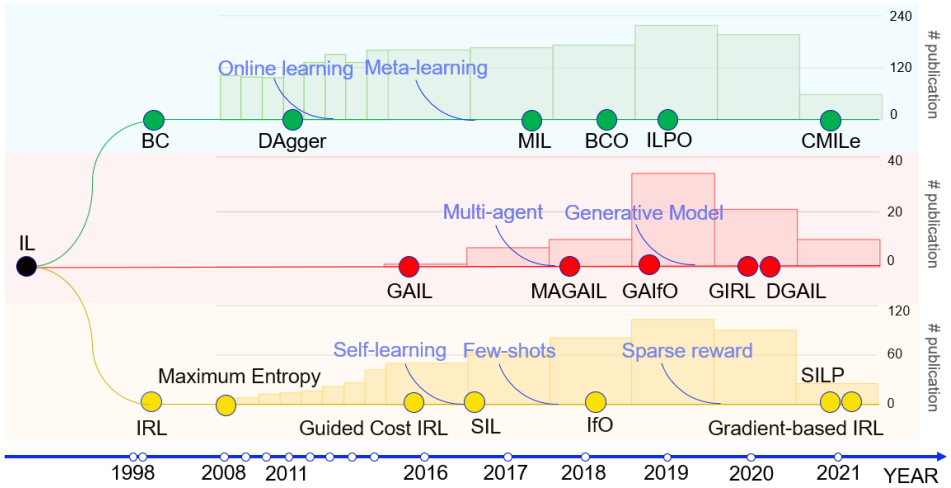

Fig. 1. Featured approaches and annual publication numbers for each class of approaches. The blue text indicates some of the most active research topics in IL and the background histogram plot is the number of annual publications. The data was collected from Web of Science until 31 May 2021, filtered by setting up each class and their abbreviation as keywords (like "Behavioural Cloning OR BC", only cover records within computer science).

and updating in a dynamic system. Besides these real-world applications, BC was also implemented into other research fields like cybernetics, for example, DAgger was used for scheduling in [75] and Liu et al. leveraged Belagy's optimal policy (proof-of-concept) as oracle to solve the cache replacement problem by predicting reuse distance when cache miss happens[38].

In terms of IRL, Ziebart et al.[82] proposed Maximum Entropy IRL, which uses maximum entropy distribution to develop a convex procedure for good promise and efficient optimization. This method played a pivotal role in the development of subsequent IRL and GAIL. In 2016, Finn et al.[21] made significant contributions to IRL and proposed a model-based IRL method called guided cost learning, neural network is used for representing cost to enhance expressive power, combining with sample-based IRL to handle the unknown dynamics. Later in 2017, Hester et al. proposed DQfD[24] which uses small amount of demonstration to significantly accelerate the training process by doing pre-training to kick-off and learning from both demonstration and self-generated data. Later methods like T-REX[9], SQIL[54], SILP[41] make improvements on IRL from different aspects.

Another novel method called Generative Adversarial Imitation Learning (GAIL), it was proposed in 2016 by Ho and Ermon[25] and became one of the hot topics in IL. Later research like[17, 33, 65, 76] were proposed inspired by GAIL and other generative models were gradually adopted in IL. Besides GAIL, another important research direction is inspired by Stadie et al.[65]. Since first-person demonstrations are hard to obtain in practice, and people usually learn by observing the demonstration of others through the perspective of a third party, learning from third-person viewpoint demonstrations was proposed. The change of viewpoint facilitates the following research like[9, 19], which includes IfO[40]. IfO focus on simplifying input to use raw video only (i.e. no longer use state-action pairs), many following methods advocate this new setting. These methods measure the distance between observations to replace the need for ground-truth actions and widen the available input for training, foe example, using YouTube videos for training[5]. Other interesting research fields like meta-learning[18, 20, 27], multi-agent learning[78] are also thrived because of

the development of IL. Figure 1 shows some featured approaches and annual publication numbers for each class and focuses on the research after 2016, it shows that the class of BC(Behavioural Cloning) has maintained a stable increment in publications, while the research in the class of Adversarial Structured IL and IRL(Inverse Reinforcement Learning) have grown rapidly due to the recent advance in other research fields like deep learning.

3 PRELIMINARY KNOWLEDGE

This section provides some basic concepts for better understanding of the IL methodology.

In IL, the demonstrated trajectories are commonly represented as pairs of states s and actions a, sometimes other parameters such as high-level commands and conditional goals will also be included to form the dataset. The way to collect the dataset could be either online or offline. Offline IL prepares the dataset in advance and obtains policies from the dataset while involves fewer interactions with the environment. This could be beneficial when interacting with the environment is expensive or risky. Contrary to offline learning, online learning assumes the data would be accessible in sequence and uses this updated data to learn the best predictor for future data. This method facilitates imitation learning to be more robust in a dynamic system. For example, in[48, 49, 57], online learning is used in surgical robotics. The online learning agent will provide a policy in iteration n, then the opponent will choose a loss function l_n based on current policy and the new observed loss will affect the choice of next iteration $n + 1$'s policy. The performance is measured through regret, i.e.

$$\sum_{n=1}^{N} l_n(\pi_n) - \min_{\pi \in \Pi} \sum_{n=1}^{N} l_n(\pi),$$

and the loss function could vary from iteration to iteration. One of the most common ways to calculate loss is Kullback-Leibler (KL) Divergence. KL Divergence measures the difference between 2 probability distribution, i.e.,

$$D_{KL}(p(x) \parallel q(x)) = \int p(x) \ln \frac{p(x)}{q(x)} dx.$$

KL divergence is not symmetric, i.e., $D_{KL}(p(x) \parallel q(x)) \neq D_{KL}(q(x) \parallel p(x))$. Many algorithms such as[8, 60] use KL divergence as the loss function as it could be useful when dealing with the stochastic policy learning problem.

For many methods, especially those under the class of IRL and Adversarial structured IL, the environment is modeled as Markov Decision Process(MDP). MDP is the process satisfying the property that the next state s_{t+1} only depends on the current state s_t at any time t. Typically, a MDP is defined as a tuple ($\mathcal{S}$,$\mathcal{A}$,$\mathcal{P}$,γ,$\mathcal{D}$,$\mathcal{R}$), where $\mathcal{S}$ is the finite set of states, $\mathcal{A}$ is the corresponding set of actions, $\mathcal{P}$ is the set of state transition probabilities and the successor states s_{t+1} is drawn from this transition model, i.e. $s_{t+1} = P(\cdot|s_t, a_t)$, $\gamma \in [1, 0)$ is the discount factor, $\mathcal{D}$ is the set of initial state distribution and $\mathcal{R}$ is the reward function $\mathcal{S} \mapsto \mathbb{R}$, and in IL setting, the reward function is not available. The Markov property assists imitation learning to simplify the input since the earlier state is helpless to determine the next state. The use of MDP inspires research to make use of other MDP variants to solve various problems, for example, Partially Observable MDP is used to model the scheduling problem in [75] and Markov games is used in multi-agent scenario[63].

The learning process of IL could be either on-policy or off-policy (there exists research using a hierarchical combination of these two[13]). On-policy learning estimates the return and updates the action using the same policy, the agent adopting on-policy will pick actions by themselves and rollout their own policy while training; Off-policy learning estimates the return and chooses the action using different policy, the agent adopting off-policy will update their policy greedily and

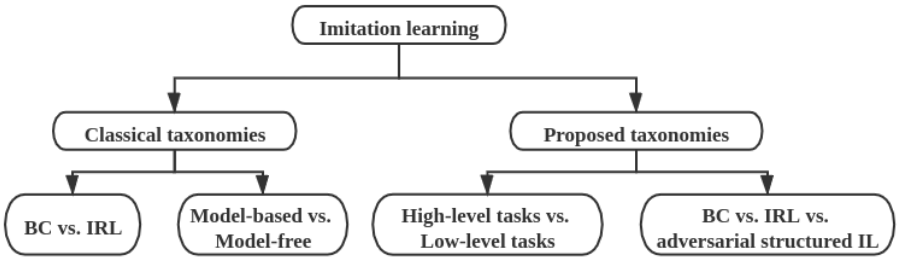

Fig. 2. Taxonomies in this review.

Table 1. Categorization of IL: BC vs. IRL

Classes	Examples and Publications
Behavioural Cloning	Few-shots learning[18]
	Input optimization[13]
	Latent policy learning[42]
	Real-world application[79]
Inverse Reinforcement Learning	Improving efficiency[9]
	Raw video as inputs[61]
	Adversarial structured[66]
	Sparse reward problem[44]

imitate action with the help of other sources. Some recent IL research such as [84**? ?**] advocates off-policy actor-critic architecture to optimize the agent policy and achieve sample efficiency comparing with on-policy learning.

4 CATEGORIZATION AND FRAMEWORKS

In this section, four kinds of taxonomies are presented (see Figure 2). The first two taxonomies (BC vs. IRL and model-free vs. model-based) follow the classifications in[47, 72] and the other two (Low=level Manipulation Tasks vs. High-Level Tasks and BC vs. IRL vs. adversarial structured IL are new proposed taxonomies.

4.1 Behavioural Cloning vs. Inverse Reinforcement Learning

IL is conventionally divided into BC and IRL. These two classes flourish by combining various techniques and then extend into different domains. Generally speaking, BC and IRL methods use different methodology to reproduce the expert behavior. BC commonly uses a direct mapping from the states to the actions, while IRL tries to recover the reward function from the demonstrations. This difference could be why BC methods are commonly applied to real-world problems while most IRL methods still do simulations in the environment with less invention.

Compared with direct mapping, recovering a reward function needs stronger computational power and technologies to obtain the unique reward function and solve the sparse reward problem.

Table 2. Categorization of IL: Model-based vs. Model-free

Classes	Examples and Publications
Model-based IL	Forward model[19, 21]
	Inverse model[43]
Model-free IL	BC method[42]
	Reward engineering[9]
	Adversarial style[70]

The inner loop reinforcement learning could also cause IRL methods to be impractical in real-world problems. For the computational problem, recent development in GPU gradually alleviate the problem of high-dimensional computation; for the technology aspect, recent algorithms like Trust Region Policy Optimization[60] and attention models[26] provide more robust and efficient approaches for IRL methods; as for the sparse reward function, Hindsight Experience Replay[2] is commonly adopted for this problem. On the other hand, BC also suffers from the "compounding error"[57] where a small error could destroy the final performance. Besides these problems, other problems like better representation and diverse behavior learning are still open, many approaches are proposed for these problems, such as[29, 39, 76].

Table 1 lists some of the recent research in IL categorized into BC and IRL. Recent BC methods mainly focus on the topics such as: meta-learning that the agent is learning to learn by pretraining on a broader range of behaviors[18]; combining BC with other technique like VR equipment[79]. On the other hand, recent IRL methods mainly focus on the topics such as: extending GAIL with other methods or problem settings[17]; recovering reward function from raw videos[5]; developing more efficient model-based IRL approaches by using the current development in reinforcement learning like TRPO[60] and HER[2].

4.2 Model-Based vs. Model-Free

Another classical taxonomy divides IL into model-based and model-free methods. The main difference between these two classes is whether the algorithm adopts a forward model to learn from the environmental context/dynamics. Before GAIL[25] was proposed, most IRL methods are developed in the model-based setting because IRL methods commonly involve iterative algorithms evaluate the environment, while BC methods are commonly model-free since the low-level controller is commonly available. After GAIL was proposed, various adversarial structured IL are proposed following the GAIL's model-free setting. Although learning from the environment sounds beneficial for all kinds of methods, it might not be necessary for a given problem setting or impractical to apply. Integrating environment context/dynamics could obtain more useful information so that the algorithm can achieve data-efficiency and feasibility, while the drawback is learning the model is expensive and challenging. For example, in robotics, the equipment is commonly precise, the spatial position, velocity and other parameters could be easily obtained, the system dynamics might provides relatively little help to reproduce the behavior. On the other hand, in autonomous car tasks, the system dynamics might be crucial to avoid hitting pedestrians. In this case, the choice of model-free or model-based depends on the tasks. Table 2 lists some of the recent research topics in IL categorized into model-based and model-free.

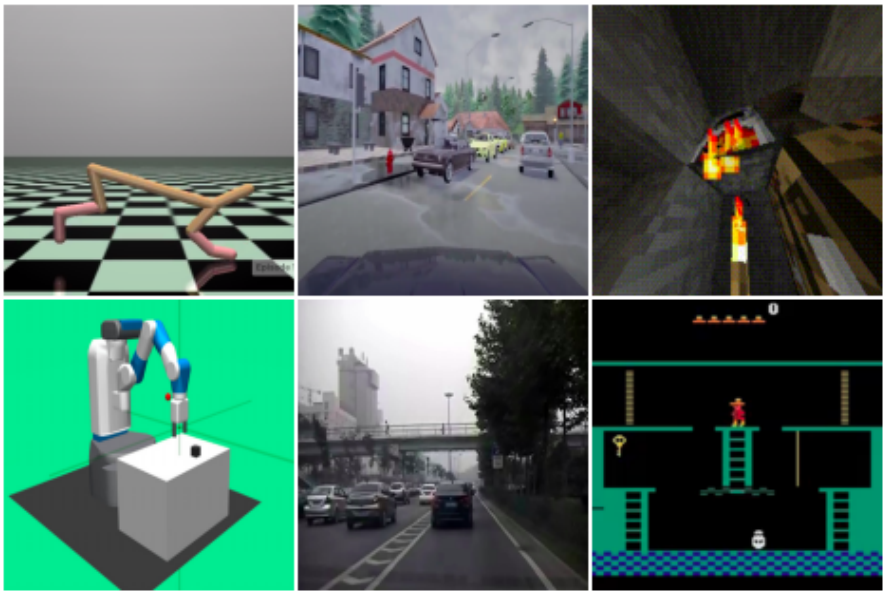

Fig. 3. Prevalent Tasks in IL. Top-left: HalfCheetah in Mujoco; Top-mid: CARLA simulator; Top-right: Minecraft scenario in MineRL dataset; Bottom-left: FetchPickAndPlace-v1 in OpenAI Gym; Bottom-mid: Driving scenario in Xi'an [80]; Bottom-right: Atari game–MontezumaRevenge

4.3 Low-Level Tasks vs. High-Level Tasks

This subsection introduces a novel taxonomy, which divides IL into manipulation tasks and high-level tasks according to their evaluation approach. The idea is inspired by a control diagram (See Figure 4) in[47]. Although some IL benchmark systems are proposed, such as[34], there is still no widely accepted one. In this case, the evaluation approaches and focus could vary from method to method, ranging from performance in sparse reward scenario to the smoothness of autonomous driving in dynamic environment. This taxonomy could draw clearer boundary and might alleviate the difficulty of designing appropriate benchmark from performance perspective.

The low-level manipulator tasks could be either real-world or virtual, and are not limited to robotics and autonomous driving problems. The robotic task can be object manipulation by robotic arm like PR2, KUKA robot arm, and simulation tasks commonly experimented on OPEN AI gym, MuJoCo simulation platform and so on. For real-world object manipulation tasks, the tasks could be push the object to the desired area, avoiding obstacles and operation soft object like rope. The autonomous driving tasks commonly implemented by simulation, and which is more related to the high-level planning. There are two widely-used benchmark system for simulation: CARLA CoRL2017 and NoCrash benchmark system, these two benchmark systems mainly focus on the urban scenario under various weather condition while the agent is evaluated on whether it can reach the destination on time, but CARLA CoRL2017 ignores the collision and traffic rules violation. Besides simulation, there are also some research doing experiment in real-world using cars[80] and smaller remote-controlled cars[14], but other kinds of equipment are also used like remote control helicopter[1]. As for the high-level controller, the tasks could be navigation tasks and gameplay. The navigation tasks are mainly route recommendation and in-door room-to-room navigation. Most of the evaluated games are 2D Atari games on OpenAI Gym, such as MontezumaRevenge is commonly

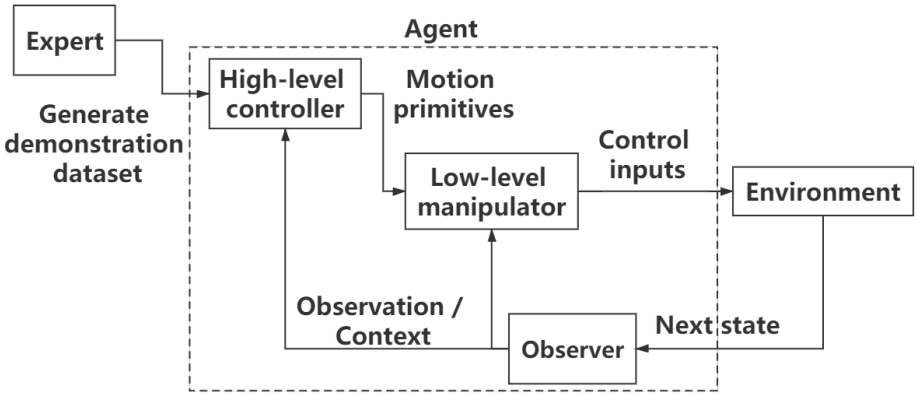

Fig. 4. Control diagram adapted from [47]

evaluated for performance on hard expolration and sparse reward scenario. Others are evaluated on 3D games like GTAV or Minecraft for evaluation. This taxonomy could be meaningful since it clearly reflects the target domain of the proposed algorithm, as the variance on their evaluation methods could be smaller, this may help to design a unified evaluation metric for IL. Figure 3 provides various popular evaluation tasks in IL.

From the Figure 4, the target of imitation could be either learning a policy for high-level controllers while assuming the low-level manipulator is working correctly or learning a policy to reproduce the simpler behavior on the low-level controller. Generally speaking, the high-level controller learns a policy to plan a sequence of motion primitives, such as [49]. As for the low-level controller, it learns a policy to reproduce the primitive behavior, such as [61], this forms the hierarchical structure of IL. Although some of the methods propose general frameworks which are evaluated on both domains, most of them are presenting “bias" on selecting tasks to demonstrate their improvement in either higher-level or low-level domain. For example, in [10], the proposed algorithm is evaluated on both Atari and Mujoco environments, but the amount of the evaluated tasks in each environment is obviously unequal. In this case, the ambiguity of classifying these general methods could be simply eliminated based on their tendency on evaluation tasks.

Table 3 lists some of the recent research under this taxonomy. The majority of current imitation methods tend to use low-level manipulation tasks to evaluate the proposed method, since reinforcement learning performs acceptably in high-level controller tasks like games, and commonly performs poorly on the low-level manipulation tasks where the reward function might be impractical to obtain. Nevertheless, IL in the high-level controller tasks is non-trivial, since for the 3D tasks or hard exploration games, reinforcement learning can be time-consuming on the huge state and action space.

4.4 BC vs. IRL vs. Adversarial Structured IL

This taxonomy is extended from the first taxonomy (BC vs. IRL). This new taxonomy divides IL into three categories: Behavioural Cloning (BC), Inverse Reinforcement Learning (IRL) and adversarial structured IL. With the recent development of IL, adversarial structured IL brings new insights for researchers and alleviate problems existing in previous work, such as high-dimensional problem. Inspired by the presence of GAIL, many recent papers adopt this adversarial structure, and inevitably, GAIL becomes baseline for comparison. But this is not enough to establish an

Table 3. Categorization of IL: Low-level Tasks vs. High-level Tasks

Classes	Examples and Publications
Low-level manipulation	Surgical assistance[49, 68]
	Vehicle manipulation[80]
	Robotic arm[61]
	VR teleoperation[79]
High-level tasks	2D gameplay[59]
	3D gameplay[4]
	Navigation[28]
	Sports analysis[78]

independent category in IL, the true reason making it distinguishable is that GAIL is not belongs to either BC or IRL. Although adversarial structured IL has close connection with IRL, most adversarial structured IL does not recover the reward function. In this case, the taxonomy of IL could be more specific. GAIL and its derivations are separated from the traditional IRL category and classified as adversarial structured IL in this survey. Compared with the traditional taxonomies, the proposed new taxonomy is more adapted to the development of IL and eliminates the vagueness of classifying these adversarial structured methods.

Figure 5 roughly evaluate the proposed three classes through two kinds of aspects which are commonly compared between research. Since different methods evaluate on various tasks, the overall performance is hard to quantify and rank, in this case, we evaluate three classes from Efficiency and Robustness from an empirical perspective.

In terms of Efficiency, we mainly focus on environmental interaction, computation, and expert interaction. BC methods commonly take advantage of interaction with expert while have less interaction in the environment, and due to these characteristics, the computational cost for BC is more likely to be the lowest; IRL methods commonly have abundant interaction with the environment in their inner-loop, and the evaluation on system dynamic makes IRL suffers from high computational cost, but IRL methods hardly enquiry the expert during training; Adversarial structured IL methods also involve frequent interaction with the environment when they iteratively update the policy parameter and discriminator parameter, and get rid of the interaction with expert. As adversarial structured IL methods are commonly model-free, in the evaluation of computational efficiency, we rank it as the second.

In terms of Robustness, we mainly focus on robustness in high-dimensional space, robustness when demonstrations are suboptimal (includes the consideration on noise in demonstration), and robustness in dynamic system. BC methods commonly have better performance in high-dimensional space so that they are widely evaluation on robotics, while the performance in dynamic environment and suboptimal dataset are limited; IRL methods optimize the parameter in their inner-loop, which becomes a burden limiting their performance in high-dimensional space, but the recovered reward function would benefit the agent to do prediction in dynamic system. Since adversarial structured IL methods commonly derive from GAIL, they inherit the merits of GAIL: robustness in high-dimensional space and when changes occur in distribution. Because recent research such as[9, 17, 84] in both IRL and Adversarial structured IL make progress in suboptimal

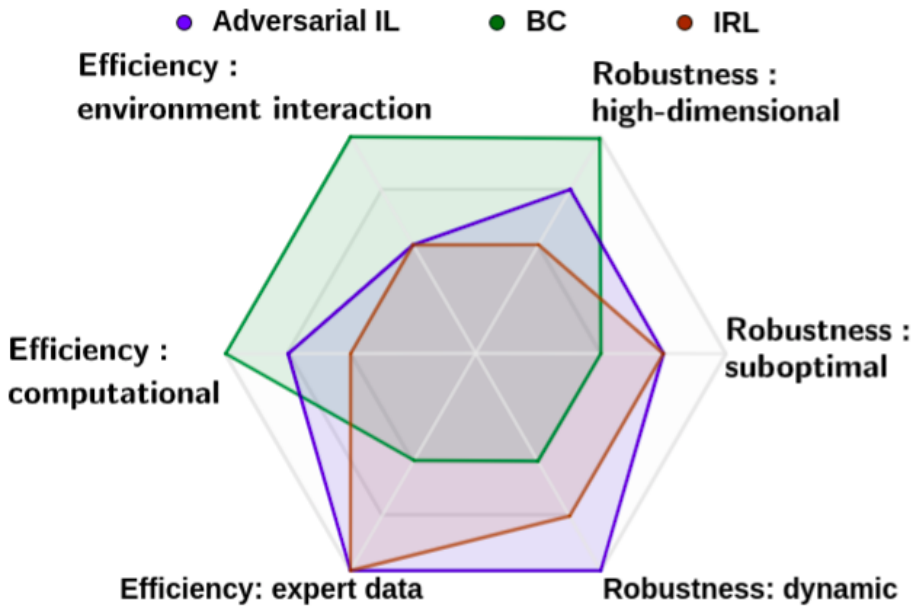

Fig. 5. Web plot for taxonomy: BC vs. IRL vs. Adversarial Structured IL. We collected 6 popular evaluation criteria from the research and empirically ranked them into three levels based on research consensus. The outer the point, the higher the ranking, which means that it scores higher in the evaluation from the empirical perspective.

demonstration problem, we give them the same rank in the evaluation of robustness on suboptimal demonstration.

5 MAIN RESEARCH TOPICS AND METHODS

5.1 Behavioural Cloning

Behavioural Cloning directly maps the states/contexts to actions/trajectories by leveraging the demonstration provided by expert/oracle. After generating the control input or trajectories, the loss function $\mathcal{L}$ will be designed according to the problem formulation and optimized in a supervised learning fashion. The state-of-the-art behavioural cloning uses negative log-likelihood loss to update the policy, i.e.

$$\underset{\pi}{\operatorname{argmin}}\, \mathcal{L}(\pi) = -\frac{1}{N}\sum_{k=1}^{N} \log \pi(a_k|s_k)$$

Algorithm 1 outlines the state-of-the-art behavioural cloning process. As traditional BC has less connection to MDP comparing with other prevalent methods, its efficiency is guaranteed, the trade-off is that it suffers from the scenario when the agent visits an unseen state. Loss function $\mathcal{L}$ could be customized for specific problem formulation. Loss function (objective function) significantly influences the training process and there are many existing lost function available to measure the differences (in most cases, the difference means the 1 step deviation) such as ℓ_1 loss, ℓ_2 loss, KL divergence, Hinge Loss, etc. For example, when using KL divergence as the loss function, the objective policy could be obtained by minimizing the deviation between expert distribution $q\pi_E$

Algorithm 1 Basic behavioural cloning method

1: Collect expert demonstration into dataset $\mathcal{D}$;
2: Select policy representation π_θ and loss function $\mathcal{L}$;
3: Use $\mathcal{D}$ to optimize the loss function $\mathcal{L}$ based on policy representation π_θ;
4: **return** optimized policy representation π_θ;

and induced distribution $q(\pi)$, i.e.

$$\pi^* = \underset{\pi}{\operatorname{argmin}} D_{KL}(q(\pi_E) \| q(\pi)).$$

BC could be subdivided into model-free BC and model-based BC methods. The main difference is whether the method learns a forward model to estimate the system dynamics. Since model-free BC methods take no consideration on the context, model-free BC methods perform well in industry applications where accurate controllers are available and experts could control and modify the robot joints. However, model-free BC methods typically are hard to predict future states and could not guarantee the output's feasibility under the environment that an accurate controller is not available. Under this kind of "imperfect" environment, the agent would have limited information of system dynamics and usually gets stuck into the unseen scenarios due to the "compounding error"[55]. While model-based BC methods leverage the environment information and learn the dynamics iteratively to produce feasible output, the trade-off is that model-based BC methods usually have greater time-complexity since the iterative learning involvement process.

One of the significant BC method is DAgger, which is a model-free BC method proposed by Ross et al.[57] and the idea is to use dataset aggregation to improve the generalization on unseen scenario. Algorithm 2 presents the abstract process of DAgger. DAgger adopts iterative learning process and mixes a new policy $\hat{\pi}^{n+1}$ with probability β to construct the next policy. The mixing parameter is a set of $\{\beta_i\}$ that satisfies $\frac{1}{N}\sum_{i=1}^{N}\beta_i \rightarrow 0$. The start-up policy is learned by BC and records the trajectory into the dataset. Since a small difference can lead to compounding error, new unseen trajectories will be recorded combining with the expert's corrections. In this case, the algorithm gradually updates the possible state and fully leverages the presence of expert. Later research like[29, 38, 56, 67, 73, 75] were proposed to make improvements on DAgger. This method alleviates the problem that traditional BC methods perform poorly on the unseen scenario and achieve data-efficiency comparing with previous methods like SMILe[55]. However, it does have drawbacks, such as DAgger involves frequent interaction with the expert which might not be available and expensive in some cases (e.g., enquiring expert correction could be expensive in interdisciplinary tasks). Recent methods such as[13, 25] successfully alleviate this problem. Another problem of DAgger could be that cost of each action is ignored. Since DAgger is evaluated on video games where the actions have equal cost, the cost of implementing each action is not obvious like tasks such as navigation tasks. This problem is solved later by Ross and Bagnell[56].

5.2 Inverse Reinforcement Learning

Inverse reinforcement learning was firstly proposed by Russell[58]. Unlike BC, the IRL agent is recovering and evaluating the reward function from expert demonstrations iteratively instead of establishing a mapping from states to actions. The choice of choosing BC or IRL depends on the problem settings. When the problem setting weights more on system dynamics and future prediction is necessary, choosing IRL methods can be more likely to evaluate the given context iteratively and provide a more accurate prediction. On the other hand, when an accurate controller

Algorithm 2 DAgger [57]

Initialize $\mathcal{D} \leftarrow \varnothing$;
Initialize $\hat{\pi}_1$ to any policy in Π;
for $i = 1 \rightarrow N$ **do**
    Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
    Sample T-step trajectory using π_i.
    Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by π_i and action given by expert.
    Aggregate dataset $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.
    Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
end for
return best $\hat{\pi}_i$ on validation.

Algorithm 3 Classic feature matching IRL method

Require: The set of demonstrated trajectories $\mathcal{D}$;
Initialize reward function parameter ω and policy parameter θ;
repeat
    Evaluate current policy π_θ state-action visitation frequency u;
    Evaluate loss function $\mathcal{L}$ w.r.t. u and the dataset $\mathcal{D}$ distribution;
    Update the reward function parameter ω based on the loss function;
    Update the policy parameter θ in the inner loop RL method using the updated reward parameter ω;
until
return optimized policy representation π_θ;

and abundant demonstrations are available, choosing BC methods usually takes less time and performs better.

IRL commonly assumes that the demonstrations are under Markov Decision Process setting and since the reward $\mathbb{R}$ is unknown, the set of states is used to estimate the feature vector (i.e. $\phi : \mathcal{X} \mapsto [0, 1]^k$) instead of the true reward function (i.e. $\mathcal{X} \mapsto \mathbb{R}$). The process of classic IRL method (see Algorithm 3) is based on iteratively update the reward function parameter ω and policy parameter θ. The reward function parameter ω is updated after the state-action visitation frequency u are evaluated, and the way that ω is updated could vary, for example, Ziebart et al.[82] updated ω by maximizing the likelihood of the demonstration over maximum entropy distribution, i.e. $\omega^* = \text{argmax}_\omega \sum_{\tau \in D} \log P(\tau \| \omega)$. On the other hand, the policy parameter θ is updated in the inner loop reinforcement learning process. This iterative and embedded structure can be problematic: the learning process could be time-consuming and impractical for high-dimensional problems like the high Degree Of Freedom (DOF) robotic problem. Another significant problem is "ill-posed" which means the many different cost functions could lead to the same action. In this case, the good IRL methods need to have more expressive power and a more efficient framework. Research such as[9, 15, 21, 30, 41, 50, 54] was proposed to alleviate the above problems by using more expressive models like neural network and optimizing the input like ranking the demonstration in advance.

Several recent IRL methods are gradually integrated with various novel methods such as self-supervised learning. Self-supervised learning means learning a function from a partially given context to the remaining or surrounding context. Nair et al.[43] could be one of the earliest researchers who adopt self-supervised learning into imitation learning. One important problem that integrating self-supervised learning with imitation learning has to solve is the huge amount of data, since the state and action space is extensive for real-world manipulation tasks. Nair et al. solved this problem by using the Baxter robot which automatically records data for a rope manipulation task. This method achieves practical improvement and provides a novel viewpoint for later research and leads the tendency of learning from the past. In 2018, Oh et al.[45] proposed self-IL, which tries to leverage past good experience to get better exploration result. The proposed method takes a initial policy as input. It then iteratively uses the current policy to generate trajectories, calculates the accumulated return value R, update the dataset $D \leftarrow D \bigcup \{(s_t, a_t, R)\}_{t=0}^{T}$ and finally uses the deviation between accumulated return and the agent estimate value $R - V_\theta$ to optimize the policy parameter θ. The process gradually ranks the state-action pairs and updates the policy parameter from the high-ranked pairs. In addition, Self-IL integrates Q learning with policy gradient under the actor-critic framework. As the component of the loss function, policy gradient loss was used to determine the good experience and lower bound Q learning was used to exploit the good experience, this helps Self-IL perform better in the hard exploration tasks. Similarly, in[74], Self-supervised Imitation Learning (SIL) also tries to learn from its good experience but in a different structure. SIL creatively uses voice instruction in the imitation learning process. One language encoder is used to extract textual feature $\{\omega_i\}_{i=1}^{n}$ and an attention-based trajectory encoder LSTM is use to encode the previous state-action as a history context vector from visual state $\{v_j\}_{j=1}^{m}$, i.e. $h_t = LSTM([v_t, a_{t-1}], h_{t-1})$. Then visual context c_t^{visual} and language context c_t^{text} could be obtained based on the historical context vector, finally the action is predicted based on these parameters. The obtained experience is evaluated on a match critic, and the "good" experience is stored in a replay buffer for future prediction.

5.3 Generative Adversarial Imitation Learning (GAIL)

In order to mitigate problems in BC and IRL, Ho and Ermon[25] proposed a novel general framework called Generative adversarial imitation learning in 2016. GAIL builds a connection between GAN[23] and maximum entropy IRL[82]. Inheriting from the structure of GAN, GAIL consists of a generative model G and a discriminator D, while G generates data distribution ρ_π integrating with true data distribution $\rho_{\pi E}$ to confuse D. GAIL works in an iterative fashion, and the formal objective of GAIL could be denoted as

$$\min_{\pi} \max_{D \in (0,1)^{S \times \mathcal{A}}} \hat{\mathbb{E}}_{\tau_i}[\log(D_\omega(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\log(1 - D_\omega(s,a))].$$

GAIL firstly samples trajectories from initial policy, then these generated trajectories are used to update the discriminator weight ω by applying an Adam gradient step on equation

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_\omega \log(D_\omega(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_\omega \log(1 - D_\omega(s,a))],$$

and maximize this equation with respect to D. Then adopting the TRPO[60] with the cost function $\log(D_{\omega_{i+1}}(s,a))$ to update the policy parameter θ and minimize the above function with respect to π, combining with a causal entropy regularizer controlled by non-negative parameter λ, i.e.

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_\theta \log \pi_\theta(a|s) \boldsymbol{Q}(s,a))] - \lambda \nabla_\theta H(\pi_\theta)$$

where $\boldsymbol{Q}(\bar{s}, \bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{\omega_{i+1}}(s,a))|s_0 = \bar{s}, a_0 = \bar{a}]$.

The abstract training process is presented in Algorithm 4. By adopting TRPO, the policy could be more resistant and stable to the noise in the policy gradient. Unlike DAgger and other previous

Algorithm 4 GAIL [25]

Require: Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameter θ_0, ω_0
  for $i = 0, 1, 2, ...$ **do**
    Sample trajectories $\tau_i \sim \pi_{\theta_i}$.
    Update the discriminator parameters ω_i to ω_{i+1}.
    Update the policy parameter θ_i to θ_{i+1}.
  end for

Table 4. Different Kinds of Derivative on GAIL

GAILs	Methods
Make further improvement	MGAIL[7], InfoGAIL[35]
Apply to other research question	MAGAIL[63], GAIfO[70]
Other generative model	Diverse GAIL[76], GIRL[77]

algorithms, GAIL is more sample-efficiency from the perspective of using expert data and does not require expert interaction during the training process, it also presents adequate capacity dealing with the high-dimensional domain and changes in distribution. While the trade-off is the training process involves frequent interaction with the environment and could be more fragile and not stable for saddle point problem. As for the first problem, the authors suggested to initialize the policy with BC so that the amount of environment interaction would reduce. As for the second problem, recent research such as[3] tries to alleviate this problem by formulating the distribution-matching problem as an iterative lower-bound optimization problem.

Inspired by GAIL's presence, there is a bunch of research proposed to make further development on GAIL (see Table 4) and adversarial structured IL gradually becomes a category. In terms of "make further improvement", many proposed methods modify and improve GAIL from different perspectives. For example, MGAIL[7] uses an advanced forward model to make the model differentiable so that the Generator could use the exact gradient of the Discriminator. InfoGAIL[35] modifies GAIL by adopting WGAN instead of GAN. Other recent work like GoalGAIL [17], TRGAIL[33] and DGAIL[83] are all making improvement on GAIL by combining with other method like hindsight relabeling and Deep Deterministic Policy Gradient (DDPG) [36] to achieve faster convergence and better final performance. In terms of "apply to other research question", some of the proposed methods combine other method with GAIL and apply to various problems. For example, in[66], FAIL outperforms GAIL on sparse reward problem without using the ground truth action and achieves both sample and computational efficiency. It integrates adversarial structure with minimax theory, which is used to determines the next time step policy π_h under the assumption that $\{\pi_1, \pi_2, ..., \pi_{h-1}\}$ is learned and fixed. GAIL is also applied into the other research area, such as multi-agent settings[8, 63, 78] and IfO settings[70] to effectively deal with more dynamic environment. In terms of "combine IL with other generative model", a number of recent research adopt other generative models to facilitate learning process, for example, in[76], Variational AutoEncoder(VAE) is integrated with IL by using encoder to map from trajectories to an embedding vector z, which makes the proposed algorithm to behave diversely with relatively less demonstration and achieve one-shot learning for the new trajectory. Other research like GIRL[77] also achieves the outstanding performance from limited demonstrations using VAE.

Table 5. Publication Related to IfO

Publication	Description
IfO[40]	Learning policy from aligned observation only
BCO[69]	Adopting IfO setting and integrating with BC
TCN[61]	Multi-viewpoint self-supervised IfO method
One-shot IfO[5]	Extracting features from unlabeled and unaligned gameplay footage
Zero-Shot Visual Imitation[51]	Using distance between observations to predict and penalize the actions
IfO survey[72]	Detailed classified recent IfO methods
Imitating Latent Policies from Observation[19]	Infering latent policies directly from state observations
GAIfO[70]	Generative adversarial structure aggregating with IfO
IfO Leveraging Proprioception[71]	Leveraging internal information of the agent
OPOLO[81]	Using dual-form of the expectation function and adversarial structure to achieve off-policy IfO

5.4 Imitation from Observation (IfO)

The prevalent methods introduced above is almost using sequences of state-action pairs to form trajectories as the input data. This kind of data preparation process could be laborious and this is a kind of waste for the abundant raw unlabeled videos. This problem got mitigated after IfO[40] was proposed, and IL algorithms start to advocate this novel settings and make use of raw videos to learn policies. Comparing with traditional IL methods, this algorithm is more intuitive, and it follows the nature of how human and animal imitate. For example, people learn to dance by following a video, this kind of following process is achieved though detecting the changes of poses and taking actions to match the pose, which is similar to how IfO solves the problem. Different from traditional IL, the ground truth action sequence is not given. Similar to IRL, the main objective of IfO is the reward function from demonstration videos. Imitation from observation tries to build connection for different context so that the VAE structure is adopted to encode both the context (environment) of demonstrator (expert) s_1 and target context s_2. The proposed model has four components: a source observation encoder $Enc_1(o_t^i)$ which extracts feature vector z_1, a target observation encoder $Enc_2(o_0^j)$ which extracts feature vector z_2, a translator z_3 and a target context decoder $Dec(z_3)$. The model takes two sets of observations ($D_i = [o_t^i]_{t=0}^T$ and $D_j = [o_t^j]_{t=0}^T$ as source observation and target observation respectively) as input, then using these two sets to predict the future observation in target context under the assumption that source observation and target observation are time aligned. The translator z_3 translates features in z_1 produced by source encoder into the context of z_2 produced by another encoder, i.e. $z_3 = T(z_1, z_2)$, then the translated feature vector z_3 is decoded into the observation $\hat{o}_t^j$. The model is working in a supervised learning process with the loss function $\mathcal{L}_{trans} = \|(\hat{o}_t^j) - o_t^j\|_2^2$. To improve the performance, the final objective of the proposed model is

combined with the loss of VAE reconstruction and the loss of time alignment, i.e.

$$\mathcal{L} = \sum_{(i,j)} (\mathcal{L}_{trans} + \lambda_1 \mathcal{L}_{rec} + \lambda_2 \mathcal{L}_{align}),$$

where λ_1 and λ_2 are the hyperparameter predetermined in advance. The output reward function consists of two parts, the first one is deviation penalty on squared Euclidean distance, which measures the difference between the encoded learner's observation feature and translated expert observation feature in learner's context, i.e.

$$\hat{R}_{feat}(0_t^l) = -\|Enc_1(o_t^l) - \frac{1}{n}\sum_{t=0}^{T} T(o_t^i, o_0^l)\|_2^2$$

The second part is the penalty which ensures the current observation keeping similar with translated observations, i.e.

$$\hat{R}_{img}(0_t^l) = -\|o_t^l - \frac{1}{n}\sum_{t=0}^{T} M(o_t^i, o_0^l)\|_2^2,$$

where M is the full observation translation model. The proposed reward function could be applied into the any reinforcement learning algorithm, Liu et al. uses TRPO[60] for the simulation experiments.

After IfO being proposed, measuring observation distance to replace the ground truth action becomes a prevalent setting in imitation learning. In Table 5, we present some of the research advocate this new insight and apply this idea into various domain. Both BC, IRL and GAIL start to adopt this setting to simplify the input. For example, in[5], raw unaligned YouTube videos are used for imitation to reproduce the behavior for games. YouTube videos are relatively noisy and varying in settings like resolution. The proposed method successfully handled these problems by using a novel self-supervised objective to learn a domain-invariant representation from videos. Similarly, in[61], multi-viewpoint self-supervised IL method Time-Contrastive Network (TCN) was proposed. Different viewpoints introduce a wide range of contexts about the task environment and the goal is to learn invariant representation about the task. By measuring the distance between the input video frames and "looking at itself in the mirror", the robot could learn its internal joint to learn the mapping and achieve imitating demonstration.

6 CHALLENGES AND OPPORTUNITIES

Although improvements like integrating novel techniques, reducing human interaction during training and simplifying inputs alleviate difficulties in learning behaviour, there are still some open challenges for IL:

Diverse behavior learning: Current IL methods commonly use task-specific training datasets to learn to reproduce single behavior. Research like[76] presented diverse behavior learning by combining adversarial structure and variational autoencoder, but this is still an open challenge. Other methods could be adopted to optimize IL, such as transfer learning might help the agent to learn from similar tasks so that the training process could be more efficient.

Sub-optimal demonstration for training: Current IL methods generally require a high-quality set of demonstrations for training. However, the number of high-quality demonstrations could be limited and expensive to obtain. Existing research like[9, 17, 64] have shown the possibility to use sub-optimal demonstration for training, but performance can be improved by extracting common intent from the dataset.

Imitation not just from observation: Current IfO methods commonly use raw videos and the deviation of observations to recover the reward function. But the video is not just observation,

maybe the voice instruction could also be used to get a better reward function. Wang et al.[74] demonstrated using natural language for navigation tasks, but it could be an interesting topic to explore in the IfO settings.

Better representation: Good policy representation could benefit the training process to achieve data-efficiency and computation-efficiency. Finding better policy representation is still an active research topic for IL. Besides policy representation, how to represent the demonstration is another problem in IL. The representation of demonstration needs to be more efficient and expressive.

Find globally optimal solution: Most research is finding a locally optimal solution based on demonstration, which might set the upper-bound for the agent performance. The future direction could be finding the global optimal for a specific task, which requires the agent to understand the intent of the behavior instead of copy-pasting. Current research like[77] successfully surpasses the demonstrator's performance, but finding the global optimal still needs effort.

7 CONCLUSION

Imitation learning achieves outstanding performance in a wide range of problems, ranging from solving hard exploration Atari games to achieving object manipulation while avoiding obstacles by robotic arm. Different kinds of imitation learning methods make contribution to this significant development, such as BC methods replicate behavior more intuitively where the environmental parameters could be easily obtained; IRL methods achieve data-efficiency and future behavior prediction when problems weight more on environment dynamics and care less about training time; adversarial structured IL methods eliminate expert interaction during the training process and present adequate capacity dealing with the high-dimensional problem. While IL methods continue to grow and develop, IL is also seeking breakthroughs in settings, like IfO methods simplify the input by replacing the need of action labels when the input demonstrations are raw video. Although recent work presents a superior advantage in replicating behavior, taxonomy ambiguity exists as the presence of GAIL and its derivatives break out of the previous classification framework. To alleviate this ambiguity, we analyzed the traditional taxonomies of IL and proposed new taxonomies that draw clearer boundaries between methods. Despite the success of IL, challenges and opportunities exist, such as diverse behavior learning, leveraging sub-optimal demonstration and voice instruction, better representation, and finally finding the globally optimal solution. Future work is expected to unravel IL and its practical applications.

REFERENCES

[1] Pieter Abbeel, Adam Coates, and Andrew Y. Ng. 2010. Autonomous Helicopter Aerobatics through Apprenticeship Learning. *The International Journal of Robotics Research* 29, 13 (Nov. 2010), 1608–1639. https://doi.org/10.1177/0278364910371999

[2] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. 2017. Hindsight experience replay. In *Advances in neural information processing systems*. 5048–5058.

[3] Oleg Arenz and Gerhard Neumann. 2020. Non-Adversarial Imitation Learning and its Connections to Adversarial Methods. *arXiv:2008.03525 [cs, math, stat]* (Aug. 2020). http://arxiv.org/abs/2008.03525 arXiv: 2008.03525.

[4] Dilip Arumugam, Jun Ki Lee, Sophie Saskin, and Michael L Littman. 2019. Deep reinforcement learning from policy-dependent human feedback. *arXiv preprint arXiv:1902.04257* (2019).

[5] Yusuf Aytar, Tobias Pfaff, David Budden, Thomas Paine, Ziyu Wang, and Nando de Freitas. 2018. Playing hard exploration games by watching YouTube. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 2930–2941. http://papers.nips.cc/paper/7557-playing-hard-exploration-games-by-watching-youtube.pdf

[6] Michael Bain and Claude Sammut. 1999. A framework for behavioural cloning. In *Machine Intelligence 15*. Oxford University Press, 103–129.

[7] Nir Baram, Oron Anschel, Itai Caspi, and Shie Mannor. 2017. End-to-end differentiable adversarial imitation learning. In *International Conference on Machine Learning*. 390–399.

[8] Raunak P. Bhattacharyya, Derek J. Phillips, Blake Wulfe, Jeremy Morton, Alex Kuefler, and Mykel J. Kochenderfer. 2018. Multi-Agent Imitation Learning for Driving Simulation. *arXiv:1803.01044 [cs]* (March 2018). http://arxiv.org/abs/1803.01044 arXiv: 1803.01044.

[9] Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. 2019. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International Conference on Machine Learning*. PMLR, 783–792.

[10] Daniel S. Brown, Wonjoon Goo, and Scott Niekum. 2019. Better-than-Demonstrator Imitation Learning via Automatically-Ranked Demonstrations. arXiv:1907.03976 [cs.LG]

[11] Andreas Bühler, Adrien Gaidon, Andrei Cramariuc, Rares Ambrus, Guy Rosman, and Wolfram Burgard. 2020. Driving Through Ghosts: Behavioral Cloning with False Positives. arXiv:2008.12969 [cs.CV]

[12] Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daume, and John Langford. 2015. Learning to search better than your teacher. In *International Conference on Machine Learning*. PMLR, 2058–2066.

[13] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. 2020. Learning by cheating. In *Conference on Robot Learning*. PMLR, 66–75.

[14] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. 2018. End-to-End Driving Via Conditional Imitation Learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 4693–4700. https://doi.org/10.1109/ICRA.2018.8460487 ISSN: 2577-087X.

[15] Neha Das, Sarah Bechtle, Todor Davchev, Dinesh Jayaraman, Akshara Rai, and Franziska Meier. 2020. Model-Based Inverse Reinforcement Learning from Visual Demonstrations. *arXiv:2010.09034 [cs]* (Oct. 2020). http://arxiv.org/abs/2010.09034 arXiv: 2010.09034.

[16] Hal Daumé, John Langford, and Daniel Marcu. 2009. Search-based structured prediction. *Machine learning* 75, 3 (2009), 297–325.

[17] Yiming Ding, Carlos Florensa, Mariano Phielipp, and Pieter Abbeel. 2019. Goal-conditioned imitation learning. *arXiv preprint arXiv:1906.05838* (2019).

[18] Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. 2017. One-Shot Imitation Learning. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 1087–1098. http://papers.nips.cc/paper/6709-one-shot-imitation-learning.pdf

[19] Ashley Edwards, Himanshu Sahni, Yannick Schroecker, and Charles Isbell. 2019. Imitating latent policies from observation. In *International Conference on Machine Learning*. PMLR, 1755–1763.

[20] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*. PMLR, 1126–1135.

[21] Chelsea Finn, Sergey Levine, and Pieter Abbeel. 2016. Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*. PMLR, 49–58.

[22] Laurent George, Thibault Buhet, Emilie Wirbel, Gaetan Le-Gall, and Xavier Perrotton. 2018. Imitation Learning for End to End Vehicle Longitudinal Control with Forward Camera. *arXiv:1812.05841 [cs]* (Dec. 2018). http://arxiv.org/abs/1812.05841 arXiv: 1812.05841.

[23] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. arXiv:1406.2661 [stat.ML]

[24] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Gabriel Dulac-Arnold, Ian Osband, John Agapiou, Joel Z. Leibo, and Audrunas Gruslys. 2017. Deep Q-learning from Demonstrations. *arXiv:1704.03732 [cs]* (Nov. 2017). http://arxiv.org/abs/1704.03732 arXiv: 1704.03732.

[25] Jonathan Ho and Stefano Ermon. 2016. Generative Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 4565–4573. http://papers.nips.cc/paper/6391-generative-adversarial-imitation-learning.pdf

[26] X. Hu, J. Liu, J. Ma, Y. Pan, and L. Zhang. 2020. Fine-Grained 3D-Attention Prototypes for Few-Shot Learning. *Neural Computation* 32, 9 (2020), 1664–1684. https://doi.org/10.1162/neco_a_01302

[27] Z. Hu, Z. Gan, W. Li, J. Z. Wen, D. Zhou, and X. Wang. 2020. Two-Stage Model-Agnostic Meta-Learning With Noise Mechanism for One-Shot Imitation. *IEEE Access* 8 (2020), 182720–182730. https://doi.org/10.1109/ACCESS.2020.3029220 Conference Name: IEEE Access.

[28] Ahmed Hussein, Eyad Elyan, Mohamed Medhat Gaber, and Chrisina Jayne. 2018. Deep imitation learning for 3D navigation tasks. *Neural Comput & Applic* 29, 7 (April 2018), 389–404. https://doi.org/10.1007/s00521-017-3241-z

[29] Mostafa Hussein, Brendan Crowe, Marek Petrik, and Momotaz Begum. 2021. Robust Maximum Entropy Behavior Cloning. *arXiv preprint arXiv:2101.01251* (2021).

[30] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. 2018. Reward learning from human preferences and demonstrations in atari. *arXiv preprint arXiv:1811.06521* (2018).

[31] Parham M. Kebria, Abbas Khosravi, Syed Moshfeq Salaken, and Saeid Nahavandi. 2020. Deep imitation learning for autonomous vehicles based on convolutional neural networks. *IEEE/CAA Journal of Automatica Sinica* 7, 1 (Jan. 2020), 82–95. https://doi.org/10.1109/JAS.2019.1911825 Conference Name: IEEE/CAA Journal of Automatica Sinica.

[32] Beomjoon Kim and Joelle Pineau. 2013. Maximum Mean Discrepancy Imitation Learning. In *Robotics: Science and Systems IX*. Robotics: Science and Systems Foundation. https://doi.org/10.15607/RSS.2013.IX.038

[33] Akira Kinose and Tadahiro Taniguchi. 2020. Integration of imitation learning using GAIL and reinforcement learning using task-achievement rewards via probabilistic graphical model. *Advanced Robotics* (June 2020), 1–13. https://doi.org/10.1080/01691864.2020.1778521

[34] A. Lemme, Y. Meirovitch, M. Khansari-Zadeh, T. Flash, A. Billard, and J. J. Steil. 2015. Open-source benchmarking for learned reaching motion generation in robotics. *Paladyn, Journal of Behavioral Robotics* 6, 1 (Jan. 2015). https://doi.org/10.1515/pjbr-2015-0002

[35] Yunzhu Li, Jiaming Song, and Stefano Ermon. 2017. InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 3812–3822. http://papers.nips.cc/paper/6971-infogail-interpretable-imitation-learning-from-visual-demonstrations.pdf

[36] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2019. Continuous control with deep reinforcement learning. arXiv:1509.02971 [cs.LG]

[37] Rudolf Lioutikov, Gerhard Neumann, Guilherme Maeda, and Jan Peters. 2017. Learning movement primitive libraries through probabilistic segmentation. *The International Journal of Robotics Research* 36, 8 (July 2017), 879–894. https://doi.org/10.1177/0278364917713116 Publisher: SAGE Publications Ltd STM.

[38] Evan Liu, Milad Hashemi, Kevin Swersky, Parthasarathy Ranganathan, and Junwhan Ahn. 2020. An imitation learning approach for cache replacement. In *International Conference on Machine Learning*. PMLR, 6237–6247.

[39] Mengyue Liu, Jun Liu, Yihe Chen, Meng Wang, Hao Chen, and Qinghua Zheng. 2019. AHNG: representation learning on attributed heterogeneous network. *Information Fusion* 50 (2019), 221–230.

[40] YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. 2018. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1118–1125.

[41] Sha Luo, Hamidreza Kasaei, and Lambert Schomaker. 2021. Self-Imitation Learning by Planning. *arXiv preprint arXiv:2103.13834* (2021).

[42] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. 2020. Learning latent plans from play. In *Conference on Robot Learning*. PMLR, 1113–1132.

[43] Ashvin Nair, Dian Chen, Pulkit Agrawal, Phillip Isola, Pieter Abbeel, Jitendra Malik, and Sergey Levine. 2017. Combining self-supervised learning and imitation for vision-based rope manipulation. In *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2146–2153.

[44] Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. 2018. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 6292–6299.

[45] Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. 2018. Self-imitation learning. In *International Conference on Machine Learning*. PMLR, 3878–3887.

[46] Takayuki Osa, Amir M. Ghalamzan Esfahani, Rustam Stolkin, Rudolf Lioutikov, Jan Peters, and Gerhard Neumann. 2017. Guiding Trajectory Optimization by Demonstrated Distributions. *IEEE Robotics and Automation Letters* 2, 2 (April 2017), 819–826. https://doi.org/10.1109/LRA.2017.2653850 Conference Name: IEEE Robotics and Automation Letters.

[47] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, and Jan Peters. 2018. An Algorithmic Perspective on Imitation Learning. *FNT in Robotics* 7, 1-2 (2018), 1–179. https://doi.org/10.1561/2300000053

[48] Takayuki Osa, Naohiko Sugita, and Mamoru Mitsuishi. 2014. Online Trajectory Planning in Dynamic Environments for Surgical Task Automation. In *Robotics: Science and Systems X*. Robotics: Science and Systems Foundation. https://doi.org/10.15607/RSS.2014.X.011

[49] Takayuki Osa, Naohiko Sugita, and Mamoru Mitsuishi. 2018. Online Trajectory Planning and Force Control for Automation of Surgical Tasks. *IEEE Transactions on Automation Science and Engineering* 15, 2 (April 2018), 675–691. https://doi.org/10.1109/TASE.2017.2676018 Conference Name: IEEE Transactions on Automation Science and Engineering.

[50] Malayandi Palan, Nicholas C. Landolfi, Gleb Shevchuk, and Dorsa Sadigh. 2019. Learning Reward Functions by Integrating Human Demonstrations and Preferences. *arXiv:1906.08928 [cs]* (June 2019). http://arxiv.org/abs/1906.08928 arXiv: 1906.08928.

[51] Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Fred Shentu, Evan Shelhamer, Jitendra Malik, Alexei A. Efros, and Trevor Darrell. 2018. Zero-Shot Visual Imitation. In *2018 IEEE/CVF Conference*

on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, Salt Lake City, UT, USA, 2131–21313. https://doi.org/10.1109/CVPRW.2018.00278

[52] Dean A. Pomerleau. 1989. ALVINN: An Autonomous Land Vehicle in a Neural Network. In *Advances in Neural Information Processing Systems 1*, D. S. Touretzky (Ed.). Morgan-Kaufmann, 305–313. http://papers.nips.cc/paper/95-alvinn-an-autonomous-land-vehicle-in-a-neural-network.pdf

[53] Dean A. Pomerleau. 1991. Efficient training of artificial neural networks for autonomous navigation. *Neural computation* 3, 1 (1991), 88–97.

[54] Siddharth Reddy, Anca D. Dragan, and Sergey Levine. 2019. SQIL: Imitation Learning via Reinforcement Learning with Sparse Rewards. *arXiv:1905.11108 [cs, stat]* (Sept. 2019). http://arxiv.org/abs/1905.11108 arXiv: 1905.11108.

[55] Stéphane Ross and Drew Bagnell. 2010. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 661–668.

[56] Stephane Ross and J Andrew Bagnell. 2014. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979* (2014).

[57] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. 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, 627–635.

[58] Stuart Russell. 1998. Learning agents for uncertain environments (extended abstract). In *Proceedings of the eleventh annual conference on Computational learning theory - COLT' 98*. ACM Press, Madison, Wisconsin, United States, 101–103. https://doi.org/10.1145/279943.279964

[59] Tim Salimans and Richard Chen. 2018. Learning Montezuma's Revenge from a Single Demonstration. *arXiv preprint arXiv:1812.03381* (2018).

[60] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*. PMLR, 1889–1897.

[61] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. 2018. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1134–1141.

[62] Zhenyu Shou, Xuan Di, Jieping Ye, Hongtu Zhu, Hua Zhang, and Robert Hampshire. 2020. Optimal passenger-seeking policies on E-hailing platforms using Markov decision process and imitation learning. *Transportation Research Part C: Emerging Technologies* 111 (Feb. 2020), 91–113. https://doi.org/10.1016/j.trc.2019.12.005

[63] Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. 2018. Multi-Agent Generative Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 7461–7472. http://papers.nips.cc/paper/7975-multi-agent-generative-adversarial-imitation-learning.pdf

[64] Lingyun Song, Jun Liu, Mingxuan Sun, and Xuequn Shang. 2020. Weakly Supervised Group Mask Network for Object Detection. *International Journal of Computer Vision* (2020). https://doi.org/10.1007/s11263-020-01397-w

[65] Bradly C Stadie, Pieter Abbeel, and Ilya Sutskever. 2017. Third-person imitation learning. *arXiv preprint arXiv:1703.01703* (2017).

[66] Wen Sun, Anirudh Vemula, Byron Boots, and Drew Bagnell. 2019. Provably efficient imitation learning from observation alone. In *International Conference on Machine Learning*. PMLR, 6036–6045.

[67] Wen Sun, Arun Venkatraman, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. 2017. Deeply aggrevated: Differentiable imitation learning for sequential prediction. In *International Conference on Machine Learning*. PMLR, 3309–3318.

[68] Ajay Kumar Tanwani, Pierre Sermanet, Andy Yan, Raghav Anand, Mariano Phielipp, and Ken Goldberg. 2020. Motion2Vec: Semi-supervised representation learning from surgical videos. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2174–2181.

[69] Faraz Torabi, Garrett Warnell, and Peter Stone. 2018. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954* (2018).

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

[71] Faraz Torabi, Garrett Warnell, and Peter Stone. 2019. Imitation Learning from Video by Leveraging Proprioception. *arXiv:1905.09335 [cs, stat]* (June 2019). http://arxiv.org/abs/1905.09335 arXiv: 1905.09335.

[72] Faraz Torabi, Garrett Warnell, and Peter Stone. 2019. Recent advances in imitation learning from observation. *arXiv preprint arXiv:1905.13566* (2019).

[73] Stephen Tu, Alexander Robey, and Nikolai Matni. 2021. Closing the closed-loop distribution shift in safe imitation learning. *arXiv preprint arXiv:2102.09161* (2021).

[74] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. 2019. Reinforced cross-modal matching and self-supervised imitation learning for vision-language

navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6629–6638.

[75] Xiaojie Wang, Zhaolong Ning, Song Guo, Miaowen Wen, and Vincent Poor. 2021. Minimizing the age-of-critical-information: an imitation learning-based scheduling approach under partial observations. *IEEE Transactions on Mobile Computing* (2021).

[76] Ziyu Wang, Josh S Merel, Scott E Reed, Nando de Freitas, Gregory Wayne, and Nicolas Heess. 2017. Robust Imitation of Diverse Behaviors. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 5320–5329. http://papers.nips.cc/paper/7116-robust-imitation-of-diverse-behaviors.pdf

[77] Xingrui Yu, Yueming Lyu, and Ivor Tsang. 2020. Intrinsic reward driven imitation learning via generative model. In *International Conference on Machine Learning*. PMLR, 10925–10935.

[78] Eric Zhan, Stephan Zheng, Yisong Yue, Long Sha, and Patrick Lucey. 2018. Generating multi-agent trajectories using programmatic weak supervision. *arXiv preprint arXiv:1803.07612* (2018).

[79] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. 2018. Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 5628–5635. https://doi.org/10.1109/ICRA.2018.8461249 ISSN: 2577-087X.

[80] Yang Zhou, Rui Fu, Chang Wang, and Ruibin Zhang. 2020. Modeling Car-Following Behaviors and Driving Styles with Generative Adversarial Imitation Learning. *Sensors* 20, 18 (Sept. 2020), 5034. https://doi.org/10.3390/s20185034

[81] Zhuangdi Zhu, Kaixiang Lin, Bo Dai, and Jiayu Zhou. 2021. Off-policy imitation learning from observations. *arXiv preprint arXiv:2102.13185* (2021).

[82] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. 2008. Maximum entropy inverse reinforcement learning.. In *Aaai*, Vol. 8. Chicago, IL, USA, 1433–1438.

[83] Guoyu Zuo, Kexin Chen, Jiahao Lu, and Xiangsheng Huang. 2020. Deterministic generative adversarial imitation learning. *Neurocomputing* 388 (May 2020), 60–69. https://doi.org/10.1016/j.neucom.2020.01.016

[84] Guoyu Zuo, Qishen Zhao, Kexin Chen, Jiangeng Li, and Daoxiong Gong. 2020. Off-policy adversarial imitation learning for robotic tasks with low-quality demonstrations. *Applied Soft Computing Journal* 97 (2020), 106795. https://doi.org/10.1016/j.asoc.2020.106795