Imitation Learning: Progress, Taxonomies and Challenges

Boyan Zheng†, Sunny Verma, Jianlong Zhou, Senior Member, IEEE,
Ivor W. Tsang‡, Fellow, IEEE, and Fang Chen†

Abstract—Imitation learning (IL) aims to extract knowledge from human experts’ demonstrations or artificially created agents to replicate their behaviors. It promotes interdisciplinary communication and real-world automation applications. However, the process of replicating behaviors still exhibits various problems, 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 an insightful review on IL. We first introduce the background knowledge from development history and preliminaries, followed by presenting different taxonomies within IL 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.

Index Terms—Imitation learning (IL), machine learning, taxonomies.

I. 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 like deep learning, online learning, and Generative Adversarial Network (GAN)[1] make further improvement on IL. These advances alleviate existing problems like dynamic environment, frequent inquiries, and high-dimensional computation, and achieve faster convergence, better robustness on noise, and better sample efficiency. These improvements of IL promote the applications in both continuous and discrete control domains. For example, in the continuous control domain, IL could be applied to autonomous vehicle manipulation to reproduce appropriate driving behavior in a dynamic environment [2], [3], [4], [5], [6], [7], [8], [9]. In addition, IL is also applied to robotic, ranging from basic grabbing and placing to surgical assistance [10], [11], [12], [13], [14], [15], [16], [17]. In the discrete control domain, IL makes contribution to fields like game theory [18], [19], [20], [21], navigation tasks [22], [23], [24], cache management [25], 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. [8] 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.

IL has a close relationship with Reinforcement Learning (RL). Both IL and RL commonly solve the problem under Markov Decision Process (MDP), and improvements like trust region policy optimization (TRPO) [26] 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 nonintuitive in some settings. For example, people learn to swim and walk almost from demonstration (other forms of instruction may also contribute to the learning process, such as verbal and corrective intervention) instead of math functions, and it is hard to formalize 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.

The main contributions of this survey are as follows.

1) Latest Progress in IL: We review the latest state-of-the-art research related to IL and conclude major research trends in IL. A wide range of techniques, covering Behavioral Cloning (BC), Inverse Reinforcement Learning (IRL), Adversarial structured IL, and Imitation from Observation (HO), are discussed and summarized along with their merits, limitations, and the corresponding derivatives.

2) Innovative Classification: This survey presents research in IL under categories BC versus IRL and model-free versus model-based. It then summarizes IL research into two new categories namely low-level tasks versus high-level tasks and BC versus IRL versus adversarial structured IL, which are more adapted to the development of IL and alleviate problems in current framework.

3) Challenges Identification: This article identifies the remaining challenges of IL in areas of diverse behavior learning, suboptimal demonstration, and various modalities learning. Future research opportunities are also proposed with respect to methods like transfer learning and importance sampling.

This survey is organized as follows. A brief description of IL development history is presented in Section II. In Section III, fundamental knowledge and concepts are given. Section IV details some recent achievements under four frameworks. Afterward, in Section V, we review current taxonomy frameworks and introduced two alternatives that are more adapted to IL development. Challenges and potential opportunities are discussed in Section VI, and we conclude this survey in Section VII.

II. 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 [2]. Basic autonomous driving was achieved with a forward camera input in this study. Later in 1998, IRL was firstly proposed by Russell [27]. IRL aims to recover reward function from demonstrations and develops as a distinct category in IL. A year after, a formal definition of another important category—BC was proposed in [28].

BC works in a supervised learning fashion and seeks to learn a policy that builds a direct mapping between states and actions, then outputs a control strategy for control tasks, such as object manipulation, humanoid robotic control, and simulated games. Although BC demonstrates significant advantage on its simplicity and efficiency, it also suffers from various problems, such as “compounding error” [29] (a small action deviation would lead to significant state difference so that the agent would stick into unseen states) and “causal confusion” [30] (agent establishes incorrect causal relationship with the input patterns). In order to alleviate these problems, numerous subsequent approaches were proposed. In 2010, SMiLe [18] was proposed, it mixed the learner’s policy $\pi^*(s)$ with a new estimated policy $\hat{\pi}^*$ under a small fixed probability $\alpha$ as next policy, this method demonstrated better performance guarantee in practice and set up the foundation for the later proposed DAgger. DAgger was proposed by Ross et al. [29]. It updates the dataset in each iteration and trains a new policy in the subsequent iteration based on the updated dataset. DAgger alleviates the unseen scenario’s problem and achieves data efficiency compared to previous methods. Later research like [15], [31], and [25] were proposed to make improvements on DAgger. Besides DAgger and its derivatives, other methods also make contribution to the development of BC like maximum mean discrepancy (MMD)-IL [32], locally optimal learning to search (LOLS) [33], and learning by cheating (LBC) [8]. BC was applied into a wide range of low-level problems, one of the notable applications of BC was proposed by Abbeel et al. [34], they leveraged BC to train an agent that achieves autonomous helicopter aerobatics. Osa et al. [13] also applied BC into autonomous surgical knot-tying problem, which achieved online trajectory planning and updating in a dynamic system. Besides these real-world low-level applications, BC was also implemented into other research fields like scheduling [35] and cache replacement problem [25].

In terms of IRL, before 2008, IRL methods primarily recovered the reward function by maximizing the margin of (difference to the second best) actions [36], policies [37], or feature expectations [38]. However, the maximum margin approaches suffer from the “ill-posed” (different reward functions would result in the same action) problem as they introduce a bias into the learned reward function [39], this problem got alleviated by Ziebart et al. [40], who proposed Maximum Entropy IRL. Maximum Entropy IRL maximizes the entropy to obtain the least wrong distribution over behaviors and develops a convex procedure for good promise and efficient optimization. This method played a pivotal role in developing subsequent IRL and Generative Adversarial IL (GAIL). However, the IRL methods introduced above commonly hypothesize that the representation of the reward function is linear, limiting its application in nonlinear tasks. In 2016, Finn et al. [10] proposed a model-based IRL method called guided cost learning. The neural network is used to represent the nonlinear cost to enhance expressive power, combined with sample-based IRL to handle the unknown dynamics. Later in 2017, Hester et al. proposed deep Q-learning from demonstrations (DQfD) [19] which uses a small amount of demonstration to significantly accelerate the training process by doing pretraining to kick-off and learning from both demonstration and self-generated data. Later methods like reward regularized classification for apprenticeship learning (RCAL) [41], trajectory-ranked reward extrapolation (T-REX) [42], soft Q imitation learning (SQL) [43], self-imitation learning by planning (SILP) [44] make improvements on IRL, such as using ranking for reward inferencing and integrating self-supervised learning. As for the applications, IRL presents a competitive performance in high-level tasks such as game strategy but performs limited in high-dimensional tasks due to the cost of computational capacity.

Because of the benefits of deep learning and generative models, a novel learning framework was introduced in IL. The most representative approach could be GAIL, which was
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 2021, filtered by setting up each class and their abbreviation as keywords (like “BC,” only cover records within computer science).

proposed in 2016 by Ho and Ermon [45]. GAIL works in an adversarial fashion without the need of recovering the reward function, where the generator optimizes and outputs a policy iteratively, and the discriminator distinguishes the generated and expert policies. GAIL demonstrated state-of-the-art performance guarantee over both high-level discrete and low-level continuous control tasks. It became one of the active research fields in IL and its subsequent development shows a trend toward becoming a new category. GAIL also exposed some limitations, such as sample inefficient and unstable structure, later research like [23], [46], [47], and [48] was proposed to address these problems. Apart from optimizing GAIL, the research community also incorporated other generative models with IL inspired by GAIL or applied GAIL to other specific research fields. For example, Stadie et al. [46] proposed third-person imitation learning (TPIL) which partition the discriminator into feature extractor and classifier, and learned from third-person viewpoint demonstrations. The context translation and change of viewpoint facilitate the following research, including IfO [49]. IfO focuses on simplifying input to use raw video only (i.e., no longer use state-action pairs) and measure the difference between observation representation to obtain the policy, many following methods advocate this new setting, such as [50] and [42]. These methods measure the distance between observations to replace the need for ground-truth actions and widen the available input for training, for example, deviation between embeddings of the YouTube video frames was calculated in [20]. Other research fields such as meta-learning [51], [52], [53] and multiagent learning [54] are also thrived.

Fig. 1 presents some featured approaches and annual publication numbers for each class and focuses on the research after 2016. The methods in blue are the research topics integrating with IL based on time. Fig. 1 shows that the class of BC and IRL has maintained a stable increment in publications, while the novel research direction on the adversarial structured IL has grown rapidly due to the recent advance in other research fields like deep learning. We also notice that the annual publication of each category slightly drops after 2020, and we suspect that the reason for this phenomenon is that the progress of scientific research has been slowed down due to the lockdown around the world caused by the COVID-19 pandemic.

III. PRELIMINARY CONCEPTS

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

Definition 1: A state $s$ is a vector that describes the environmental context, the agent posture, velocity, spatial position, and corresponding information about its inner joints.

Definition 2: An action $a$ could be represented as a single discrete value under discrete tasks such as Atari games; or represented as a vector while simultaneous actions are needed, such as joint movement.

Definition 3: A policy is a function that maps a given state to a specific action or a probabilistic distribution of the action. Depending on the inclusion of the time parameter, a strategy can be further subdivided into stationary and nonstationary, respectively [55].

In IL, the demonstrated trajectories are commonly represented as pairs of states $s$ and actions $a$, sometimes other parameters such as high-level commands [7] and conditional goals [47] will also be included into the demonstration dataset.

Definition 4: A demonstration is a sequence of states or state-action pairs that represents the expert behavioral trajectory.
Many IL research assumes the optimality of the demonstrations, which means the demonstrations represents the global optimal solution, while some research focuses on the suboptimal demonstration, which means the demonstrations are generated from a nonexpert and could be locally optimal.

The way to collect the dataset could be either online or offline. **Offline learning** 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 IL to be more robust in a dynamic system. For example, in [12], [29], and [13], 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.

**Definition 5:** Occupancy measure is the unnormalized distribution of state-action pairs that an agent encounters by implementing the policy \( \pi \) [56]. For IRL and adversarial structured IL methods, occupancy measure is commonly used to represent the expectation of the policy.

One of the most common ways to calculate loss is **Kullback–Leibler (KL) divergence.** KL divergence measures the difference between two 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 [26] and [57] use KL divergence as the loss function as it could be useful when dealing with the stochastic policy learning problem. In the same vein, **Jensen–Shannon (JS) divergence,** as the smoothed version of KL divergence, measures the similarity between two distributions, i.e.,

\[
D_{JS}(p(x) \parallel q(x)) = \frac{1}{2} D_{KL}(p(x) \parallel M) + \frac{1}{2} D_{KL}(q(x) \parallel M)
\]

where \( M = (1/2)(p(x) + q(x)) \). JS divergence plays a significant role in adversarial structured IL, and unlike KL divergence, it is symmetric, i.e., \( D_{JS}(p(x) \parallel q(x)) = D_{JS}(q(x) \parallel p(x)) \).

For many methods, especially those under the class of IRL and adversarial structured IL, the environment is modeled as 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, an MDP is defined as a tuple \((\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, D, \mathcal{R})\), 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., \( s_{t+1} = P(\cdot | s_t, a_t) \), \( \gamma \in [0, 1) \) is the discount factor, \( D \) is the set of initial state distribution, and \( \mathcal{R} \) is the reward function \( \mathcal{S} \rightarrow \mathbb{R} \), and in IL setting, the reward function is not available. The Markov property assists IL 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 [35] and Markov games [58] is used in multiagent scenario [59].

The learning process of IL could be either on-policy or off-policy (there exists research using a hierarchical combination of these two [8]). **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 policies, the agent adopting off-policy will update their policy greedily and imitate action with the help of other sources. Some recent IL research such as [60], [61], and [62] advocates off-policy actor-critic architecture to optimize the agent policy and achieve sample efficiency comparing with on-policy learning.

### IV. MAIN RESEARCH TOPICS AND METHODS

#### A. Behavioral Cloning

**Behavioral Cloning** (BC) 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 BC uses negative log-likelihood loss to update the policy, i.e.,

\[
\arg\min_{\pi} \mathcal{L}(\pi) = -\frac{1}{N} \sum_{k=1}^{N} \log \pi(a_{k}|s_{k}).
\]

Algorithm 1 outlines the state-of-the-art BC 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 loss functions available to measure the differences (in most cases, the difference means the one-step deviation) such as \( \ell_1 \) loss, \( \ell_2 \) loss, KL divergence, and Hinge Loss. For example, when using KL divergence as the loss function, the objective policy could be obtained by minimizing the deviation between expert distribution \( q_{\pi_E} \) and induced distribution \( q(\pi) \), i.e.,

\[
\pi^* = \arg\min_{\pi} D_{KL}(q(\pi_E) \parallel q(\pi)).
\]

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” [18]. 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. Recent research has explicitly investigated the existing challenges of BC under the autonomous driving scenario [63], the mentioned problems could also negatively impact the performance of BC in other domain, and further research is still needed to alleviate these problems.

One of the significant BC methods is DAgger, which is a model-free BC method proposed by Ross et al. [29] 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 $\hat{\pi}^{i+1}$ with probability $\beta$ 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 startup 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 [15], [25], [31], [35], [64], and [65] 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 [18]. 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., inquiring expert correction could be expensive in interdisciplinary tasks). Later research such as SafeDAgger [66] and LazyDAgger [67] learned to ask the demonstrator for help actively and minimize the context switches to improve the interaction efficiency based on DAgger. Other recent methods such as [45] and [8] also 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 [31].

### B. Inverse RL

IRL was firstly proposed by Russell [27]. 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 abundant demonstrations and accurate controllers (such as position, velocity, force feedback, steering angle, etc.) are available [68], partial state information could be collected from these controllers and choosing BC methods could benefit from these state information and the abundant information to reproduce the target behaviors [8], [69].

IRL commonly assumes that the demonstrations are under MDP setting and since the reward $\mathbb{R}$ is unknown, the set of states is used to estimate the feature vector (i.e., $\phi : \mathcal{X} \mapsto [0, 1]^d$) 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 $\omega$ and policy parameter $\theta$. The reward function parameter $\omega$ is updated after the state-action visitation frequency $u$ are evaluated, and the way that $\omega$ is updated could vary. The most intuitive approach is maximizing the margin of expected value of the optimal action to the next-best action [36], i.e.,

$$\sum_{s \in S} Q^\pi(s, a^*) - \max_{a \in A} Q^\pi (s, a).$$

The following research developed alternatives based on this form, such as minimizing the margin of learned feature expectations while assuming that the reward function could be represented as a linear combination of the feature functions,
Algorithm 3 Classic Feature Matching IRL Method [36] and Its Derivatives

Require: The set of demonstrated trajectories \( D \);
1: Initialize reward function parameter \( \omega \) and policy parameter \( \theta \);
2: repeat
3: Evaluate current policy \( \pi_\theta \) state-action visitation frequency \( u \);
4: Evaluate loss function \( L \) w.r.t. \( u \) and the dataset \( D \) distribution;
5: (GCL [10]: Evaluate loss function \( L \) using non-linear IOC with stochastic gradients through the aggregated dataset;)
6: Update the reward function parameter \( \omega \) based on the loss function;
7: Update the policy parameter \( \theta \) in the inner loop RL method using the updated reward parameter \( \omega \);
8: ( [38]: Using RL, compute optimal policy \( \pi^i \) with reward \( R = (\omega^i)^T \phi \), then compute \( \mu^i = \mu(\pi^i) \))
9: until
10: return optimized policy representation \( \pi_\theta \);

i.e., \( \min ||\mu_E - \mu|| \) [38]. However, inferring reward function through the margin suffers from the “ill-posed” problem. “Illoosed” means the many different cost functions could lead to the same action. In 2008, Ziebart et al. [40] incorporated maximum entropy principle with IRL to alleviate this problem. It updated \( \omega \) by maximizing the likelihood of the demonstration over maximum entropy distribution, i.e.,

\[
\omega^* = \arg \max_{\omega} \sum_{t \in D} \log P(\tau || \omega).
\]

However, the above methods are constrained by the linear formulation of the problem, which could be insufficient to solve some of the real-world problems. On the other hand, the policy parameter \( \theta \) is updated in the inner loop RL 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. These two aspects limit IRL’s application in real-world scenario. Research such as [10], [42], [43], [70], [71], and [44] was proposed to alleviate the problem by using the nonlinear approximator and integrating other research fields like self-supervised learning and ranking for inference.

Self-supervised learning means learning a function from a partially given context to the remaining or surrounding context. Nair et al. [14] could be one of the earliest researchers who adopt self-supervised learning into IL. One important problem that integrating self-supervised learning with IL 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. [14] 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. [73] proposed self-IL, which tries to leverage past good experience to get better exploration result. The proposed method takes an initial policy as input. It then iteratively uses the current policy to generate trajectories, calculates the accumulated return value \( R \), update the dataset

\[
D \leftarrow D \bigcup \{(s_i, a_i, R)\}_{i=0}^T
\]

and finally uses the deviation between accumulated return and the agent estimate value \( R - V_\theta \) to optimize the policy parameter \( \theta \). 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 IL (SIL) also tries to learn from its good experience but in a different structure. SIL creatively uses voice instruction in the IL process. One language encoder is used to extract textual feature \( \{o_t\}_{t=1}^m \) and an attention-based trajectory encoder long short-term memory (LSTM) [56] is used to encode the previous state action as a history context vector from visual state \( \{v_j\}_{j=1}^m \), i.e.,

\[
h_i = \text{LSTM}([v_j, a_{j-1}], h_{i-1}).
\]

Then visual context \( c_{\text{visual}} \) and language context \( c_{\text{text}} \) 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. In [75], self-supervised learning was adopted to pretrain a low-dimensional feature encoding for high-dimensional IRL problems and then leverage preferences over demonstrations to perform efficient Bayesian inference.

C. Generative Adversarial IRL

In order to mitigate problems in BC and IRL, Ho and Ermon [45] proposed a novel general framework called GAIL in 2016. GAIL builds a connection between GAN [1] and maximum entropy IRL [40]. Inheriting from the structure of GAN, GAIL consists of a generative model \( G \) and a discriminator \( D \), while \( G \) generates data distribution \( \rho_G \) integrating with true data distribution \( \rho_x \) to confuse \( D \). GAIL works in an iterative fashion, and the formal objective of GAIL could be denoted as

\[
\min \pi \max_{D \in \mathcal{D}(0,1)^{S \times A}} \mathbb{E}_{\tau \sim \pi} \left[ \log (D_m(s, a)) \right] + \mathbb{E}_{\tau \sim \pi} \left[ \log (1 - D_m(s, a)) \right].
\]

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

\[
\mathbb{E}_{\tau \sim \pi} \left[ \nabla_{\omega} \log (D_m(s, a)) \right] + \mathbb{E}_{\tau \sim \pi} \left[ \nabla_{\omega} \log (1 - D_m(s, a)) \right]
\]

and maximize this equation with respect to \( D \). Then adopting the TRPO [26] with the cost function \( \log (D_{m+1}(s, a)) \) to update the policy parameter \( \theta \) and minimize the above function.
Algorithm 4 GAIL [45] and Its Derivatives

Require: Expert trajectories $T_E \sim \pi_E$, initial policy and discriminator parameter $(\theta_0, \omega_0)$
for $i = 0, 1, 2, \ldots$ do
    Sample trajectories $t_i \sim \pi_{\theta_i}$.
    (GoalGAIL [47]: relabel the transitions using future HER strategy, then add annealed GAIL reward for updating policy parameter.)
    Update the discriminator parameters $\omega_i$ to $\omega_{i+1}$.
    Update the policy parameter $\theta_i$ to $\theta_{i+1}$.
(RAIL [60]: Using the actor-critic structure for generator part and updating the parameters through replay buffer.)
end for

| TABLE I DIFFERENT KINDS OF DERIVATIVE ON GAIL |
|-------------------------------|-------------------|
| GAILs                          | Methods           |
| Alleviate existing issues of GAIL | MGAIL [77], InfoGAIL [78], GoalGAIL [47], TRGAIL [48], DGAIL [79] |
| Apply to other research question | MAGAIL [59], GAIO [80], FAIL [81], PS-GAIL [57], MA-GAIL [82] |
| Other generative model         | Diverse GAIL [23], GIRL [83], LISA [35], GIRL [83] |

with respect to $\pi$, combining with a causal entropy regularizer controlled by non-negative parameter $\lambda$, i.e.,

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

where

$$Q(\bar{s}, \bar{a}) = \widehat{\mathbb{E}}_\pi [\log(D_{a_{\pi_{h+1}}}(s, a))|s_0 = \bar{s}, a_0 = \bar{a}]$$

The abstract training process is presented in Algorithm 4. By adopting TRPO, the policy could be more robust and stable to the noise in the policy gradient. Unlike DAgger and other previous 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-offs are the on-policy training process involves frequent interaction with the environment. Its sample efficiency could be further improved, and the framework is more fragile when encountering the saddle point problem. As for the first problem, the authors suggested initializing the policy with BC to reduce the amount of environmental interaction and adopt an actor-critic off-policy learning framework, that the research community proves its effectiveness [60]. As for the second problem, recent research such as [76] tries to alleviate this problem by formulating the distribution-matching problem as an iterative lower-bound optimization problem.

Inspired by GAIL’s presence, much research was proposed to make further development on GAIL (see Table I) 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, model-based generative adversarial IL (MGAIL) [77] uses an advanced forward model to make the model differentiable so that the Generator could use the exact gradient of the Discriminator. InfoGAIL [78] modifies GAIL by adopting Wasserstein generative adversarial network (WGAN) [84] instead of GAN. Other recent works like GoalGAIL [47], task-achievement reward generative adversarial IL (TRGAIL) [48], and deterministic generative adversarial IL (DGAIL) [79] are all making improvement on GAIL by combining with other methods like hindsight relabeling and Deep Deterministic Policy Gradient (DDPG) [85] 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 [81], 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 mini-max theory, which is used to determine the next time step policy $\pi_h$ under the assumption that $[\pi_1, \pi_2, \ldots, \pi_{h-1}]$ is learned and fixed. GAIL is also applied into the other research areas, such as multiagent settings [54], [57], [59] and IFO settings [80] 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 [23], 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 generative intrinsic reward learning (GIRL) [83] also achieves the outstanding performance from limited demonstrations using VAE.

D. Imitation From Observation

Most methods introduced above use 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. Stadie et al. [46] proposed TPIL which learned from third-person viewpoint demonstrations. TPIL alleviates the input requirements and introduces an approach similar to how people learn, as the 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. The context translation and change of viewpoint facilitate the following research, including IFO [49]. Comparing with traditional IL methods, IFO 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 accessible and the input data is raw image frames. IFO methods could be roughly decomposed into two parts: perception and control [88]. The perception part commonly co-opted the recent advancements in deep learning and image feature extraction. Methods such as convolutional neural network and encoder-decoder structure
are used to translate the raw image into the state information. Although leveraging state information to train the policy has been investigated before, imitating from raw image input is nontrivial as using the raw images needs to solve problems like time alignment, viewpoint deviation, and kinematic shifting problems. Assuming the perception part works well and clear state information is extracted from the raw images, the next challenging part is to learn imitation policy from the state information, which could be achieved by learning an inverse dynamic model to infer the action or making use of the adversarial structure. The prevalent approach to learn imitation policy is measuring the deviation between states to compose the reward function and obtaining the behavior policy by RL, for example, Liu et al. [49] uses TRPO [26] for the simulation experiments.

After IfO being proposed, measuring observation distance to replace the ground truth action becomes a prevalent setting in IL. In Table II, we present some of the research advocate this new insight and detail the approaches they used to calculate deviation between target observation and observed observation. BC, IRL, and adversarial structured IL start to adopt this setting to simplify the input and alleviate the existing problems. For example, in [86], multiviewpoint SIL 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 alleviate the kinematic shifting problem and reproduce the demonstrated behaviors. Similarly, in [49], the input raw video needs to be prealigned to ensure the scenario encountered over time is the same. While in [20], 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 visual input together with an audio embedding to align the time between trajectories. Not only did Aytar et al. [20] take advantage of the verbal instruction, but other existing research such as Wang et al. [74] also leveraged the natural language for navigation tasks to facilitate the learning process. Besides audio input, other available interaction forms are integrated with IL. For example, expert preferences are used IRL domain to infer reward functions [71], [90].

V. CATEGORIZATION AND FRAMEWORKS

In this section, four kinds of taxonomies are presented (see Fig. 2). The first two taxonomies (BC versus IRL and model-free versus model-based) follow the classifications in [68], [88], and [55] and the other two (Low-level Manipulation Tasks versus High-Level Tasks and BC versus IRL versus adversarial structured IL) are newly proposed taxonomies.

A. BC Versus IRL

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 virtual simulations in the environment with less invention and less amount of states [39].

| Publication | Domain | Input | How? |
|-------------|--------|-------|------|
| IFO [49]    | Real-world & simulated robotics | Demonstration video | using squared loss to compare encoded source observation and target observation |
| BCO [50]    | classic control & simulated robotics | State-only demonstration | maximizing the likelihood of distribution induced by the inverse dynamic model |
| TCN [86]    | Real-world & simulated robotics | unlabeled multi-viewpoints video | L2 loss between video TNC embedding and observation TNC embedding |
| One-shot IFO [20] | 2D gameplay | Youtube video | Temporal difference between two embeddings, then applied to cross-entropy loss |
| Zero Shot Visual Imitation [87] | Real-world robotic | sparse landmark image | Cross-entropy loss between actions predicted by input images |
| IFO survey [88] | — | — | — |
| Imitating Latent Policies from Observation [21] | 2D gameplay | noisy state observation | Combination loss between Squared Euclidean distance from states, generative model, and the difference between states |
| GAFO [80]   | classic control & simulated robotics | State-only demonstration | using similar loss function in GAIL to update discriminator |
| OPOLO [89]  | simulated robotics | state-action pairs from expert policy | using dual-form of the expectation function and adversarial structure to derive an upper-bound of IFO objective |
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. The inner loop RL could also cause IRL methods to be impractical in real-world problems. For the computational problem, recent development in GPU gradually alleviates the problem of high-dimensional computation; for the technology aspect, recent algorithms like TRPO [26] and attention models [91] provide more robust and efficient approaches for IRL methods; as for the sparse reward function, Hindsight Experience Replay (HER) [92] is commonly adopted for this problem. On the other hand, BC also suffers from the “compounding error” [29] 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 [23], [93], and [64].

### B. Model-Based Versus 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 [45] was proposed, most IRL methods were developed in the model-based setting because they involve iterative algorithms to evaluate the environment, while BC methods were commonly model-free since the low-level controllers are 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. Understanding the characteristics of each class could allow us to make targeted choices when faced with different problems. For example, in robotics, the equipment is commonly precise, the spatial position, velocity, and other parameters could be easily obtained, and the system dynamics might provide 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 best choice of model-free or model-based design depends on the tasks.

Table III lists some of the recent research in IL under the classic two taxonomies. From Table III, we can see that most BC methods belong to model-free, which means the BC methods take less consideration on the environment dynamics, while there is a more significant proportion of IRL methods belonging to model-based as the accuracy of IRL is dependent on the quality of modeling dynamics. It is also interesting to note that the choice of the loss function is closely related to the action representation. The loss function tends to be a cross-entropy loss when the task is under a discrete

| Class                  | Example and Publication | Model-based or Model-Free | Action Representation | Loss Function | Evaluation |
|------------------------|-------------------------|---------------------------|-----------------------|---------------|------------|
|                        |                         |                           |                       |               | Multi-tasks | End-to-End | Sub-optimal |
| Few(One)-shots learning [52] | Model-Free             | continuous or discrete    | L2 or cross-entropy   | ✓             | ✓          | ✓          |
| Input optimization [8]  | Model-Free              | continuous                | L1                    | ✓             | ✓          | ✓          |
| Latent policy learning [21] | Model-based            | discrete                  | L2 and cross-entropy  | ✓             | ✓          | ✓          |
| Real-world application [16] | Model-Free            | continuous                | mixed                 | ✓             | ✓          | ✓          |
| Improve efficiency [42] | Model-Free              | discrete                  | cross-entropy         | ✓             | ✓          | ✓          |
| sparse reward [20]      | Model-Free              | discrete                  | cross-entropy         | ✓             | ✓          | ✓          |
| high-dimensional [10]   | Model-based             | continuous                | squared hinge         |              |            |            |
| Inverse dynamic model [14] | Model-based            | continuous                | not mentioned         | ✓             | ✓          | ✓          |
BC methods belong to model-free and recent BC methods seek to answer the combination of the above questions. Most require input data. More research could be done that leveraging suboptimal demonstration, which tries to lower the representation by directly mapping the pixels to actions; and end structure, which provides a more learnable intermediate one of these three active research questions: multitask learning, We also noticed that most of the research focuses on solving one of these three active research questions: multitask learning, which tries to eliminate the task-specific restriction; end-to-end structure, which provides a more learnable intermediate representation by directly mapping the pixels to actions; and leveraging suboptimal demonstration, which tries to lower the requirement for input data. More research could be done that seeks to answer the combination of the above questions. Most BC methods belong to model-free and recent BC methods mainly focus on the topics such as: autonomous driving [7]; meta-learning that the agent is learning to learn by pretraining on a broader range of behaviors [52]; and combining BC with other techniques like virtual reality (VR) equipment [16]. On the other hand, IRL methods are more likely to be model-based and recent IRL methods mainly focus on the topics such as: extending GAIL with other methods or problem settings [47]; recovering reward function from raw videos [20]; and developing more efficient model-based IRL approaches by using the current development in RL like TRPO [45], [65], [89], [94], [95] and HER [14], [44], [60], [63].

C. Low-Level Tasks Versus High-Level Tasks

This section introduces a novel taxonomy, which divides IL into manipulation tasks and high-level tasks according to their evaluation approach and could be further subdivided employing other criteria such as model-based or model-free. The idea is inspired by a control diagram (See Fig. 3) in [68]. Although some IL benchmark systems are proposed, such as [109], there is still no widely accepted one. In this case, the focus and evaluation approaches could vary from method to method, ranging from performance in sparse reward scenarios to the smoothness of autonomous driving in a dynamic environment. The diversity in evaluation could lead to problems like existing methods might be compared under different task levels, and low-quality research could be more likely to be published, which is not conducive to new research proposals. In addition to the variety in tasks and objectives, different input data modalities also increase the difficulties of proposing an overall benchmark for comparing IL methods. Research such as [110] was proposed to benchmark and justify the use of data for high-level discrete video games. This taxonomy could help to define a clearer boundary and reduce the complexity of designing appropriate benchmarks from a policy performance perspective. Readers also benefit from this taxonomy, as they can more easily comprehend the target domain of a method and related methods of its kind suitable for comparison.

The low-level tasks could be either real-world or virtual, and are not limited to robotic manipulation and autonomous driving problems. Common low-level controllers in these tasks provide feedback about the agent position, velocity, force, joint angle, etc. The robotic task can be object manipulation by robotic arm like PR2, KUKA robot arm, and simulation tasks commonly experimented on OPEN artificial intelligence (AI) gym [see Fig. 4(e)], MuJoCo simulation platform [see Fig. 4(d)], 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 and NoCrash [63] benchmark system [see Fig. 4(c)]; 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 ignores the collision and traffic rules violation. Besides simulation, some research experiments in real-world use of cars [6] and smaller remote-controlled cars [7] [see Fig. 4(f)] and other kinds of equipment are also used like remote control helicopter [34].

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 2-D Atari games on OpenAI Gym, such as MontezumaRevenge is commonly evaluated for performance on hard exploration and sparse reward scenario [see Fig. 4(a)]. Others are evaluated on 3-D games like Grand Theft Auto V (GTAV) or Minecraft for evaluation [see Fig. 4(b)]. 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, and this may help to design a unified evaluation metric for IL.

From Fig. 3, 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 [13]. As for the low-level controller, it learns a policy to reproduce the primitive behavior, such as [86], 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 [100], 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 IV lists some of the prevalent benchmark system and recent research under this taxonomy. The majority of current imitation methods use OpenAI and Mujoco simulation to evaluate the methods, and only a small number of methods involve real-world implementation. Compared with high-level planning, low-level manipulation tasks seem to be more popular to be evaluated. Since RL 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 nontrivial, because RL can be time-consuming on the vast state and action space for the 3-D tasks or hard exploration games. From Table IV, we can conclude that proposing an overall benchmark for high-level tasks seems more achievable than low-level ones. The benchmark for each task is more mature, and the score for a specific high-level task could be more accessible than the low-level tasks. But the input demonstration’s form, quantity, and other factors need to be unified for a specific question. On the other hand, many tasks for low-level evaluation are self-designed such as rope manipulation and real-world object operation, which could be impractical to make a parallel comparison between methods.

D. BC Versus IRL Versus Adversarial Structured IL

This taxonomy is extended from the first taxonomy (BC versus IRL). This new taxonomy divides IL into three categories: BC, IRL, and adversarial structured IL, which could be further subdivided by other taxonomies’ criteria. With the recent development of IL, adversarial structured IL brings new insights for researchers and alleviates existing problems, such as high-dimensional problems. Inspired by the presence of GAIL, many recent papers adopt this adversarial structure, and inevitably, GAIL becomes the baseline for comparison. Nevertheless, this is not enough to establish an independent category in IL. The true reason making it distinguishable is
that GAIL does not belong to either BC or IRL. Although adversarial structured IL has a close connection with RL, most adversarial structured IL does not recover the reward function. Apart from the involvement of reward function, adversarial structured IL obtains a policy from the adversarial structure, and most of them are model-free, while IRL methods learn under a more linear process and commonly have a forward model to evaluate the system dynamics. As adversarial structured IL becomes more and more prevalent, the taxonomy of IL needs to be more specific. In this survey, GAIL and its derivations are separated from the traditional IRL category and classified as adversarial structured IL. 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.

Fig. 5 roughly evaluates the proposed three classes through six aspects which are commonly compared between research. Since existing methods target different criteria and evaluate through various tasks, the overall performance is hard to quantify and rank.

In terms of efficiency, we mainly focus on environmental interaction, computation, and expert data. The expert data includes considerations of the demonstration quantity and the interactions with the expert during the learning process, such as active queries and passive interventions. BC methods commonly take advantage of interaction with an expert and sufficient demonstration while having 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 of system dynamic makes IRL suffer from high computational cost, but IRL methods might inquire the expert during training to speed up the inference process. Adversarial structured IL methods also involve frequent interaction with the environment when they iteratively update the policy parameter and discriminator parameter. Nevertheless, adversarial structured IL is query-free, eliminating interaction with the expert, and the use of off-policy frameworks significantly improves its sample efficiency. 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 focus on robustness when demonstrations are suboptimal (including the consideration of noise in demonstration) and robustness in a complex environment (including the high-dimensional and dynamic environment). BC methods commonly have better performance in high-dimensional space so that they are widely evaluated on real-world robotics, while the performance in a 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. However, the recovered reward function would benefit the agent to make predictions in the 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 [42], [47], and [60] in both IRL and adversarial structured IL make progress in suboptimal demonstration problems, we give them the same rank in the evaluation of robustness on suboptimal demonstration.

In terms of explainability and transferability, BC works in a comprehensive approach that maps between states and actions.
However, the learned policy is less transferable compared to the succinct reward functions, as it is more sensitive to the changes in the environment [39]. This problem also limits adversarial structured IL approaches’ performance for this metric. Research such as [115] explored the explainability of GAIL on geographical tasks, but future works are still needed. IRL presents outstanding capability in explainability and transferability, as the recovered reward function ensures the interpretation of the agent decision-making and robustness to domain shifting.

VI. CHALLENGES AND OPPORTUNITIES

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

Diverse Behavior Learning: Most IL methods use task-specific datasets to train and reproduce a single behavior. However, an agent with multiple skills seems more favorable in practice. Wang et al. [23] validate the feasibility of achieving multiple skills learning. They combined adversarial structured IL with VAE, and the experiment result shows that the agent could work on various control problems. Following research such as [95] also contributes to this problem, but this is still an open challenge. Other methods could be adopted to optimize IL, such as real-world experience and transfer learning might help the agent to minimize the reality gap and learn from similar tasks so that the training process could achieve both efficiency and practicality.

Suboptimal 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, while the suboptimal demonstrations from amateurs are more accessible. Existing research like [42], [47], and [116] have shown the possibility of using suboptimal demonstrations for training. The agent could learn from the suboptimal demonstrations by setting alternative goals from the demonstration trajectories or ranking the dataset for reward inference. But this problem is still challenging. The quantity could be an ignored advantage of suboptimal demonstrations, and more information could be extracted from the abundant suboptimal data and facilitate the agent learning process.

Imitation Not Just From Demonstration: Current IL methods commonly use expert demonstrations and raw videos to reproduce the desired behavior. However, there are abundant interaction forms from the human expert that could be used in IL. They could be either explicit types such as corrective interventions and preferences, or implicit forms like gesture and facial expression of the expert [117], [118]. These explicit/implicit interactions could be applied individually or hierarchically into IL. With the help of these extra information, IL agent might obtain more satisfactory policy performance.

Better Representation: Good policy representation could lead to data-efficiency and computation-efficiency. Finding better policy representation is still an active research topic for IL. For example, Lynch et al. [119] achieved more compact plan representation in latent space to learn high-level behaviors from plays. Besides policy representation, how to represent the demonstration is another problem in IL. Demonstration representation keeps exploring new parameters such as high-level command and has evolved and improved from the initial state-action pair to the current popular one that only requires state observation. Future research on representations of both policy and demonstrations is needed to achieve better efficiency and expression.

VII. CONCLUSION

IL 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 IL methods make contributions 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 weigh 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 IIO 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 suboptimal 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] I. Goodfellow et al., “Generative adversarial networks,” Commum. ACM, vol. 63, no. 11, pp. 139–144, 2020.
[2] D. A. Pomerleau, “ALVINN: An autonomous land vehicle in a neural network,” in Proc. Adv. Neural Inf. Process. Syst., D. S. Touretzky, Ed. Burlington, MA, USA: Morgan-Kaufmann, 1989, pp. 305–313.
[3] D. A. Pomerleau, “Efficient training of artificial neural networks for autonomous navigation,” Neural Comput., vol. 3, no. 1, pp. 88–97, May 1991.
[4] L. George, T. Buhet, E. Wirbel, G. Le-Gall, and X. Perrotton, “Imitation learning for end to end vehicle longitudinal control with forward camera,” 2018, arXiv:1812.05841.
[5] P. M. Kebria, A. Khosravi, S. M. Salakan, and S. Nahavandi, “Deep imitation learning for autonomous vehicles based on convolutional neural networks,” IEEE/CAA J. Autom. Sinica, vol. 7, no. 1, pp. 82–95, Jan. 2020.
[6] Y. Zhou, R. Fu, C. Wang, and R. Zhang, “Modeling car-following behaviors and driving styles with generative adversarial imitation learning,” Sensors, vol. 20, no. 18, p. 5034, Sep. 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/18/5034
[7] F. Codella, M. Müller, A. Lopez, V. Kolunt, and A. Dosovitskiy, “End-to-end driving via conditional imitation learning.” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2018, pp. 4693–4700.
[8] D. Chen, B. Zhou, V. Kolunt, and P. Krähenbühl, “Learning by cheating,” in Proc. Conf. Robot Learn., 2020, pp. 66–75.
[28] M. Bain and C. Sammut, “A framework for behavioural cloning,” in Proc. Int. Conf. Mach. Learn., 2016, pp. 49–58.

[13] T. Osa, N. Sugita, and M. Mitsuishi, “Online trajectory planning and force control for imitation of surgical tasks,” IEEE Trans. Autom. Sci. Eng., vol. 15, no. 2, pp. 675–691, Apr. 2018.

[20] Y. Aytar, T. Pfaff, D. Budden, T. Paine, Z. Wang, and N. de Freitas, “Playing hard exploration games by watching YouTube,” in Proc. Adv. Neural Inf. Process. Syst., vol. 29, no. 7, pp. 1755–1763, 2016.

[22] A. Hussein, E. Elyan, and M. M. Gaber, “Deep imitation learning for 3D navigation tasks,” Neural Comput. Appl., vol. 29, no. 7, pp. 2050–2060, 2018.

[49] A. Buhler, A. A. Coates, and S. Levine, “Model-agnostic meta-learning with noise mechanism for one-shot learning,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2018, pp. 6237–6247.

[40] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, “Maximum entropy inverse reinforcement learning,” in Proc. AAM, vol. 8, Chicago, IL, USA, 2008, pp. 1433–1438.

[47] Y. Ding, C. Florensa, M. Phielipp, and P. Abbeel, “Goal-conditioned imitation learning,” 2019, arXiv:1904.03732.

[43] S. Reddy, A. D. Dragan, and S. Levine, “SQL: Imitation learning via reinforcement learning with sparse rewards,” 2019, arXiv:1905.11108.

[46] B. C. Stadie, P. Abbeel, and I. Sutskever, “Goal-conditioned imitation learning,” 2019, arXiv:1906.05838.

[45] B. Piot, M. Geist, and O. Pietquin, “Bridge the gap between imitation learning and inverse reinforcement learning,” IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 8, pp. 1814–1826, Aug. 2017.

[41] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, “Maximum entropy inverse reinforcement learning,” in Proc. AAM, vol. 8, Chicago, IL, USA, 2008, pp. 1433–1438.

[48] A. Kinose and T. Taniguchi, “Integration of imitation learning using reinforcement learning,” in Proc. ICML, vol. 1, pp. 6237–6247.

[44] B. Piot, M. Geist, and O. Pietquin, “Bridge the gap between imitation learning and inverse reinforcement learning,” IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 8, pp. 1814–1826, Aug. 2017.

[42] J. Ho and S. Ermon, “Generative adversarial imitation learning,” in Proc. Adv. Neural Inf. Process. Syst., D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2016, pp. 4565–4573. [Online]. Available: http://papers.nips.cc/paper/6391-generative-adversarial-imitation-learning.pdf

[49] A. Buhler, A. A. Coates, and S. Levine, “Model-agnostic meta-learning with noise mechanism for one-shot learning,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2018, pp. 1118–1125.

[50] F. Torabi, G. Warnell, and P. Stone, “Behavioral cloning from observation,” 2018, arXiv:1805.01954.

[51] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 1126–1135.

[52] Y. Liu, A. Gupta, P. Abbeel, and S. Levine, “Imitation from observation: Learning to imitate behaviors from raw video through context translation,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2018, pp. 1118–1125.

[53] B. C. Stadie, P. Abbeel, and I. Sutskever, “Third-person imitation learning,” 2017, arXiv:1703.01703.

[54] Z. Hu, Z. Gan, W. Li, J. Z. Wen, D. Zhou, and X. Wang, “Two-stage model-agnostic meta-learning with noise mechanism for one-shot learning,” IEEE Access, vol. 8, pp. 182720–182730, 2020.

[55] E. Zhan, S. Zheng, Y. Yue, L. Sha, and P. Lucey, “Generating multi-agent trajectories using programmatic weak supervision,” 2018, arXiv:1803.07612.
data mining, fairness, accountability, transparency and ethics in AI (FATE), and Data61, CSIRO, Canberra ACT, Australia. His research interests include and interpretable deep learning.

A. Lemme, Y. Meirowitch, M. Khansari-Zadeh, T. Flash, A. Billard, and J. J. Steil, “Open-source benchmarking for learned reaching motion generation in robotics,” J. Paladyn Behav. Robot., vol. 6, no. 1, pp. 1–152, Jan. 2015, doi: 10.1515/pjbr-2015-0002.

A. Kanervisto, J. Pussinen, and V. Hautamaki, "Benchmarking end-to-end behavioural cloning on video games," in Proc. IEEE Conf. Games (CoG), Aug. 2020, pp. 558–565.

W. H. Guiss et al., “MineRL: A large-scale dataset of minecraft demonstrations.” 2019. arXiv:1907.13440.

A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “ImageNet Classification with Random Fully-Connected Layers.” 2014. arXiv:1409.1556.

E. Todorov, T. Erez, and Y. Tassa, “MuJoCo: A physics engine for model-based control,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Oct. 2012, pp. 5026–5033.

G. Brockman et al., “OpenAI gym,” 2016, arXiv:1606.01540.

M. Pan, W. Huang, Y. Li, X. Zhou, and J. Luo, “XGAIL: Explainable generative adversarial imitation learning for explainable human decision analysis,” in Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2020, pp. 1334–1343.

L. Song, J. Liu, M. Sun, and X. Shang, “Weakly supervised group mask network for object detection,” Int. J. Comput. Vis., vol. 129, no. 3, pp. 681–702, Mar. 2021, doi: 10.1007/s11263-020-01397-w.

Y. Cui et al., “Understanding the relationship between interactions and outcomes in human-in-the-loop machine learning,” in Proc. 13th Int. Joint Conf. Artif. Intell., Aug. 2021, pp. 4382–4391.

H. J. Jeon, S. Milli, and A. Dragan, “Reward-rational (implicit) choice: A unifying formalism for reward learning,” in Proc. Adv. Neural Inf. Process. Syst., vol. 33, 2020, pp. 4415–4426.

C. Lynch et al., “Learning latent plans from play,” in Proc. Conf. Robot Learn., 2020, pp. 1113–1132.

Boyan Zheng received the master’s degree in computer science from The Australian National University, Canberra, ACT, Australia, in 2020. He is currently pursuing the Ph.D. degree at the UTS Human Centred Artificial Intelligence (AI) Research Laboratory, School of Computer Science, Faculty of Engineering and IT, University of Technology Sydney, Sydney, NSW, Australia.

His research direction focuses on imitation learning, ethics of AI, AI fairness, AI explainability, behavior analytics, and human–computer interaction.

Sunny Verma received the Ph.D. degree in computer science from the University of Technology Sydney, Ultimo, NSW, Australia, in 2020. He is currently a Data Scientist with the Commonwealth Bank of Australia, Sydney, NSW, Australia, developing machine learning models for financial crime. He previously worked as a Post-Doctoral Research Fellow on identifying cyberattacks with Human-Bot Teams, AUSMURI, Macquarie University, Sydney. Prior to that, he worked with the Data Science Institute, University of Technology Sydney; and Data61, CSIRO, Canberra ACT, Australia. His research interests include data mining, fairness, accountability, transparency and ethics in AI (FATE), and interpretable deep learning.

Jianlong Zhou (Senior Member, IEEE) received the Ph.D. degree in computer science from The University of Sydney, Camperdown, NSW, Australia, in 2011.

He is currently an Associate Professor with the School of Computer Science, Faculty of Engineering and IT, University of Technology Sydney, Sydney, NSW, Australia.

Fang Chen is currently a prominent Leader in artificial intelligence (AI)/data science with international reputation and industrial recognition, and the Leader of the Data Science Institute, University of Technology Sydney, Sydney, NSW, Australia. She has more than 300 refereed publications, including several books. She has filed more than 30 patents in Australia, USA, Canada, Europe, Japan, South Korea, Mexico, and China. Her main research interests include behavior analytics, especially in data mining, machine learning, human and system performance prediction and evaluation.

Prof. Chen was the Winner of the “Oscars” of Australian Science and the 2018 Australian Museum Eureka Prize for Excellence in Data Science.

Ivor W. Tsang (Fellow, IEEE) is currently an ARC Future Fellow and a Professor of artificial intelligence (AI) with the University of Technology Sydney (UTS), Sydney, NSW, Australia. He is also the Research Director of the Australian Artificial Intelligence Institute, Melbourne, VIC, Australia. His research interests include transfer learning, generative models, computer vision, and big data analytics for data with extremely high dimensions.