applied sciences

MDPI

Review

Data-Driven Policy Learning Methods from Biological Behavior: A Systematic Review

Yuchen Wang [1,2,*], Mitsuhiro Hayashibe [1] and Dai Owaki [1,*]

[1] Department of Robotics, Graduate School of Engineering, Tohoku University, Sendai 980-8579, Japan; mitsuhiro.hayashibe.e6@tohoku.ac.jp
[2] Japan Society for the Promotion of Science, Tokyo 102-0083, Japan
* Correspondence: wang.yuchen.q5@dc.tohoku.ac.jp (Y.W.); owaki@tohoku.ac.jp (D.O.)

Abstract: Policy learning enables agents to learn how to map states to actions, thus enabling adaptive and flexible behavioral generation in complex environments. Policy learning methods are fundamental to reinforcement learning techniques. However, as problem complexity and the requirement for motion flexibility increase, traditional methods that rely on manual design have revealed their limitations. Conversely, data-driven policy learning focuses on extracting strategies from biological behavioral data and aims to replicate these behaviors in real-world environments. This approach enhances the adaptability of agents to dynamic substrates. Furthermore, this approach has been extensively applied in autonomous driving, robot control, and interpretation of biological behavior. In this review, we survey developments in data-driven policy-learning algorithms over the past decade. We categorized them into the following three types according to the purpose of the method: (1) imitation learning (IL), (2) inverse reinforcement learning (IRL), and (3) causal policy learning (CPL). We describe the classification principles, methodologies, progress, and applications of each category in detail. In addition, we discuss the distinct features and practical applications of these methods. Finally, we explore the challenges these methods face and prospective directions for future research.

Keywords: policy learning; behavior strategy; imitation learning; inverse reinforcement learning; causal inference

Citation: Wang, Y.; Hayashibe, M.; Owaki, D. Data-Driven Policy Learning Methods from Biological Behavior: A Systematic Review. *Appl. Sci.* **2024**, *14*, 4038. https://doi.org/10.3390/app14104038

Academic Editor: Paolino Di Felice

Received: 11 April 2024
Revised: 7 May 2024
Accepted: 7 May 2024
Published: 9 May 2024

1. Introduction

Humans and animals can respond instantaneously when faced with complex and unpredictable environments by developing different strategies to adapt to the needs of any environment. Such behavioral strategies are not derived solely from responses to stimuli received by the senses but combined with decisional control from the brain/nervous system, enabling humans and animals to respond similarly to similar environments and stimuli [1]. The development of behavioral strategies is a process by which the brain continuously organizes its actions in response to environmental changes. The behavioral strategies enable them to maximize the use of their conditions and environmental resources to effectively solve the difficulties they face in the current environment.

Understanding behavioral strategies is useful for taking full advantage of flexibility and adaptability. Some examples include the development of autonomous driving systems by studying drivers' driving behavior habits [2], automatic robot navigation algorithms based on pedestrian movement strategies in crowds [3], dexterous robot movement control learned from animal locomotor behavior [4], and the design of virtual reality (VR) games by studying human motor behavior [5]. Extracting behavioral strategies facilitates agent training and further development in understanding animal behaviors. Therefore, understanding behavioral strategies from multiple perspectives in the biological behavior and engineering domains is of significant importance.

Previously, training agents to develop strategies for adapting to environments was mainly achieved through defined behavioral rules, where the behavioral patterns are summarized by human observations of behavior, e.g., motor coordination can be achieved by artificially predefining phase relationships between gait legs. This approach has been widely used owing to its simplicity and effectiveness. However, human judgments are low-dimensional and subjective [1]. With the increasing environmental adaptability and flexibility requirements, the limitations of relying on human-designed behavioral strategy models have gradually emerged. This is because behavior is inherently complicated. Even a single behavior is controlled by sophisticated neurological and motor systems [6], which makes the representation and modeling of behavioral strategies challenging.

However, with the development of machine learning techniques, neural networks with stronger representational capabilities make behavioral policy learning possible. reinforcement learning (RL) iteratively trains agents to learn strategies through pre-designed reward functions. It can be used in relatively complex environments, but it requires human experts with a very good understanding of the task to design hard-coded reward functions. Thus, for relatively complex biological behaviors (i.e., running and swimming), it is difficult to represent them intuitively in mathematical form. In addition, the data-driven policy-learning approach learns policies through different methods. It attempts to reproduce behaviors from quantitative behavioral data, emphasizing natural human or animal behavior as the basis. This approach may have problems relying on the quality of behavioral data and the ability to migrate tasks. However, it excludes behavioral strategies from artificial definitions and attempts to understand the behavioral strategies and mechanisms behind the data, which is expected to solve the problem of difficulty in designing reward functions under complex behavioral tasks.

Artificially designing simple behavioral patterns or mathematical reward functions may be impractical for representing relatively complex behaviors. In addition, for different biological behaviors, it is also difficult to have access to human experts in related fields who are familiar with behavioral strategies to assist in the design of reward functions. However, data-driven strategy learning approaches can address this dilemma. They also promote interdisciplinary research so that researchers in the engineering field can contribute to the design, evaluation, and understanding of autonomous behavioral strategies. Therefore, in this review, we focus on summarizing advanced computational approaches for data-driven behavioral strategy identification. We provide a comprehensive integration and analysis of the existing literature, categorizing and outlining the key theories, methods, and developments in this field. The recent developments and emerging trends are investigated to contribute to the ongoing discussion and exploration of data-driven policy learning methods.

This paper reviews computational methods that can be employed to identify potential behavioral strategies. Depending on the purpose and technical approach of the algorithm, we classify the current computational approaches into three categories: (1) imitative learning (IL), (2) inverse reinforcement learning (IRL), and (3) causal policy learning (CPL). The first group of algorithms aims to learn the policy underlying behavioral strategies by direct imitation. It is similar to the human learning process, where behavior is reproduced by learning from and imitating an expert's demonstration. The second group aims to recover transferable reward functions for learning policy from behavioral data. It is the inverse process of RL and can represent an expert's behavioral strategy via a reward function. The third group aims to combine causal models with other policy learning methods. Focusing on the causal relationship between states and actions, it improves safety and reliability in practical applications. The ultimate goal of these techniques is to learn behavioral strategies from data. The remainder of this paper is organized as follows: Section 2 describes the methods for obtaining relevant literature on behavioral strategy identification and mentions the selection criteria in the literature. Sections 3–5 present the results of the literature review for the IL, IRL, and CPL algorithms, respectively. Section 6 discusses different strategy

learning methods regarding algorithmic features, applications, current challenges, and future developments. Finally, Section 7 concludes the paper.

2. Methods

2.1. Literature Selection Standard

This review aims to provide an overview of policy learning methods for learning strategies from biological behavior. The keywords "behavioral strategy/strategic learning" cover the concept of policy learning while emphasizing the behavioral domain and accurately reflect the core focus of this review. Therefore, using the main keywords "behavior strategy/policy learning", we searched three databases: Web of Science, IEEE Xplore, and Scopus. The results were screened for relevant computational methods in the most recent decade to ensure that the investigated techniques were updated. We systematically screened the relevant literature, and the review process was conducted according to the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) 2022 Statement [7], as shown in Figure 1.

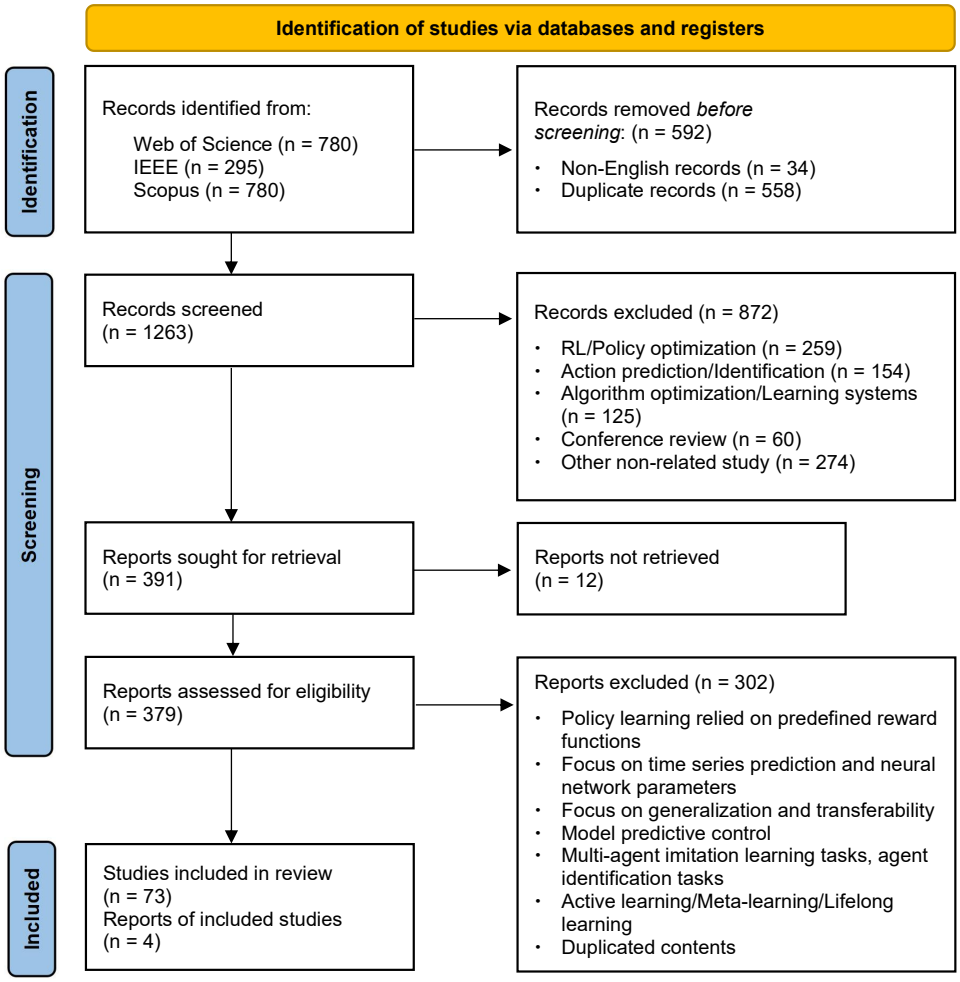

Figure 1. Literature screening process via the PRISMA statement.

The literature search was completed in January 2023, and 1855 records were obtained from all databases. The search records were exported to the reference management software, Zotero (version 6.0.18), for integration. We excluded 34 non-English literature items and 558 duplicate items automatically detected by Zotero. The preliminary literature identification resulted in 1263 records.

In the literature screening phase, 872 results were removed based on titles and abstracts. The main topics of these studies were RL, action recognition, and algorithm optimization, which are not discussed in this review. Among these, 12 records could not be retrieved and were therefore excluded. In the full-text eligibility assessment phase, we preferred strategy learning approaches based on behavioral data-driven methods rather than manually designed reward functions, neural network-based predictions, or model predictive control. Additionally, this review focused on learning single-agent behavioral strategies and did not discuss multi-agent interactive behavioral strategies. In addition, the topics in this review did not engage in active learning, meta-learning, or lifelong learning; thus, 302 records were excluded. In total, 77 articles were used for the systematic evaluation. The overall framework of this review is shown in Table 1. In addition to the literature under review, some of the early classic works are also mentioned in the text.

Table 1. Overall framework of this review.

Category	Sub-Category	References
Imitation Learning (IL)	Behavior Cloning (BC)	[4,8–20]
	Generative Adversarial Imitation Learning (GAIL)	[21,22]
	Improved GAIL Algorithms	[2,23–37]
Inverse Reinforcement Learning (IRL)	Maximum Margin Method	[5,38]
	Entropy Optimization Method	[3,5,39–50]
	Bayesian Optimization Method	[51–63]
	Adversarial Method	[64–68]
Causal Policy Learning (CPL)	Traditional Structural Causal Model (SCM) Modeling	[69–73]
	Other SCM Modeling Methods	[74–76]

2.2. Existing Reviews

During the literature review, four review papers related to this topic were published in the last two years. Among them, two summarize algorithms of strategy learning. Arora S. et al. systematically reviewed the IRL problem, including the methods, challenges, and future directions [77]. Zheng B. et al. comprehensively reviewed the categories and development processes of IL, with a focus on popular research topics [78]. Two other papers review strategy learning for certain behaviors. Di X. et al. focused on the driving behavior of human drivers and reviewed the learning methods for autonomous driving strategies [79]. Gajjar P. et al. focused on reviewing the strategy learning for path planning and obstacle avoidance behaviors of unmanned aerial vehicle systems [80].

The existing literature reviews focused on policy learning based on either a specific aspect of the algorithm or application area. In contrast to existing reviews, our review aims to examine computational approaches to extracting strategies from behaviors; it is not limited to the progress of a specific policy-learning algorithm or behavior. Thus, from a practical application perspective, it covers a more comprehensive range of policy learning methods, including IL methods incorporating RL or causal inference. Furthermore, this review does not focus on considering the development of policy learning in a particular application area but reviews the policy learning methods while incorporating additional application examples (e.g., autonomous driving, robot techniques, and biological behavior understanding), and analyzing the performance of these approaches in different application areas. In addition, this review summarizes the research progress on this topic over the past decade and discusses future research directions.

3. Strategy Learning from Demonstration: Imitation Learning

By maximizing the cumulative rewards from the environment, RL learns strategies to achieve specific goals [81]. It has been extensively employed to solve various control problems involving the interactions between agents and environments. However, experts

have designed reward functions used for strategy learning. Although they are feasible for handling relatively well-defined tasks (e.g., Go games [82]); in more complicated environments, it can be extremely difficult to artificially design a reasonable reward function for animal or human behavior. However, IL can avoid this problem by directly learning from the demonstration, that is, by imitating the object behavior in the given tasks.

IL takes the observed state and action sequences as decision data and trains the agent to learn the mapping relationships between each set of decision data. Thus, we obtain an optimal policy model to perform the task in the demonstration. IL does not rely on designing explicit reward functions; it simplifies the process of teaching agents to learn by providing demonstrations to them. Thus, it has several advantages in achieving complex tasks with minimal expert knowledge [83]. Integrated with high-speed sensors that enable instantaneous data collection, IL has great potential for applications in bio-inspired robots, autonomous driving, and VR games.

In this section, we review policy-learning methods using IL. These IL methods include behavior cloning (BC), generative adversarial imitation learning (GAIL), and improved GAIL techniques. BC learns the mapping of behaviors from states to actions through a supervised learning approach using labeled behavioral demonstration data as the training set, which allows IL to be most simple. Unlike BC, GAIL trains policies and reward functions by generating adversaries that can learn policies directly from demonstrations and has stronger representational power in IL involving environmental interactions. However, the improved GAIL methods improve learning and increase the efficiency of sample utilization based on the original GAIL algorithm, making it adaptable to real-world problems.

3.1. Behavior Cloning

As a basic form of IL, simple BC can be viewed as a supervised learning method that transforms the imitation problem into a learning process to map states to actions. While supervised learning uses labeled training data, BC uses the training dataset $\{(s_1, a_1), \cdots, (s_n, a_n)\}$, which consists of demonstration examples in the form of $(state, action)$, where n represents the length of the demonstration trajectory. BC uses deep neural networks to train the demonstration data and learns implicitly from it. Thus, obtaining realistic behavioral strategies depends heavily on accurately mapping decision pairs in the dataset.

Most BC studies learn strategies and understand behaviors from human or animal demonstrations. Here, the human or animal providing the behavioral demonstration is defined as the "teacher", and the computer or robot is defined as the "agent". Learning from observation (LfO) demonstration data can be categorized into (1) first-person and (2) third-person views:

(1) In the first-person view LfO, the demonstrating teacher records the states and actions during the activity through the worn sensors. The agent can directly imitate the teacher's decision from the first-person view without transforming the demonstration data. This approach has been extensively used in humanoid robotics and autonomous driving, such as imitation learning using the VR teleoperation approach [8], and end-to-end autonomous driving BC [9,10]. The advantage of this approach is that it does not require the remapping of demonstration data and can provide accurate measurements. However, because of the size of the sensors and the customization environment, collecting demonstration behaviors by mounting sensors on the teacher cannot be used in all cases.

(2) The third-person view LfO collects demonstration data through external behavioral observation, which requires the perspective transformation of the observed data, i.e., mapping the recorded data from the third perspective to the agent from the first perspective [11,12]. It is primarily based on external cameras and thus can be applied to robot manipulation without being limited to customized experimental environments [12,13]. Compared to the former method, the third-person view LfO is more flexible regarding the applicable environments; however, the collected demonstration data may not be sufficiently precise.

Dataset aggregation (DAgger) improves the interaction between the algorithm and the environment through data augmentation and aggregation based on simple BC [84]. DAgger is an online learning algorithm that continuously enables acquired strategies to interact with the environment to generate new data. Subsequently, the newly acquired data are aggregated to improve the dataset. It can learn expert behavioral strategies in complex environments, such as autonomous driving tasks [14,15] and strategy learning in complex 3D game scenarios [16].

In the practical application of strategy learning, IL with a combination of RL is employed in addition to the above methods. This integrates the advantages of both algorithms. This approach first performs IL with demonstration data as pre-training and then uses RL in addition [4,17,18]. It is difficult for policies learned via IL to cope with situations not encountered in demonstration data; however, this method compensates for this shortcoming by combining RL. In addition, pre-training through IL avoids the limitation of manually designing reward functions for RL and significantly reduces the RL training time. Therefore, a strategy learning method that combines IL and RL has become popular and is a powerful alternative to RL. This enables robots to perform long-horizon tasks in challenging simulated kitchen environments [18] and also mimics the agile locomotion skills of animals [4].

3.2. Generative Adversarial Imitation Learning

The purpose of simple BC is to match the behavior of the demonstrating teacher, and it has significant limitations in terms of environmental interactions. This defect was improved by the emergence of GAIL [21]. GAIL is used to extract strategies directly from expert demonstration data. It has the advantage of being a generative adversarial network (GAN) [85], thus outperforming BC in complex applications and gradually becoming a popular model-free IL framework.

First, the concept of GAN is introduced. GAN is a deep generative model that generates generative samples for fitting training samples by transforming deep features abstracted from random input noise to the underlying features, which can be regarded as the inverse process of deep classification models. The deep generative model, which aims to generate realistic samples, is called a "Generator", and the deep classification model, which aims to discriminate the source of the samples, is called a "Discriminator". They are trained adversarially by playing a game with each other such that the samples generated by the Generator can fit the real training samples well.

Similarly, GAIL adopts the idea of adversarial training. It utilizes two neural networks to represent the strategy and reward function and continuously optimizes the parameters of these two networks by adversarial training. In other words, the strategy that outputs the action according to the input state is considered a Generator, and the reward function that outputs the reward value according to the input generation samples is considered a Discriminator. The learning process of the strategy and the solving process of the reward function are the training processes of the Generator and the Discriminator, respectively. The parameters are updated during the training process using a gradient

$$\min_{\pi} \max_{D} \mathbb{E}\pi[log(D(s, a))] + \mathbb{E}\pi_E[log(1 - D(s, a))], \quad (1)$$

where π is the policy to be learned, π_E is the policy of the experts, and $D(s, a)$ is the probability determined by the Discriminator. The Generator aims to minimize the cross-entropy loss of the generative samples, whereas the Discriminator aims to maximize the cross-entropy loss of the discriminative model. In addition, the learning objective of GAIL is to optimize the strategy directly instead of computing the reward function before solving it; thus, it avoids the high complexity of the computation process in IRL (Section 4) and can adapt to IL tasks with large-scale complex behaviors.

Based on the GAN framework, the strategy and reward function in GAIL can automatically extract abstract features from training samples with higher representational capability,

which alleviates the dependence of IL on large datasets. Moreover, GAIL significantly reduces computational complexity by learning strategies directly from data and, thus, can better adapt to the complexity of real applications. The method has exhibited excellent performance in autonomous driving [22]. Nevertheless, GAIL also faces the problems of model collapse [86] and low sample efficiency in terms of environmental interactions [23,87].

3.3. Improved GAIL Algorithms

The drawbacks of GAIL in practical applications limit its adaptation to applications with high sample acquisition costs. The problem of model collapse originates from GAN, e.g., the Generator finds a data type that easily deceives the Discriminator and thus continues to generate that type, that is, resulting in the Generator having a similar single type of output, thus depriving the generated samples of diversity. Low sample efficiency for environmental interactions relates to how the policy is learned. The stochastic policy of GAIL causes the agent to randomly select behaviors (including undesirable ones) according to the probability distribution, which may lead to low efficiency for the agent to search the environment. In addition, GAIL learns policies directly through a model-free policy instead of using environmental models. Hence, agents require many environmental interactions to maximize the reward, and the expensive training process leads to low sample efficiency. To improve the performance of imitation learning, many attempts have been made to solve these problems in addition to GAIL.

The improvement in the GAIL model collapse caused by GAN is also inspired by an improved approach to GAN. Extending imitation learning from a single-model assumption to learning from multiple models can effectively avoid model collapse and satisfy the requirements of practical applications. In the case of autonomous driving, for example, an intelligent agent is expected to imitate not only the fast-driving model but also learn multiple driving models, including safety.

- Conditional GAIL (cGAIL): Based on the idea of conditional GAN (cGAN) [88], i.e., when multiple model labels for imitation learning can be obtained directly from expert samples, these model label data can be directly used as conditional constraints in the process of policy and reward function construction. Hence, cGAIL [24,25], which can perform imitation learning under multiple model conditional constraints, is proposed.
- GAIL with Auxiliary Classifier (AC-GAIL): AC-GAIL [26–28] was inspired by GAN with auxiliary classifier (AC-GAN) [89], which introduced an auxiliary network model for multiple model imitation learning. AC-GAIL does not use model label data directly but uses an additional auxiliary network to train the model label data, which can extract abstract deep features from the samples and efficiently use the model label data.
- Information Maximizing GAIL (InfoGAIL): Inspired by information maximizing GAN (InfoGAN) [90], InfoGAIL [29,30] employs the idea of mutual information to judge the correlation between samples and maximizes the mutual information from the relationship between the generated samples and the randomly sampled hidden modal data to interpret the meaningful implicit modal data in the expert samples. Thus, it realizes unsupervised multimodal imitation learning and provides a certain degree of interpretability.
- Variational Auto-Encoder GAIL (VAE-GAIL): Based on the principle of variational auto-encoder GAN (VAE-GAN) [91], VAE-GAIL [31,32] replaces the generative model in the original GAIL with a variational self-encoder to obtain meaningful modalities in the sample by maximizing the mutual information between the expert samples and the implicit modal data inferred by VAE. Because of the continuity of the implicit modal data inferred by VAE, the strategies learned by VAE-GAIL have a higher level of diversity.
- Wasserstein GAIL: Based on the improvement of GAN by Wasserstein distance [92], Wasserstein GAIL is proposed [33]. It employs Wasserstein distance instead of KL divergence or JS divergence to compute the distance between the generated distribution and the true distribution, i.e., the loss function of the Generator. In contrast,

the difference measured by Wasserstein distance is continuous; hence, the training is smoother. Furthermore, no modal collapse was observed in the experiment [92].

Among the above methods, cGAIL and AC-GAIL based on supervised learning cannot work without the modal labeled data. In contrast, InfoGAIL and VAE-GAIL, based on unsupervised learning, can handle the condition against the modal labeled data hidden in expert samples. Wasserstein GAIL improves the GAIL by obtaining smooth gradients to smooth the training.

The problem of the low efficiency of sample environment interaction utilization can be improved from the perspective of RL.

- Model-based GAIL (MGAIL): differing from the model-free policy learning approach of GAIL, model-based GAIL (MGAIL) [34,35] effectively improves the utilization efficiency of samples in the environment interaction process by modeling the dynamic environment and random sampling process [93].
- Actor–Critic Policy Searching Based GAIL: Another improvement idea is to replace the stochastic policy with the Actor–Critic policy searching method [94], where Actor is a policy that executes actions based on state, and Critic is used to evaluate actions. The learning process is similar to the generative adversarial learning process of GAIL. GAIL based on Actor–Critic policy searching can improve the efficiency of sample utilization and also achieve end-to-end gradient update from reward function to policy [2,23,36,37].

These two approaches improve GAIL by introducing an environmental model and improving the policy search methods. Compared to MGAIL, GAIL with an Actor–Critic-based policy search method, avoids the complex recursive computation involved in using a model.

Improved GAIL with multimodal learning capabilities performs well in policy learning and has been corroborated to apply to complex real-world tasks. This method can perform autonomous driving tasks, including learning the policy preferences of different taxi drivers to find passengers in various areas [25], learning driving strategies to change lanes based on traffic conditions [26], and imitating driving strategies [29]. In terms of motion control, it can imitate human motion patterns [28] and complete high-dimensional robotic simulation tasks [95]. Moreover, it can be extended to real-time strategy (RTS) game strategy learning [26]. In addition, improved GAIL with increased sample utilization efficiency has enabled large-scale applications in dense urban autonomous driving tasks [35] and imitation learning involving multiple agents [96].

3.4. Section Summary

Through direct imitation, strategies can be learned from demonstrations using IL. This section describes two specific methods by which behavior can be imitated directly from demonstrations, namely BC and GAIL, and their improved methods. Supervised-learning-based simple BC uses labeled demonstration data for learning. Depending on the needs of the demonstration subject and the experimental environment, the BC can use demonstration data not only from a first-person viewpoint but also from a third-person viewpoint. The first-person-view BC is designed for occasions where the accuracy of the measurement data is important, and the sensor can be easily mounted directly on the expert. When customizing the experimental environment is costly, and it is difficult to fix the sensor to the experimental subject, third-person-view BC based on video measurement is easier to apply. BC is theoretically feasible as the simplest IL method. It performs well in imitating demonstrative robotic manipulations in simple environments. However, relying on the properties of supervised learning and the policy-learning process without environmental interaction, it was found in early practice that simple BC often fails to mimic expert behaviors in real-world high-dimensional environments. This is because of problems such as covariate shifts and causal confusion [97]. To address the problem of environmental interaction, DAgger employs the policy learned by the BC to interact with the environment and generate new data, iterating online to augment the dataset. The algorithm considers

the differences between demonstration and test results. Thus, it exhibits a generalization performance superior to simple BC in complex environments. However, DAgger does not inherently depart from the BC algorithm. Owing to its online learning mechanism, it requires long hours of expert assistance to supplement the demonstration, which can result in high workloads. Moreover, BC is often used for pre-training and is combined with RL owing to its simplicity. This approach of combining IL and RL introduces a reward obtained from the imitation of an expert into the RL framework, which can be used to obtain an agent that exceeds that of a demonstration expert. It has been extensively applied in robot motion control. By contrast, GAIL uses a generative adversarial approach to extract abstract features from training samples, and its stronger representation ability alleviates the dependence of the algorithm on large datasets. However, the original GAIL experiences the problems of GAN modal collapse and inefficiency of stochastic policy search methods. To solve the modal collapse problem, cGAIL and AC-GAIL constrain the explicit modal labels. In contrast, through unsupervised learning, InfoGAIL and VAE-GAIL determine the implicit modal labels in the presentation data. In addition, Wasserstein GAIL improves the smoothness of the training process. MGAIL and Actor–Critic-based GAIL improve the environment sample interaction utilization by introducing environment models and changing the strategy search method, respectively. GAIL is widely employed in automatic driving, motion control, robot manipulation in high-dimensional environments, and game strategy learning. It has become a popular strategy learning algorithm in recent years.

4. Explaining Behavior via Reward Function: Inverse Reinforcement Learning

The goal of RL is for agents to attempt to learn optimal behaviors through experience. Reward functions are required for RL to determine policies during the interaction between the agent and the environment. IRL, in contrast, is the reverse process of RL; that is, the agent learns the corresponding reward function based on the observed behavior [98]. Although some studies have classified the IRL under the category of IL, in this review, IRL is listed as an independent section. This is because, unlike IL methods, such as BC and GAIL, which learn policies directly from demonstration data, IRL extracts reward functions from behavioral data in a transferable manner. If the reward function is recovered, the target strategy can be implemented efficiently using the reward function and RL. Compared with the human-designed reward function approach, IRL automates the reward design, and the learned reward function can handle complex and multimodal presentations. In addition, by learning the reward functions, the behavioral preferences of experts can be obtained, contributing to an understanding of biological behavior.

IRL assumes that the behavior is consistent with a Markov decision process (MDP) and infers reward functions from behavioral data. MDP is a sequential decision process that simulates an agent [99]. The Markov property simplifies the state dependencies in real-world problems by ensuring that the action decision for the next state depends only on the current state–action pair and is independent of the previous state, where the transition from state to action $\pi(a_t \mid s_t)$ is determined by the policy of the agent and the transition from action to state $P(s_{t+1} \mid s_t, a_t)$ is determined by the environment. The long-term reward based on MDP can be expressed as

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0} \gamma^k r_{t+k+1}, \tag{2}$$

which is the cumulative value after the current state is obtained by multiplying all reward values by the corresponding discount rate γ. Value functions are used to quantify the contribution of the current state to the end goal. The state value function $v_\pi(s_t)$ and the state–action value function $q_\pi(s_t, a_t)$ are defined as follows:

$$v_\pi(s) = \mathbb{E}_\pi[R_t \mid s_t = s], \tag{3}$$

$$q_\pi(s, a) = \mathbb{E}_\pi[R_t \mid s_t = s, a_t = a]. \tag{4}$$

They demonstrate an extension of MDP. On this basis, IRL inversely estimates the reward function through a data-driven approach. IRL performs well in autonomous driving, path planning, and human–computer interaction (HCI) applications. It has also been used in the research of learning animal behavioral strategies.

In this section, we review policy-learning methods using IRL. The listed IRL methods are divided into four main categories based on the optimization method: the maximum margin algorithm, entropy optimization, Bayesian optimization, and adversarial methods. The maximum margin IRL determines the ideal reward function for an expert trajectory by maximizing the margin between optimal and suboptimal policies. This IRL algorithm performs well in recovering expert behavioral trajectories. Entropy optimization and Bayesian optimization IRL are based on probabilistic methods. They can find a strategy that outperforms experts' suboptimal behavior when the demonstrated behavior is not optimal. In addition, IRL using adversarial methods builds on the entropy optimization IRL to make reward learning efficient and robust in dynamic environments.

4.1. Maximum Margin Method

The original IRL algorithm assumes a problem consistent with MDP, where a policy can decide an action for the next state using the current state-action pair. The reward function must be estimated from a portion of the basis functions that represent the important features of the states using a linear approximation. Subsequently, we generalize to more unknown states as follows:

$$R(s) = \alpha_1\phi_1(s) + \alpha_2\phi_2(s) + \cdots + \alpha_n\phi_n(s) = \alpha\phi(s), \tag{5}$$

where each basis function $\phi_i(s)$ maps the state s to a scalar value, and the coefficient α determines the contribution of each basis function to the state. According to Equations (2) and (5), the value function $v_\pi(s_t)$ can be expressed as

$$v(\pi) = \mathbb{E}[\sum_{t=0}^{\infty}\gamma^t\sum_{i=1}^{n}\alpha_i\phi_i(s_t) \mid \pi], \tag{6}$$

and is further simplified to the following form:

$$v(\pi) = \sum_{i=1}^{n}\alpha_i\mathbb{E}[\sum_{t=0}^{\infty}\gamma^t\phi_i(s_t) \mid \pi] = \sum_{i=1}^{n}\alpha_i\mu_i(\pi) = \alpha\mu(\pi). \tag{7}$$

It is worth mentioning that IRL is an ill-posed problem according to its definition. Multiple optimal policies may exist under the same reward function, and multiple reward functions may correspond to the same optimal policy.

When the optimal policy is unknown, the original IRL algorithm estimates it using multiple expert demonstrations, also known as apprenticeship learning (AL) [100]:

$$\hat{\mu}(\pi^E) = \frac{1}{n}\sum_{i=1}^{n}\mu(\pi^i). \tag{8}$$

For the reward function to have only one corresponding optimal policy, AL solves the ill-posed problem using the maximum margin method, that is, maximizing the difference between the first and second optimal policies. The process of optimizing the reward function can be represented as follows:

$$\max\{t_i = \min_{j\in\{0,1,\ldots,i-1\}}\alpha(\mu(\pi^E) - \mu(\pi^j))\}, \tag{9}$$

where t_i denotes the minimum value of the transition difference in the expected rewards among all previous strategies up to step $i-1$. By maximizing this term, a strategy significantly better than any previous strategy is identified to make it as close as possible to

the expert strategy, as shown in Figure 2A. The optimization process of this algorithm is randomized, and thus, may not be sufficiently accurate in the face of expert behavior with deterministic behavior.

Rather than returning to an optimal policy, AL uses the maximum margin method to select a randomly returned alternative policy with performance approximately similar to expert behavior. Thus, maximum margin planning (MMP) provides a learning method that can directly reproduce expert behavior [101]. MMP avoids the ill-posed problem by introducing a loss function $L(\pi)$. The difference from AL is the margins in the MMP scale with the loss function. $\max_{\pi'}$ denotes the maximum value of all strategies, except π. In other words, $L(\pi)$ penalizes strategies with a small margin between the expected value and the expected value of the best strategy, thus selecting more robust strategies with larger margins:

$$\max_{\pi}\{v_{\pi}(s_0) - \min_{s'} v_{\pi}(s')\}, \tag{10}$$

$$L(\pi) = \max_{\pi'}[v_{\pi}(s_0) - v_{\pi'}(s_0)]. \tag{11}$$

In the optimization process, MMP employs the $q_{\pi}(s, a)$ function to estimate the expected reward for each action in each state. It is continuously updated iteratively, such that the action with the highest reward is selected to obtain an optimal policy. In subsequent improvements, LEARCH was introduced and extended the learning of the reward function to nonlinear using an exponential function gradient descent method for optimization [102], as shown in Figure 2B.

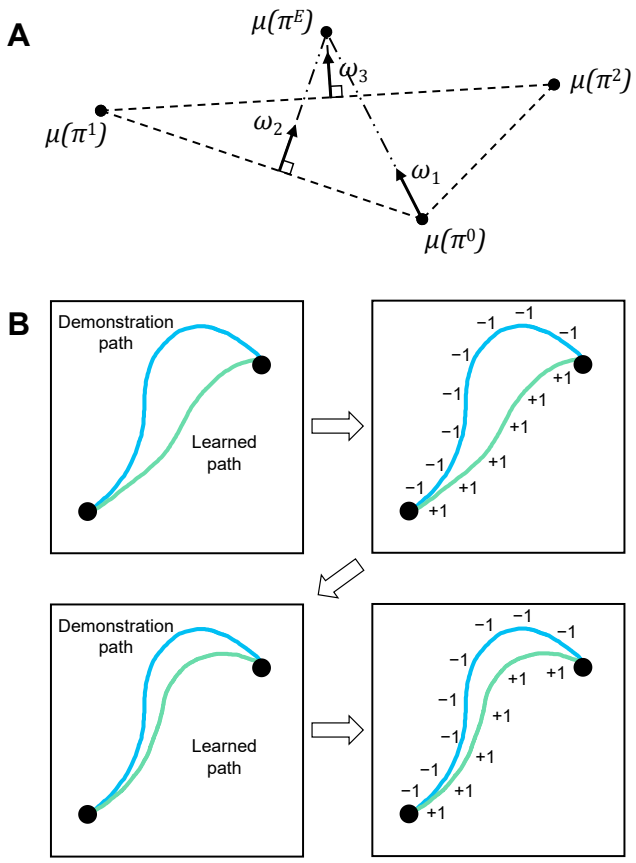

Figure 2. Direct illustrations of the maximum margin method. (**A**) An example of three iterations for AL using the maximum margin method to approach expert policy (Redrawn from [100]). (**B**) MMP methods with LEARCH searching (Redrawn from [102]).

For some complex tasks (e.g., autonomous helicopter aerobatic flight, which needs to consider other factors such as dynamics in addition to the target trajectory), the sample tra-

jectories are difficult to describe directly, so supervised learning-based simple BC methods are no longer applicable. However, the core of the maximum margin method-based IRL is to extract desirable trajectories from demonstrations that robots can follow, which can be used to solve such problems. Although AL still exhibits problems such as sensitivity to demonstration data and stochastic optimization strategies that may lead to inaccuracies, it has achieved good results in fields such as aerobatic helicopter flight [103] and game simulation [5,38]. Compared with AL, which selects the reward function using linear programming, the optimization process of MMP is more flexible and can reproduce the expert trajectory. It exhibits better performance in robot navigation on rough terrain [104]. However, this approach relies on a complete sequence to achieve optimization, which has limited scalability and is computationally expensive for high-complexity tasks.

4.2. Entropy Optimization Method

Although MMP has enabled the replication of expert trajectories, this maximum margin method-based approach suffers from labeling bias in practical applications. The estimated reward function will be biased towards actions or trajectories consistent with the expert's demonstrated behavior while ignoring other potentially desirable alternatives [105]. The over-preference for expert behavior when learning reward functions from demonstrations makes diverse strategy exploration challenging. Hence, a series of IRL methods employ entropy optimization. The policy with the highest reward derived by MMP is not necessarily the policy with the highest probability among the entropy optimization methods.

To reduce this label bias, the maximum entropy IRL (MaxEnt-IRL) [106] learns the reward function using a probabilistic method to solve the ill-posed problem of IRL. MaxEnt-IRL uses the maximum entropy principle [107] to solve this uncertainty, which allows for estimating the minimum deviation from the given information when only partial information is available. The probability of an observed expert trajectory is weighted by the estimated reward:

$$Pr(\tau \mid w) = \frac{1}{Z} e^{\alpha \mathbb{E}(\tau, w)}, \tag{12}$$

where τ represents the trajectory, w is the weight vector, and Z is the normalizing constant. α indicates the confidence level at which the expert selects an action with a high reward value. Thus, the optimal weight vector w is derived by maximizing the likelihood of the observed trajectory using the maximum-entropy approach:

$$w^* = arg \max_{w} \sum_{\tau \in D} \log Pr(\tau \mid w), \tag{13}$$

where D represents the demonstration. MaxEnt-IRL uses the maximum entropy approach to select the reward function based on the premise that the calculated strategy reward is comparable to the expert strategy reward. Despite the superiority of the probabilistic approach, MaxEnt-IRL remains limited to the representational capacity of linear reward functions. On this basis, the maximum entropy deep IRL (MEDIRL) [39] extends the representation of the reward function to nonlinearity. MEDIRL employs the same basic algorithm as MaxEnt-IRL; however, it represents the reward function through a flexible neural network, and the features in the reward function do not need to be set manually.

However, MaxEnt-IRL does not adequately consider stochastic transformation models. Thus, the maximum causal entropy IRL (MCE-IRL) extends the maximum entropy principle to side information [40,41,108]. It focuses on inferring an agent's preferences from observed behaviors and further understanding the causality between actions and outcomes. MCE-IRL allows stochastic transformation models. Hence, it considers the causality between actions and outcomes while estimating the reward function.

IRL algorithms based on the maximum entropy framework are model-based. In contrast, relative entropy IRL (RE-IRL) is a model-free algorithm [5,42,109]. It overcomes the computational cost problem by minimizing the KL scatter between the learned and real

trajectories to determine the optimal policy. Therefore, it is more suitable for IRL tasks that use large datasets.

These probability-based maximum entropy methods enable experts to demonstrate strategies as suboptimal and, therefore, learn potentially optimal models that are more flexible in practical applications without the concern of less-than-perfect demonstration samples. For autonomous driving problems, the MaxEnt-IRL can learn the reward function [43,44] from driving data, including the vehicle speed, distance from the front vehicle, and relative speed. It has also been used to learn pedestrian navigation [3,45], animal behavior [46–48], and human preference [49], which provided new inspirations for HCI developments and the understanding of behavior mechanisms. On this basis, MEDIRL can handle large datasets [50]. Moreover, as a model-free algorithm, RE-IRL is capable of learning different strategy styles [5]. It can also be used as a highly scalable method for the inverse reinforcement learning of large samples [42] and achieves good results with a relatively lower computational cost.

4.3. Bayesian Optimization Method

Bayesian IRL (BIRL) [51,52,110] uses a probability distribution approach to address uncertainty when estimating the reward function for inverse reinforcement learning. BIRL uses a Bayesian framework and prior knowledge to determine the posterior distribution preferred in expert behavior, producing alternative strategies that may outperform the suboptimal strategy demonstrated by the expert. The probability of demonstrating the entire trajectory to an expert is expressed as $Pr(\tau \mid R)$. Thus, the posterior probability $Pr(R \mid \tau)$ of the reward function R can be derived:

$$Pr(\tau \mid R) = \frac{1}{Z} e^{\alpha \mathbb{E}(\tau, R)}, \tag{14}$$

$$Pr(R \mid \tau) = \frac{1}{Z \cdot P(\tau)} e^{\alpha \mathbb{E}(\tau, R)} P(R). \tag{15}$$

Bayesian ideas can also be embodied in Gaussian process IRL (GP-IRL) and maximum likelihood IRL (ML-IRL). The GP-IRL models the reward function as a Gaussian process and treats the distribution of the reward function as a posterior distribution, computed by Bayesian updating [111]. It employs a Gaussian process to learn the nonlinear reward function. The uncertainty is addressed by determining the posterior distribution. Unlike BIRL, the ML-IRL directly uses the likelihood function to perform the estimation. Boltzmann exploration makes the likelihoods differentiable, thus inferring a reward function [53].

As IRL methods based on the probabilistic approach, BIRL and GP-IRL impose a prior on the relevant parameters to learn the degree of suboptimality in expert demonstrations explicitly and provide a posterior distribution. In contrast, the ML-IRL is based on maximum likelihood estimation, which cannot incorporate prior knowledge and handle uncertainty. IRL with the Bayesian optimization method has been used to learn driving strategies [54], mobile robot navigation [55,56], and robot demonstrative learning [57] with good performance. Hierarchical BIRL extended on the original basis outperforms MaxEnt-IRL in cab driver route selection based on maps and GPS data [58]. This method is also used to learn the behavior of animals [59,60], the behavior of pedestrians [61], user behavior modeling [62], and market trading strategies [63]. Nevertheless, IRL based on the Bayesian framework or Gaussian process significantly increases the computational difficulty and remains difficult to extend to applications in large environments before solving complex computational volumes. The ML-IRL, in contrast, could be computationally simpler but lacks robustness to uncertainty.

4.4. Adversarial Method

Compared to GAIL, an efficient algorithm that allows for direct strategy imitation, IRL methods are often criticized for their high computational costs. Nevertheless, the GAIL framework cannot estimate transferable reward functions. On some occasions, learning the

reward function may be more useful than directly imitating the behavior. This is because the reward function is transferable to new environments and agents and can represent the agent's behavioral preference to some extent. Therefore, IRL could be a better choice when the reward function needs to be recovered.

The adversarial IRL (AIRL) is an IRL algorithm based on adversarial learning. This is based on the maximum entropy principle of the IRL method (see Section 4. *B*). Similar to the generative adversarial concept mentioned in Section 3. *B*, AIRL treats the process of updating the Discriminator as an update of the reward function, which is shown as follows:

$$D(s, a, s') = \frac{e^{f(s,a,s')}}{e^{f(s,a,s')} + \pi(a \mid s)}, \tag{16}$$

where f denotes the learning function. In the absence of a priori knowledge of dynamic information, IRL can only learn rewards from ideal agent demonstrations, which cannot guarantee robustness against dynamic environmental changes. Considering this problem, AIRL recovers a policy-invariant reward $\hat{r}(s, a, s')$ by studying the policy invariance in the two MDPs (M, M'). Thus, AIRL can avoid an entangled reward-learning process and make the learned reward dynamically robust to the environment.

AIRL is a practical, efficient, and robust IRL algorithm. This was validated in simulated environments for locomotion control [64,65] and robotic manipulation [66,67]. It has also been applied to cognitive studies of collective animal behavior [68].

4.5. Section Summary

By recovering the reward function from behavioral data, IRL-based policy learning achieves better generalization performance and avoids some problems exhibited in BC. In this section, we introduced four methods for IRL: the maximum margin algorithm, entropy optimization, Bayesian optimization, and the adversarial method. Assuming that expert demonstration is the optimal strategy, the AL uses the maximum margin algorithm to maximize the reward difference between the optimal and suboptimal strategies. It exhibits excellent performance in learning high-dimensional complex behavioral trajectories (e.g., helicopter flying). However, the reward function derived by AL is only a close alternative to the expert's behavior. MMP improves AL and can reproduce an expert's behavioral trajectory. However, when the expert demonstration is suboptimal, probability-based entropy optimization and Bayesian optimization methods can obtain a policy that exceeds that of the expert demonstration. For entropy optimization IRL, MaxEnt-IRL uses the principle of maximum entropy to find the optimal policy, which is widely applied in autonomous driving, navigation, human–computer interaction, and understanding biological behavior. Accordingly, MEDIRL generalizes the estimation of the reward function to be nonlinear by combining the advantages of neural networks. MCE-IRL further considers the causal relationship between actions and outcomes. In contrast, the RE-IRL changes the model-based algorithm and uses a model-free approach for optimization. This reduces the computational cost and enables the algorithm to apply to real-world scenarios with large-scale data. However, for Bayesian optimization IRL, BIRL determines the posterior distribution of expert behavioral preferences through prior knowledge. Similarly, GP-IRL and ML-IRL employ this idea to learn the reward functions. IRL based on Bayesian optimization has a higher computational cost than entropy optimization methods, thus limiting its application in large-scale scenarios. Furthermore, because IRL learns behavioral strategies by recovering the reward function, it is computationally intensive compared to IL, which directly imitates behavior. Considering this problem, AIRL takes advantage of the algorithmic advantages of GAIL based on MaxEnt-IRL and uses an adversarial approach to update the reward function. Robust rewards can be learned without prior knowledge of the dynamic environment.

5. Learning Based on Interpretability and Safety: Causal Policy Learning

As mentioned in Section 3. *A*, supervised learning-based BC methods suffer from causal confusion [97] as they ignore the causal consequences of expert–environment interactions. Although GAIL improves the interaction between the agent and environment, according to the architecture and complexity of its generative models, more complex and powerful models create black-box models that lack interpretability. Deep learning-based neural networks can output strategies or reward functions from data inputs; however, they cannot provide an intrinsic causal relationship between the behavior and the reward behind them [112]. This means that while uninterpretable strategies may be valid, there is no guarantee that such uninterpretable strategies are free from errors [112,113]. This lack of transparency and interpretability limits the further application of models in the industry because users cannot fully trust them. If targeted interventions can be made to determine the correct causal relationship in the model design, this will help provide interpretability to the algorithm.

Causal inference is the process of determining causality based on the causes that lead to the occurrence of an outcome [114]. However, it is sometimes difficult to draw direct causal relationships from observations. To explain with a simple example, even if the collected data exhibit a simultaneous increase in the number of people eating ice cream and the number of people drowning, it does not show that there is a causal relationship between the two, ice cream eating and drowning. This is because the summer temperature increase might be the cause. That is, "correlation" is not "causation". Determining causality is an important and challenging task in policy learning because the relationship between two different actions is often difficult to determine.

In this section, we reviewed the policy-learning approach using a causal inference framework. The basic concepts of the causal inference framework and policy learning methods using causal relationships between variables were introduced. The listed methods for constructing the structural causal model (SCM) include the basic method based on the directed acyclic graph (DAG) and the extended method based on Bayesian networks and neural network difference equations. After determining the causal effects between the variables, the policy can be optimized using the GAIL or IRL algorithms. The advantages of CPL are its robustness and interpretability. Therefore, it has a good scope for industrial applications.

5.1. Causal Model

Many frameworks have been proposed to explore the causal relationships between different actions observed. The most commonly used models include the SCM [115] and the potential outcome framework [116]. The SCM, also known as the DAG model, understands the causal structure between observed actions by representing the dependencies between variables. DAG uses nodes to represent the variables of interest and directed edges to represent the causal relationships between the variables, as shown in Figure 3. SCM can be represented by the tuple M: $\{U, V, F, P(u)\}$, where U, V, and F are the sets of exogenous variables, endogenous variables, and structural equations, respectively, and $P(u)$ is the exogenous distribution. The potential outcome framework, also known as the counterfactual model, considers the potential outcome of applied or unapplied treatment to analyze the causal effects of treatment. Both models aim to understand and analyze the causal relationships of observed actions: SCM focuses on presenting causal outcomes, and the potential outcome framework focuses on the inference and estimation of potential outcomes.

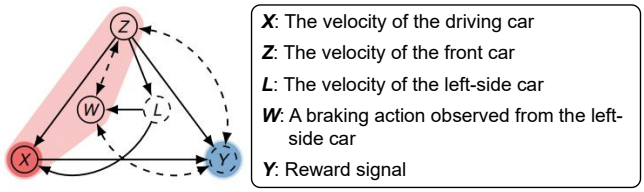

Figure 3. Example of a highway driving causal diagram, where X and Y represent the action (shaded red) and reward (shaded blue), respectively. Input covariates are shaded in light red (CC BY 4.0 image credit: [69]).

5.2. Causal Policy Learning

The IL and IRL mentioned in Sections 3 and 4, respectively, exhibit excellent performance in behavioral strategy learning methods by learning from the demonstration and learning the reward function of behavior. In these machine learning-based policy-learning methods, although more input data can make the model estimate more accurately, it is difficult to ensure that the estimation result is unbiased [114]. This is because IL and IRL follow the assumption that the observations demonstrated by the expert and those available to the agent match when implementing policy learning. However, in the real world, it is difficult to satisfy this assumption fully; that is, the expert may not observe all covariates. Under such conditions, it is difficult for IL and IRL to perform as desired. A causal inference framework improves the accuracy and interpretability of machine learning.

CPL incorporates causal learning frameworks. It aims to learn causal relationships and strategies from the observed data by estimating the causal effects of actions on potential future states and considering potential confounders in the data. The common approach in CPL is to use SCMs to represent the causal relationships between variables. These models allow agents to perform counterfactuals on the variables of interest and estimate their causal effects. An agent can thereby use this causal knowledge to guide the process of adaptive and flexible behavioral generation.

The traditional CPL framework for SCM [69–73] first defines the SCM M as $\{U, V, F, P(u)\}$ to capture the causal relationship between the variables of interest. It divides the endogenous variable V into observed variables O and latent variables L, where $O \subseteq V$ and $L = V \setminus O$. Thus, SCM M can be implemented to model the unobserved nature of endogenous variables. The expected reward and reward of the expert demonstrated are denoted by $\mathbb{E}[Y \mid do(\pi)]$ and $\mathbb{E}[Y] = 1$, respectively, where π denotes the policy. The function $do()$ represents the intervention or action being performed in causal inference. In policy learning tasks, the expected reward $\mathbb{E}[Y \mid do(\pi)]$ cannot be determined directly from the observed data or DAG because the reward Y is typically latent. Additionally, there are cases in which confounders or other variables exist. For example, for ordered variables (X, Y) in a DAG, exploring the causal effects between $X \to Y$ can be difficult if there are variables that point to or influence X that are difficult to measure in the experiment. The backdoor criterion is used to address this problem [70]. It states that the set of variables Z satisfies the backdoor criterion for ordered variables (X, Y) when no element of the set Z is a successor node of X and Z blocks all paths between X and Y that towards X. In subsequent refinements, the backdoor criterion applicable to single-stage decision problems is extended to the sequential backdoor criterion by considering the time dependence between variables; that is, X affects Y at different points in time and can be viewed as a recursive form of the single-stage backdoor criterion [70]. If Z satisfies the ordered backdoor criterion for ordered variables (X, Y), then the causal effect between $X \to Y$ can be deduced.

$$P(Y \mid do(X)) = \sum_{z} P(Y \mid X, Z)P(Z) \tag{17}$$

Strategy learning is then guided by the collected observed data and estimated causal effects. Policies can be obtained by solving the equation $P(s \mid do(\pi)) = P(s)$, where s belongs to a subset S of the observed variables [69]:

$$\pi(x_0) = \frac{P(s_1) - P(s_1 \mid do(x_0))}{P(s_1 \mid do(x_1)) - P(s_1 \mid do(x_0))}, \tag{18}$$

$$\pi(x_1) = \frac{P(s_1 \mid do(x_1)) - P(s_1)}{P(s_1 \mid do(x_1)) - P(s_1 \mid do(x_0))}. \tag{19}$$

However, the computation becomes challenging when the observed variable O is high-dimensional. Considering that expert demonstrations are sometimes not optimal strategies, CPL frameworks are often combined with the GAIL or IRL optimization methods.

In addition to the DAG-based causal representation of SCM mentioned above, SCM can also be represented using Bayesian networks [74,75] or continuous-time neural networks based on differential equations [76]. The construction of SCMs through Bayesian networks requires consideration of prior knowledge to quantify the probabilistic dependencies between each variable in the causal model. This probabilistic modeling framework allows uncertainty quantification and probabilistic inference compared to the traditional DAG-based SCM. Another approach that uses the probabilistic assignment of constraints between variables is maximum causal entropy, as mentioned in Section 4. *B*. Unlike the conditions explored in this section, the causal structure defined by the maximum causal entropy is based on the fact that experts and agents can receive the same observed data and will not be further discussed here. By contrast, the SCM construction method based on continuous-time neural networks was implemented using neural networks to construct differential equations that conformed to the causal model. Its advantage is the stronger representation of the causal structure in a complex time series.

CPL performed well on several synthetic datasets, including highway-driving vehicle trajectories (Figure 3), MNIST digits [69,73], and visual navigation tasks [76]. They also exhibit promising applications in autonomous driving and industrial automation. This is because the safety and robustness of automated decision algorithms are currently being emphasized in the industry. CPL based on causal inference is more robust than unexplainable black box model systems.

5.3. Section Summary

CPL introduces causal models into strategy learning algorithms. In this section, we introduced preliminary knowledge of causal inference and the CPL framework. Causal inference investigates the causal relationships between variables of interest by defining the SCM. The unobserved properties between variables can be modeled using three methods: DAG, Bayesian networks, and continuous-time neural networks. The DAG uses nodes and directed sides to represent causal relationships between variables. In contrast, SCM constructed using Bayesian networks can quantify the probabilistic dependencies between variables. In addition, SCM based on neural networks has a stronger ability to characterize causal structures. For IL and IRL algorithms that lack interpretability, there is no guarantee that their definitions of the relationships between variables are correct. However, CPL frameworks can avoid causal confusion by introducing causal reasoning. Therefore, CPL can learn strategies safely, even if there is a difference between the observed behavior of the expert and the agent. Nonetheless, updating the causal structure might be very necessary when encountering a new environment or task, which could become a limitation of CPL. The computational cost depends on whether CPL employs model-free or model-based approaches. The high causal inference will also cause high computational costs according to tasks. In industrial environments (e.g., autonomous driving and navigation), where increasing emphasis is placed on algorithmic interpretability and safety, policy learning methods with causal inference intervention will have broader applications.

6. Discussion

6.1. Algorithmic Property Analysis

Policy learning enables the generation of expert behavior when the reward function is unknown or difficult to specify. This review examined three approaches based on data-driven policy-learning algorithms. The features of the algorithms reviewed in this paper are summarized in Table 2. Based on the literature review results, combined with their respective recent improvements, we selected relatively popular algorithms among the three groups of approaches: BC, MaxEnt-IRL, DAgger, GAIL, and AIRL. Furthermore, we discuss the approaches based on task requirements, data availability, and computational cost. The robustness in high-dimensional spaces, dataset quality, and dynamic systems is also discussed.

Table 2. Comparison of properties for reviewed policy learning algorithms.

Category	Sub-Category	Algorithm	Task Requirements	Data Availability	Computational Cost
IL	BC	Simple BC	Simple tasks with clear, static goals and easily accessible expert demonstrations	The quality, quantity, and diversity of data sets are important	Low computational cost
		DAgger	Challenging tasks involved in environmental interaction	Experts need to continuously supplement demonstration data during the iteration process	Computational costs can be high owing to the iterative process of collecting data and refining the model
		BC as RL pre-training	High-dimensional tasks where the reward function is hard to determine, suitable for complex and dynamic environments	The quality and quantity of data can significantly affect the effectiveness of pre-training	Lower computational cost than RL, can speed up training
	GAIL	GAIL	Complex and high-dimensional tasks with single-model assumption	Less dependent on large-scale datasets because of the adversarial framework	Relatively small computational cost
		cGAIL	Tasks that require learning diverse behaviors	Compared with GAIL, explicit modal labels are required for diverse behavior learning	Similar to GAIL but the computational cost depends on conditional information
		AC-GAIL	Tasks that require learning diverse behaviors	Compared with GAIL, explicit modal labels are required for diverse behavior learning	Relatively higher computational cost than GAIL due to auxiliary classifiers
		InfoGAIL	Tasks where diverse and information-rich behavior need to be captured	Similar to GAIL, multimodal labels are not required for diverse behavior learning	Depending on the mutual information estimation, the computational cost may be higher than GAIL
		VAE-GAIL	Tasks that require diverse behavior learning and more effective representation	Similar to GAIL, multimodal labels are not required for diverse behavior learning	Higher computational cost than GAIL due to the variational autoencoder
		Wasserstein GAIL	Tasks that require stable and robust imitation learning	Similar to GAIL	Depending on the efficiency of calculating the Wasserstein distance, the computational cost may be higher than GAIL
		MGAIL	Complex and high-dimensional tasks that emphasize environment interaction	Similar to GAIL	Higher computational cost than GAIL owing to the introduction of the environmental model
		Actor–Critic Policy Searching Based GAIL	Tasks that require efficient exploration and exploitation	Compared with GAIL, it improves sample efficiency and stability in policy learning	Relatively higher computational cost than GAIL due to the training of Actor–Critic networks
IRL	Maximum Margin Method	AL	High dimensional tasks where the reward function is required and hard to determine, suitable for complex and dynamic environments	Can learn an alternative policy of the expert policy, the robustness of suboptimal demonstration is poor	High computational cost owing to model-based algorithm and reward function searching
		MMP	Compared with AL, it can learn more complicated expert trajectories	Can reproduce the expert policy, the robustness of suboptimal demonstration is poor	High computational cost owing to model-based algorithm and reward function searching
	Entropy Optimization Method	MaxEnt-IRL	Tasks that emphasize uncertainty capture in expert behavior	Can learn optimal policies that outperform expert suboptimal policies	High computational cost similar to other IRL methods
		MCE-IRL	Compared with MaxEnt-IRL, it emphasizes the causal relationship in expert behavior	Based on MaxEnt-IRL, it considers maximizing causal entropy, involving the causal impact of actions on the environment and subsequent rewards	Relatively higher computational cost than MaxEnt-IRL owing to maximum causal entropy

Table 2. *Cont.*

Category	Sub-Category	Algorithm	Task Requirements	Data Availability	Computational Cost
		RE-IRL	Compared with other IRL methods, it is suitable for tasks with large-scale datasets because of less computational cost	Can learn optimal policies that outperform expert suboptimal policies	Lower computational cost than other IRL methods because of the model-free algorithm
	Bayesian Optimization Method	BIRL	Tasks that emphasize the uncertainty in reward function estimation	Prior knowledge is required to determine the posterior distribution, suitable for datasets with limited quantity	Compared with other IRL methods, Bayesian framework greatly increases computational cost
		GP-IRL	Tasks where the reward function has a smooth structure and the uncertainty of the reward function needs to be modeled explicitly	Kernel function may be needed to define the covariance structure of the Gaussian process when modeling reward functions	Compared with other IRL methods, Gaussian processes greatly increase computational cost
		ML-IRL	Tasks where the point estimate of the reward function is sufficient and uncertainty does not need to be modeled explicitly	Similar to other IRL methods	The computation cost is lower than BIRL and ML-IRL but lacks robustness to uncertainty
	Adversarial Method	AIRL	Complex and high-dimensional tasks that require defining a reward function efficiently and robustly	Compared with other IRL methods, less dependent on the quantity of dataset because of the adversarial framework	Lower computational cost than MaxEnt-IRL
CPL			Complex and high-dimensional tasks that emphasize causality and safety	It requires access to data that captures causal relationships between actions and outcomes	The computational cost depends on whether it is a model-free or model-based approach

Simple BC employs expert presentation datasets for supervised state-action mapping. BC is suitable for tasks where the presentation data are easily accessible and can completely cover the state-action space. Although computationally efficient, it is highly dependent on the quality and diversity of large datasets. Subtle inaccuracies can eventually lead to poor prediction results, known as covariate shifts and causal confusion. Hence, it is limited in dynamic environments and suboptimal dataset performance. When the task focuses more on the dynamics of the environment, IRL methods can recover the reward function from the presentation data that explains the behavior, helping the agent to make predictions in a dynamic system. In contrast, MaxEnt-IRL exploits the maximum entropy principle and learns strategies outperforming suboptimal expert demonstrations. MaxEnt-IRL has a high generalization ability to avoid many problems in BC. However, MaxEnt-IRL inevitably incurs high computational costs because the function search is ill-posed. This limits the computational efficiency in high-dimensional scenarios. Extending this method to large-scale datasets and environments remains challenging for the current IRL research. IRL is fundamentally about learning a cost function to explain expert behavior rather than telling the agent how to act. For cases that need to mimic behavior directly, imitating the demonstrated behavior through policy optimization can avoid high computational expenses. By iterating online to refine the policy based on BC, DAgger compensates for BC's shortcomings and enables interaction with the environment. It is also used for pre-training with RL. However, iterative iterations involve a high demand for supplementary presentation data, and the online learning approach cannot apply to specific presentation experts. Depending on the number of iterations, DAgger can be computationally demanding. In contrast, GAIL learns policies through generative adversarial training and is thus less dependent on large-scale datasets. Hence, it can be used to learn complex, high-dimensional spatial tasks and is robust to the distribution changes. To avoid redundant computation, GAIL is more computationally efficient. Other IL methods that utilize adversarial structures are mainly derived from GAIL and inherit its advantages, avoiding expert interactions during training. Similarly, for complex behavioral policy learning that requires recovery of the reward function, AIRL utilizes an adversarial approach to update the reward function based on MaxEnt-IRL, which improves computational efficiency while obtaining a robust reward function in dynamic environments.

These algorithms are widely used and accepted in policy learning. Each algorithm has its advantages and disadvantages. Depending on the specific requirements of different

tasks, data availability, and computational resources, users and researchers can select appropriate algorithms according to their corresponding needs.

6.2. Application Scenario Analysis

For the reviewed policy learning approaches, their applications in autonomous driving, robot control, and biological behavior understanding were analyzed. In this section, we highlight how the diverse characteristics of these algorithms lead to varied applications across these domains, summarizing the relevant policy learning methods for each application scenario.

(1) Autonomous driving: Autonomous driving tasks often involve autonomous driving systems and driver behavior in various traffic situations. Considering the visual information input, drivers' style modeling, and system safety of the autonomous driving system, policy learning methods such as DAgger, IRL, GAIL, and CPL are often used for autonomous driving. In contrast, DAgger is often used for automatic driving strategy planning from visual information [14,15] and can be used for simple lane-keeping tasks [19]. However, modeling drivers based on their gender, age, and driving style is a difficult task. IRL and GAIL model drivers' driving preferences from driver data, which enables optimal driving strategies in different driving situations (e.g., car following, lane changing, and overtaking) [2,35,43,44]. In addition, safety is also an important factor, as CPL improves the safety of autonomous driving strategies by modeling the causal relationship between variables such as speed and distance to the vehicle in front [69,73].

(2) Robot control: Robot control applications, including robot manipulation, motion control, and navigation, are discussed. According to different test conditions, BC and IRL can be used in robot manipulation with trajectory reproduction. For applications that require high environmental interaction conditions, BC combined with RL methods, IRL, and GAIL are often used to implement motion control and navigation tasks. Robot manipulation requires an agent to learn the presentation trajectories of humans. BC is an effective and simple method for low-dimensional robot manipulation without interactions [20]. However, for complex and high-dimensional robot manipulation trajectories, IRL performs significantly better than the BC [57,66,67]. For dexterous robot motion control, we found that using BC or DAgger as pre-training and combining it with RL's policy learning method can flexibly mimic animal motion and learn the ideal motion strategy, which is more practical [4]. Both IRL and GAIL were used for robot navigation tasks. The former focuses on reproducing human navigation trajectories [45,55,56], whereas the latter focuses on autonomous robot navigation in dynamic environments [33].

(3) Biological behavior understanding: Reward functions reproduced through behavioral data can explain behavioral preferences; therefore, with the biological behavior data input, IRL can thus be used to help understand biological behaviors. The study of behavioral strategies has been widely discussed as a fundamental topic in neuroscience and behavior [1]. IRL has been used to study foraging strategies in worms [60], olfactory searching of silk moths [46], migratory routes of migratory birds [47], and the behavior of group animals [59,68]. The recognition of behavior preferences is fundamental to understanding the mechanisms of animal behavior and analyzing the control activities of the nervous system [1,117,118]. It is not limited to animal behavior but also includes human behavior. IRL has also been used to learn user preferences from user data to help relevant organizations adjust user strategies to maximize their benefits [42,49,62].

In addition to the three typical examples mentioned above, these policy-learning methods are also used in strategy learning for game agents, HCI, and helicopter aerobatics. Practitioners and researchers can select appropriate strategy learning methods depending on the requirements of different application contexts.

6.3. Existing Challenges and Development Trends

In general, we believe that the current trend is to combine different techniques to bridge the barriers between algorithms. The latest IL approaches focus on achieving generalization capabilities across different domains with meta-learning. IRL techniques focus on combining algorithms with adversarial frameworks. Causal structure invariance or invariant causal features can help improve the generalization ability of CPL across various tasks by meta-learning or domain transfer. In addition, we analyze the challenges, their corresponding solutions, and future work methods in existing policy learning approaches from demonstration samples, diversification strategies, and computational costs.

- Data availability: Many studies face difficulties in accessing demonstration data when using data-driven policy learning, including obtaining large amounts of high-quality demonstration data, and the accessibility or interactivity of demonstration experts. Therefore, it is important to be able to take advantage of datasets to the maximum, i.e., learning through a limited amount of non-perfect demonstration data. Combining generative adversarial methods [64], self-supervision, meta-learning, transfer learning, or domain adaptation [119] is a new trend in the development of policy learning methods to solve the problems of difficulty in collecting expert data and reducing the dependence on large datasets.
- Learning diverse behaviors: Limited by algorithms, many methods currently can only learn a single behavioral strategy from expert demonstrations. Learning multimodal behaviors from data is still a challenge in current research. As mentioned in Section 3. *C*, some improved GAIL methods can effectively learn multimodal behavioral strategies. Besides, other algorithms are also considering learning diverse behaviors by combining VAE (e.g., BC with VAE [120–122]) or dividing multiple subtasks (i.e., multi-task learning) [123,124].
- High computational cost: Although the development of GPUs has eased the pressure of high computational cost, for most IRLs developed in model-based environments, DAgger involving multiple iterations, and certain GAIL algorithms using models, the high computational costs might limit their application in large-scale datasets. There have been proposed approaches that consider improving function search methods [64] or using model-free algorithms [23] to improve computational efficiency. Further computational cost reduction is also a priority that needs to be addressed in future work.
- Transition to real-world applications: Most algorithms are based on the following ideal assumptions: (1) the agent can access complete information about the environment; (2) the information observed by the expert and the agent is consistent. These are difficult to realize in practice. Simple BC can apply low-dimensional tasks to reality, while most policy learning methods are still tested in simulated environments. Reducing the gap between the simulated and real environments requires the generalization ability of the algorithm. Moreover, tasks involving HCI place more importance on algorithmic security. Therefore, using partially observable MDPs (POMDPs) [32,119], causal modeling of variables [69], and sim-to-real transfer techniques [125] will be key to extending policy learning to real-world problem solutions.

7. Conclusions

This review investigates the computational approaches used to learn strategies from behavior. This includes research findings from the past decade. The policy-learning approaches reviewed in the literature are divided into three categories based on their focus: IL, IRL, and CPL. This review provides a comprehensive review of the development, characteristics, and applications of these three algorithms. It also compares and discusses policy learning methods in terms of their properties and applications, with the analysis of existing challenges and future development directions. Data-driven behavioral learning-based algorithms change the adaptive and flexible behavioral generation method for manually defined rewards such that agents can maximize the use of expert experience and the environment

to achieve desirable behaviors with maximum rewards. They contribute to developing robot dexterity control, HCI algorithms, and even biological behavior understanding.

Funding: This work was supported by JSPS KAKENHI a Grant-in-Aid for Scientific Research (A) (JP23H00481), a Grant-in-Aid for Challenging Research (Exploratory) (JP23K18472), and JST SPRING (JPMJSP2114).

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

References

1. Anderson, D.J.; Perona, P. Toward a Science of Computational Ethology. *Neuron* **2014**, *84*, 18–31. [CrossRef]
2. Zhou, Y.; Fu, R.; Wang, C.; Zhang, R. Modeling Car-Following Behaviors and Driving Styles with Generative Adversarial Imitation Learning. *Sensors* **2020**, *20*, 5034. [CrossRef] [PubMed]
3. Fahad, M.; Chen, Z.; Guo, Y. Learning how pedestrians navigate: A deep inverse reinforcement learning approach. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 819–826.
4. Peng, X.B.; Coumans, E.; Zhang, T.; Lee, T.W.; Tan, J.; Levine, S. Learning agile robotic locomotion skills by imitating animals. *arXiv* **2020**, arXiv:2004.00784.
5. Muelling, K.; Boularias, A.; Mohler, B.; Schölkopf, B.; Peters, J. Learning strategies in table tennis using inverse reinforcement learning. *Biol. Cybern.* **2014**, *108*, 603–619. [CrossRef] [PubMed]
6. Dürr, V.; Theunissen, L.M.; Dallmann, C.J.; Hoinville, T.; Schmitz, J. Motor Flexibility in Insects: Adaptive Coordination of Limbs in Locomotion and near-Range Exploration. *Behav. Ecol. Sociobiol.* **2017**, *72*, 15. [CrossRef]
7. Liberati, A.; Altman, D.G.; Tetzlaff, J.; Mulrow, C.; Gøtzsche, P.C.; Ioannidis, J.P.; Clarke, M.; Devereaux, P.J.; Kleijnen, J.; Moher, D. The PRISMA Statement for Reporting Systematic Reviews and Meta-Analyses of Studies That Evaluate Health Care Interventions: Explanation and Elaboration. *PLoS Med.* **2009**, *6*, e1000100. [CrossRef] [PubMed]
8. Zhang, T.; McCarthy, Z.; Jow, O.; Lee, D.; Chen, X.; Goldberg, K.; Abbeel, P. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 5628–5635.
9. Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L.D.; Monfort, M.; Muller, U.; Zhang, J.; et al. End to end learning for self-driving cars. *arXiv* **2016**, arXiv:1604.07316.
10. Codevilla, F.; Santana, E.; López, A.M.; Gaidon, A. Exploring the limitations of behavior cloning for autonomous driving. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9329–9338.
11. Edwards, A.; Sahni, H.; Schroecker, Y.; Isbell, C. Imitating latent policies from observation. In Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; Volume 97, pp. 1755–1763.
12. Torabi, F.; Warnell, G.; Stone, P. Behavioral cloning from observation. In Proceedings of the 27th International Joint Conferences on Artificial Intelligence (IJCAI), Stockholm, Sweden, 13–19 July 2018; pp. 4950–4957.
13. Liu, Y.; Gupta, A.; Abbeel, P.; Levine, S. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 1118–1125.
14. Zhang, J.; Cho, K. Query-efficient imitation learning for end-to-end simulated driving. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
15. Abeysirigoonawardena, Y.; Shkurti, F.; Dudek, G. Generating adversarial driving scenarios in high-fidelity simulators. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 8271–8277.
16. Harmer, J.; Gisslén, L.; del Val, J.; Holst, H.; Bergdahl, J.; Olsson, T.; Sjöö, K.; Nordin, M. Imitation learning with concurrent actions in 3D games. In Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games (CIG), Maastricht, The Netherlands, 14–17 August 2018; pp. 1–8.
17. Reddy, S.; Dragan, A.D.; Levine, S. SQIL: Imitation learning via reinforcement learning with sparse rewards. In Proceedings of the International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26 April–1 May 2020.
18. Gupta, A.; Kumar, V.; Lynch, C.; Levine, S.; Hausman, K. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In Proceedings of the Conference on Robot Learning (CoRL), Osaka, Japan, 30 October–1 November 2019; Volume 100, pp. 1025–1037.
19. Gao, Y.; Liu, Y.; Zhang, Q.; Wang, Y.; Zhao, D.; Ding, D.; Pang, Z.; Zhang, Y. Comparison of control methods based on imitation learning for autonomous driving. In Proceedings of the 10th International Conference on Intelligent Control and Information Processing (ICICIP), Marrakesh, Morocco, 14–19 December 2019; pp. 274–281.
20. Rajaraman, N.; Yang, L.; Jiao, J.; Ramchandran, K. Toward the fundamental limits of imitation learning. In Proceedings of the Advances Neural Information Processing Systems (NeurIPS), 6–12 December 2020; Volume 33, pp. 2914–2924.

21. Ho, J.; Ermon, S. Generative adversarial imitation learning. In Proceedings of the Advances Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; Volume 29.
22. Kuefler, A.; Morton, J.; Wheeler, T.; Kochenderfer, M. Imitating driver behavior with generative adversarial networks. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Redondo Beach, CA, USA, 11–14 June 2017; pp. 204–211.
23. Blondé, L.; Kalousis, A. Sample-efficient imitation learning via generative adversarial nets. In Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS), Naha , Japan, 16–18 April 2019; pp. 3138–3148.
24. Hawke, J.; Shen, R.; Gurau, C.; Sharma, S.; Reda, D.; Nikolov, N.; Mazur, P.; Micklethwaite, S.; Griffiths, N.; Shah, A.; et al. Urban driving with conditional imitation learning. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), 31 May–31 August 2020; pp. 251–257.
25. Zhang, X.; Li, Y.; Zhou, X.; Luo, J. CGAIL: Conditional generative adversarial imitation learning—An application in taxi Drivers' strategy learning. *IEEE Trans. Big Data* **2020**, *8*, 1288–1300. [CrossRef]
26. Fei, C.; Wang, B.; Zhuang, Y.; Zhang, Z.; Hao, J.; Zhang, H.; Ji, X.; Liu, W. Triple-GAIL: A multi-modal imitation learning framework with generative adversarial nets. In Proceedings of the 29th International Joint Conferences on Artificial Intelligence (IJCAI), 19–26 August 2021; pp. 2929–2935.
27. Lin, J.; Zhang, Z. ACGAIL: Imitation learning about multiple intentions with auxiliary classifier GANs. In Proceedings of the PRICAI 2018: Trends Artificial Intelligence, Nanjing, China, 28–31 August 2018; pp. 321–334.
28. Merel, J.; Tassa, Y.; TB, D.; Srinivasan, S.; Lemmon, J.; Wang, Z.; Wayne, G.; Heess, N. Learning human behaviors from motion capture by adversarial imitation. *arXiv* **2017**, arXiv:1707.02201.
29. Li, Y.; Song, J.; Ermon, S. InfoGAIL: Interpretable imitation learning from visual demonstrations. In Proceedings of the Advances Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
30. Kuefler, A.; Kochenderfer, M.J. Burn-in demonstrations for multi-modal imitation learning. In Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems, Stockholm, Sweden, 10–15 July 2018; pp. 1071–1078.
31. Wang, Z.; Merel, J.S.; Reed, S.E.; de Freitas, N.; Wayne, G.; Heess, N. Robust imitation of diverse behaviors. In Proceedings of the Advances Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
32. Rafailov, R.; Yu, T.; Rajeswaran, A.; Finn, C. Visual adversarial imitation learning using variational models. In Proceedings of the Advances Neural Information Processing Systems (NeurIPS), 6–14 December 2021; Volume 34, pp. 3016–3028.
33. Tai, L.; Zhang, J.; Liu, M.; Burgard, W. Socially compliant navigation through raw depth inputs with generative adversarial imitation learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 1111–1117.
34. Baram, N.; Anschel, O.; Mannor, S. Model-based adversarial imitation learning. *arXiv* **2016**, arXiv:1612.02179.
35. Bronstein, E.; Palatucci, M.; Notz, D.; White, B.; Kuefler, A.; Lu, Y.; Paul, S.; Nikdel, P.; Mougin, P.; Chen, H.; et al. Hierarchical Model-Based Imitation Learning for Planning in Autonomous Driving. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 8652–8659.
36. Kostrikov, I.; Agrawal, K.K.; Dwibedi, D.; Levine, S.; Tompson, J. Discriminator-Actor-Critic: Addressing Sample Inefficiency and Reward Bias in Adversarial Imitation Learning. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
37. Sasaki, F.; Yohira, T.; Kawaguchi, A. Sample efficient imitation learning for continuous control. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
38. Lee, G.; Luo, M.; Zambetta, F.; Li, X. Learning a Super Mario controller from examples of human play. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 1–8.
39. Wulfmeier, M.; Ondruska, P.; Posner, I. Maximum entropy deep inverse reinforcement learning. *arXiv* **2015**, arXiv:1507.04888.
40. Tschiatschek, S.; Ghosh, A.; Haug, L.; Devidze, R.; Singla, A. Learner-aware teaching: Inverse reinforcement learning with preferences and constraints. In Proceedings of the Advances Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
41. Herman, M.; Gindele, T.; Wagner, J.; Schmitt, F.; Burgard, W. Inverse reinforcement learning with simultaneous estimation of rewards and dynamics. In Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS), Cadiz, Spain, 9–11 May 2016; pp. 102–110.
42. Pan, M.; Huang, W.; Li, Y.; Zhou, X.; Liu, Z.; Song, R.; Lu, H.; Tian, Z.; Luo, J. DHPA: Dynamic human preference analytics framework: A case study on taxi drivers' learning curve analysis. *ACM Trans. Intell. Syst. Technol. (TIST)* **2020**, *11*, 1–19. [CrossRef]
43. Sadigh, D.; Sastry, S.; Seshia, S.A.; Dragan, A.D. Planning for autonomous cars that leverage effects on human actions. In Proceedings of the Robotics: Science and System, Cambridge, MA, USA, 18–22 June 2016; Volume 2, pp. 1–9.
44. You, C.; Lu, J.; Filev, D.; Tsiotras, P. Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning. *Robot. Auton. Syst.* **2019**, *114*, 1–18. [CrossRef]
45. Martinez-Gil, F.; Lozano, M.; García-Fernández, I.; Romero, P.; Serra, D.; Sebastián, R. Using inverse reinforcement learning with real trajectories to get more trustworthy pedestrian simulations. *Math.* **2020**, *8*, 1479. [CrossRef]
46. Hernandez-Reyes, C.; Shigaki, S.; Yamada, M.; Kondo, T.; Kurabayashi, D. Learning a Generic Olfactory Search Strategy From Silk Moths by Deep Inverse Reinforcement Learning. *IEEE Trans. Med. Robot. Bionics* **2021**, *4*, 241–253. [CrossRef]
47. Hirakawa, T.; Yamashita, T.; Tamaki, T.; Fujiyoshi, H.; Umezu, Y.; Takeuchi, I.; Matsumoto, S.; Yoda, K. Can AI predict animal movements? Filling gaps in animal trajectories using inverse reinforcement learning. *Ecosphere* **2018**, *9*, e02447. [CrossRef]

48. Ermon, S.; Xue, Y.; Toth, R.; Dilkina, B.; Bernstein, R.; Damoulas, T.; Clark, P.; DeGloria, S.; Mude, A.; Barrett, C.; et al. Learning large-scale dynamic discrete choice models of spatio-temporal preferences with application to migratory pastoralism in East Africa. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29.
49. Lage, I.; Lifschitz, D.; Doshi-Velez, F.; Amir, O. Exploring computational user models for agent policy summarization. In Proceedings of the 28th International Joint Conferences on Artificial Intelligence (IJCAI), Macao, China, 10–16 August 2019; Volume 28, pp. 1401–1407.
50. Wulfmeier, M.; Rao, D.; Wang, D.Z.; Ondruska, P.; Posner, I. Large-scale cost function learning for path planning using deep inverse reinforcement learning. *Int. J. Robot. Res.* **2017**, *36*, 1073–1087. [CrossRef]
51. Zheng, J.; Liu, S.; Ni, L.M. Robust Bayesian inverse reinforcement learning with sparse behavior noise. In Proceedings of the AAAI Conference on Artificial Intelligence, Quebec, Canada, 27–31 July 2014; Volume 28.
52. Brown, D.; Coleman, R.; Srinivasan, R.; Niekum, S. Safe imitation learning via fast Bayesian reward inference from preferences. In Proceedings of the 37th International Conference on Machine Learning (ICML), 12–18 July 2020; pp. 1165–1177.
53. Mourad, N.; Ezzeddine, A.; Nadjar Araabi, B.; Nili Ahmadabadi, M. Learning from demonstrations and human evaluative feedbacks: Handling sparsity and imperfection using inverse reinforcement learning approach. *J. Robot.* **2020**, *2020*, 3849309. [CrossRef]
54. Brown, D.; Niekum, S. Efficient probabilistic performance bounds for inverse reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
55. Okal, B.; Arras, K.O. Learning socially normative robot navigation behaviors with Bayesian inverse reinforcement learning. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 2889–2895.
56. Xia, C.; El Kamel, A. Neural inverse reinforcement learning in autonomous navigation. *Robot. Auton. Syst.* **2016**, *84*, 1–14. [CrossRef]
57. Batzianoulis, I.; Iwane, F.; Wei, S.; Correia, C.G.P.R.; Chavarriaga, R.; Millán, J.d.R.; Billard, A. Customizing skills for assistive robotic manipulators, an inverse reinforcement learning approach with error-related potentials. *Commun. Biol.* **2021**, *4*, 1406. [CrossRef]
58. Choi, J.; Kim, K.E. Hierarchical Bayesian inverse reinforcement learning. *IEEE Trans. Cybern.* **2014**, *45*, 793–805. [CrossRef]
59. Schafer, T.L.; Wikle, C.K.; Hooten, M.B. Bayesian inverse reinforcement learning for collective animal movement. *Ann. Appl. Statist.* **2022**, *16*, 999–1013. [CrossRef]
60. Yamaguchi, S.; Naoki, H.; Ikeda, M.; Tsukada, Y.; Nakano, S.; Mori, I.; Ishii, S. Identification of Animal Behavioral Strategies by Inverse Reinforcement Learning. *PLoS Comput. Biol.* **2018**, *14*, e1006122. [CrossRef]
61. Nasernejad, P.; Sayed, T.; Alsaleh, R. Modeling pedestrian behavior in pedestrian-vehicle near misses: A continuous Gaussian Process Inverse Reinforcement Learning (GP-IRL) approach. *Accid. Anal. Prev.* **2021**, *161*, 106355. [CrossRef] [PubMed]
62. Massimo, D.; Ricci, F. Harnessing a generalised user behaviour model for next-POI recommendation. In Proceedings of the 12th ACM Conference on Recommender Systems, Vancouver, BC, Canada, 2–7 October 2018; pp. 402–406.
63. Yang, S.Y.; Qiao, Q.; Beling, P.A.; Scherer, W.T.; Kirilenko, A.A. Gaussian process-based algorithmic trading strategy identification. *Quant. Financ.* **2015**, *15*, 1683–1703. [CrossRef]
64. Fu, J.; Luo, K.; Levine, S. Learning robust rewards with adversarial inverse reinforcement learning. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
65. Qureshi, A.H.; Boots, B.; Yip, M.C. Adversarial imitation via variational inverse reinforcement learning. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
66. Zhang, X.; Sun, L.; Kuang, Z.; Tomizuka, M. Learning variable impedance control via inverse reinforcement learning for force-related tasks. *IEEE Robot. Automat. Lett.* **2021**, *6*, 2225–2232. [CrossRef]
67. Ghasemipour, S.K.S.; Zemel, R.; Gu, S. A divergence minimization perspective on imitation learning methods. In Proceedings of the Conference on Robot Learning (CoRL), Cambridge, MA, USA, 16–18 November 2020; pp. 1259–1277.
68. Yu, X.; Wu, W.; Feng, P.; Tian, Y. Swarm inverse reinforcement learning for biological systems. In Proceedings of the 2021 IEEE International Conference on Bioinformatics Biomedicine (BIBM), Houston, TX, USA, 9–12 December 2021; pp. 274–279.
69. Zhang, J.; Kumor, D.; Bareinboim, E. Causal imitation learning with unobserved confounders. In Proceedings of the Advances Neural Information Processing Systems (NeurIPS), Online, 6–12 December 2020; Volume 33, pp. 12263–12274.
70. Kumor, D.; Zhang, J.; Bareinboim, E. Sequential causal imitation learning with unobserved confounders. In Proceedings of the Advances Neural Information Processing Systems (NeurIPS), Online, 6–14 December 2021; Volume 34, pp. 14669–14680.
71. Bica, I.; Jarrett, D.; van der Schaar, M. Invariant causal imitation learning for generalizable policies. In Proceedings of the Advances Neural Information Processing Systems (NeurIPS), Online, 6–14 December 2021; Volume 34, pp. 3952–3964.
72. Swamy, G.; Choudhury, S.; Bagnell, D.; Wu, S. Causal imitation learning under temporally correlated noise. In Proceedings of the 39th International Conference on Machine Learning (ICML), Honolulu, HI, USA, 17–23 July 2022; pp. 20877–20890.
73. Ruan, K.; Di, X. Learning human driving behaviors with sequential causal imitation learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 22 February–1 March 2022; Volume 36, pp. 4583–4592.
74. Swamy, G.; Choudhury, S.; Bagnell, J.; Wu, S.Z. Sequence model imitation learning with unobserved contexts. In Proceedings of the Advances Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 28 November–9 December 2022; Volume 35, pp. 17665–17676.

75. Vuorio, R.; Brehmer, J.; Ackermann, H.; Dijkman, D.; Cohen, T.; de Haan, P. Deconfounded Imitation Learning. In Proceedings of the Deep Reinforcement Learning Workshop NeurIPS, New Orleans, LA, USA, 9 December 2022.
76. Vorbach, C.; Hasani, R.; Amini, A.; Lechner, M.; Rus, D. Causal navigation by continuous-time neural networks. In Proceedings of the Advances Neural Information Processing Systems (NeurIPS), 6–14 December 2021; Volume 34, pp. 12425–12440.
77. Arora, S.; Doshi, P. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artif. Intell.* **2021**, *297*, 103500. [CrossRef]
78. Zheng, B.; Verma, S.; Zhou, J.; Tsang, I.; Chen, F. Imitation learning: Progress, taxonomies and challenges. *arXiv* **2021**, arXiv:2106.12177.
79. Di, X.; Shi, R. A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to AI-guided driving policy learning. *Transp. Res. Part C Emerg. Technol.* **2021**, *125*, 103008. [CrossRef]
80. Gajjar, P.; Dodia, V.; Mandaliya, S.; Shah, P.; Ukani, V.; Shukla, M. Path Planning and Static Obstacle Avoidance for Unmanned Aerial Systems. In Proceedings of the International Conference on Advances in Smart Computing and Information Security, Rajkot, India, 24–26 November 2022; pp. 262–270.
81. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285. [CrossRef]
82. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef] [PubMed]
83. Hussein, A.; Gaber, M.M.; Elyan, E.; Jayne, C. Imitation Learning: A Survey of Learning Methods. *ACM Comput. Surv.* **2017**, *50*, 1–35. [CrossRef]
84. Ross, S.; Gordon, G.; Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 627–635.
85. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [CrossRef]
86. Arjovsky, M.; Bottou, L. Towards Principled Methods for Training Generative Adversarial Networks. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
87. Baram, N.; Anschel, O.; Caspi, I.; Mannor, S. End-to-end differentiable adversarial imitation learning. In Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; Volume 70, pp. 390–399.
88. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
89. Nowozin, S.; Cseke, B.; Tomioka, R. f-GAN: Training generative neural samplers using variational divergence minimization. In Proceedings of the Advances Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; Volume 29.
90. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In Proceedings of the Advances Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; Volume 29.
91. Larsen, A.B.L.; Sønderby, S.K.; Larochelle, H.; Winther, O. Autoencoding beyond pixels using a learned similarity metric. In Proceedings of the 33rd International Conference on Machine Learning (ICML), New York, NY, USA, 19–24 June 2016; Volume 48, pp. 1558–1566.
92. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; Volume 70, pp. 214–223.
93. Rezende, D.J.; Mohamed, S.; Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In Proceedings of the 31st International Conference on Machine Learning (ICML), Beijing, China, 21–26 June 2014; Volume 32, pp. 1278–1286.
94. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
95. Zhu, J.; Jiang, C. TAC-GAIL: A Multi-modal Imitation Learning Method. In Proceedings of the Neural Information Processing: 27th International Conference (ICONIP), Bangkok, Thailand, 23–27 November 2020; pp. 688–699.
96. Song, J.; Ren, H.; Sadigh, D.; Ermon, S. Multi-agent generative adversarial imitation learning. In Proceedings of the Advances Neural Information Processing Systems (NeurIPS), Montreal, Canada, 2–8 December 2018; Volume 31.
97. De Haan, P.; Jayaraman, D.; Levine, S. Causal confusion in imitation learning. In Proceedings of the Advances Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
98. Ng, A.Y.; Russell, S. Algorithms for inverse reinforcement learning. In Proceedings of the 17th International Conference on Machine Learning (ICML), Standord, CA, USA, 29 June–2 July 2000; pp. 663–670.
99. Puterman, M.L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*; John Wiley & Sons: New York, NY, USA, 2014.
100. Abbeel, P.; Ng, A.Y. Apprenticeship learning via inverse reinforcement learning. In Proceedings of the 21st International Conference on Machine Learning (ICML), Banff, AB, Canada, 4–8 July 2004; p. 1.
101. Ratliff, N.D.; Bagnell, J.A.; Zinkevich, M.A. Maximum margin planning. In Proceedings of the 23rd International Conference on Machine Learning (ICML), Pittsburgh, PA, USA, 25–29 June 2006; pp. 729–736.
102. Ratliff, N.D.; Silver, D.; Bagnell, J.A. Learning to search: Functional gradient techniques for imitation learning. *Auton. Robot.* **2009**, *27*, 25–53. [CrossRef]

103. Abbeel, P.; Coates, A.; Quigley, M.; Ng, A. An application of reinforcement learning to aerobatic helicopter flight. In Proceedings of the Advances Neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 4–9 December 2006; Volume 19.
104. Valencia-Murillo, R.; Arana-Daniel, N.; López-Franco, C.; Alanís, A.Y. Rough terrain perception through geometric entities for robot navigation. In Proceedings of the 2nd International Conference on Advances in Computer Science and Engineering (CSE 2013), Los Angeles, CA, USA, 1–2 July 2013; pp. 309–314.
105. Lafferty, J.D.; McCallum, A.; Pereira, F.C.N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the 18th International Conference on Machine Learning (ICML), Williamstown, MA, USA, 28 June–1 July 2001; pp. 282–289.
106. Aghasadeghi, N.; Bretl, T. Maximum entropy inverse reinforcement learning in continuous state spaces with path integrals. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 25–30 September 2011; pp. 1561–1566.
107. Jaynes, E.T. Information theory and statistical mechanics. II. *Phys. Rev.* **1957**, *108*, 171. [CrossRef]
108. Ziebart, B.D.; Bagnell, J.A.; Dey, A.K. Modeling Interaction via the Principle of Maximum Causal Entropy. In Proceedings of the 27th International Conference on Machine Learning (ICML), Haifa, Israel, 21–24 June 2010; pp. 1255–1262.
109. Boularias, A.; Kober, J.; Peters, J. Relative entropy inverse reinforcement learning. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 182–189.
110. Ramachandran, D.; Amir, E. Bayesian Inverse Reinforcement Learning. In Proceedings of the 20th International Joint Conferences on Artificial Intelligence (IJCAI), Hyderabad, India, 6–12 January 2007; Volume 7, pp. 2586–2591.
111. Levine, S.; Popovic, Z.; Koltun, V. Nonlinear inverse reinforcement learning with Gaussian processes. In Proceedings of the Advances Neural Information Processing Systems (NIPS), Granada, Spain, 12–17 December 2011; Volume 24.
112. Puiutta, E.; Veith, E.M. Explainable reinforcement learning: A survey. In Proceedings of the International Cross Domain Conference for Machine Learning & Knowledge Extraction (CD-MAKE), Dublin, Ireland, 25–28 August 2020; pp. 77–95.
113. Lee, J.H. Complementary reinforcement learning towards explainable agents. *arXiv* **2019**, arXiv:1901.00188.
114. Yao, L.; Chu, Z.; Li, S.; Li, Y.; Gao, J.; Zhang, A. A survey on causal inference. *ACM Trans. Knowl. Discov. Data (TKDD)* **2021**, *15*, 1–46. [CrossRef]
115. Pearl, J. Causal diagrams for empirical research. *Biometrika* **1995**, *82*, 669–688. [CrossRef]
116. Rubin, D.B. Estimating causal effects of treatments in randomized and nonrandomized studies. *J. Educ. Psychol.* **1974**, *66*, 688–701. [CrossRef]
117. Manoonpong, P.; Patanè, L.; Xiong, X.; Brodoline, I.; Dupeyroux, J.; Viollet, S.; Arena, P.; Serres, J.R. Insect-inspired robots: Bridging biological and artificial systems. *Sensors* **2021**, *21*, 7609. [CrossRef] [PubMed]
118. Wang, Y.; Hayashibe, M.; Owaki, D. Prediction of whole-body velocity and direction from local leg joint movements in insect walking via LSTM neural networks. *IEEE Robot. Automat. Lett.* **2022**, *7*, 9389–9396. [CrossRef]
119. Raychaudhuri, D.S.; Paul, S.; Vanbaar, J.; Roy-Chowdhury, A.K. Cross-domain imitation from observations. In Proceedings of the 38th International Conference on Machine Learning (ICML), Online, 18–24 July 2021; pp. 8902–8912.
120. Kipf, T.; Li, Y.; Dai, H.; Zambaldi, V.; Sanchez-Gonzalez, A.; Grefenstette, E.; Kohli, P.; Battaglia, P. Compile: Compositional imitation learning and execution. In Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 3418–3428.
121. Mandlekar, A.; Xu, D.; Martín-Martín, R.; Savarese, S.; Fei-Fei, L. GTI: Learning to Generalize across Long-Horizon Tasks from Human Demonstrations. In Proceedings of the Robotics: Science and Systems XVI, Corvalis, OR, USA, 12–16 July 2020; p. 061.
122. Bonatti, R.; Madaan, R.; Vineet, V.; Scherer, S.; Kapoor, A. Learning visuomotor policies for aerial navigation using cross-modal representations. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 1637–1644.
123. Wang, L.; Tang, R.; He, X.; He, X. Hierarchical imitation learning via subgoal representation learning for dynamic treatment recommendation. In Proceedings of the 15th ACM International Conference on Web Search and Data Mining, Online, 21–25 February 2022; pp. 1081–1089.
124. Shimosaka, M.; Nishi, K.; Sato, J.; Kataoka, H. Predicting driving behavior using inverse reinforcement learning with multiple reward functions towards environmental diversity. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Jeju Island, Republic of Korea, 2–5 June 2015; pp. 567–572.
125. Zhu, W.; Guo, X.; Owaki, D.; Kutsuzawa, K.; Hayashibe, M. A Survey of Sim-to-Real Transfer Techniques Applied to Reinforcement Learning for Bioinspired Robots. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 3444–3459. [CrossRef] [PubMed]