sensors MDPI

Review

Learning for a Robot: Deep Reinforcement Learning, Imitation Learning, Transfer Learning

Jiang Hua [1], Liangcai Zeng [1], Gongfa Li [1] and Zhaojie Ju [2,*]

[1] Key Laboratory of Metallurgical Equipment and Control Technology, Ministry of Education, Wuhan University of Science and Technology, Wuhan 430081, China; huajiang@wust.edu.cn (J.H.); zengliangcai@wust.edu.cn (L.Z.); ligongfa@wust.edu.cn (G.L.)

[2] School of Computing, University of Portsmouth, Portsmouth 03801, UK

[*] Correspondence: zhaojie.ju@port.ac.uk

Abstract: Dexterous manipulation of the robot is an important part of realizing intelligence, but manipulators can only perform simple tasks such as sorting and packing in a structured environment. In view of the existing problem, this paper presents a state-of-the-art survey on an intelligent robot with the capability of autonomous deciding and learning. The paper first reviews the main achievements and research of the robot, which were mainly based on the breakthrough of automatic control and hardware in mechanics. With the evolution of artificial intelligence, many pieces of research have made further progresses in adaptive and robust control. The survey reveals that the latest research in deep learning and reinforcement learning has paved the way for highly complex tasks to be performed by robots. Furthermore, deep reinforcement learning, imitation learning, and transfer learning in robot control are discussed in detail. Finally, major achievements based on these methods are summarized and analyzed thoroughly, and future research challenges are proposed.

Keywords: dexterous manipulation; adaptive and robust control; deep reinforcement learning; imitation learning; transfer learning

check for updates

Citation: Hua, J.; Zeng, L.; Li, G.; Ju, Z. Learning for a Robot: Deep Reinforcement Learning, Imitation Learning, Transfer Learning. *Sensors* **2021**, *21*, 1278. https://doi.org/10.3390/s21041278

Academic Editor: Sašo Blažič

Received: 6 January 2021
Accepted: 5 February 2021
Published: 11 February 2021

1. Introduction

The concept of robot gripping originated in 1962 with industrial robot Unimate which used a two-finger to grab wooden blocks and stack them together. The robot is designed to mimic the function of humans, so the pioneers of the field have done a lot of research on the grasp and manipulation mechanism. Human beings can manipulate objects and explore the world in various environments, so we also want robots to be as capable as humans. However, manipulation of the robot is not as simple as we think even though studies have been conducted for decades [1]. Although robotics has gained vast progress in mechanical design, perception, and robust control targeted to grasp and handle objects, robotic manipulation is still a poor proxy for human dexterity. To date, no robots can easily hand-wash dishes, button a shirt, or peel a potato.

Children are born with the ability to grab, and then get the adult-equivalent competence for planning sequences of manipulation skills after the learning of 9 years [2]. Neuroscience studies have shown that humans can grasp steadily and perform a variety of dexterous manipulations based on rich perceptual information and intelligence, so researchers want robots to have human-like abilities. Yaxu et al. analyze and compare existing human grasp taxonomies and synthesize them into a single new taxonomy [3]. Although a variety of research is carried out, how to implement various grasps and manipulation is still a problem of its own [4].

In order to realize the intelligent operation of the robot, it can be summarized into two main functional requirements, the first is the visual perception, the other is the intelligence of the robot. In the early stage, robots did not have the ability of perception. They grasped the robot mainly by means of manual teaching, hard coding, data gloves, and other tactile sensors. With the breakthrough of hardware technology, the integration of multi-model information such as vision, touch, and perception enables robots to identify the pose of the target more accurately [5]. So far, the biggest challenge at present is how to learn the optimal grasping strategy based on visual information.

At present, although the robot can perform some simple repetitive tasks well, it still cannot adapt to the complex environment with shielding or changing lighting conditions in real time. With the increasing demand for intelligent robots, it is urgent to design a robot grasping solution with independent ability of decision-making and learning. Therefore, the robot is a high-level embodiment of artificial intelligence in the physical world, and automation is the basis of intelligence [6]. The rapid development of artificial intelligence technology that encapsulates models of uncertainty further advances in adaptive and robust control. These machine learning algorithms for object grasps mainly include analytical and empirical approaches [7]. These methods are effective, but simplify the grasping environment and are based on hand-crafted features. Therefore, they are arduous, time-consuming, and cannot adapt to complex environments [8]. It is necessary to create a universal robotic solution for various environments, which have the ability to make decisions and learn independently. At present, deep reinforcement learning is the main method of intelligent decision and control of robots, which enables robots to learn a task from scratch. This method requires a lot of trials and incurs many errors, which is difficult to apply to actual robot manipulation [9]. To solve this problem, imitation learning and transfer learning are proposed. Ultimately, it is hoped that an end-to-end neural network can be constructed to output the motor control of each joint simply by inputting the observed image [10].

To sum up, this paper will present a state-of-the-art survey on an intelligent robot with the capability of autonomous deciding and learning. The paper first reviews the main achievements and research in adaptive and robust control. The survey reveals that the latest research in deep learning and reinforcement learning has paved the way for highly complex tasks to be performed by robots. Furthermore, three main methods of deep reinforcement learning, imitation learning, and transfer learning are discussed for a robot. Finally, major achievements based on these methods are summarized and analyzed thoroughly, and future research challenges are proposed.

The remainder of the paper is arranged as follows. In Section 2, we survey the theory of how to form stable manipulation and introduce the research background. Section 3 focuses on how a robot can learn a motor control policy via deep reinforcement learning as a complete solution to a task. Section 4 describes approaches of imitation learning to master skills by observing movements from only a small number of samples. Section 5 describes approaches that knowledge can be transferred to the real robot by building a robot virtual simulation system based on transfer learning. Finally, latest applications and future research directions are discussed.

2. The Background

For decades, researchers have worked to establish the theory of how to form a stable manipulation. However, manipulating an object is a far more daunting problem. At present there are mainly two directions, one way is to set up a mathematical model aimed at determining the minimal number and optimal positions of the fingertips on the object's surface to ensure stability [11]. The second way is data-driven methods by establishing a database about the manual grasping type, the optimal solution of grasping can be obtained by analyzing and understanding the data with sensors information and prior knowledge [12]. The survey is structured as Figure 1.

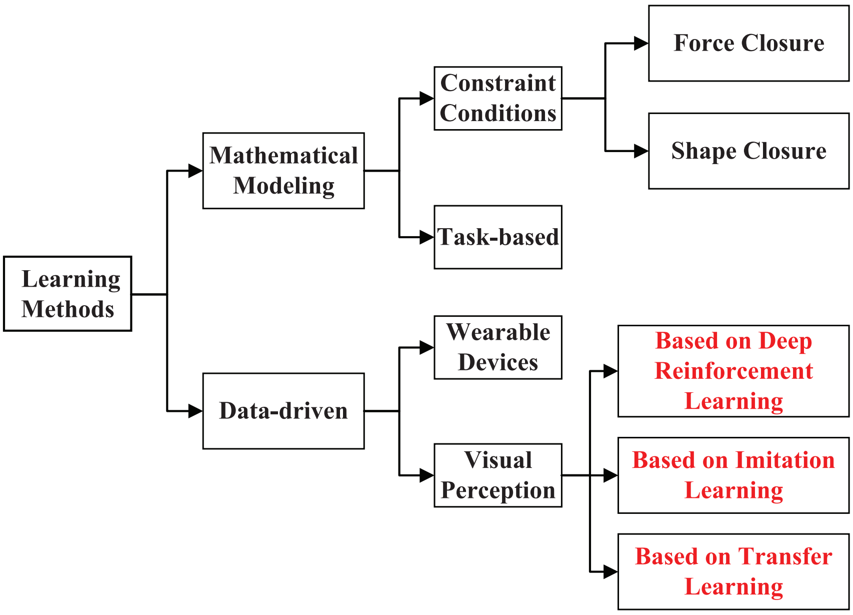

Figure 1. Overview of the structure of the survey.

The method of mathematical modeling needs to take many constraints into consideration and obtains the optimal value by establishing the objective function [13]. As shown in the Figure 1, the closed conditions are the major factors to be considered. Force closure and shape closure are two important manifestations of closed conditions, which are widely used in the plan of manipulation [14]. Force closure means that the contact force spiral on the surface of the object is in equilibrium with the external force spiral. Shape closure is a stronger constraint than force closure, but it increases the complexity of calculation and the difficulty of control accordingly [15]. Therefore, grasping stability is evaluated by force closure in most cases. Another scheme of mathematical modeling is that establishing an extremely suitable policy based on specific tasks [16]. For instance, a statistical model of interference distribution based on the grasping task was proposed, so the optimal grasping pose for the specified task can be obtained [17]. This solution greatly reduces the complexity of manipulation and improves the efficiency of policy planning. Yet these methods of mathematical modeling have to rely on the accurate geometric model of the target, so they are difficult to meet the actual need [18]. Moreover, the consumption of optimizing the objective function is very large and cannot ensure the real-time update of robot systems.

With the advancement of hardware and machine learning technology, data-driven methods that can reduce the complexity of the computation without listing all possibilities are widely used in robot manipulation [11]. The ability of perception and understanding are improved via feature recognition and classification, then the probability model of manipulation can be learned to perform the task [19]. Nowadays, there are two main solutions for data-driven methods. One scheme is that delivering body information of manipulation to the robot via some wearable sensing devices, and the other is extracting object features to plan the policy based on visual perception [20]—collecting the data via wearable devices, and analyzing the coordinated movement relationship among multiple joints of the human hand [21]. Then the features of the manipulation pose can be extracted, so as to establish the mapping between the human hand and the dexterous manipulator [22]. This scheme can explore the deep mechanism of human hand, simplify the space dimension of the robot manipulation, and provide a theoretical basis for human–machine collaboration [23].

Currently, learning to manipulate objects based on the scheme of visual perception has been a research focus of data-driven methods [24]. The method of extracting features from images provides a new direction for learning robot manipulation, but traditional methods of feature extraction mainly rely on the prior knowledge, so merely part of the information can be utilized effectively [22]. Owing to the great breakthrough of deep learning, the robot can extract more generalized features autonomously[25]. Due to the excellent capability of feature extraction, the deep learning network has achieved fantastic results in machine perception and image processing [26]. At the same time, deep learning can also be combined with the method of mathematical modeling to learn the robot manipulation, but the biggest shortcoming is still the lack of the entire system model [27]. Therefore, deep reinforcement learning (DRL) is proposed to realize the end-to-end learning from perception to robot manipulation.

However, it is difficult for agents to ensure the effectiveness of deep reinforcement learning in complex scenarios due to the limitations of sparse rewards. There, researchers put forward the idea of hierarchy according to the characteristics of human intelligence [28]. Hierarchical deep reinforcement learning can decompose the whole task, and then implement it step by step by lower levels of policy. According to the latest research in recent years, it is found that the effect of hierarchical deep reinforcement learning is far better than previous algorithms, which can not only adapt to complex problems, but also solve the problem of sparse rewards [29].

Reinforcement learning enables the robot to interact with the environment through trial and error, then the optimal strategy can be learned by maximizing the total return [30]. The method of deep reinforcement learning require a large number of samples and trials, so they are feasible for the field of image recognition but hardly suit for real robot manipulation. Nowadays, there are two ways forward to solve this problem [31]. One is imitation learning, in which machines can quickly learn to manipulate by observing a demonstration or a small amount of data. The method can reduce the complexity of robot strategy space and improve the learning efficiency [32]. The other one is transfer learning, in which the robot firstly learns to manipulate in the simulation environment, and then transfer the knowledge to the real. During the training of the real robot, valuable information is extracted from the simulated neural network, which greatly accelerates and strengthens the learning effect [33]. These three methods of robot learning will be described and analyzed in detail in this paper.

3. Deep Reinforcement Learning

Traditional manipulation learning methods need to know the model of the whole system in advance, but it is impossible in most cases in practice. Therefore, the method of reinforcement learning is inevitable, which enables the robot to make policy independently [34]. The traditional algorithm of reinforcement learning is the dynamic planning that deals with finite state space—then the optimal strategy can be obtained based on the accurate model, but it cannot solve the problem of robot manipulation. Therefore, deep reinforcement learning independent of the dynamic model that can adapt to the environment well is proposed to handle the task of continuous state space [35]. Deep reinforcement learning combines the perception ability of deep learning and the decision-making ability of reinforcement learning, which can learn the actions of the robot directly from images. Nowadays, deep reinforcement learning has become a key research direction in the field of robotics. Markov decision process (MDP) is the basis of reinforcement learning, the function of action-state value can be obtained from the expected sum of rewards [36]. The formula of value function is shown as Formula (1).

$$Q_\pi(s,a) = \mathrm{E}_\pi\left[\sum_{t=0}^{T} \gamma^t r_t \mid s_t = s, a_t = a\right] \tag{1}$$

In the formula, the expected sum of discounted rewards is defined as the function of action state value $Q_\pi(s,a)$. E_π represents the expected value in the case of motion strategy π, r_t represents the reward value for the corresponding moment, and γ^t represents the discount factor. On the basis of whether the state transition probability and return are known, reinforcement learning also can be sorted into model-based and mode-free methods as shown in Figure 2. Model-based methods can generate an environment model via sample data. Model-free reinforcement learning algorithms do not need to model the environment, but interact directly with the environment to learn relevant strategies. These two types of reinforcement learning algorithms can be divided into two categories based on the solution approach: the value-based learning method and the policy-based learning method [37]. At the same time, these two methods also can be combined to get a new method, actor-critic. This section will introduce representative algorithms of deep reinforcement learning in the field of robot manipulation.

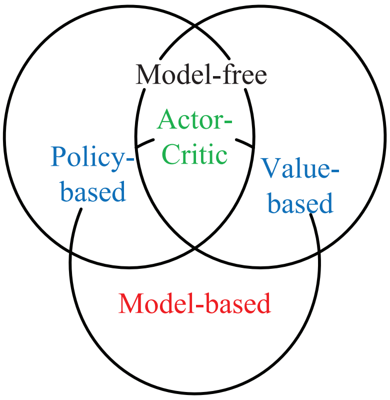

Figure 2. The classification of reinforcement learning.

3.1. Model-Based Methods

The model-based method of deep reinforcement learning can construct a dynamic probabilistic model via lots of data, and learn the best strategy from the value function of sate [38]. At the same time, methods can avoid interaction with the environment and train the strategy based on learned dynamic models. Therefore, the prior knowledge is an advantage of the model-based approach. Therefore, the development of predictive models based on prior knowledge of tasks and environment is the focus of subsequent research. The optimal solution can be obtained by the algorithm of value iteration and the algorithm of policy iteration when the model is known [39].

Some researches of robot manipulation via value-based deep reinforcement learning can be found. Todd et al. enabled the robot to play football via the state transition probability model of decision tree (DT) [40]. Rudolf et al. build a state transfer probability model based on the local linear system estimation (LLSE). The method to gain the value function is converted into a problem of solving linear programming that enables the two-link mechanical arm to play table tennis [41]. Connor et al. built the model of manipulation based on the convolutional neural network (CNN) and the mechanical arm can dig beans [42]. Methods of value function can adjust the strategy in time with the state value, which greatly reduces the time of iteration.

Learning the optimal strategy by policy improvement and policy evaluation is the core of policy iteration [43]. The sum of expected rewards is calculated in the stage of policy evaluation and the stage of policy improvement is used to optimize the strategy via the result of policy evaluation. These algorithms work by perturbing the policy parameters in many different ways, and then moving in the direction of good performance [44]. Jan et al. trained the manipulation skill of hitting the baseball via combining the policy gradient with the motor primitive [45]. Gen et al. learned the walking skill of a bipedal robot based on the policy gradient [46]. Marc et al. proposed a model-based algorithm of probabilistic inference for learning control (PILCO) for robot grasping, which incorporated the image information provided and the spatial constraints of manipulation into the learning process [47]. Currently, mainstream methods of policy iteration include Guided Policy Search (GPS) [48] and Cross-entropy method (CEM) [49].

The GPS proposed by Sergey Levine is a representative example of robot control achieved by combining traditional control algorithms with deep learning [50]. By the traditional control algorithm to create an end-to-end neural network, the tasks such as hanging clothes and opening bottle caps can be completed autonomously. Feature points are outputted through a convolutional neural network and in series with the basic parameters, and then the motor torques are output from two fully connected layers. Mechanical arm model information and precise information of the door can generate an optimal trajectory by traditional robot control algorithms such as linear quadratic regulator. The trained neural network can optimize the control trajectory based on these samples, and then explore the state and action space. However, the efficiency of the traditional method is very low, so the method of CEM is proposed to take samples and gain the probability of picking up the object [51].

The algorithms of policy iteration can be used to initialize the parameters with expert knowledge and accelerate the convergence process of strategy optimization. They are easy to implement and work very well for policies with a small number of parameters. Model-based methods of reinforcement learning can greatly improve the utilization of data and effectively reduce the cost of learning [52].

3.2. Model-Free Methods

The model-based methods can approximate the current value via the previous state value function, but not suitable for robot manipulation that accurate kinetic models are difficult to build. Therefore, model-free methods will be the focus of research in which agents interact with the environment via trial and error to gradually optimize the strategy [53]. At present, there are mainly two research directions, methods of value-based and policy-based. The representative algorithm based on value function is Q-learning in which the selection policy of action is greedy [54]. The algorithm updates the action-value function in accordance with the following formula:

$$Q(s,a) = Q(s,a) + \alpha\left[r + \gamma \max Q(s',a') - Q(s,a)\right] \tag{2}$$

Minoru et al. adopted the Q-leaning algorithm to realize the robot hitting the ball to the designed position based on visual enhancement [55]. Q is a tabular solution to evaluate the quality of each action. However, most scenes of robot control have so huge a state-space or action-space that the cost of using Q table is a big consumption. The method of function approximation is the solution to upper problem, which can be expressed by the function of linear or nonlinear [56]. Therefore, deep Q-network (DQN) that is combined of Q-learning and deep neural network is proposed to explore the high-dimensional space [57]. Zhang et al. train the grasping strategy of a three-joint robot based on DQN. Due to the difference between the simulation environment and real scene, the grasping effect of the controller is not good enough [58]. In order to perform dexterous manipulation of the robot, the improved algorithm of DQN was proposed [59].

Value-based methods cannot enumerate the quality of every action in continuous action-space, so it is impossible to calculate the optimal value. Therefore, another more direct way is needed to solve this problem, namely the policy gradient. Policy-based methods can directly parameterize the strategy and optimize the parameters based on the evaluation function [60]. The estimation of value function is still needed in the policy-based method, but the difference lies in whether the final strategy is directly measured by parameters or derived from the value function [61]. The policy-based algorithm could solve the problem of high cost in a real scenario and generate guided training samples by optimizing the trajectory distribution [62]. Schulman et al. proposed the algorithm of trust region policy optimization (TRPO), which updated policy parameters by optimizing the objective function [63]. Then the improved algorithm of proximal policy optimization (PPO) achieved a better result than TRPO when learning robot manipulation in the virtual simulation environment [33]. Mirowski et al. came up with that the agent learned to navigate in a complex environment based on the algorithm of asynchronous advantage actor-critic (A3C) [64]. In addition, Levine et al. learned the robot manipulation skills by optimizing the parameterized strategy based on various methods of policy gradient [65].

Policy gradient can select the appropriate strategy from continuous actions, but it can only be updated at the end of the round. Therefore, the algorithm of actor-critic was proposed which combined the advantages of value-based methods and policy-based methods. Lillicrap et al. proposed an algorithm of deep deterministic policy gradient (DDPG) based on the actor-critic framework, and realized robot manipulation in the simulation environment [66]. However, the algorithm of DDPG needed to train two networks, so the normalized advantage function (NAF) of one network was proposed that applied the algorithm of Q-learning into continuous action space [67]. Gu et al. proposed an algorithm of asynchronous NAF that had a trainer thread and multiple collector threads, in which the latest parameters of neural network were continuously shared with each robot [68]. The above achievements indicate that trained predictive models can be used by real robotic systems to manipulate unseen tasks in the past.

To conclude, motion planning of the robot is a tedious and complex task, so traditional algorithms of reinforcement learning cannot fulfill the task of high degree of freedom in continuous action space. If it is in discrete action space, the method of DQN can achieve high-performance. The method of DDPG can solve the tasks of continuous space and low action dimension. The algorithm of A3C is recommended when the action dimension is high and data are easy to obtain.

For more complex tasks, a stable and efficient algorithm of soft actor-critic (SAC) is proposed for real-world robot learning [69]. What is more, the algorithm of SAC can perform robotic tasks in a matter of hours and work in a variety of environments using the same set of hyperparameters. By comparison, the policy-based approach can more easily integrate the expert knowledge to accelerate the convergence process of the strategy. At the same time, policy-based methods has fewer parameters than value-based methods, so the learning efficiency is higher. The strategy obtained from the model-based algorithm of deep reinforcement learning depends on the accuracy of the model, while the model-free algorithm can improve the robustness of the learned strategy by a large number of interactions with the environment. Therefore, model-free methods can learn more generalized strategies. Various methods of deep reinforcement learning have their own advantages and disadvantages. It is necessary to make a trade-off among computational complexity, sample complexity, and strategy performance. Therefore, the effective combination of the advantages of various methods of deep reinforcement learning is the current research focus for improving the performance of robot manipulation. The characteristics of robot algorithms based on reinforcement learning are summarized in Table 1.

Table 1. Robot algorithms based on reinforcement learning.

Model-Based/Free	Ref.	Year	Authors	Algorithm	Value/Policy-Based
Model-based	[40]	2010	Hester et al.	DT	Value-based
	[41]	2014	Lioutikov et al.	LLSE	Value-based
	[42]	2017	Schenck et al.	CNN	Value-based
	[47]	2011	Deisenroth et al.	PILCO	Policy-based
	[50]	2016	Levine et al.	GPS	Policy-based
	[51]	2018	Levine et al.	CEM	Policy-based
Model-free	[58]	2015	Zhang et al.	DQN	Value-based
	[63]	2015	Schulman et al.	TRPO	Policy-based
	[33]	2018	Marcin et al.	PPO	Policy-based
	[64]	2016	Mirowski et al.	A3C	Policy-based
	[66]	2016	Lillicrap et al.	DDPG	Both
	[67]	2016	Gu et al.	NAF	Both
	[68]	2017	Gu et al.	Asynchronous NAF	Both
	[69]	2018	Haarnoja et al.	SAC	Both

It can be seen from the above research that deep reinforcement learning can successfully enable robots to master task skills through learning. The method will become the most promising way to realize a universal robot. However, methods based on deep reinforcement learning have the disadvantages of slow convergence and long computation time in the field of robot learning. It is a great challenge to perfectly match the rewards with a series of actions and achieve the rapid convergence of the entire network. In order to solve the problem of high consumption in training data and cost, the method of imitation learning has been further explored.

4. Imitation Learning

Imitation learning, in which the robot learns manipulation by observing the expert's demonstration, and skills can be generalized to other unseen scenarios. This process not only extracts information of the behavior and surrounding environment, but also learns the mapping between the observation and the performance. The task of robot manipulation can be viewed as a Markov decision process, then encoding action sequence of the expert into state-action pairs that are consistent with the expert. Imitation learning can train data from good samples instead of learning from scratch, so the learning efficiency is further improved [70]. By combining with reinforcement learning mechanisms, the speed and accuracy of imitation learning can be improved. Currently, the methods of imitation learning can be divided into behavior cloning (BC), inverse reinforcement learning (IRL), and generative adversarial imitation learning (GAIL) [71]. The classification of imitation learning can be seen in Figure 3.

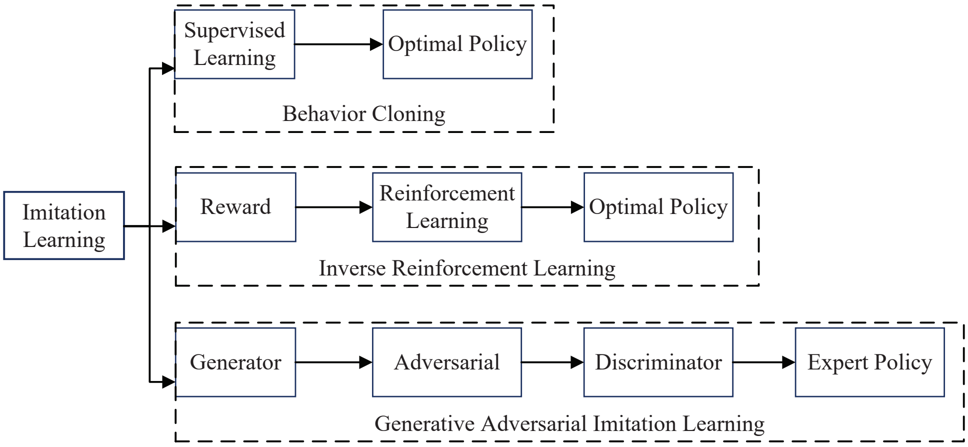

Figure 3. Classification of imitation learning.

4.1. Behavior Cloning

The essence of BC is direct policy learning, which enables the distribution of state-action trajectory generated by the agent to match the given teaching trajectory [72]. The traditional method of behavioral cloning is that the robotic arm learns the trajectory of the movement by manual guidance or teaching box. However, it can only simply repeat learned motions, not adapt to environmental changes. With the development of statistical learning, methods of machine learning have been introduced to identify basic units of robot manipulation. Takeda et al. trained a robot manipulation policy that can dance with humans based on hidden Markov model (HMM) [73]. However, such methods describe the trajectory through a series of discrete states and transitions between states, which does not allow for continuous smooth path and direct control of the robot motion. In order to solve the related problems, Calinon et al. enabled the robot to complete a series of operations from simple to complex based on the Gaussian mixture model (GMM) and Gaussian mixture regression (GMR) [74,75]. Multiple Gaussian distributions are used to model different stages of the trajectory and the covariance can be used to describe the uncertainty.

Gams et al. proposed dynamic motion primitives (DMPs) to generate a stable and generalizable strategy based on trajectories [76]. Methods of DMPs can generate trajectory of arbitrary complexity that can be used to describe robot manipulation. The disadvantage of DMPs is the need of a deterministic model, yet the fact that demonstrations cannot be completely alike. Therefore, it is difficult for this method to model the uncertainty of multiple demonstrations, resulting in a poor fit of the system as a whole. Zhang et al. proposed a virtual reality teleoperation system to collect high-quality demonstrations of robot manipulation, then the control strategy can be obtained via visuomotor learning (VL) [77]. The result shows that imitation learning can be surprisingly effective in learning deep policy that map directly from pixel values to actions, only with a small amount of learning data.

However, the problem of BC is that the number of samples is not enough, so the agent cannot learn situations that are not included in samples. Therefore, in the case of a small amount of samples, the strategy obtained by behavioral cloning is not generalizable. In order to solve the learning problem of insufficient samples, the method of inverse reinforcement learning is proposed [78].

4.2. Inverse Reinforcement Learning

Inverse reinforcement learning is a method of evaluating how well an action is performed via reward function, which is an abstract description of behavior. Compared to methods of behavioral cloning, IRL is an efficient paradigm of imitation learning that is more adaptable in responding to different environments. When the execution environment or robot model changes significantly, the resulting mapping function will be difficult to apply and will need to learn again [79]. Whereas the method of IRL is more task-related, the appropriate strategy can be obtained based on the previous reward function after receiving new information from the environment and the model [80]. Inverse reinforcement learning can be classified according to the algorithms it depends on.

Abbeel et al. proposed the max-margin principle (MP) of obtaining a reward function based on teaching data, in which the difference between the optimal strategy and other suboptimal strategies can be maximized [81]. Ratliff et al. suggested the framework of maximum marginal planning (MMP) based on the principle of maximum margin, and transformed the learning of reward function into a structural prediction [82]. The method of maximum marginal programming is very expensive to solve the MDP, so Klein et al. proposed a method of structured classification (SC) to learn the reward function without solving MDP [83]. Ho et al. proposed a neural network on the basis of apprenticeship learning (AL), and updated it via the method of policy gradient. It is hard to determine the quality of actions in actual scenarios [84]. The above methods are all artificial design features of the reward function that are difficult to generalize to the high dimensional and continuous robot state space. Therefore, Xia et al. proposed the neural inverse reinforcement learning (NIRL), which is still based on the framework of maximum margin [85].

The disadvantage of maximum margin methods is that different reward functions will lead to the same expert strategy in many cases, thus resulting in ambiguity. Therefore, many algorithms of inverse reinforcement learning are proposed based on the probabilistic model to overcome this problem [86]. Ziebart et al. constructed a probabilistic model for sequential policy by maximum entropy inverse reinforcement learning, which can ensure the manipulation strategy has better performance when the teaching data are not optimal and the reward function is random deviation [87]. Finn et al. updated the policy based on the maximum entropy IRL and constructed reward function to help training via expert data [88]. The method of inverse reinforcement learning based on maximum entropy needs to know the state transition probability of the system. Therefore, Boularias et al. established the maximum relative entropy model to solve the model-free problem [89]. Peng et al. presented a data-driven deep reinforcement learning framework to train humanoid robots in virtual environment via the algorithm of DeepMimic and then learned a series of difficult manipulation skills [90]. The resulting strategy is highly robust and the generated natural motion is almost indistinguishable from the original motion capture data in the absence of perturbations.

Methods of behavioral cloning and IRL learn strategies from demonstrations, but can not interact with the expert to further optimize the policy [91]. Therefore, the method of generative adversarial imitation learning is proposed to solve the problem based on adversarial networks [92].

4.3. Generative Adversarial Imitation Learning

The method of GAIL is implemented by comparing the difference between the generated strategy and the expert strategy. Iterative confrontation training can be performed to make the distribution between the expert and the agent as close as possible [93]. Generative adversarial networks (GANs) have been successfully applied to policy imitation problems under model-free settings. Baram et al. proposed the algorithm of model-based generative adversarial imitation learning (MGAI) based on a forward model to make the calculations completely divisible, which allows the use of accurate discriminator gradients to train strategies [94]. The use of pure learning methods with simple reward functions often results in non-human and too rigid movement behaviors. Merel et al. extended the algorithm of GAIL that the training of general neural network strategies can generate human-like motion patterns from limited demonstrations without access to actions. This method constructs strategies and shows that they can be reused to solve tasks when controlled by a higher-level controller [95]. They are vulnerable to cascading failures when the agent trajectory diverges from the demonstrations. Wang et al. added a variation auto-encoder (VAE) to learn semantic policy embeddings that made the algorithm of GAIL more robust than the supervised controller especially with few demonstrations. Leveraging these policies, a new version of GAIL can be developed to avoid mode collapse and capture many different behaviors [96].

Unfortunately, methods of imitation learning tend to require that demonstrations are supplied in the first-person that is limited by the relatively hard problem of collecting first-person demonstrations. Stadie et al. presented a method of unsupervised third-person imitation learning (TPIL) to train agent to correctly achieve goal in a simple environment when the demonstration is provided from a different viewpoint [97]. Standard imitation learning methods assume received examples that could be provided in advance, which stands in contrast to how humans and animals imitate. Liu et al. proposed an learning method of imitation from observation (IFO) based on video prediction with context translation, which ensured output of different domains consistent [98]. The assumption in imitation learning is lifted that shows the effectiveness of our approach in learning a wide range of real-world robot manipulation.

The way that robots learn desired strategies based on deep reinforcement learning in real scenario will face the problem of large data requirement, high cost of trial and error and long training process. To enable learning of robot manipulation, roboticists focused their efforts on imitation learning that coincided with the learning process of human. Methods of imitation learning combined expert demonstrations with appropriate algorithms of machine learning, which can provide a simple and intuitive framework of robot learning and reduce the cost of deployment. Therefore, imitation learning is an effective method for the system to obtain control strategies when an explicit reward function is insufficient, using supervision provided as demonstrations of the expert.

Although the method of BC is intuitive and simple to implement, a large amount of data is required and the learned policy cannot adapt to the new environment. Although the method of IRL makes up for shortcomings of above situations, the consumption of training time is still costly. The method of GAIL introduces the idea of generative adversarial networks for imitation learning, which has better performance than the other two methods in high-dimensional situations. A major drawback of GAIL is the problem of model collapse because the diversity of generated images is often smaller than the real data. To summarize, imitation learning has been a key method in the field of robot manipulation.

Current algorithms solve the problem of designing the reward function to a certain extent and accelerate the rate of learning by initializing strategies based on teaching data. Robot algorithms based on imitation learning are summarized in Table 2. However, there are still some problems in imitation learning, such as the high consumption of collecting data and the local optimal solution of policy, which may lead to the poor effect of learning. Therefore, some scholars have put forward the method of transfer learning, in which the model of learning is trained in a simulation environment and then the knowledge is transferred to the real robot, so as to acquire the skills of robot manipulation more efficiently.

Table 2. Robot algorithms based on imitation learning.

Categories	Ref.	Year	Authors	Algorithms
Behavior Cloning	[73]	2007	Takeda et al.	HMM
	[74]	2007	Calinon et al.	GMM
	[75]	2010	Calinon et al.	GMR
	[76]	2014	Gams et al.	DMPs
	[77]	2018	Zhang et al.	VL
Inverse Reinforcement Learning	[81]	2004	Abbeel et al.	MP
	[82]	2006	Ratliff et al.	MMP
	[83]	2012	Klein et al.	SC
	[84]	2016	Ho et al.	AL
	[85]	2016	Xia et al.	NIRL
	[87]	2008	Ziebart et al.	Maximum Entropy IRL
	[89]	2011	Boularias et al.	Relative Entropy IRL
	[90]	2018	Peng et al.	DeepMimic
Generative Adversarial Imitation Learning	[94]	2017	Baram et al.	MGAI
	[95]	2017	Merel et al.	Extended GAIL
	[96]	2017	Wang et al.	VAE
	[97]	2017	Stadie et al.	TPIL
	[98]	2017	Liu et al.	IFO

5. Transfer Learning

Robot manipulation is so complex that the consumption of obtaining an optimal solution is costly. Obtained policy based on deep reinforcement learning can only be applied in one task and have to start from scratch whenever the environment changes slightly. By introducing transfer learning into robot deep reinforcement learning, the data in the simulated environment can be used to help the robot better learn control strategies (Figure 4). The method of transfer learning with learning ability can divert knowledge from the source task to the target task by sharing learned parameters with the new model [99]. This optimization method of transfer learning can greatly improve the generalization of the original model and the speed of modeling new tasks. Data sets of the task which include position, velocity, and force are collected and then used to learn the skill model. Then the knowledge of the learned model can be transferred into real robot so as to obtain the new model that can reproduce the robot manipulation in new environments [100].

However, it is not easy for robots to transfer and learn, because there is a reality gap between the simulation and reality. The policy will not adapt to changes in the external environment if trained in a flawed simulation. In addition, the physics of sliding friction and contact forces also cannot be perfectly simulated [101]. Several improved methods of transfer learning are proposed that will be elaborated briefly in this section.

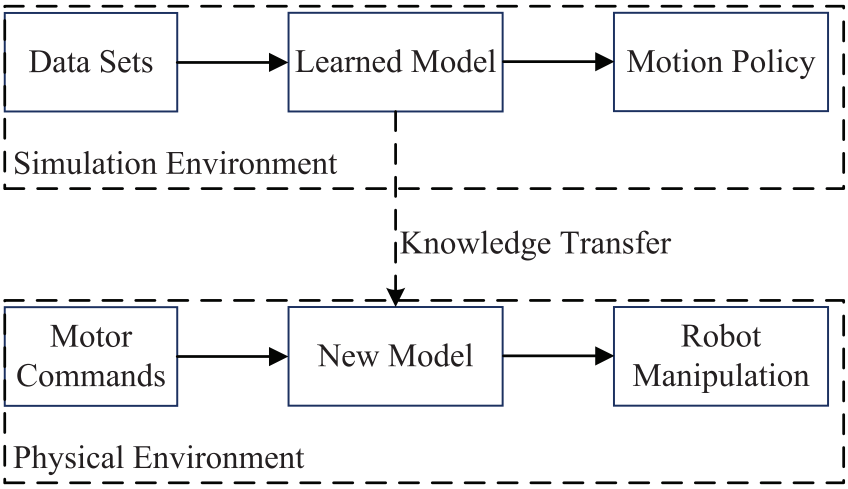

Figure 4. Principle of transfer learning for robot manipulation.

5.1. Better Simulation

For many robot manipulations, data sets of real world are costly to obtain, but easy to collect in the simulation environment. Tzeng et al. proposed a novel method of domain adaptation for robot perception without expensive manual data annotation before policy search [102]. The improved method of transfer learning compensates for domain shift more effectively than previous techniques by using weakly paired images. Zhu built a highly similar simulation framework named AI2-THOR in which the optimal strategy was trained in high-quality 3D scenes [103]. Agents can manipulate and interact with objects in the framework, so a huge number of samples are collected. At the same time, the method is end-to-end trainable and converges faster than other methods. In the robot simulation environment, only limited parameters can be used to simulate the physical environment, so there are errors compared with the real situation. Peng et al. proposed a recurrent neural network to reduce the gap between virtual and real, which improved the effect of robot transfer learning by narrowing the training error [104]. The neural-augmented simulation can improve the effect of robot transfer learning by narrowing the training error.

According to the above analysis, these methods can construct a better simulation environment, but they are all measured in definite states and actions to train the agent. Another idea is that highly adaptable strategy can be trained through randomized processing of states and actions. Then the system of the robot will respond to dynamic changes effectively in the real world without adjustments.

5.2. Policy Randomization

Although the simulation environments provide an abundant source of data and reduce the potential safety concerns during the training process, policies that are successful in simulation may not transfer to real world because of modeling errors. Algorithms of policy gradient are very effective in solving the high-dimensional sequential task of robot manipulation. Ammar et al. proposed a method of multi-task policy gradient to learn policy, which can transfer knowledge between tasks to improve learning efficiency [105]. The realization of end-to-end pixel-driven control for complex robot manipulation is an unresolved problem. Rusu et al. proposed progressive networks, which are a general framework that can reuse everything from the low-level visual functions to the high-level strategies. The speed of each robot joint can be obtained by input image only, which further verifies the feasibility of progressive neural networks [106]. Peng et al. proposed that dynamic and highly adaptive strategies could be obtained by randomizing the dynamics of the simulator during training, which can adapt to the significantly different situation [90].

Both above approaches have done extensive processing of the virtual environment to improve the performance of the simulator. However, none of studies can guarantee the adaptive capability required for real-world robots. The approach presented below is a higher-level complementary approach of enhanced transfer learning that produces policies which generalize across tasks.

5.3. Robust Policy

Methods of transfer learning have difficulty in obtaining policies which can generalize across tasks, despite collecting a large amount of data. He et al. proposed an attempt to learn a robust policy directly on a real robot based on model-predictive control (MPC), adapting to unseen tasks [107]. A continuous parameterization and policy can be learned simultaneously in simulation instead of end-to-end learning policy for single task. Then the multi-skill policy can be transferred directly to a real robot that is actuated by choosing sequences of skill latents. The model of MPC is composed of the pre-trained policy executed in the simulation, run in parallel with the real robot. Agents trained in the simulator may not be invalid in the real world when performing actions due to the gap between training and execution environments. Ramakrishnan proposed the oracle feedback to learn a predictive model of blind spots in order to reduce costly errors [108]. By evaluating the application of the method in two domains, it was demonstrated that the predictive performance has been improved and the learned model can be used to query oracle selectively to prevent errors. Although a general simulator is needed for flexible learning approaches, control policy in simulation directly applied to robot will yield model errors. In order to overcome cases of severe mismatch, Raileanu et al. proposed a novel way to regularize a decoder of a variational autoencoder to a black-box simulation, with the latent space bound to a subset of simulator parameters [109]. Encoder training from real-world trajectories can yield a latent space with simulation parameter distribution that matches the real setting.

The above methods are mainly to improve the adaptability of state and action in the virtual environment, and try to introduce parameters in the physical environment into the strategy training of the simulated environment [110]. In addition to the three methods mentioned above, there are other ways to improve transfer learning for robot manipulation. Jeong directly introduced the state-related generalized forces to capture the difference between the simulated environment and the real world, thus realizing the transfer learning of robot manipulation [111]. Hwangbo et al. built a perfect actuator model by adding stochastic dynamic parameters, which strengthened the generalization of the neural network [112]. Matas et al. studied the manipulation of non-rigid objects in simulation [113]. Sadeghi et al. studied transfer learning based on multiple domains and proposed a simulation benchmark for robot grasping, which played an important role in promoting the research on robot [114]. Mees et al. proposed an adversarial skill network to find the embedded space suitable for different task domains. This method is not only applicable to the transfer learning of robots, but also to other tasks of finding and learning transferable skills [115].

In summary, methods of transfer learning help us to find out the commonality of problems and deal with the newly encountered problems. The advantage of robot transfer learning lies in learning control strategies based on sufficient data in the simulated environment, while the difficulty of research lies in transferring control strategies to real robots. In the field of robotics, data from simulation can be used to solve problems in which there are few or no sample in the target domain. Dominant approaches for ameliorating transfer learning include building better simulation environment, policy randomization, and direct training of robust policy. Improved methods for robot transfer learning are summarized in Table 3.

Table 3. Improved methods for robot transfer learning.

Improved Methods	Ref.	Year	Authors	Approaches
Better Simulation	[102]	2015	Tzeng et al.	Neural-augmented simulation
	[103]	2017	Zhu et al.	Weak pairwise constraints
	[104]	2018	Peng et al.	Framework of AI2-THOR
Policy Randomization	[105]	2014	Ammar et al.	Randomizing the dynamics of the simulator
	[106]	2016	Rusu et al.	Multi-task policy gradient
	[90]	2018	Peng et al.	Progressive neural networks
Robust Policy	[107]	2018	He et al.	Model-predictive control
	[108]	2020	Ramakrishnan et al.	The oracle feedback
	[109]	2020	Hwasser et al.	Variational auto-regularized alignment

6. Discussion

The above learning methods can enable robots to make decisions autonomously and adapt various complex environments dynamically. The approach of reinforcement learning generate data from trial-and-error experiments that may damage the robot. Therefore, imitation learning is proposed that robot learns from images, videos, or an expert. Nevertheless, an expert cannot be found anytime, especially when robot manipulation skills are difficult to learn or require extreme precision. In view of this, transfer learning is the appropriate algorithm that train the data in simulation, then the policy refined can be reused on a physical platform [116]. Most notably, robot manipulation leverages the immense progress in learning methods to achieve wonderful developments in many applications. Robot learning application domains can be found in Table 4.

Table 4. Robot learning application domains.

Applications	Classical Demos	References
Industrial Robot	Peg-in-hole	[117]
	grinding and polishing	[118]
	welding	[119]
	human-machine collaboration	[120]
Personal Robot	Ironing clothes	[121,122]
	pouring water	[123–125]
	autonomous navigation	[53,126]
	obstacle avoidance	[76,127,128]
Medical Robot	Rehabilitation training	[129,130]
	surgical operation	[131–133]

As shown in the above table, the manipulation environment can be classified into three situations, such as industrial robot, personal robot, and medical robot. Previously, robots worked in a structured environment, mainly for delivering, painting, welding, etc. At the same time, they could only perform simple and repetitive tasks with little variation. Currently, robots are gradually able to perform dexterous tasks that ranges from the simple interaction of parts to the complex interaction between humans and the environment. Methods of robot learning address the lack of accurate object models and dynamic changes in complex environments. The learning process is also simplified by visually extracting information from expert presentations [134].

In the training process of deep reinforcement learning, there are two very disturbing problems, namely the design of neural network structure and the setting of hyperparameters. The neural network needs to solve such problems as gradient vanishing, gradient explosion, and overfitting. The appropriate loss functions and activation functions are needed to solve the above two problems. Common loss functions include mean square error, cross entropy error, mean absolute value error, etc. Proper activation functions can make the deep neural network fit the nonlinear model better which mainly include Sigmoid, Tanh, ReLU etc. Moreover, data regularization and dropout are the main methods to solve overfitting. By inserting these processes before the activation function, the deviation of data distribution can be reduced and the accuracy of network can be effectively improved. Neural network architectures need to be experimented and inferred from experimental results. It is recommended to use proven architectures such as VGG, ResNet, Inception, etc. The hyperparameters are the values that initialize the neural network, and these values cannot be learned during training. These super parameters include the number of neural network layers, the size of the batch, the number of trained epochs, etc. Each neural network will have an optimal combination of hyperparameters, which will achieve the maximum accuracy. There is no direct way to get it but usually through trial and error.

It is a challenge to ensure that the learned model is valid, given the interference of the real environment. Much of the collected data are meaningless, so constructing an accurate simulator is hard. Generally, humans solve the new problem via some basic skills. Inspired from this, the method of meta-learning is proposed to generate correct motion sequences that adapts to scene changes based on existing models. Meta-learning is the foundation of both transfer learning and imitation learning that utilizes the previous knowledge and experience to form a core value network [135].

The existing meta-learning neural network structure can be used to accelerate learning when facing new tasks. Model-agnostic meta-learning (MAML) is a meta-learning algorithm for supervised learning and reinforcement learning [136]. The method of MAML makes a back-propagation update of the neural network with the sample, and then completes supervised learning based on the updated parameters. The neural network is forced to learn some task information by adding external data [137,138]. Santoro et al. added external memory to the neural network, which obtained the relevant images for comparison [139]. Marcin et al. trained a general neural network to predict the gradient by the regression problem of equation. As long as the gradient is predicted correctly, this method significantly speeds up the training [140]. Oriol et al. constructed an attention mechanism via imitating humans, which directly focus on the most important parts [141]. Sachin et al. trained an update mechanism of neural networks via the long short-term memory (LSTM) structure, and obtained new parameters by inputting current network parameters [142]. Flood et al. constructed a model to learn and predict the function of loss via previous tasks, which sped up the learning rate [143].

Hence, meta-learning is not a simple mapping, but a way to connect different information. Meta-learning enables the neural network to learn a kind of meta-knowledge based on samples, so that the change factors have been completely separated from the invariant factors in the learned representation space, and the decisive factors can be learned. Although the method has made some progress in the field of robot learning, a large amount of training data is still required in the training phase of meta-learning. Meta-learning is the basis of imitation learning and transfer learning, and one shot learning is an extreme form of the two methods. Therefore, designing a one-shot learning neural network structure with high learning efficiency and excellent performance is an important research direction in the future.

7. Conclusions

In view of this method, the robot can understand the intention of samples and map directly to joint control without a lot of training data [144]. With fast learning ability, such a robot system has strong universality. Finn et al. used the visual information to obtain the control information of joints based on the MAML algorithm of Meta-Learning [145]. Tianhe et al. proposed a method of one-shot learning to build prior knowledge by using human and robot demonstration data based on meta-learning. Next, combining this prior knowledge with a person's video presentation, the robot can perform the tasks demonstrated by the person [146]. Feifei et al. proposed a novel framework of robot learning called neural task programming (NTP), which used neural program induction to do few-shot imitation learning. NTP decomposed the robot's manipulation into multi-step motions, and the neural network learned how to compose these motions and then execute them. To some extent, it greatly simplifies the difficulty of the problem [147].

To summarize, compared with traditional methods, the methods of robot learning based on deep learning can enable the robot to have the ability of decision-making and learning, which dynamically adapt to many complex situations and greatly improve production efficiency. An end-to-end, completely learned robot with strong imitation learning ability will be the basis for robots to be used in various fields widely. In the future, the complexity of tasks will need to be further increased, such as the one-shot imitation learning in the third person. Improving the efficiency and the generalization in robot learning is also seeking further research attention.

Author Contributions: Conceptualization, J.H. and Z.J.; writing—original draft, J.H.; writing—review and editing, L.Z. and G.L. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the support from the AiBle project co-financed by the European Regional Development Fund and National Natural Science Foundation of China (grant No. 52075530 and No. 51975425).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Billard, A.; Kragic, D. Trends and challenges in robot manipulation. *Science* **2019**, *364*. . [CrossRef] [PubMed]
2. Thibaut, J.; Toussaint, L. Developing Motor Planning over Ages. *J. Exp. Child Psychol.* **2010**, *105*, 116–129. [CrossRef] [PubMed]
3. Xue, Y.; Ju, Z.; Xiang, K.; Chen, J.; Liu, H. Multimodal Human Hand Motion Sensing and Analysis—A Review. *IEEE Trans. Cogn. Dev. Syst.* **2019**, *11*, 162–175.
4. Feix, T.; Romero, J.; Schmiedmayer, H.; Dollar, A.M.; Kragic, D. The GRASP Taxonomy of Human Grasp Types. *IEEE Trans. Hum. Mach. Syst.* **2016**, *46*, 66–77. [CrossRef]
5. Homberg, B.S.; Katzschmann, R.K.; Dogar, M.R.; Rus, D. Haptic identification of objects using a modular soft robotic gripper. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–3 October 2015; pp. 1698–1705.
6. Edwards, C.; Edwards, A.; Stoll, B.; Lin, X.; Massey, N. Evaluations of an artificial intelligence instructor's voice: Social Identity Theory in human-robot interactions. *Comput. Hum. Behav.* **2019**, *90*, 357–362. [CrossRef]
7. Sahbani, A.; Elkhoury, S.; Bidaud, P. An overview of 3D object grasp synthesis algorithms. *Robot. Auton. Syst.* **2012**, *60*, 326–336. [CrossRef]
8. Pinto, L.; Gupta, A. Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 3406–3413.
9. Vecerik, M.; Hester, T.; Scholz, J.; Wang, F.; Pietquin, O.; Piot, B.; Heess, N.; Rothorl, T.; Lampe, T.; Riedmiller, M. Leveraging Demonstrations for Deep Reinforcement Learning on Robotics Problems with Sparse Rewards. *arXiv* **2017**, arXiv:1707.08817.
10. Kroemer, O.; Niekum, S.; Konidaris, G. A Review of Robot Learning for Manipulation: Challenges, Representations, and Algorithms. *arXiv* **2019**, arXiv:1907.03146.
11. Bohg, J.; Morales, A.; Asfour, T.; Kragic, D. Data-Driven Grasp Synthesis—A Survey. *IEEE Trans. Robot. Autom.* **2014**, *30*, 289–309. [CrossRef]
12. Li, Y.H.; Lei, Q.J.; Cheng, C.; Zhang, G.; Wang, W.; Xu, Z. A review: Machine learning on robotic grasping. In Proceedings of the Eleventh International Conference on Machine Vision (ICMV 2018), International Society for Optics and Photonics, Munich, Germany, 1–3 November 2019; Volume 11041.
13. Bicchi, A.; Kumar, V. Robotic grasping and contact: A review. In Proceedings of the 2000 ICRA, Millennium Conference, IEEE International Conference on Robotics and Automation, Symposia Proceedings (Cat. No. 00CH37065), San Francisco, CA, USA, 24–28 April 2000; Volume 1, pp. 348–353.
14. Colomé, A.; Pardo, D.; Alenya, G.; Torras, C. External force estimation during compliant robot manipulation. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation,Karlsruhe, Germany, 6–10 May 2013; pp. 3535–3540.
15. Bicchi, A. On the closure properties of robotic grasping. *Int. J. Robot. Res.* **1995**, *14*, 319–334. [CrossRef]

16. Li, J.; Liu, H.; Cai, H. On computing three-finger force-closure grasps of 2-D and 3-D objects. *IEEE Trans. Robot. Autom.* **2003**, *19*, 155–161.
17. Lin, Y.; Sun, Y. Grasp planning to maximize task coverage. *Int. J. Robot. Res.* **2015**, *34*, 1195–1210. [CrossRef]
18. Kemp, C.C.; Edsinger, A.; Torres-Jara, E. Challenges for robot manipulation in human environments [grand challenges of robotics]. *IEEE Robot. Autom. Mag.* **2007**, *14*, 20–29. [CrossRef]
19. Fang, B.; Jia, S.; Guo, D.; Xu, M.; Wen, S.; Sun, F. Survey of imitation learning for robotic manipulation. *Int. J. Intell. Robot. Appl.* **2019**, *3*, 362–369. [CrossRef]
20. Alexandrova, S.; Cakmak, M.; Hsiao, K.; Takayama, L. Robot Programming by Demonstration with Interactive Action Visualizations. *Robot. Sci. Syst.* **2014**, *10*, doi:10.15607/RSS.2014.X.048. [CrossRef]
21. Huang, D.; Ma, M.; Ma, W.; Kitani, K.M. How do we use our hands? Discovering a diverse set of common grasps. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 666–675.
22. Pérez-D'Arpino, C.; Shah, J.A. Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 6175–6182.
23. Yang, Y.; Fermuller, C.; Li, Y.; Aloimonos, Y. Grasp type revisited: A modern perspective on a classical feature for vision. *Comput. Vis. Pattern Recognit.* **2015**, 400–408.
24. Lenz, I.; Lee, H.; Saxena, A. Deep Learning for Detecting Robotic Grasps. *Int. J. Robot. Res.* **2013**, *34*, 705–724 [CrossRef]
25. Lecun, Y.; Bengio, Y.; Hinton, G.E. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
26. Redmon, J.; Angelova, A. Real-Time Grasp Detection Using Convolutional Neural Networks. *Int. Conf. Robot. Autom.* **2015**, 1316–1322.
27. Varley, J.; Weisz, J.; Weiss, J.; Allen, P.K. Generating multi-fingered robotic grasps via deep learning. *Intell. Robot. Syst.* **2015**, 4415–4420. [CrossRef]
28. BCS, M. Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. *arXiv* **2016**, arXiv:1604.06057.
29. Hou, Z.; Fei, J.; Deng, Y.; Xu, J. Data-efficient Hierarchical Reinforcement Learning for Robotic Assembly Control Applications. *IEEE Trans. Ind. Electron.* **2020**. [CrossRef]
30. Kiran, B.R.; Sobh, I.; Talpaert, V.; Mannion, P.; Sallab, A.A.A.; Yogamani, S.; Perez, P. Deep Reinforcement Learning for Autonomous Driving: A Survey. *arXiv* **2020**, arXiv:2002.00444.
31. Zhu, Y.; Wang, Z.; Merel, J.; Rusu, A.A.; Erez, T.; Cabi, S.; Tunyasuvunakool, S.; Kramar, J.; Hadsell, R.; De Freitas, N.; et al. Reinforcement and Imitation Learning for Diverse Visuomotor Skills. *arXiv* **2018**, arXiv:1802.09564.
32. Zhu, Z.; Hu, H. Robot Learning from Demonstration in Robotic Assembly: A Survey. *Robotics* **2018**, *7*, 17.
33. Andrychowicz, O.M.; Baker, B.; Chociej, M.; Jozefowicz, R.; Mcgrew, B.; Pachocki, J.; Petron, A.; Plappert, M.; Powell, G.; Ray, A.; et al. Learning Dexterous in-Hand Manipulation. *Int. J. Robot. Res.* **2020**, *39*, 3–20. [CrossRef]
34. Levine, S.; Popovic, Z.; Koltun, V. Nonlinear Inverse Reinforcement Learning with Gaussian Processes. *Adv. Neural Inf. Process. Syst.* **2011**, *24*, 19–27.
35. Kober, J.; Bagnell, J.A.; Peters, J. Reinforcement learning in robotics: A survey. *Int. J. Robot. Res.* **2013**, *32*, 1238–1274. [CrossRef]
36. Luong, N.C.; Hoang, D.T.; Gong, S.; Niyato, D.; Wang, P.; Liang, Y.C.; Kim, D.I. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Commun. Surv. Tutorials* **2019**, *21*, 3133–3174. [CrossRef]
37. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]
38. Nguyen, H.; La, H. Review of deep reinforcement learning for robot manipulation. In Proceedings of the 2019 Third IEEE International Conference on Robotic Computing (IRC), Naples, Italy, 25–27 February 2019; pp. 590–595.
39. Fan, L.; Zhu, Y.; Zhu, J.; Liu, Z.; Zeng, O.; Gupta, A.; Creus-Costa, J.; Savarese, S.; Fei-Fei, L. Surreal: Open-source reinforcement learning framework and robot manipulation benchmark. In Proceedings of the Conference on Robot Learning, Zürich, Switzerland, 29–31 October 2018; pp. 767–782.
40. Hester, T.; Quinlan, M.; Stone, P. Generalized model learning for Reinforcement Learning on a humanoid robot. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 2369–2374.
41. Lioutikov, R.; Paraschos, A.; Peters, J.; Neumann, G. Sample-Based Information-Theoretic Stochastic Optimal Control. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–5 June 2014; pp. 3896–3902.
42. Schenck, C.; Tompson, J.; Fox, D.; Levine, S. Learning Robotic Manipulation of Granular Media. In Proceedings of the Conference on Robot Learning, PMLR, Mountain View, CA, USA, 13–15 November 2017.
43. Ross, S.; Gordon, G.J.; Bagnell, J.A. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011; Volume15, pp. 627–635.
44. Sutton, R.S.; Mcallester, D.; Singh, S.; Mansour, Y. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In Proceedings of the NIPs, Denver, CO, USA, 29 November–4 December 1999; Volume 12, pp. 1057–1063.
45. Peters, J.; Schaal, S. 2008 Special Issue: Reinforcement learning of motor skills with policy gradients. *Neural Netw.* **2008**, *21*, 682–697. [CrossRef] [PubMed]

46. Endo, G.; Morimoto, J.; Matsubara, T.; Nakanishi, J.; Cheng, G. Learning CPG-based Biped Locomotion with a Policy Gradient Method: Application to a Humanoid Robot. *Int. J. Robot. Res.* **2008**, *27*, 213–228. [CrossRef]
47. Deisenroth, M.P.; Rasmussen, C.E. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), Bellevue, WA, USA, 28 June–2 July 2011; pp. 465–472.
48. Yahya, A.; Li, A.; Kalakrishnan, M.; Chebotar, Y.; Levine, S. Collective robot reinforcement learning with distributed asynchronous guided policy search. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 79–86.
49. Gass, S.I.; Harris, C.M. Encyclopedia of Operations Research and Management Science. *J. Am. Stat. Assoc.* **1997**, *92*, 800.
50. Levine, S.; Finn, C.; Darrell, T.; Abbeel, P. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.* **2016**, *17*, 1334–1373.
51. Levine, S.; Pastor, P.; Krizhevsky, A.; Quillen, D. Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection. *Int. J. Robot. Res.* **2018**, *37*, 421–436. [CrossRef]
52. Shi, H.; Shi, L.; Xu, M.; Hwang, K.S. End-to-end navigation strategy with deep reinforcement learning for mobile robots. *IEEE Trans. Ind. Inform.* **2019**, *16*, 2393–2402. [CrossRef]
53. Kroemer, O.; Detry, R.; Piater, J.; Peters, J. Combining active learning and reactive control for robot grasping. *Robot. Auton. Syst.* **2010**, *58*, 1105–1116. [CrossRef]
54. Tan, X.; Chng, C.B.; Su, Y.; Lim, K.B.; Chui, C.K. Robot-assisted training in laparoscopy using deep reinforcement learning. *IEEE Robot. Autom. Lett.* **2019**, *4*, 485–492. [CrossRef]
55. Asada, M.; Noda, S.; Tawaratsumida, S.; Hosoda, K. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Mach. Learn.* **1996**, *23*, 279–303. [CrossRef]
56. Wulfmeier, M.; Ondruska, P.; Posner, I. Maximum Entropy Deep Inverse Reinforcement Learning. *arXiv* **2015**, arXiv:1507.04888.
57. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602.
58. Zhang, F.; Leitner, J.; Milford, M.; Upcroft, B.; Corke, P. Towards vision-based deep reinforcement learning for robotic motion control. *arXiv* **2015**, arXiv:1511.03791.
59. Hausknecht, M.; Stone, P. Deep Recurrent Q-Learning for Partially Observable MDPs. *arXiv* **2015**, arXiv:1507.06527.
60. Cao, J.; Liu, W.; Liu, Y.; Yang, J. Generalize Robot Learning From Demonstration to Variant Scenarios with Evolutionary Policy Gradient. *Front. Neurorobot.* **2020**, *14*. [CrossRef]
61. Yang, C.; Chen, C.; Wang, N.; Ju, Z.; Fu, J.; Wang, M. Biologically Inspired Motion Modeling and Neural Control for Robot Learning From Demonstrations. *IEEE Trans. Cogn. Dev. Syst.* **2019**, *11*, 281–291.
62. Levine, S.; Koltun, V. Learning Complex Neural Network Policies with Trajectory Optimization. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 829–837.
63. Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.I.; Moritz, P. Trust Region Policy Optimization. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1889–1897.
64. Mirowski, P.; Pascanu, R.; Viola, F.; Soyer, H.; Ballard, A.J.; Banino, A.; Denil, M.; Goroshin, R.; Sifre, L.; Kavukcuoglu, K.; et al. Learning to Navigate in Complex Environments. *arXiv* **2016**, arXiv:1611.03673.
65. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Harley, T.; Lillicrap, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1928–1937.
66. Lillicrap, T.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2016**, arXiv:1509.02971.
67. Gu, S.; Lillicrap, T.; Sutskever, I.; Levine, S. Continuous Deep Q-Learning with Model-based Acceleration. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016.
68. Gu, S.; Holly, E.; Lillicrap, T.; Levine, S. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In Proceedings of the 2017 IEEE International conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3389–3396.
69. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018.
70. Bojarski, M.; Testa, D.D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L.D.; Monfort, M.; Muller, U.A.; Zhang, J.; et al. End to End Learning for Self-Driving Cars. *arXiv* **2016**, arXiv:1604.07316.
71. Kumar, V.; Gupta, A.; Todorov, E.; Levine, S. Learning Dexterous Manipulation Policies from Experience and Imitation. *arXiv* **2016**, arXiv:1611.05095.
72. Wu, Y.; Charoenphakdee, N.; Bao, H.; Tangkaratt, V.; Sugiyama, M. Imitation Learning from Imperfect Demonstration. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 6818–6827.
73. Takeda, T.; Hirata, Y.; Kosuge, K. Dance Step Estimation Method Based on HMM for Dance Partner Robot. *IEEE Trans. Ind. Electron.* **2007**, *54*, 699–706. [CrossRef]
74. Calinon, S.; Billard, A. Incremental learning of gestures by imitation in a humanoid robot. In Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction, Arlington, VA, USA, 10–12 March 2007; pp. 255–262.

75. Calinon, S.; Dhalluin, F.; Sauser, E.L.; Caldwell, D.G.; Billard, A. Learning and Reproduction of Gestures by Imitation. *IEEE Robot. Autom. Mag.* **2010**, *17*, 44–54. [CrossRef]
76. Gams, A.; Nemec, B.; Ijspeert, A.J.; Ude, A. Coupling Movement Primitives: Interaction with the Environment and Bimanual Tasks. *IEEE Trans. Robot.* **2014**, *30*, 816–830. [CrossRef]
77. Zhang, T.; Mccarthy, Z.; Jowl, 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, Australia, 21–25 May 2018; pp. 1–8.
78. Ng, A.Y.; Russell, S. *Algorithms for Inverse Reinforcement Learning*; ICML: Stanford, CA, USA, 2000; Volume 67, pp. 663–670.
79. Krishnan, S.; Garg, A.; Liaw, R.; Thananjeyan, B.; Miller, L.; Pokorny, F.T.; Goldberg, K. SWIRL: A sequential windowed inverse reinforcement learning algorithm for robot tasks with delayed rewards. *Int. J. Robot. Res.* **2019**, *38*, 126–145. [CrossRef]
80. Jiang, Y.; Yang, C.; Wang, Y.; Ju, Z.; Li, Y.; Su, C.Y. Multi-hierarchy interaction control of a redundant robot using impedance learning. *Mechatronics* **2020**, *67*, 102348. [CrossRef]
81. Abbeel, P.; Ng, A.Y. Apprenticeship learning via inverse reinforcement learning. In Proceedings of the Twenty-First International Conference on Machine Learning, Banff Alberta, AL, Canada, 4–8 July 2004; p. 1.
82. Ratliff, N.; Bagnell, J.A.; Zinkevich, M. Maximum margin planning. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; Volume 3, pp. 729–736.
83. Klein, E.; Geist, M.; Piot, B.; Pietquin, O. Inverse Reinforcement Learning through Structured Classification. In Proceedings of the NIPS, Lake Tahoe, NV, USA, 3–6 December 2012, pp. 1007–1015.
84. Ho, J.; Gupta, J.K.; Ermon, S. Model-Free Imitation Learning with Policy Optimization. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016.
85. Xia, C.; Kamel, A.E. Neural inverse reinforcement learning in autonomous navigation. *Robot. Auton. Syst.* **2016**, *84*, 1–14. [CrossRef]
86. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
87. Ziebart, B.D.; Maas, A.L.; Bagnell, J.A.; Dey, A.K. *Maximum Entropy Inverse Reinforcement Learning*; The AAAI Press: Menlo Park, CA, USA, 2008; pp. 1433–1438.
88. Finn, C.; Levine, S.; Abbeel, P. Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016.
89. Boularias, A.; Kober, J.; Peters, J. Relative Entropy Inverse Reinforcement Learning. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 182–189.
90. Peng, X.B.; Abbeel, P.; Levine, S.; De Panne, M.V. DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills. *ACM Trans. Graph.* **2018**, *37*, 143. [CrossRef]
91. Cai, Q.; Hong, M.; Chen, Y.; Wang, Z. On the Global Convergence of Imitation Learning: A Case for Linear Quadratic Regulator. *arXiv* **2019**, arXiv:1901.03674.
92. Ho, J.; Ermon, S. Generative Adversarial Imitation Learning. *arXiv* **2016**, arXiv:1606.03476.
93. Kuefler, A.; Morton, J.; Wheeler, T.A.; Kochenderfer, M.J. Imitating driver behavior with generative adversarial networks. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 204–211.
94. Baram, N.; Anschel, O.; Caspi, I.; Mannor, S. End-to-End Differentiable Adversarial Imitation Learning. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 390–399.
95. Merel, J.; Tassa, Y.; Dhruva, T.B.; Srinivasan, S.; Lemmon, J.; Wang, Z.; Wayne, G.; Heess, N. Learning human behaviors from motion capture by adversarial imitation. *arXiv* **2017**, arXiv:1707.02201.
96. Wang, Z.; Merel, J.; Reed, S.; Wayne, G.; De Freitas, N.; Heess, N. Robust Imitation of Diverse Behaviors. *arXiv* **2017**, arXiv:1707.02747.
97. Stadie, B.C.; Abbeel, P.; Sutskever, I. Third-Person Imitation Learning. *arXiv* **2017**, arXiv:1703.01703.
98. 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, Brisbane, Australia, 21–25 May 2018. [CrossRef]
99. Gupta, A.; Devin, C.; Liu, Y.; Abbeel, P.; Levine, S. Learning Invariant Feature Spaces to Transfer Skills with Reinforcement Learning. *arXiv* **2017**, arXiv:1703.02949.
100. Raileanu, R.; Goldstein, M.; Szlam, A.; Fergus, R. Fast Adaptation via Policy-Dynamics Value Functions. *arXiv* **2020**, arXiv:2007.02879.
101. Charles, R.N.; Guillaume, D.; Andreialexandru, R.; Koray, K.; Thais, H.R.; Razvan, P.; James, K.; Josef, S.H. Progressive Neural Networks. *arXiv* **2017**, arXiv:1606.04671.
102. Tzeng, E.; Devin, C.; Hoffman, J.; Finn, C.; Abbeel, P.; Levine, S.; Saenko, K.; Darrell, T. Adapting Deep Visuomotor Representations with Weak Pairwise Constraints. In *Algorithmic Foundations of Robotics XII*; Springer: Cham, Switzerland, 2015._44. [CrossRef]
103. Zhu, Y.; Mottaghi, R.; Kolve, E.; Lim, J.J.; Gupta, A.; Feifei, L.; Farhadi, A. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3357–3364.

104. Peng, X.B.; Andrychowicz, M.; Zaremba, W.; Abbeel, P. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 3803–3810.
105. Ammar, H.B.; Eaton, E.; Ruvolo, P.; Taylor, M.E. Online Multi-Task Learning for Policy Gradient Methods. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1206–1214.
106. Rusu, A.A.; Vecerik, M.; Rothorl, T.; Heess, N.; Pascanu, R.; Hadsell, R. Sim-to-Real Robot Learning from Pixels with Progressive Nets. In Proceedings of the Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; pp. 262–270.
107. He, Z.; Julian, R.; Heiden, E.; Zhang, H.; Schaal, S.; Lim, J.J.; Sukhatme, G.S.; Hausman, K. Zero-Shot Skill Composition and Simulation-to-Real Transfer by Learning Task Representations. *arXiv* **2018**, arXiv:1810.02422.
108. Ramakrishnan, R.; Kamar, E.; Dey, D.; Horvitz, E.; Shah, J.A. Blind Spot Detection for Safe Sim-to-Real Transfer. *J. Artif. Intell. Res.* **2020**, *67*, 191–234. [CrossRef]
109. Hwasser, M.; Kragic, D.; Antonova, R. Variational Auto-Regularized Alignment for Sim-to-Real Control. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May– 31 August 2020.
110. Golemo, F.; Taiga, A.A.; Courville, A.; Oudeyer, P.Y. Sim-to-real transfer with neural-augmented robot simulation. In Proceedings of the Conference on Robot Learning, New York, NY, USA, 29–31 October 2018; pp. 817–828.
111. Jeong, R.; Kay, J.; Romano, F.; Lampe, T.; Rothorl, T.; Abdolmaleki, A.; Erez, T.; Tassa, Y.; Nori, F. Modelling Generalized Forces with Reinforcement Learning for Sim-to-Real Transfer. *arXiv* **2019**, arXiv:1910.09471.
112. Hwangbo, J.; Lee, J.; Dosovitskiy, A.; Bellicoso, D.; Tsounis, V.; Koltun, V.; Hutter, M. Learning agile and dynamic motor skills for legged robots. *Sci. Robot.* **2019**, *4*. [CrossRef] [PubMed]
113. Matas, J.; James, S.; Davison, A.J. Sim-to-Real Reinforcement Learning for Deformable Object Manipulation. In Proceedings of the Conference on Robot Learning, Zürich, Switzerland, 29–31 October 2018.
114. Sadeghi, F.; Toshev, A.; Jang, E.; Levine, S. Sim2Real View Invariant Visual Servoing by Recurrent Control. *arXiv* **2017**, arXiv:1712.07642.
115. Mees, O.; Merklinger, M.; Kalweit, G.; Burgard, W. Adversarial skill networks: Unsupervised robot skill learning from video. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 30 May–5 June 2020; pp. 4188–4194.
116. Ogenyi, U.E.; Liu, J.; Yang, C.; Ju, Z.; Liu, H. Physical human-robot collaboration: robotic systems, learning methods, collaborative strategies, sensors and actuators. *IEEE Trans. Syst. Man Cybern.* **2019**, 1–14. [CrossRef]
117. Gribovskaya, E.; Kheddar, A.; Billard, A. Motion learning and adaptive impedance for robot control during physical interaction with humans. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 4326–4332.
118. Rozo, L.; Calinon, S.; Caldwell, D.G.; Jimenez, P.; Torras, C. Learning physical collaborative robot behaviors from human demonstrations. *IEEE Trans. Robot.* **2016**, *32*, 513–527. [CrossRef]
119. Calinon, S.; Evrard, P.; Gribovskaya, E.; Billard, A.; Kheddar, A. Learning collaborative manipulation tasks by demonstration using a haptic interface. In Proceedings of the 2009 International Conference on Advanced Robotics, Munich, Germany, 22–26 June 2009; pp. 1–6.
120. Evrard, P.; Gribovskaya, E.; Calinon, S.; Billard, A.; Kheddar, A. Teaching physical collaborative tasks: Object-lifting case study with a humanoid. In Proceedings of the 2009 9th IEEE-RAS International Conference on Humanoid Robots, Paris, France, 7–10 December 2009; pp. 399–404.
121. Levine, S.; Wagener, N.; Abbeel, P. Learning contact-rich manipulation skills with guided policy search. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 156–163.
122. Kaelbling, L.P.; Lozanoperez, T. Unifying perception, estimation and action for mobile manipulation via belief space planning. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, St. Paul, MN, USA, 14–18 May 2012; pp. 2952–2959.
123. Platt, R.W.; Kaelbling, L.P.; Lozanoperez, T.; Tedrake, R. Efficient Planning in Non-Gaussian Belief Spaces and Its Application to Robot Grasping. In *Robotics Research*; Springer: Cham, Switzerland, 2017; pp. 253–269.
124. Li, Z.; Zhao, T.; Chen, F.; Hu, Y.; Su, C.; Fukuda, T. Reinforcement Learning of Manipulation and Grasping Using Dynamical Movement Primitives for a Humanoidlike Mobile Manipulator. *IEEE-ASME Trans. Mechatron.* **2017**, *23*, 121–131. [CrossRef]
125. Kulvicius, T.; Biehl, M.; Aein, M.J.; Tamosiunaite, M.; Worgotter, F. Interaction learning for dynamic movement primitives used in cooperative robotic tasks. *Robot. Auton. Syst.* **2013**, *61*, 1450–1459. [CrossRef]
126. Zhao, T.; Deng, M.; Li, Z.; Hu, Y. Cooperative Manipulation for a Mobile Dual-Arm Robot Using Sequences of Dynamic Movement Primitives. *IEEE Trans. Cogn. Dev. Syst.* **2020**, *12*, 18–29. [CrossRef]
127. Xue, Z.; Ruehl, S.W.; Hermann, A.; Kerscher, T.; Dillmann, R. Autonomous grasp and manipulation planning using a ToF camera. *Robot. Auton. Syst.* **2012**, *60*, 387–395. [CrossRef]
128. Jonschkowski, R.; Brock, O. Learning state representations with robotic priors. *Auton. Robot.* **2015**, *39*, 407–428. [CrossRef]
129. Meng, J.; Zhang, S.; Bekyo, A.; Olsoe, J.; Baxter, B.; He, B. Noninvasive Electroencephalogram Based Control of a Robotic Arm for Reach and Grasp Tasks. *Sci. Rep.* **2016**, *6*, 38565. [CrossRef]
130. Hahne, J.M.; Schweisfurth, M.A.; Koppe, M.; Farina, D. Simultaneous control of multiple functions of bionic hand prostheses: Performance and robustness in end users. *Sci. Robot.* **2018**, *3*. . [CrossRef]

131. Li, Z.; Xu, C.; Wei, Q.; Shi, C.; Su, C. Human-Inspired Control of Dual-Arm Exoskeleton Robots with Force and Impedance Adaptation. *IEEE Trans. Syst. Man Cybern.* **2019**, *50*, 5296–5305 [CrossRef]
132. Calinon, S.; Bruno, D.; Malekzadeh, M.S.; Nanayakkara, T.; Caldwell, D.G. Human-robot skills transfer interfaces for a flexible surgical robot. *Comput. Methods Programs Biomed.* **2014**, *116*, 81–96. [CrossRef]
133. Hu, D.; Gong, Y.; Hannaford, B.; Seibel, E.J. Semi-autonomous simulated brain tumor ablation with RAVENII Surgical Robot using behavior tree. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; Volume 2015, pp. 3868–3875.
134. Deng, M.; Li, Z.; Kang, Y.; Chen, C.L.P.; Chu, X. A Learning-Based Hierarchical Control Scheme for an Exoskeleton Robot in Human—Robot Cooperative Manipulation. *IEEE Trans. Cybern.* **2020**, *50*, 112–125. [CrossRef] [PubMed]
135. Nguyen, T.T.; Nguyen, N.D.; Nahavandi, S. Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Trans. Cybern.* **2020**, *50*, 3826–3839. [CrossRef] [PubMed]
136. Wen, Y.; Si, J.; Brandt, A.; Gao, X.; Huang, H.H. Online Reinforcement Learning Control for the Personalization of a Robotic Knee Prosthesis. *IEEE Trans. Cybern.* **2020**, *50*, 2346–2356. [CrossRef] [PubMed]
137. Duan, Y.; Schulman, J.; Chen, X.; Bartlett, P.L.; Sutskever, I.; Abbeel, P. Fast Reinforcement Learning via Slow Reinforcement Learning. *arXiv* **2017**, arXiv:1611.02779.
138. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; The MIT Press: Cambridge, MA, USA, 1999.
139. Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; Lillicrap, T. Meta-learning with memory-augmented neural networks. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19– 24 June 2016; pp. 1842–1850.
140. Andrychowicz, M.; Denil, M.; Gomez, S.; Hoffman, M.W.; Pfau, D.; Schaul, T.; Shillingford, B.; De Freitas, N. Learning to learn by gradient descent by gradient descent. *arXiv* **2016**, arXiv:1606.04474.
141. Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; Wierstra, D. Matching networks for one shot learning. *arXiv* **2016**, arXiv:1606.04080.
142. Ravi, S.; Larochelle, H. *Optimization as a Model for Few-Shot Learning*; ICLR: Toulon, France, 2017.
143. Sung, F.; Zhang, L.; Xiang, T.; Hospedales, T.M.; Yang, Y. Learning to Learn: Meta-Critic Networks for Sample Efficient Learning. *arXiv* **2017**, arXiv:1706.09529.
144. Duan, Y.; Andrychowicz, M.; Stadie, B.C.; Ho, J.; Schneider, J.; Sutskever, I.; Abbeel, P.; Zaremba, W. One-Shot Imitation Learning. *Neural Inf. Process. Syst.* **2017**, 1087–1098.
145. Finn, C.; Yu, T.; Zhang, T.; Abbeel, P.; Levine, S. One-Shot Visual Imitation Learning via Meta-Learning. In Proceedings of the Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; pp. 357–368.
146. Yu, T.; Finn, C.; Xie, A.; Dasari, S.; Zhang, T.; Abbeel, P.; Levine, S. One-Shot Imitation from Observing Humans via Domain-Adaptive Meta-Learning. *arXiv* **2018**, arXiv:1802.01557.
147. Xu, D.; Nair, S.; Zhu, Y.; Gao, J.; Garg, A.; Feifei, L.; Savarese, S. Neural Task Programming: Learning to Generalize Across Hierarchical Tasks. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 3795–3802.