

Review

# Deep Reinforcement Learning for Soft, Flexible Robots: Brief Review with Impending Challenges

Sarthak Bhagat <sup>1,2,†</sup>, Hritwick Banerjee <sup>1,3,†</sup> and Zion Tsz Ho Tse <sup>4</sup> and Hongliang Ren <sup>1,3,5,\*</sup>

<sup>1</sup> Department of Biomedical Engineering, Faculty of Engineering, 4 Engineering Drive 3, National University of Singapore, Singapore 117583, Singapore; sarthak16189@iitd.ac.in (S.B.); biehb@nus.edu.sg (H.B.)

<sup>2</sup> Department of Electronics and Communications Engineering, IndraPrastha Institute of Information Technology, New Delhi 110020, India

<sup>3</sup> Singapore Institute for Neurotechnology (SINAPSE), Centre for Life Sciences, National University of Singapore, Singapore 117456, Singapore

<sup>4</sup> School of Electrical & Computer Engineering, College of Engineering, The University of Georgia, Athens, GA 30602, USA; ziontse@uga.edu

<sup>5</sup> National University of Singapore (Suzhou) Research Institute (NUSRI), Suzhou Industrial Park, Suzhou 215123, China

\* Correspondence: ren@nus.edu.sg; Tel.: +65-6601-2802

† These authors equally contributed towards this manuscript.

Received: 30 September 2018; Accepted: 1 January 2019; Published: 18 January 2019



**Abstract:** The increasing trend of studying the innate softness of robotic structures and amalgamating it with the benefits of the extensive developments in the field of embodied intelligence has led to the sprouting of a relatively new yet rewarding sphere of technology in intelligent soft robotics. The fusion of deep reinforcement algorithms with soft bio-inspired structures positively directs to a fruitful prospect of designing completely self-sufficient agents that are capable of learning from observations collected from their environment. For soft robotic structures possessing countless degrees of freedom, it is at times not convenient to formulate mathematical models necessary for training a deep reinforcement learning (DRL) agent. Deploying current imitation learning algorithms on soft robotic systems has provided competent results. This review article posits an overview of various such algorithms along with instances of being applied to real-world scenarios, yielding frontier results. Brief descriptions highlight the various pristine branches of DRL research in soft robotics.

**Keywords:** deep reinforcement learning; imitation learning; soft robotics

## 1. Introduction

### 1.1. Soft Robotics: A New Surge in Robotics

The past decade has seen engineering and biology coming together [1–5], leading to cropping up of a relatively newer field of research—Soft Robotics (SoRo). SoRo has been enhancing physical potentialities of robotic structures amplifying the flexibility, rigidity and the strength and hence, accelerating their performance. Biological organisms use their soft structure to good advantage to maneuver in complex environments, hence giving the motivation to exploit such physical attributes that could be incorporated to perform tasks that demand robust interactions with uncertain environments [6]. SoRo with three-dimensional bio-inspired structures [7] are capable of self-regulated homeostasis, resulting in robotics actuators that have the potential to mimic biomimetic motions

with inexpensive actuation [5,8–10]. These developments enabling such robotic hardware present an opportunity to couple with imitation learning techniques by exploiting the special properties of these materials to clinch precision and accuracy. Various underlying physical properties including body shape, elasticity, viscosity, softness, density enable such unconventional structures and morphologies in robotic systems with embodied intelligence. Developing such techniques would certainly lead to fabrication of robots that could invulnerably communicate with the environment. SoRo presents future prospects of being melded with Tissue Engineering, giving rise to composite systems that could find vast applications in the medical domain [11].

Soft Robots are fabricated from materials that are easily deformable and possess the pliability and rheological characteristics of biological tissue. This fragment of the bio-inspired class of machines represents an interdisciplinary paradigm in engineering capable of aiding human assistance in varied domains of research. There are applications wherein SoRo have accelerated the performance and expanded the potentialities. These robots have shown promise from being used as wearables for prosthetics to replacing human labor in industries involving large-scale manipulation and autonomous navigation.

### 1.2. Deep Learning for Controls in Robotics

There has been a certain incline towards utilization of deep learning techniques for creating autonomous systems. Deep learning [12] approaches have shown benefits when combined with reinforcement learning (RL) tasks in the past decade and are known to produce state-of-the-art results in various diverse fields [13]. There have been several pioneer algorithms in this domain in tasks that were difficult to handle with former methods. The need for creating completely autonomous intelligent robotic systems has led to the heavy dependence on the use of Deep RL to solve a set of complex real-world problems without any prior information about the environment. These continuously evolving systems aid the agent to learn through a sequence of multiple time steps, gradually moving towards an optimal solution.

Robotics tasks can be broken down into two different fragments, namely perception [14,15] and control. The task of perception can get necessary information about the environment via sensory inputs, from which they extract desired target quantities or properties. However, in the case of learning a control policy, the agent actively interacts with the environment, trying to achieve an optimal behavior based on the rewards received.

The problem of soft robotic control goes one step further than the former due to the following factors:

- **Data distribution:** In the case of Deep RL for perception, the observations are independent and identically distributed. While in the case of controls, they are accumulated in an online manner due to their continuous nature where each one is correlated to the previous ones [16].
- **Supervision Signal:** Complete supervision is provided in case of perception in the form of ground truth labels. While in controls, there are only sparse rewards available.
- **Data Collection:** Dataset collection can be done offline in perception; it requires online collection in case of controls. Hence, this affects the data we can collect due to the fact that the agent needs to execute actions in the real world, which is not a primitive task.

### 1.3. Deep Learning in SoRo

The task of control of bio-inspired robots requires additional efforts due to the heavy demand of large training data, expensive interactions between a soft robot and the environment, a large action space dimension and the persistently varying structure of the robot due to bending, twisting, warping and other deformations and variations in chemical composition. Such challenges can be simplified as a straightforward problem to design adaptive and evolving models that learn from previous experiences and are capable of handling prodigious-sized datasets.

The implementation of Deep Learning Techniques for solving compound problems in the task of controls [17–19] in soft robotics has been one of the hottest topics of research. Hence, there has been the development of various algorithms that have surpassed the accuracy and precision of earlier approaches. The last decade has seen dependence on soft robotics (and/or bio-robotics) for solving the control related tasks, and applying such DRL techniques on these soft robotics systems has become a focal point of ongoing research. Some of them are depicted in Figure 1.



**Figure 1.** Various applications of SoRo.

Hence, the amalgamation of these budding fields presents the potential of building smarter control systems [11] that can handle objects of varying shapes [20], adapt to continuously diverging environments and perform substantial tasks combining soft robots. Hence, in this paper, we focus on applying DRL and imitation learning techniques to soft robots to perform the task of control of robotic systems.

#### 1.4. Forthcoming Challenges

Artificial Intelligence is the development of machines that are capable of making independent decisions that normally require human aid. The next big step towards learning control policies for robotic applications is imitation learning. In such approaches, the agent learns to perform a task by mimicking the actions of an expert, gradually improving its performance with time as a Deep Reinforcement Learning agent. Even still, it is hard to design the reward/loss function in these cases due to the dimension of the action space, pertaining to the wide variety of motions possible in soft robots. These approaches are valuable for humanoid robots or manipulators with high degrees of freedom where it is accessible to demonstrate desired behaviors because of the magnified flexibility and tensile strength. Manipulation tasks, especially the ones that involve the use of soft robotics, are effectively integrated with such imitation learning algorithms giving rise to agents that are able to imitate expert's actions [3,21,22]. Various factors including the large dimension of the action space, varying composition, and structure of the soft robot and environment alterations presents a variety of challenges that require intense surveillance and deep research. Attempts have similarly been made to reduce the amendments required in models when transferring from one trained on a simulation to the one that functions effectively in the real world. These challenges not only appear as hindrances to achieving complete self-sufficiency but also act as strong candidates for the future of artificial intelligence and Robotics research. In the paper, we list various DRL and Imitation Learning algorithms applied to solve real-world problems, besides mentioning various such challenges that prevail and could act as centers of upcoming research.

This review article is comprised of various sections that focus on applying deep reinforcement learning and imitation learning algorithms to various tasks of manipulation and navigation utilizing soft flexible robots. The beginning sections give a basic overview of the reinforcement learning (Section 2.2) and Deep RL (Section 3.1) followed by descriptive explanation about the application of Deep RL in Navigation (Section 3.3) and Manipulation (Section 3.4) mainly on SoRo environments. The succeeding section (Section 4) talks about behavioural cloning followed by inverse RL and generative adversarial imitation learning applied to solve real world problems. The paper incorporates separate sections on problems faced while transferring learnt policies from simulation to real world

and possible solutions to avoid observing such a *reality gap* (Section 3.5) which gives way to section (Section 3.6) that talks about various simulation softwares available for soft robots. We include a section (Section 5) at the end on challenges of such technologies and budding areas of global interest that can be future frontiers of DRL research for soft robotic systems.

## 2. Brief Overview of Reinforcement Learning

### 2.1. Introduction

Soft robots intend to solve non-trivial tasks that are generally required to have adaptive capabilities in interacting with constantly varying environments. Controlling such soft-bodied agents with the aid of Reinforcement Learning involves making machines that are able to execute and identifying optimal behavior in terms of a certain reward (or loss) function. Therefore, Reinforcement Learning can be expressed as a procedure in which at each state  $s$  the agent performs an action  $a$ , receiving a response in the form of a reward from the environment. It decides the goodness of the previous state-action pairs and this process continues until the agent has learned a policy well enough. This is a process that involves both explorations that refer to exploring different ways to achieve a particular task as well as exploitation, which is the method of utilizing the current information gained and trying to receive the largest reward possible at that given state.

Robotics tasks can be modeled as a Markov Decision Processes (MDP), consisting of a 5-tuple such as: (i)  $S$ : set of states; (ii)  $A$ : set of actions; (iii)  $P$ : transition dynamics; (iv)  $R$ : set of rewards; and  $\gamma$ : discount factor. Episodic MDPs have a terminal state which once obtained ends the learning procedure. Episodic MDPs with time horizon  $T$ , ends after  $T$  time steps regardless of whether it has reached its goal or not. In the problem of controls for robots, the information about the environment is gathered through the sensors which are not enough to make a decision about the action, such MDPs are called Partially Observable MDPs. These are countered either by stacking observations up to that time step before processing or by using a recurrent neural network (RNN). In any RL task, we intend to maximize the expected discount return that is the weighted sum of rewards received by the agent [23]. For this purpose, we have two types of policies, namely stochastic ( $\pi(a|s)$ ) where actions are drawn from a probability distribution and deterministic ( $\mu(s)$ ) where they are selected specifically for every state. Then, we have Value functions ( $V_{\pi}(s)$ ) that depict the expected outcome starting from state  $s$  and following policy  $\pi$ .

Reinforcement Learning tasks can be broadly classified as model-based and models-free approaches. The framework in which the agents learn optimal actions for each state based on the rewards and observations is a model-based technique. In these methods, we make use of supervised learning algorithms to minimize a cost function based on what the agent observes from the environment. It is not necessary to learn a model for predicting the optimal actions. In approaches like Actor-Critic and Policy-based methods (will be described in detail in the next sections), we can simply estimate the optimal Q-values for any particular action at a state from which it is trivial to choose the policy with the highest Q-values. Such methods are beneficial when dealing with SoRo due to the difficulty and cost of the interaction of SoRo with the environment.

### 2.2. Reinforcement Learning Algorithms

This section provides an overview of major RL algorithms that have been extended by using deep learning frameworks.

- **Value-based Methods:** These methods estimate the probability of being in a given state, using which the control policy is determined. The sequential state estimation is done by making use of Bellman's Equations (Bellman's Expectation Equation and Bellman's Optimality Equation). Value-based RL algorithms include State-Action-Reward-State-Action (SARSA) and Q-Learning, which differ in their targets, that is the target value to which Q-values are recursively updated by a step size at each time step. SARSA is an on-policy method where the value estimations are updated

towards a policy while Q-Learning, being an off-policy method, updates the value estimations towards a target optimal policy. This algorithm is a complex algorithm that is used to expound various multiplex-looking problems but computational constraints act as stepping stones to utilizing it. Detailed explanation can be found in recent works like Dayan [24], Kulkarni et al. [25], Barreto et al. [26], and Zhang et al. [27].

- Policy-based Methods: In contrast to the Value-based methods, Policy-based methods directly update the policy without looking at the value estimations. Some of key differences between Value-based and Policy-based are listed in Table 1. They are slightly better than value-based methods in the terms of convergence, solving problems with continuous high dimensional data, and effectiveness in solving deterministic policies. They perform in two broad ways—gradient-based and gradient-free [28,29] methods of parameter estimation. We focus on gradient-based methods where gradient descent seems to be the choice of optimization algorithm. Here, we optimize the objective function as:

$$J(\pi_\theta) = E_{\pi_\theta}[f_{\pi_\theta}(\cdot)] \quad (1)$$

wherein the score function [30] for the policy  $\pi_\theta$  is given by  $f_{\pi_\theta}(\cdot)$ . Using Equation (1), we can comment on the performance of the model with respect to the task in hand. A RL algorithm is the REINFORCE algorithm [31], that simply plugs in the sample return equal to the score function given by:

$$f_{\pi_\theta}(\cdot) = G_t. \quad (2)$$

A baseline term  $b(s)$  is subtracted from the sample return to reduce the variance of estimation which updated the equation in the following manner:

$$f_{\pi_\theta}(\cdot) = G_t - b_t(s_t). \quad (3)$$

while using Q-value function, score function can make use of either stochastic policy gradient (Equation (4)) or deterministic policy gradient (Equation (5)) [32] given by:

$$\nabla_\theta(\pi_\theta) = E_{s,a}[\nabla_\theta \log \pi_\theta(a|s) \cdot Q^\pi(s, a)] \quad (4)$$

and

$$\nabla_\theta(\mu_\theta) = E_s[\nabla_\theta \mu_\theta(s) \cdot Q^\mu(s, \mu_\theta(s))]. \quad (5)$$

It is observed that this method certainly overpowers the former in terms of computational time and space limitations. Still, it cannot be extended to tasks involving interaction with continuously evolving environments that require the agent to be adaptive.

It has been noted that it is not practically suitable to follow the policy gradient because of safety issues and hardware restrictions. Therefore, we optimize using the policy gradient on stochastic policies wherein integration is done over state space due to the large dimension of the action space in case of soft robots that can sustain movements in directions and angles possible.

- Actor Critic Method: These algorithms keep a clear representation of the policy and state estimations. The score function for this is obtained by replacing the return  $G_t$  from Equation (3) of policy based methods with  $Q^{\pi_\theta}(s_t, a_t)$  and baseline  $b(s)$  with  $V^{\pi_\theta}(s_t)$  that results in the following equation:

$$f_{\pi_\theta}(\cdot) = Q_{\pi_\theta}(s_t, a_t) - V_{\pi_\theta}(s_t). \quad (6)$$

The advantage function  $A(s, a)$  is given by:

$$A(s, a) = Q(s, a) - V(s). \quad (7)$$

Actor-critic methods could be described as an intersection of policy-based and value-based methods, wherein it combines iterative learning methods of both the methods.

- Integrating Planning and Learning: There exist methods wherein the agent learns from experiences itself and can collect imaginary roll-outs [33]. Such methods have been upgraded by using alongside DRL methods [34,35]. They are essential in extending RL techniques to soft robotic systems, as the droves of degrees of freedom lead to expensive interaction with the environment and hence compromise on the training data available.

**Table 1.** Value Based and Policy Based (along with Actor Critic Methods) based on various factors.

Algorithm	Value Based Methods	Policy Based Methods (and Actor Critic Methods)
Examples	Q-Learning, SARSA, Value Iteration	Advantage Actor Critic, Cross Entropy Method
Steps Involved	Finding optimal value function and find the policy based on that (policy extraction)	Policy evaluation and policy improvement
Iteration	The two processes (listed in above cell) are not repeated after once completed	The above two processes (listed in above cell) are iteratively done to achieve convergence
Convergence	Relatively Slower	Relatively Faster
Type of Problem Solved	Relatively Harder control problems	Relatively basic control problems
Method of Learning	Explicit Exploration	Innate Exploration and Stochasticity
Advantages	Basic to train off-policy	Blends well with supervised way of learning
Process Basis	Based on Optimality Bellman Operator- is non-linear operator	Based on Bellman Operator

### 3. Deep Reinforcement Learning Algorithms Coupled with SoRo

#### 3.1. Introduction

The benefits in terms of physical and mechanical properties allow a wide range of actions with soft robots. There are sectors wherein soft robots have found extensive applications:

**Bio-medical:** Soft Robots have found enormous applications in the domain of bio-medicine, that include the development of soft robotic tools for surgery, diagnosis, drug delivery, wearable medical devices, and prostheses, active simulators that copy the working of human tissues for training and biomechanical research. The fact that they are durable and flexible makes them apt for applications involving maneuvering in close and delicate areas where a possible human error could cause heavy damage. Certain special properties of being completely water-soluble or ingestible make them a candidate for an effective delivery agent.

**Manipulation:** Another application domain of soft robots is autonomous picking, sorting, distributing, classifying and grasping capabilities in various workplaces including warehouses, factories, and industries.

**Mobile Robotics:** Various types of diverse domain-specific robots that possess the ability to move have been employed for countless purposes. Robots that could walk, climb, crawl or jump, having structures inspired from other animals that portray special movement capabilities find applications in inspection, monitoring, maintenance and cleaning tasks. The recent works in the field of swarm technology have greatly enhanced the performance of robots that are mobile and possess such flexible and adaptive structures.

**Edible Robotics:** The considerable developments in the edible robotic materials and 3D printing have led to a sharp rise in ease of prototyping soft robotic structure that is ingestible and water

soluble. Such biodegradable robotic equipment could relieve damages that are incurred because of the interaction of these machines with the environment, contributing to pollution (especially the damage done to the water bodies). These unique type of robots are generally composed of edible polymers competitive for use in the medical and food industry.

Origami Mechanics: Origami, a concept that has been in use for hundreds of years, has been employed to enhance the physical strength of soft robotic systems [22,36]. These robots that have structures are capable of having varied sizes, shapes, and appearances and are proficient in lifting weight up to 1000 times their own weight. These robots can find intensive applications in various diverse industries that require lifting of heavy material.

Apart from these, soft robots have found applications in including motor control in machines, assistive robots, military reconnaissance, natural disaster relief, pipe inspection and more.

The vast domain applications of soft robotics have made its study alongside DRL-based methods necessary. It can perform compound tasks because of their special mechanical capabilities and incorporate self-adaptive and evolving models that learn from interactions from the environment. The following table shows various domains wherein soft robots are utilized and DRL techniques will be discussed in detail in the sections to follow.

**Table 2.** SoRo applied to achieve state-of-the-art results alongside sub-domains where its utilization with deep reinforcement learning (DRL) and imitation learning techniques presently occur.

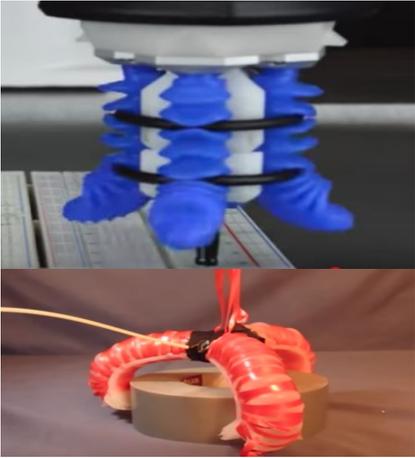
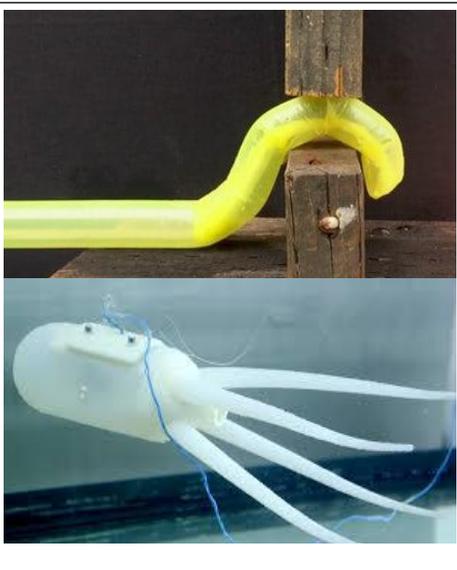
Domain of Application	Basic Applications	Methods in Which DRL/Imitation Learning Algorithms Can Be Incorporated
<p><i>Biomedical</i> [37,38]</p> <ul style="list-style-type: none"> <li>• Equipment for Surgeries, endoscopy, laparoscopy etc.</li> <li>• Prosthetics for the impaired</li> </ul>		<ul style="list-style-type: none"> <li>• Autonomous surgeries, endoscopy, laparoscopy etc. via Imitation Learning</li> <li>• Vision Capabilities via DRL techniques for analysis of ultrasounds, X-Rays etc.</li> <li>• Automatic and in dependant Manipulation abilities for soft robotic gloves or other wearable prosthetics using DRL techniques in human-machine interface meant for impaired.</li> </ul>
<p><i>Manipulation</i> [39]</p> <ul style="list-style-type: none"> <li>• Automation of various picking, placing, grasping, sorting tasks in industries, factories and other workspaces</li> <li>• Picking and placing heavy objects using strength of super strong soft robots</li> </ul>		<ul style="list-style-type: none"> <li>• Imitation Learning techniques for manipulation tasks accompanied by autonomous selection/classification using DRL-based vision capabilities</li> <li>• Meta-Learning—an effective amalgamation of Imitation Learning and DRL-based learning for complex manipulation tasks</li> </ul>

Table 2. Cont.

Domain of Application	Basic Applications	Methods in Which DRL/Imitation Learning Algorithms Can Be Incorporated
<p><i>Mobile Robotics</i> [40]</p> <ul style="list-style-type: none"> <li>• Various tasks in warehouses, industries, factories etc. could be automated that require maintenance, inspection and cleaning applications</li> <li>• Surveillance and disaster management applications</li> </ul>	 <p>The image block contains two photographs. The top photograph shows a bright yellow, flexible, tube-like robotic arm with a curved end, mounted on a wooden post. The bottom photograph shows a white, octopus-like robot with multiple long, thin tentacles, floating in a blue liquid environment.</p>	<ul style="list-style-type: none"> <li>• Autonomous path planning using DRL-based techniques in compound environments to perform perverse tasks</li> <li>• Imitation Learning techniques for teaching tasks like walking, crawling, sprinting etc.</li> </ul>

Neural networks can approximate optimal value functions in Reinforcement Learning Algorithms and hence, have been extensively applied to predict the control policy in robotics. Systems involving soft robots generally have challenges in policy optimization due to large action and state spaces. Hence, incorporating neural networks in models alongside adaptive reinforcement learning techniques enhances the performance. The last decade has seen a sharp rise in the usage of DRL methods for performing a variety of tasks to make use of the bio-inspired structures of soft robots. The following are the common DRL algorithms in practise solving such control problems:

- Deep Q-Network (DQN) [41]: In this approach the optimal value of Q-function is obtained using a deep neural network (generally a CNN), like we do in other value-based algorithms. We denote the weights of this Q-network by  $Q^*(s, a)$  and the error and target are given by equations:

$$\delta_t^{DQN} = y_t^{DQN} - Q(s_t, a_t; \theta_t^Q) \tag{8}$$

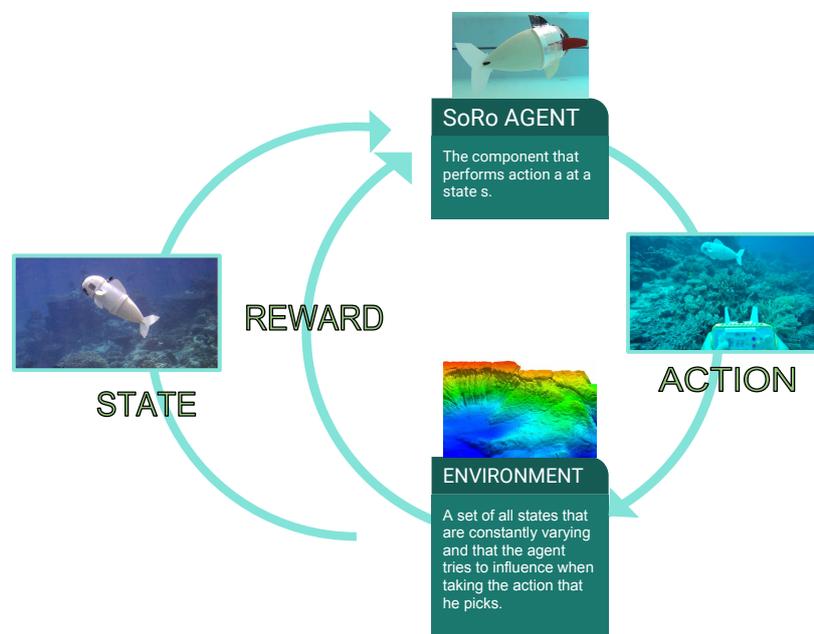
and

$$y_t^{DQN} = R_{t+1} + \gamma. \tag{9}$$

The weights are recursively updated by the equation:

$$\theta_{t+1} \leftarrow \theta_t - \alpha (\partial(\delta_t^{DQN}(\theta_t^Q))^2 / \partial \theta_t^Q). \tag{10}$$

moving in the direction of decreasing gradient with a rate equal to the learning rate. These are capable of solving problems involving high dimensional state spaces but restricted to discrete low dimensional action space. These manipulation methods using soft robotics that have structure inspired from biological creatures can interact with the environment, alongside additional flexibility and adaptation to changing situations. The training architecture is given by Figure 2.



**Figure 2.** Training architecture of a Deep Q-Network (DQN) agent.

Two main methods employed in DQN for learning are:

- Target Network: Target network Q- has the same architecture as the Q-network but while learning the weights of only the Q-network are updated, while repeatedly being copied to

weights of the  $\theta^-$  network. In this procedure, a target is computed from the output of  $\theta^-$  function [18].

- Experience Replay: The collected data in form of state-action pairs with their rewards are not directly utilized but are stored in a replay memory. While actually training, samples are picked up from the memory to serve as mini-batches for the learning. The further learning task follows the usual steps of using gradient descent to reduce loss between learned Q-network and target Q-network.

These two methods are used to stabilize the learning in DQN by reducing the correlation between estimated and target Q-values, and between consecutive observations respectively. Advanced techniques for stabilizing and creating efficient models include Double DQN [42] and Dueling DQN [43].

- Deep Deterministic Policy Gradients (DDPG) [18]: This is a modification of the DQN combining techniques from actor-critic methods to model problems with continuous high dimensional action spaces. The training procedure of a DDPG agent is depicted in Figure 3. The equations for stochastic (Equation (11)) and deterministic (Equation (12)) policies are given by equations:

$$Q^\pi(s_t, a_t) = E_{R_{t+1}, s_{t+1} \sim E} [R_{t+1} + \gamma E_{a_{t+1} \sim \pi} [Q^\pi(s_{t+1}, a_{t+1})]] \quad (11)$$

and

$$Q^\mu(s_t, a_t) = E_{R_{t+1}, s_{t+1} \sim E} [R_{t+1} + \gamma Q^\mu(s_{t+1}, \mu(s_{t+1}))]. \quad (12)$$

The difference between this and DQN lies in the dependence of Q-value on the action where it is represented by giving one value from each action in DQN and by taking action as input to theta Q in case of DDPG. This method remains to be one of the premiere algorithms in the field of DRL applied to systems utilizing soft robots.

- Normalised Advantage Function (NAF) [44]: This functions in a similar way as DDPG in the sense that it enables Q-learning in continuous high dimensional action spaces by employing the use of deep learning. In NAF, Q-function  $Q(s, a)$  is represented so as to ensure that its maximum value can easily be determined during the learning procedure. The difference in NAF and DQN lies in the network output, wherein it outputs  $\theta^V$ ,  $\theta^\mu$  and  $\theta^L$  in its last linear layer of the neural network.  $\theta^\mu$  and  $\theta^L$  predict the advantage necessary for the learning technique. Similar to a DQN, it makes use of Target Network and Experience Replays to ensure there is the least correlation in observations collected over time. The advantage term in NAF is given by:

$$A(s, a; \theta^\mu, \theta^L) = -(1/2)(a - \mu(s; \theta^\mu))^T P(s; \theta^L)(a - \mu(s; \theta^\mu)) \quad (13)$$

wherein,

$$P(s; \theta^L) = L(s; \theta^L)L(s; \theta^L)^T. \quad (14)$$

Asynchronous NAF approach has been introduced in the work by Gu et al. [45].

- Asynchronous Advantage Actor Critic (A3C) [46]: In asynchronous DRL approaches, various actors-learners are utilized to collect observations, each storing gradient for their respective observations that used to update the weights of the network. A3C, as a commonly used algorithm of this type, always maintains a policy representation  $\pi(a|s; \theta^\pi)$  and a value estimation  $V(s; \theta^V)$  making use of score function in the form of an advantage function that is obtained by observations that are provided by the action-learners. Each actor-learner collects roll-outs of observations

of its local environment up to  $T$  steps, accumulating gradients from samples in the roll-outs. The approximation of advantage function used in this approach is given by equation:

$$A(s_t, a_t; \theta^\pi, \theta^V) = \left[ \sum_{k=t}^{T-1} [\gamma^{k-t}] + \gamma^{T-t} V(s_T; \theta^V) - V(s_t; \theta^V); \theta^\pi \right]. \quad (15)$$

The network parameters  $\theta^V$  and  $\theta^\pi$  are updated repeatedly according to the equations given by:

$$d\theta^\pi \leftarrow d\theta^\pi + \nabla_{\theta^\pi} \log \pi(a_t | s_t; \theta^\pi) A(s_t, a_t; \theta^\pi, \theta^V) \quad (16)$$

and

$$d\theta^V \leftarrow d\theta^V + \partial A(s_t, a_t; \theta^\pi, \theta^V)^2 / \partial \theta^V. \quad (17)$$

Training architecture is shown in Figure 4.

This approach does not require learning stabilization techniques like memory replay as the parameters are updated simultaneously rather than sequentially, hence eliminating the correlation factor between them. Furthermore, there are action-learners involved in this method that tend to explore a wider view of the environment and helping to learn an optimal policy. A3C has proven to be the stepping stone for DRL research and to be efficient in providing state-of-art results alongside reduced time and space complexity and its range of problem-solving capabilities.

- Advantage Actor-Critic (A2C) [47,48]: It is not necessary that asynchronous methods lead to better performance. It has been shown in various papers, that synchronous version of the A3C algorithm provides fine results wherein each actor-learner finishes collecting observation after which they are averaged and an update is made.
- Guided Policy Search (GPS) [49]: This approach involves collecting samples making use of current policy, generating a training trajectory at each iteration that is utilized to update the current policy according to supervised learning. The change is bounded by adding it like a regularization term in the cost function, to prevent sudden changes in policies leading to instabilities.
- Trust Region Policy Optimization (TRPO) [50]: In Schulman et al. [50], an algorithm was proposed for optimization of large nonlinear policies which gave improvement in the accuracy. Discount cost function for an infinite horizon MDP is given by replacing reward function with cost function giving the equation:

$$\eta(\pi) = E_\pi \left[ \sum_{t=0}^{\infty} \gamma^t c(s_t) | s_0 \sim \rho_0 \right]. \quad (18)$$

Similarly, the same replacement made to state-value functions give the following Equation (19). Hence, resulting in advantage function given by:

$$A^\pi = Q^{\pi_i}(s, a) - V^\pi(s). \quad (19)$$

Optimizing Equation (19) would result in giving an updating rule for the policy as follows:

$$\eta(\pi) = \eta(\pi_{old}) + E \left[ \sum_{t=0}^{\infty} \gamma^t A^{\pi_{old}}(s_t, a_t) | s_0 \sim \rho_0 \right]. \quad (20)$$

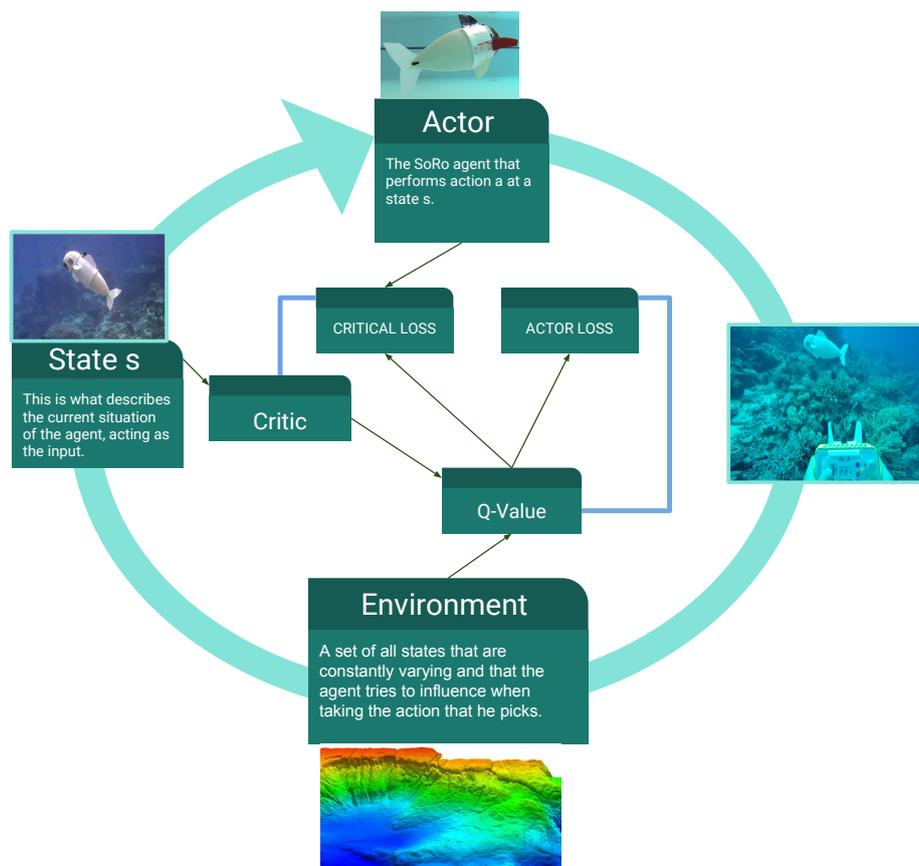
This has been mathematically proven via advanced literatures by Kakade and Langford [51] that this method improves the performance. This algorithm requires advanced optimization problem solving techniques using conjugate gradient and then using line search [50].

- Proximal Policy Optimization (PPO) [52]: These methods solve soft constraint optimization problem making use of standard Stochastic Gradient Descent problem. Due to its simplicity and

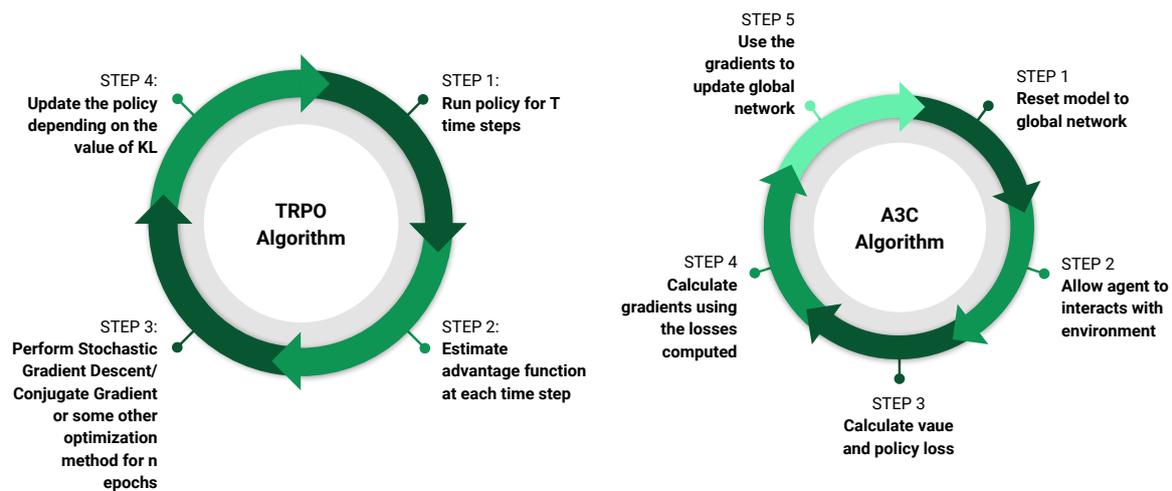
effectiveness in solving control problems, it has been applied to policy estimation in OpenAI. Training architecture is shown in Figure 4.

- Actor-Critic Kronecker-Factored Trust-Region (ACKTR) [48]: This uses a form of trust region gradient descent algorithm for an actor-critic with curvature estimated using a Kronecker-Factored approximation.

These methods have shown prospect when combined with the innumerable physical capabilities of the soft structure of bio-inspired robots [3,21], and are a topic of interest. Some of these algorithms applied successfully to SoRo are listed in Table 2.



**Figure 3.** Training architecture of a Deep Deterministic Policy Gradients (DDPG) agent. The blue lines portray the updation equations.



**Figure 4.** Training architectures of Asynchronous Advantage Actor Critic (A3C) and Trust Region Policy Optimization (TRPO) agents.

### 3.2. Deep Reinforcement Learning Mechanisms

Mechanisms have been proposed that can enhance the learning procedure while solving control problems involving soft robots through the aid of DRL algorithms. These act as catalysts to the task in hand acting orthogonally to the actual algorithm. They ensure that the task of solving DRL problems to obtain nearly optimal actions with respect to each state for soft robotics is computationally less expensive. A diverse set of DRL control tasks include:

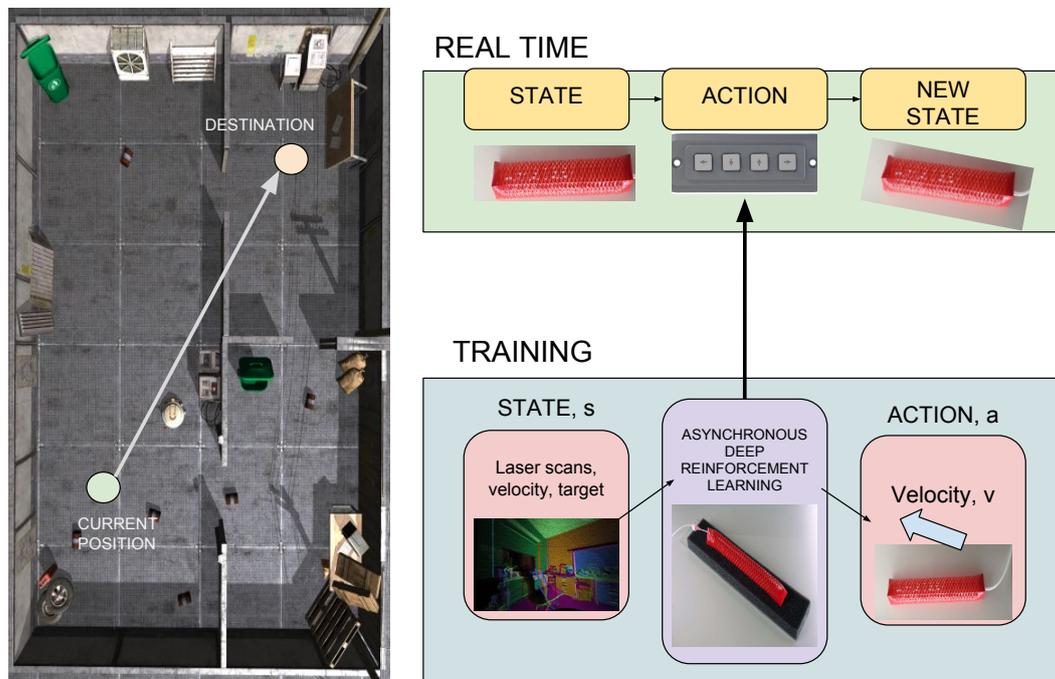
- Auxiliary Tasks [53–57]: Usage of other supervised and unsupervised machine learning methods like regressing depth images from color images, detecting loop closures etc., besides the main algorithm to receive information from sparse supervision signals in the environment.
- Prioritized Action Replay [58]: Prioritizing memory replay according to error
- Hindsight Action Replay [13]: Relabeling the rewards for the collected observations by effective use of failure trajectories along with using binary/sparse labels that speed the off-policy methods.
- Curriculum Learning [59–61]: Exposing the agent to a sophisticated set of the environment, helping it to learn to solve complex tasks.
- Curiosity-Driven Exploration [62]: Incorporating internal rewards besides external ones collected from observations.
- Asymmetric Action Replay for Exploration [63]: The interplay between two forms of the same learner generates curricula, hence, driving exploration
- Noise in Parameter Space for Exploration [64,65]: Inserts additional noise so as to aid exploration in the learning procedure.

The rising demands for creating structures in physical capabilities in terms of flexibility, strength, rigidity etc. Autonomous Systems lead to growth in the developments in DRL that could be applied to soft robots. This brings about a need for incorporating such mechanisms (as mentioned in this section) in models that strengthen the impact of DRL algorithms that are coupled with the soft robots.

### 3.3. Deep Reinforcement Learning for Soft Robotics Navigation

Deep RL approaches have turned out to be an aid for navigation tasks, helping to generate such trajectories by learning from observations taken from the environment in form of state-action pairs. Similar to other types of robots, Soft Robots require autonomous navigation capabilities that can be coupled with their mechanical properties allowing them to execute onerous looking tasks with

ease. Soft robots used for investigation, maintenance or monitoring purposes at various workplaces have the climbing or crawling capabilities that require self-sufficient path planning potentialities. Completely reliable and independent movement is necessary for creating systems that perform tasks requiring continuous interaction with the environment in order to find the path for desired movements. One such example of DRL being utilized in order to navigate between two points within a room by a mobile soft robot is depicted in Figure 5.



**Figure 5.** Expected application of DRL techniques in the task of navigation.

Like other DRL problems, the navigation problem is considered as an MDP. Input sensors (LIDAR scans and depth images from an onboard camera) get readings and in return output a trajectory (policy in the form of actions to be taken in a particular state), that will complete the task of reaching the goal in the given span of time.

Experiments have been carried out in this growing field of research as stated below:

- Zhu et al. [66] gave the A3C system the first-person view alongside the target image to conceive the goal of reaching the destination point, by the aid of universal function approximators. The network used for learning is a ResNet [67] that is trained using a simulator [68] that creates a realistic environment consisting of various rooms each as a different scene for scene-specific layers [66].
- Zhang et al. [27] implemented a deep successor representation formulation for predicting Q-value functions that learn representations interchangeable between navigation tasks. Successor feature representation [25,26] breaks down the learning into two fragments-learning task-specific reward functions and task-specific features alongside their evolution for getting the task in hand done. This method takes motivation from other DRL algorithms that make use of optimal function approximations to relate and utilize the information gained from previous tasks for solving the tasks we currently intend to perform [27]. This method has been observed to work effectively in transferring current policies to different goal positions in varied/scaled reward functions and to newer complex situations like new unseen environments.

Both these methods intend to solve the problem of navigation for autonomous robots that have inputs in the form of RGB images of the environment by either getting the target image [66] or by transferring information that is gained through previous processes [27,69]. Such models are trained via asynchronous DDPG for a varied set of real and simulations of real environments.

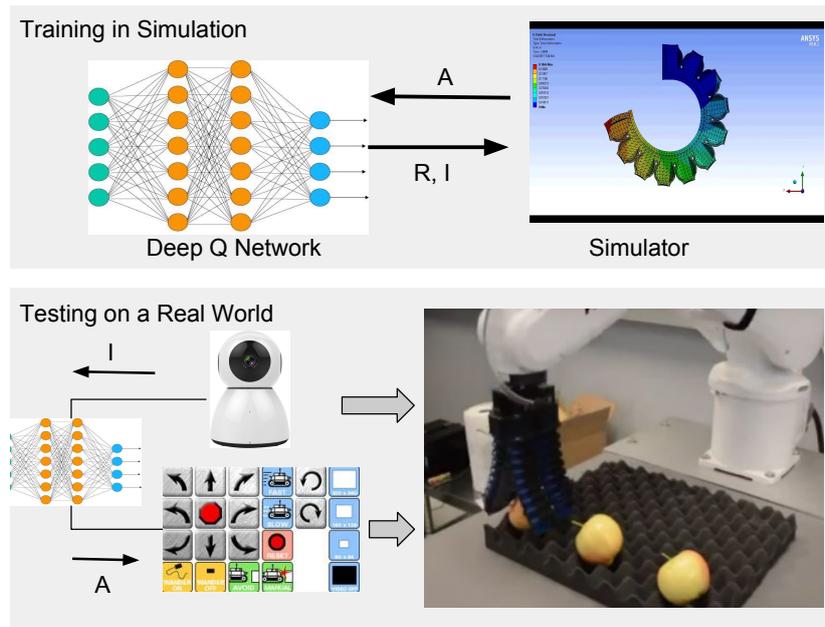
- Mirowski et al. [53] made use of a DRL mechanism with additional supervision signals available (especially loop closures and depth prediction losses) in the environment, allowing the robot to freely move between a varying start and end.
- Chen et al. [70] proposed a solution for compound problems involving dynamic environment (essentially obstacles) like navigating on a path with pedestrians as moving obstacles. It utilizes a set of hardware to demonstrate the proposed algorithm in which LIDAR sensor readings are used to predict the different properties associated with pedestrians (like speed, position, radius) that contribute to forming the reward/loss function. Long et al. [71] make use of PPO to conduct multi-agent obstacle avoidance task.
- Simultaneous Localisation and Mapping (SLAM) [72] has been at the heart of recent robotic advancements with loads of papers been written regularly in this field in the last decade. SLAM makes use of DRL methodologies partially or completely and have shown to produce one of the best results in such tasks of localization and navigation.
- An imitation learning problem, that will be dealt with later in detail, trains a Cognitive Mapping and Planning (CMP) [73] model with DAGGer that inputs 3-channeled images and with the aid of Value Iteration Networks (VIN) creates a map of the environment. The work of Gupta et al. [74] has further introduced a new form of amalgamation of spatial reasoning and path planning.
- Zhang et al. [60] further introduced a new form of SLAM called Neural SLAM that took inspiration from works of Parisotto and Salakhutdinov [75] that allowed interacting with Neural Turing Machine (NTM). Graph-based SLAM [72,76] led the way for Neural Graph Optimiser by Parisotto et al. [77] which inserted this global pose optimiser in the network.

Further advanced readings on Navigation using DRL techniques include Zhu et al. [66], Schaul et al. [78], He et al. [67], Kolve et al. [68], Zhu et al. [79], Long et al. [71], Gupta et al. [73], Khan et al. [80], Bruce et al. [81], Chaplot et al. [82] and Savinov et al. [83].

#### 3.4. Deep Reinforcement Learning for Soft Robotics Manipulation

DRL techniques have been applied to soft robotic manipulation tasks like picking, dropping, reaching, etc. [18,46,50,84]. The enhanced rigidity, flexibility, strength and adaptation capability of soft robots over hard robots [85,86] has extensive applications in the manipulation field and combining it with DRL has been observed to give precise and satisfactory results. The coming of the soft robotics technologies and its blend with deep learning technologies has contributed to it becoming a crucial part of manipulation tasks. One such example of DRL being utilized in a manipulator with soft end-effectors for picking objects of varied sizes and shapes is depicted in Figure 6.

Recent advancements in the grasping capabilities aided by vision-based complex systems have certainly provoked the growth in the utilization of models involving the employment of artificially intelligent robots. Various techniques of robotic grasp detection [87] and delicate control of soft robots for grasping objects of varying shape, size, and composition [88] have evolved, making way for deep learning and DRL algorithms integrated with benefits of the robust yet limber structure of soft robots. These models take as input 3-channel images of the scene along with the object to be picked, passing it through a deep convolutional network (CNN) that outputs a grasp predictor. This predictor is the input to control system that intends to maneuver the end-effector in order to pick the object.



**Figure 6.** Application of DRL techniques (DQN Method) in the task of manipulation.

After developments in the domain of soft robotics, we require learning algorithms that can solve complex manipulation tasks alongside taking care to follow constraints that are enforced as a result of the physical structure of such bio-inspired robots [3,21]. DRL technologies have certainly enhanced the performance of such agents.

In the past few years, we have witnessed a drastic increase in research focus on DRL techniques while dealing with soft robots as listed below:

- Gu et al. [45]: Gave a modified form of NAF that works in an asynchronous fashion in the task of door opening taking state inputs like joint angles, end effector position and position of the target. It gave a whopping 100% accuracy in this task and learned it in a mere 2.5 h.
- Levine et al. [56]: Proposed a visuomotor policy-based model that is an extended deep version of the GPS algorithm studied earlier. The architecture of the network consists of convolutional layers along with softmax activation function taking in as input-images of the environment and concatenating the necessary information gained along with the robot's state information. The required torques are predicted by passing this concatenated input to linear layers at the end of the network. These experiments were carried out with a PR2 robot for various tasks like screwing a bottle, inserting the block into a shape sorting cube etc. Despite giving desirable results, it is not widely used in real-world applications as it requires complete observability in state space that is difficult to obtain in real life.
- Finn et al. [89] and Tzeng et al. [90]: Made use of DRL techniques for predicting optimal control policies by studying state representations.
- Fu et al. [91]: Introduced the use of a dynamic network for creating a one-shot model for manipulation problems. There have been advancements in the area of manipulation using multi-fingered bio-inspired hands that are model-based [92,93] or model-free [94].
- Riedmiller et al. [54]: Gave a new algorithm that enhanced the learning procedure from the time complexity as well as accuracy point of view. It said that sparse rewards for the model in attaining optimal policy faster than providing binary rewards, that lead to policy that did not have the desired trajectories for the end effector. For this, another policy (referred to as intentions) was learned for auxiliary tasks whose inputs are easily attainable via basic sensors [95]. Besides this, a scheduling policy is further learned for scheduling the intention policies. Such a system has

better results than a normal DRL algorithm for the task of lifting that took about 10 h to learn from scratch.

Soft Robotics has turned out to be an emerging field especially for manipulation tasks, turning out to be superior in terms of accuracy and efficiency in comparison to the human or hard robotic counterparts [96]. Involving such soft robots in place of humans has further lead to a drastic dip in chances of industrial disasters due to human error (as a result of their environment adaptation property) and they have proven to be valuable for working environments that are unsuitable.

### 3.5. Difference between Simulation and Real World

Collecting training samples is not an easy task while solving the problem of controls in soft robotics as it is in perception for similar systems. Collection of the real-world dataset (state-action pair) is a costly operation due to the high dimensionality of the control spaces and the lack of availability of a central source of control data for every environment. This inflicts various challenges while bringing models that have been trained in simulation to real-world scenarios. Even though we have various simulation software for soft robotics especially designed for manipulation tasks like picking, dropping, placing etc. as well as navigation tasks like walking, jumping, crawling etc., there are still challenges that act as hindrances in this problem of solving control tasks making use of these flexible robots.

Soft robots having bio-inspired designs that make use of DRL techniques like the ones listed in the previous sections are known to yield satisfactory results, but still, they face various obstacles that hinder their performance when tested on the real-world problems after being trained on simulation settings. Such a gap is often viewed in disparities in visual data [27] or laser readings [69]. The following section provides an overview:

- Domain Adaptation: This translates images from a source domain to the destination domain. Domain confusion loss that was first proposed in the paper Tzeng et al. [97] learns a representation that is steady towards changes in the domain. However, the limitation of this approach lies in the fact that it requires the source and destination domain information before the training step, which is challenging. Visual data coming from multiple sources is represented by  $X$  (simulator) and  $Y$  (onboard sensors). The problem arises when we train the model on  $X$  and test it on  $Y$ , wherein we observe a considerable amount of performance difference between the two. This problem of *reality gap* is a genuine problem faced while dealing with systems involving soft robots due to the constant variations in the position of end-effector in numerous degrees of freedom. Hence, there is a need for a system that is invariant to changes in perspective (transform) with which the agent observes various key points in the environment. Domain adaptation is a basic yet effective approach that is widely utilized to solve problems of low accuracy due to variations between simulation and real-world environments.
- This problem can be solved if we have a mapping function that can map data from one domain to the other one. This can be done by employing a deep generative model called Generative Adversarial Network or commonly known as GANs [98–100]. GANs are deep models that have two basic components—a discriminator and a generator. The job of the generator is to produce image samples from the source domain to the destination domain, while that of the discriminator is to differentiate between true and false (generated) samples.
  - CycleGAN [79]: First proposed in Zhu et al. [79], works on the principle that it is possible and feasible to predict a mapping that maps from input domain to output domain simply by adding a cycle consistent loss term as a regulariser, for the original loss for making sure the mapping is reversible. It is a combination of two normal GANs and hence, two separate encoders, decoders and discriminators are trained according to equations:

$$L_{GAN_Y}(G_Y, D_Y; X, Y) = E_y[\log D_Y(y)] + E_x[\log(1 - D_Y(D_Y(x)))] \quad (21)$$

and

$$L_{GAN_X}(G_X, D_X; Y, X) = E_x[\log D_X(x)] + E_y[\log(1 - D_X(D_X(y)))]. \quad (22)$$

The loss term for the complete weight updating (after incorporating the cycle consistent loss terms for each GAN) step now turns out to be:

$$L(G_Y, G_X, D_Y, D_X; X, Y) = L_{GAN_Y}(G_Y, D_Y; X, Y) + L_{GAN_X}(G_X, D_X; Y, X) \\ + \lambda_{cyc} L_{cyc_Y}(G_X, G_Y; Y) + \lambda_{cyc} L_{cyc_X}(G_Y, G_X; X). \quad (23)$$

Hence, the final optimization problem turns out to be Equation (24).

$$G_Y^*, G_X^* = \arg \min_{G_Y, G_X} \max_{D_Y, D_X} L(G_Y, G_X, D_Y, D_X) \quad (24)$$

This is known to produce desirable results for scenes to draw comparisons/relations between both domains but occasionally fails on complex environments.

- CyCADA [101]: The problems that CycleGAN, that was first introduced in Hoffmann et al. [101], faced were resolved by making use of the semantic consistency loss that could be used to map complex environments. It trains a model to move from the source domain containing semantic labels, helping map the domain images from  $X$  to that in  $Y$ . The equations that are used for mapping using the decoder are given by:

$$L_{sem_Y}(G_Y; X, f_X) = CE(f_X(X), f_X(G_Y(X))) \quad (25)$$

and

$$L_{sem_X}(G_Y; Y, f_X) = CE(f_X(Y), f_X(G_X(Y))). \quad (26)$$

Here,  $CE(S_X, f_X(X))$  represents the cross-entropy loss between data-points predicted by pre-trained model and the true labels  $S_X$ .

Deep learning frameworks like GANs [98–100], VAEs [102], disentangled representations [103,104] have the potential to aid the control process of soft robots. These developing frameworks have widened the perspective of DRL for robotic controls. The combination of two such tender fields of technology-soft robotics and deep learning frameworks (especially generative models) act as stepping stones to major technological advancement in the coming decades.

- Domain Adaptation for Visual DRL Policies: In such adaptation techniques, we transform the policy from a source domain to the destination domain.

Bousmalis et al. [105] proposed a new way to solve problems of reality gap in policies trained on simulations and applying them in real life scenarios.

There have been recent developments with the aim of developing newer training techniques to avoid such a gap in efficiency while testing on simulation and real-world scenarios besides advancements in the simulation environments possible to create virtually. Tobin et al. [106] randomized the lighting conditions, viewing angles and texture of objects to ensure the agent is manifested to disparities in the factors of variation. The work by Peng et al. [107] and Rusu et al. [108] further focus on such training methods. The recent advancements in VR Goggles [109] have separated policy learning and its adaptation in order to minimize the transfer steps required for moving from simulation to the real world. A new optimization objective comprising of an additional shift loss regularisation term was further deployed on such a model, that borrows motivation from artistic style transfer proposed by Ruder et al. [110]. Works in the domain of scene adaptation include indoor scene adaptation where the semantic loss is not added (A VR Goggles [109] model tested on Gazebo [111] simulator using

a Turtlebot3 Waffle) and outdoor scene adaptation where we do add such a semantic loss term to the original loss function. Outdoor scene adaptation involves a collection of real-world data through a dataset like RobotCar [112] which is tested on a simulator (like CARLA [113]). The network is trained using DeepLab Model [114] after adding the semantic loss term. Such a model turns out to be applicable to situations where the simulation fails to accurately represent the real agent [115].

With a rise in the number of simulations software (and simulation methods [116]) available publicly and the growth in the hardware industry for soft robots-upcoming of 3D printed bio-robots [117–120] with specific tasks for which they have been designed for alongside special flexible electronics for such systems [121]. There is scope for improvement in the development of real-world soft robots with practical applications, making way for a hot topic for future research in the upcoming years.

### 3.6. Simulation Platforms

There are platforms that are available for simulation purposes of DRL agents before testing on real-world applications, selected in Table 3.

**Table 3.** The following table lists various simulation software available for simulating real-looking environments for training of DRL agents alongside the modularities available.

Simulator	Modalities	Special Purpose of Use
Gazebo (Koenig et al. [111])	Sensor Plugins	General Purpose
Vrep (Rohmer et al. [122])	Sensor Plugins	General Purpose
Airsim (Shah et al. [123])	Depth/Color/Semantics	Autonomous Driving
Carla (Dosovitskiy et al. [113])	Depth/Color/Semantics	Autonomous Driving
Torcs (Pan et al. [124])	Color/Semantics	Autonomous Drivings
AI-2 (Kolve et al. [68])	Color	Indoor Navigation
Minos (Savva et al. [125])	Depth/Color/Semantics	Indoor Navigation
House3D (Wu et al. [126])	Depth/Color/Semantics	Indoor Navigation

The readily available software contributes to the upcoming research in the field of controls. The fact that a normal DRL agent requires lots of training even before testing it in real environments makes the presence of special purpose simulation tools important.

For the task of controls, there is fewer simulation software available for soft robots [11] as compared to the hard ones. The fact that soft robotics is a relatively new field might be the reason that there is scarcely any simulation software for manipulation tasks using a soft robot.

SOFA is an efficient framework for physical simulation of soft robotics of different shapes, sizes and material that has the potential to boost the research in this budding field. The SOFA allows the user to model, simulate and control a soft robot. Soft-robotics toolkit [127] is a plugin that aids to simulate soft robots using SOFA framework [128]. Others that are capable of modeling and simulating soft robotics agents are V-REP simulator [122], Action simulation by Energid, and MATLAB (Simscape modeling) [129] (Figure 7).

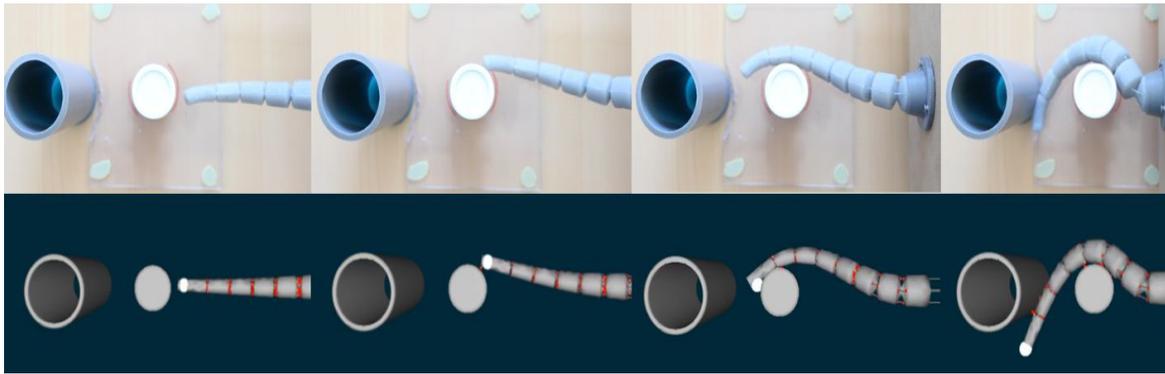


Figure 7. Soft Robot Simulation on SOFA using Soft-robotics toolkit [128].

Apart from the development of simulation software that has expanded the current state of development of soft robots, the increasing number of open-sourced software libraries enabled effortless programming of deep neural networks, creating computational graphs that are intensely optimized. The programming languages that have found large-scale applications when coupled with soft robots include Theano, Torch, Caffe, Tensorflow, and MXNet.

#### 4. Imitation Learning for Soft Robotic Actuators

There are drawbacks of training a DRL agent to perform control tasks especially for soft robots: high training time and data that requires robots to interact with the real world which is computationally expensive, a well-formulated reward function is impractical to obtain. Imitation Learning is a unique approach to perform a control task, especially the ones involving the use of bio-inspired robots without the requirement of a reward function. In such cases, it is inconvenient to formulate one where the problem is ill-posed and requires an expert whose actions are mimicked by the agent [16]. Imitation Learning is desirable in situations where an expert is present with high degrees of movement/action space leading to enormous computation time, necessity of a large training set and the difficulty to give a reward function that describes the problem. An overview of the training procedure for an imitation learning agent is shown in Figure 8.

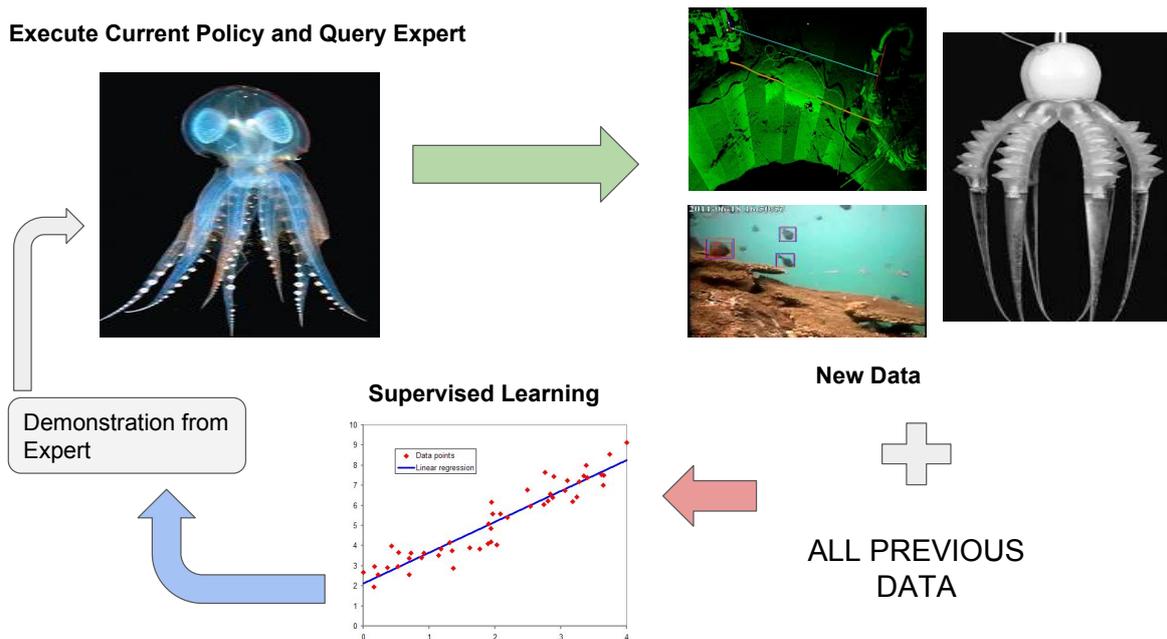


Figure 8. Training Architecture of CycleGAN and CyCADA [85,130].

The use of imitation learning for solving problems of manipulation like picking, dropping, etc. [131,132] where we can exploit the benefits of soft robotics over hard ones have become essential. Controls tasks in such situations generally have to compute cost functions due to the high dimension of action space caused by the flexibility of the soft structure of the robot introduced in the motion of the actuator/end-effector. Under such situations, the study of imitation learning becomes a topic of significance. It requires an expert agent which in cases of manipulation using soft robotics is a person who performs the same tasks that a robot is required to copy. Therefore, manipulation with soft robots and imitation learning algorithms for performing the control task in hand go hand in hand and complement each other.

A primitive imitation learning algorithm is a supervised learning problem. However, simply applying the normal steps of supervised learning to tasks involving the formulation of control policy does not work well. There are changes/variations that must be made due to the difference in common supervised learning problems and control problems. The following section provides an overview of those variations:

- **Errors—Independent or Compound:** The basic assumption of a common supervised learning task that assumes that the actions of the soft robotics agent do not affect the environment in any way is violated in the case of imitation control tasks. The presupposition that data samples (observations) collected are independent and identically distributed is not valid for imitation learning tasks, hence, causing error propagation making the system unstable due to minor errors too.
- **Time-step Decisions—Single or Multiple:** Supervised learning models generally ignore the dependence between consecutive decisions different from what primitive robotic approaches. The goal of imitation learning is different from simply mimicking the expert's actions. At times, a hidden objective is missed by the agent while simply copying the actions of the expert. These hidden objectives might be tasks like avoiding colliding with obstacles, increasing the chances to complete a specific task, or minimizing the effort by the mobile robot.

In the next section, we describe three of the main imitation learning algorithms that have been applied to real life scenarios effectively.

- **Behaviour Cloning:** This is one of the basic imitation learning approaches in which we train a supervised learning agent based on actions of the expert agent from input states to output states via performed actions. Data AGGregation (DAGGer) [133] is one of the algorithms described earlier that solves the problem of propagation of errors in a sequential system. This is similar to common supervised learning problems in which at each iteration the updated (current) policy is applied and observations recorded are labeled by expert policy. The data collected is concatenated to the data already available and the training procedure is applied to it. This technique has been readily utilized in diverse domain applications due to its simplicity and effectiveness.

Bojarski et al. [134] trained a navigation control model that collected data from 72 h of actual driving by the expert agent and tried to mimic the state (images pixels) to actions (steering commands) with a mapping function. Similarly, Tai et al. [135] and Giusti et al. [136] came up with imitation learning applications for real life robotic control. Advanced readings include Codevilla et al. [137] and Dosovitskiy et al. [113].

Imitation learning is effective in problems involving manipulation given below:

- Duan et al. [138] improved the one-shot imitation learning to formulate the low-dimensional state to action mapping, using behavioral cloning that tries to reduce the differences in agent and the expert actions. He used this method in order to make a robotic arm stack various blocks in the way the expert does it, observing the relative position of the block at each time step. The performance achieved after incorporating various other additional features like temporal dropouts and convolutions were similar to that of a DAGGer.

- Finn et al. [139] and Yu et al. [55] modified the already existing Model Agnostic Meta-Learning (MAML) [140], which is a diverse algorithm that trains a model on varied tasks and making it capable to solve a new unseen task when assigned. The updating of weights,  $\theta$  is done using a method which is quite similar to the common gradient algorithm and given by equation:

$$\theta'_i = \theta - \alpha \nabla_{\theta} L_{T_i}(f_{\theta}), \tag{27}$$

wherein  $\alpha$  is the step size for gradient descent. The learning is done to achieve the objective function given by:

$$\sum_{T_i \sim p(T)} L_{T_i}(f_{\theta'}) = \sum_{T_i \sim p(T)} L_{T_i}(f_{\theta - \alpha \nabla_{\theta} L_{T_i}(f_{\theta})}) \tag{28}$$

which leads to the gradient descent step given by:

$$\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{T_i \sim p(T)} L_{T_i}(f_{\theta'_i}) \tag{29}$$

wherein  $\beta$  represents the meta step size.

- While Duan et al. [138] and Finn et al. [139] propose a way to train a model that works on newer set of samples, the earlier described Yu et al. [55] is an effective algorithm in case of domain shift problems. Eitel et al. [141] came up with a new approach wherein he gave a new model that takes in over segmented RGB-D images as inputs and gives actions as outputs for segregation of objects in an environment.
- Inverse Reinforcement Learning: This method aims to formulate the utility function that makes the desired behavior nearly optimal. An IRL algorithm called as Maximum Entropy IRL [142] uses an objective function as given by the equation:

$$\arg \max_{c \in C} (\min_{\pi \in \Pi} -H(\pi) + E_{\pi}[c(s, a)]) - E_{\pi^E}[c(s, a)]. \tag{30}$$

For a robot following a set a constraints [143–145], it is difficult to formulate an accurate reward function but these actions are easy to demonstrate. Soft robotic systems generally have constraint issues due to the composition and configuration of different materials of the actuators resulting in the elimination of a certain part of the action space (or sometimes state space). Hence forcing the involvement of IRL techniques in such situations. This algorithm for soft robotic systems can perform the task that we want to solve by the human expert due to the flexibility and adaptability of the soft robot. The motivation for exploiting this technique in soft robots comes from the fact that their pliant movements make it difficult to formulate a cost function, hence leading to dependence of such models in systems requiring resilience and robustness. Maximum Entropy IRL [146] has been used alongside deep convolutional networks to learn the multiplex representations in problems involving a soft robot to copy the actions of a human expert for control tasks.

- Generative Adversarial Imitation Learning: Even though IRL algorithms are effective, they require large sets of data and training time. Hence, an alternative was proposed by Ho and Ermon [147] who gave Generative Adversarial Imitation Learning (GAIL) that comprises a Generative Adversarial Network (GAN) [98]. Such generative models are essential when working with soft robotic systems as they require larger sets of training data because of the wide variety of actions-state pairs possible in such cases and the fact that GANs are complex deep networks that are able to learn complex representations in the latent space.

Like other GANs, GAIL consists of two independently trained fragments-generator (or the decoder) that generate state-action pairs close to that of the expert and the discriminator that

learns to distinguish between samples created by the generator and real samples. The objective function of such a model is given by equation:

$$E_{\pi_{\theta}}[\log(D(s, a))] + E_{\pi^E}[\log(1 - D(s, a))] - \lambda H(\pi_{\theta}). \quad (31)$$

Extensions of GAIL have been proposed in recent works including Baram et al. [148] and Wang et al. [149]. GAIL solved imitation learning problems in navigation (Li et al. [150] and Tai et al. [151] applied it for finding socially compliant policies) as well as manipulation (Stadie et al. [152] used GAIL for mimicking an expert's actions through domain agnostic representation). This presents an opportunity for GAIL to be applied to systems involving soft actuators for its composite structure and unique learning technique.

Imitation learning for soft robotics, being a relatively new field of research, has not yet been explored to its fullest. It is effective in the domains of industrial applications, capable of replacing its human counterpart due to its precision, reliability, and efficiency. Expert agent's actions can be mimicked by an autonomous soft robotics agent. These algorithms overcome the tough formulation of an appropriate cost function due to high action space dimensionality of soft robots. These techniques form an amalgam that could copy the expert as well as learn on its own via exploration depending on the situation in hand, and hence, must be a center for future deep learning developments in soft robots. DRL alongside imitation learning has been applied to countless scenarios and has been observed to provide satisfactory results, as shown in Table 4.

**Table 4.** Instances of bio-inspired soft robotics applications that make use DRL or Imitation Learning technologies.

Type of Soft Bio-Inspired Robot	Features of Soft Physical Structure	Applications of DRL/Imitation Learning Algorithms
<p data-bbox="255 336 707 360"><i>MIT's Soft Robotic Fish (SoFi)</i> [153–157]</p> 	<ul style="list-style-type: none"> <li>• 3D Movements</li> <li>• Soft Fluidic Elastomer Actuators</li> <li>• Rapid Planar Movements</li> <li>• Continuum Body Motion</li> <li>• Multiple Degrees of Freedom</li> <li>• Quick Escape Capabilities</li> </ul>	<ul style="list-style-type: none"> <li>• Autonomous Maneuvering using DRL navigation techniques (described in Section 3.3)</li> <li>• DRL techniques employed for underwater vision capabilities</li> </ul>
<p data-bbox="255 759 707 783"><i>Harvard's Soft Octopus (Octobot)</i> [158,159]</p> 	<ul style="list-style-type: none"> <li>• Microfluidic logic</li> <li>• Immense Strength and Dexterity with no internal skeleton</li> <li>• Innumerable Degrees of Freedom</li> <li>• Ease in prototyping due 3D printable structure</li> <li>• Rapid Maneuvering through tight spaces</li> </ul>	<ul style="list-style-type: none"> <li>• DRL and Imitation Learning algorithms extensively employed for manipulation capabilities like grasping and picking</li> <li>• Autonomous navigation potential via the use of Deep Learning techniques</li> </ul>

Table 4. Cont.

Type of Soft Bio-Inspired Robot	Features of Soft Physical Structure	Applications of DRL/Imitation Learning Algorithms
<p>CMU's Inflatable Soft Robotic Arm [160,161]</p> 	<ul style="list-style-type: none"> <li>• Quick Planar Movement via Soft Artificial Muscle Actuators, Soft and Stretchable Elastic Films and Flexible Electronics</li> <li>• Touch Sensors and Pressure-Sensitive Skins to predict the shape of objects</li> </ul>	<ul style="list-style-type: none"> <li>• Precise grasping, holding and picking capabilities via DRL techniques</li> <li>• DRL and Deep Learning Vision abilities</li> </ul>
<p>Soft Caterpillar Micro-Robot [162]</p> 	<ul style="list-style-type: none"> <li>• Light-sensitive rubby material that harvest energy from light</li> <li>• Ease in prototyping due 3D printable structure</li> <li>• Horizontal/Vertical Movement possible in tough environment, angles and conditions</li> <li>• Strength to push items 10 times their mass</li> </ul>	<ul style="list-style-type: none"> <li>• Autonomous Movement Capabilities that could involve Imitation Learning or DRL techniques</li> </ul>

## 5. Future Scope

Deep learning has the potential to solve control problems like manipulation in soft robots. Further, we list the stepping stones in the path of using deep reinforcement approaches to solve such tasks that might be topics of great interest in the near future:

- **Sample Efficiency:** It takes efforts and resources in collecting observations for training by making agents interact with the environments especially for soft robotic systems due to the various number of actions possible at each state. The biomimetic motions [10,163] of the flexible bio-inspired actuators make way for further research in creating efficient systems that can collect experiences without the expense.
- **Strong Real-time Requirements:** Training networks with millions of neurons and tons of tunable parameters that require special hardware and loads of computational time. The current policies need to be made compact in their representation to prevent wasting time and hardware resources in training. The dimensionality of the actions as well as the state space for soft robotic actuated systems is sizeable as compared to its hard counterpart leading to a rise in the number of neurons in the deep network.
- **Safety Concerns:** The control policies designed need to be precise in their decision-making process. Like factories producing food items, soft robots are required to operate in environments where even a small error could cause loss of life and property.
- **Stability, Robustness, and Interpretability [164]:** Slight changes in configurations of parameters or robotic hardware or changes in concentration or composition of the materials of the soft robot over time affect the performance of the agent in a way, hence making it unstable. A learned representation that can detect adversarial scenarios is a topic of interest for researchers aiming to improve the performance of DRL agents on soft robotic systems.
- **Lifelong Learning:** The appearance of the environment differs drastically when observed at different moments, alongside the composition and configuration of soft robotics systems varying with time could result in a certain dip in performance of the learned policies. Hence, this problem provokes technologies that are always evolving and learning from changes in the environmental conditions caused due to actions like bending, twisting, warping and other deformations and variations in chemical composition of the soft robot, besides keeping the policies intact.
- **Generalization between tasks:** A completely trained model is able to perform well in the tasks trained on, but it performs poorly in new tasks and situations. For soft robotics systems that are required to perform a varied set of tasks that are correlated, it is necessary to come up with methods that can transfer the learning from one training procedure when being tested on tasks. Therefore, there is a requirement of creating completely autonomous systems that take up least resources for training and still are diverse in application. This challenge is of key significance in the context of soft robots due to the hefty expense of allowing them to interact with the environment and the inconsistency of their own structure and composition, leading to increased adaptation concerns.

Despite these challenges in control problems for soft robots, there are topics that are gaining the attention of DRL researchers due to the future scope of development in these areas of research. Two of them are:

- **Unifying Reinforcement Learning and Imitation Learning:** There have been quite a few developments [165–168] with the aim to combine the two algorithms and reap the benefits of both wherein the agent can learn from the actions of the expert agent alongside interacting and collecting experiences from the environment itself. The learning from experts' actions (generally a person for soft robotic manipulation problems) can sometimes lead to less-optimal solutions while using deep neural networks to train reinforcement learning agent can turn out to be an expensive task pertaining to the high dimensional action space for soft robots. Current research in

this domain focuses on creating a model where a soft robotic agent is able to learn from experts' demonstrations and then as the time progresses, it moves to a DRL-based exploration technique wherein it interacts with the continuously evolving environment to collect observations. In the near future, we could witness completely self-determining soft robotics systems that have the best of both worlds. It can learn from the expert in the beginning and equipped with capabilities to learn on its own when necessary. Hence resulting in the benefits of the amalgamated mechanical structure by exploiting its benefits.

- **Meta-Learning:** Methods proposed in Finn et al. [140] and by Nichol and Schulman [169] have found a way to find parameters from relatively fewer data samples and produce better results on newer tasks than they have not been trained on. This development can be stepping stones to further developments leading to the creation of robust and universal policy solutions. This could be a milestone research item when it comes to combining deep learning technologies with soft robotics. Generally, it is hard to retrieve a large dataset for soft robotic systems due to the expenses in allowing it to interact with its environment. Soft robotic systems are generally harder to deal with as compared to the harder ones and therefore, such learning procedures could aid soft robots to perform satisfactorily well.

Control of soft robots of enhanced flexibility and strength has become one of the premier domains of robotics researchers. There have been numerous DRL and imitation learning algorithms proposed for such systems. Recent works have shown massive span for further development including some that could branch out as separate areas of soft robotics research themselves. These challenges have opened new doors for such artificially intelligent algorithms that will be a trending topic of discussion and debate for the coming decades. Combining deep learning frameworks with soft robotic systems and extracting the benefits of both is seen a potential area of future developments. For Table 4, image source: MIT News/Youtube, NPG Press/Youtube, National Science Foundation (Credit: Siddharth Sanan, Carnegie Mellon University), Quality Point Tech/Youtube.

## 6. Conclusions

This paper demonstrate an overview of deep reinforcement learning and imitation learning algorithms applied to problems involving control of soft robots and have been observed to give state-of-the-art results in their domains of application especially manipulation where soft robots are extensively utilized. We have described learning paradigms of various learning techniques, followed by the instances of being applied to solve real-life robotic control problems. Despite the growth in research in this field of universal interest in the last decade, there are still challenges in controls of soft robots (it being a relatively new field of research in robotics) that needs concentrated attention. Soft Robotics is a constantly growing academic domain that focuses on exploiting the mechanical structure by integration of materials, structures, and software, and when combined with the boons of imitation learning and other DRL mechanisms can create systems capable of replacing humans at any discipline possible. We list the stepping stones to the development of such soft robots that are completely autonomous and self-adapting yet physically strong systems.

In a nutshell, the subject that gathers the attention of one and all remains to how the incorporation of DRL and imitation learning approaches can accelerate the ever so satisfactory performances of soft robotic systems and unveil a plethora of possibilities of creating altogether self-sufficient systems in the near future.

**Author Contributions:** H.R., and Z.T.H.T. provided the outline for the draft and critical revision; S.B. and H.B. conducted the literature search and S.B., H.B. drafted the manuscript; H.B. and S.B. also collected data for tables and figures.

**Funding:** This work was supported by the NMRC Bench & Bedsides under Grant R-397-000-245-511, Singapore Academic Research Fund under Grant R-397-000-227-112, and the NUSRI China Jiangsu Provincial Grants BK20150386 & BE2016077 awarded to H.R.

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

## References

1. Trimmer, B. A Confluence of Technology: Putting Biology into Robotics. *Soft Robot.* **2014**, *1*, 159–160. [[CrossRef](#)]
2. Banerjee, H.; Tse, Z.T.H.; Ren, H. Soft Robotics with Compliance and Adaptation for Biomedical Applications and Forthcoming Challenges. *Int. J. Robot. Autom.* **2018**, *33*. [[CrossRef](#)]
3. Trivedi, D.; Rahn, C.D.; Kier, W.M.; Walker, I.D. Soft robotics: Biological inspiration, state of the art, and future research. *Appl. Bionics Biomech.* **2008**, *5*, 99–117. [[CrossRef](#)]
4. Banerjee, H.; Ren, H. Electromagnetically responsive soft-flexible robots and sensors for biomedical applications and impending challenges. In *Electromagnetic Actuation and Sensing in Medical Robotics*; Springer: Berlin, Germany, 2018; pp. 43–72.
5. Banerjee, H.; Aaron, O.Y.W.; Yeow, B.S.; Ren, H. Fabrication and Initial Cadaveric Trials of Bi-directional Soft Hydrogel Robotic Benders Aiming for Biocompatible Robot-Tissue Interactions. In Proceedings of the IEEE ICARM 2018, Singapore, 18–20 July 2018.
6. Banerjee, H.; Roy, B.; Chaudhury, K.; Srinivasan, B.; Chakraborty, S.; Ren, H. Frequency-induced morphology alterations in microconfined biological cells. *Med. Biol. Eng. Comput.* **2018**. [[CrossRef](#)] [[PubMed](#)]
7. Kim, S.; Laschi, C.; Trimmer, B. Soft robotics: A bioinspired evolution in robotics. *Trends Biotechnol.* **2013**, *31*, 287–294. [[CrossRef](#)] [[PubMed](#)]
8. Ren, H.; Banerjee, H. A Preface in Electromagnetic Robotic Actuation and Sensing in Medicine. In *Electromagnetic Actuation and Sensing in Medical Robotics*; Springer: Berlin, Germany, 2018; pp. 1–10.
9. Banerjee, H.; Shen, S.; Ren, H. Magnetically Actuated Minimally Invasive Microbots for Biomedical Applications. In *Electromagnetic Actuation and Sensing in Medical Robotics*; Springer: Berlin, Germany, 2018; pp. 11–41.
10. Banerjee, H.; Suhail, M.; Ren, H. Hydrogel Actuators and Sensors for Biomedical Soft Robots: Brief Overview with Impending Challenges. *Biomimetics* **2018**, *3*, 15. [[CrossRef](#)]
11. Iida, F.; Laschi, C. Soft robotics: Challenges and perspectives. *Proc. Comput. Sci.* **2011**, *7*, 99–102. [[CrossRef](#)]
12. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [[CrossRef](#)]
13. Andrychowicz, M.; Wolski, F.; Ray, A.; Schneider, J.; Fong, R.; Welinder, P.; McGrew, B.; Tobin, J.; Abbeel, O.P.; Zaremba, W. Hindsight experience replay. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5048–5058.
14. Deng, L. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Trans. Signal Inf. Process.* **2014**, *3*. [[CrossRef](#)]
15. Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S.; Lew, M.S. Deep learning for visual understanding: A review. *Neurocomputing* **2016**, *187*, 27–48. [[CrossRef](#)]
16. Bagnell, J.A. *An Invitation to Imitation*; Technical Report; Carnegie-Mellon Univ Pittsburgh Pa Robotics Inst: Pittsburgh, PA, USA, 2015.
17. Levine, S. Exploring Deep and Recurrent Architectures for Optimal Control. *arXiv* **2013**, arXiv:1311.1761.
18. 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.
19. Spielberg, S.; Gopaluni, R.B.; Loewen, P.D. Deep reinforcement learning approaches for process control. In Proceedings of the 2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP), Taipei, Taiwan, 28–31 May 2017; pp. 201–206.
20. Ho, S.; Banerjee, H.; Foo, Y.Y.; Godaba, H.; Aye, W.M.M.; Zhu, J.; Yap, C.H. Experimental characterization of a dielectric elastomer fluid pump and optimizing performance via composite materials. *J. Intell. Mater. Syst. Struct.* **2017**, *28*, 3054–3065. [[CrossRef](#)]
21. Shepherd, R.F.; Ilievski, F.; Choi, W.; Morin, S.A.; Stokes, A.A.; Mazzeo, A.D.; Chen, X.; Wang, M.; Whitesides, G.M. Multigait soft robot. *Proc. Natl. Acad. Sci. USA* **2011**, *108*, 20400–20403. [[CrossRef](#)] [[PubMed](#)]
22. Banerjee, H.; Pusalkar, N.; Ren, H. Single-Motor Controlled Tendon-Driven Peristaltic Soft Origami Robot. *J. Mech. Robot.* **2018**, *10*, 064501. [[CrossRef](#)]
23. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
24. Dayan, P. Improving generalization for temporal difference learning: The successor representation. *Neural Comput.* **1993**, *5*, 613–624. [[CrossRef](#)]

25. Kulkarni, T.D.; Saeedi, A.; Gautam, S.; Gershman, S.J. Deep Successor Reinforcement Learning. *arXiv* **2016**, arXiv:1606.02396.
26. Barreto, A.; Dabney, W.; Munos, R.; Hunt, J.J.; Schaul, T.; van Hasselt, H.P.; Silver, D. Successor features for transfer in reinforcement learning. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 4055–4065.
27. Zhang, J.; Springenberg, J.T.; Boedecker, J.; Burgard, W. Deep reinforcement learning with successor features for navigation across similar environments. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 2371–2378.
28. Fu, M.C.; Glover, F.W.; April, J. Simulation optimization: A review, new developments, and applications. In Proceedings of the 37th Conference on Winter Simulation, Orlando, FL, USA, 4–7 December 2005; pp. 83–95.
29. Szita, I.; Lőrincz, A. Learning Tetris using the noisy cross-entropy method. *Neural Comput.* **2006**, *18*, 2936–2941. [[CrossRef](#)]
30. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; Abbeel, P. High-Dimensional Continuous Control Using Generalized Advantage Estimation. *arXiv* **2015**, arXiv:1506.02438.
31. Williams, R.J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **1992**, *8*, 229–256. [[CrossRef](#)]
32. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic policy gradient algorithms. In Proceedings of the ICML, Beijing, China, 21–26 June 2014.
33. Sutton, R.S. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bull.* **1991**, *2*, 160–163. [[CrossRef](#)]
34. Weber, T.; Racanière, S.; Reichert, D.P.; Buesing, L.; Guez, A.; Rezende, D.J.; Badia, A.P.; Vinyals, O.; Heess, N.; Li, Y.; et al. Imagination-Augmented Agents for Deep Reinforcement Learning. *arXiv* **2017**, arXiv:1707.06203.
35. Kalweit, G.; Boedecker, J. Uncertainty-driven imagination for continuous deep reinforcement learning. In Proceedings of the Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; pp. 195–206.
36. Banerjee, H.; Pusalkar, N.; Ren, H. Preliminary Design and Performance Test of Tendon-Driven Origami-Inspired Soft Peristaltic Robot. In Proceedings of the 2018 IEEE International Conference on Robotics and Biomimetics (IEEE ROBIO 2018), Kuala Lumpur, Malaysia, 12–15 December 2018.
37. Atalay, O.; Atalay, A.; Gafford, J.; Walsh, C. A Highly Sensitive Capacitive-Based Soft Pressure Sensor Based on a Conductive Fabric and a Microporous Dielectric Layer. *Adv. Mater.* **2017**. [[CrossRef](#)]
38. Cianchetti, M.; Ranzani, T.; Gerboni, G.; Nanayakkara, T.; Althoefer, K.; Dasgupta, P.; Menciassi, A. Soft Robotics Technologies to Address Shortcomings in Today’s Minimally Invasive Surgery: The STIFF-FLOP Approach. *Soft Robot.* **2014**, *1*, 122–131. [[CrossRef](#)]
39. Truby, R.L.; Wehner, M.J.; Grosskopf, A.K.; Vogt, D.M.; Uzel, S.G.M.; Wood, R.J.; Lewis, J.A. Soft Somatosensitive Actuators via Embedded 3D Printing. *Adv. Mater.* **2018**, *30*, e1706383. [[CrossRef](#)] [[PubMed](#)]
40. Bishop-Moser, J.; Kota, S. Design and Modeling of Generalized Fiber-Reinforced Pneumatic Soft Actuators. *IEEE Trans. Robot.* **2015**, *31*, 536–545. [[CrossRef](#)]
41. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529. [[CrossRef](#)] [[PubMed](#)]
42. Van Hasselt, H.; Guez, A.; Silver, D. Deep Reinforcement Learning with Double Q-Learning. In Proceedings of the AAAI, Phoenix, AZ, USA, 12–17 February 2016; Volume 2, p. 5.
43. Wang, Z.; Schaul, T.; Hessel, M.; Van Hasselt, H.; Lanctot, M.; De Freitas, N. Dueling Network Architectures for Deep Reinforcement Learning. *arXiv* **2015**, arXiv:1511.06581.
44. Gu, S.; Lillicrap, T.; Sutskever, I.; Levine, S. Continuous deep q-learning with model-based acceleration. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 2829–2838.
45. 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.

46. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, 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.
47. Wang, J.X.; Kurth-Nelson, Z.; Tirumala, D.; Soyer, H.; Leibo, J.Z.; Munos, R.; Blundell, C.; Kumaran, D.; Botvinick, M. Learning to Reinforcement Learn. *arXiv* **2016**, arXiv:1611.05763.
48. Wu, Y.; Mansimov, E.; Grosse, R.B.; Liao, S.; Ba, J. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5279–5288.
49. Levine, S.; Koltun, V. Guided policy search. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 16 June–21 June 2013; pp. 1–9.
50. Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; Moritz, P. Trust region policy optimization. In Proceedings of the International Conference on Machine Learning, Lille, France, 6 July–11 July 2015; pp. 1889–1897.
51. Kakade, S.; Langford, J. Approximately optimal approximate reinforcement learning. In Proceedings of the ICML, Sydney, Australia, 8–12 July 2002; Volume 2, pp. 267–274.
52. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
53. 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.
54. Riedmiller, M.; Hafner, R.; Lampe, T.; Neunert, M.; Degraeve, J.; Van de Wiele, T.; Mnih, V.; Heess, N.; Springenberg, J.T. Learning by Playing-Solving Sparse Reward Tasks from Scratch. *arXiv* **2018**, arXiv:1802.10567.
55. 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.
56. Levine, S.; Finn, C.; Darrell, T.; Abbeel, P. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.* **2016**, *17*, 1334–1373.
57. Jaderberg, M.; Mnih, V.; Czarnecki, W.M.; Schaul, T.; Leibo, J.Z.; Silver, D.; Kavukcuoglu, K. Reinforcement Learning with Unsupervised Auxiliary Tasks. *arXiv* **2016**, arXiv:1611.05397.
58. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized Experience Replay. *arXiv* **2015**, arXiv:1511.05952.
59. Bengio, Y.; Louradour, J.; Collobert, R.; Weston, J. Curriculum learning. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009; pp. 41–48.
60. Zhang, J.; Tai, L.; Boedecker, J.; Burgard, W.; Liu, M. Neural SLAM. *arXiv* **2017**, arXiv:1706.09520.
61. Florensa, C.; Held, D.; Wulfmeier, M.; Zhang, M.; Abbeel, P. Reverse Curriculum Generation for Reinforcement Learning. *arXiv* **2017**, arXiv:1707.05300.
62. Pathak, D.; Agrawal, P.; Efros, A.A.; Darrell, T. Curiosity-driven exploration by self-supervised prediction. In Proceedings of the International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; Volume 2017.
63. Sukhbaatar, S.; Lin, Z.; Kostrikov, I.; Synnaeve, G.; Szlam, A.; Fergus, R. Intrinsic Motivation and Automatic Curricula via Asymmetric Self-Play. *arXiv* **2017**, arXiv:1703.05407.
64. Fortunato, M.; Azar, M.G.; Piot, B.; Menick, J.; Osband, I.; Graves, A.; Mnih, V.; Munos, R.; Hassabis, D.; Pietquin, O.; et al. Noisy Networks for Exploration. *arXiv* **2017**, arXiv:1706.10295.
65. Plappert, M.; Houthoofd, R.; Dhariwal, P.; Sidor, S.; Chen, R.Y.; Chen, X.; Asfour, T.; Abbeel, P.; Andrychowicz, M. Parameter Space Noise for Exploration. *arXiv* **2017**, arXiv:1706.01905.
66. Zhu, Y.; Mottaghi, R.; Kolve, E.; Lim, J.J.; Gupta, A.; Fei-Fei, 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, 30–31 May 2017; pp. 3357–3364.
67. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
68. Kolve, E.; Mottaghi, R.; Gordon, D.; Zhu, Y.; Gupta, A.; Farhadi, A. AI2-THOR: An Interactive 3d Environment for Visual AI. *arXiv* **2017**, arXiv:1712.05474.
69. Tai, L.; Paolo, G.; Liu, M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 31–36.

70. Chen, Y.F.; Everett, M.; Liu, M.; How, J.P. Socially aware motion planning with deep reinforcement learning. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1343–1350.
71. Long, P.; Fan, T.; Liao, X.; Liu, W.; Zhang, H.; Pan, J. Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning. *arXiv* **2017**, arXiv:1709.10082.
72. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*; The MIT Press: Cambridge, MA, USA, 2001.
73. Gupta, S.; Davidson, J.; Levine, S.; Sukthankar, R.; Malik, J. Cognitive Mapping and Planning for Visual Navigation. *arXiv* **2017**, 3, arXiv:1702.03920.
74. Gupta, S.; Fouhey, D.; Levine, S.; Malik, J. Unifying Map and Landmark Based Representations for Visual Navigation. *arXiv* **2017**, arXiv:1712.08125.
75. Parisotto, E.; Salakhutdinov, R. Neural Map: Structured Memory for Deep Reinforcement Learning. *arXiv* **2017**, arXiv:1702.08360.
76. Kümmerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; Burgard, W. G<sub>2</sub>O: A general framework for graph optimization. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 3607–3613.
77. Parisotto, E.; Chablot, D.S.; Zhang, J.; Salakhutdinov, R. Global Pose Estimation with an Attention-Based Recurrent Network. *arXiv* **2018**, arXiv:1802.06857.
78. Schaul, T.; Horgan, D.; Gregor, K.; Silver, D. Universal value function approximators. In Proceedings of the International Conference on Machine Learning, Lille, France, 6 July–11 July 2015; pp. 1312–1320.
79. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. *arXiv* **2017**, arXiv:1703.10593.
80. Khan, A.; Zhang, C.; Atanasov, N.; Karydis, K.; Kumar, V.; Lee, D.D. Memory Augmented Control Networks. *arXiv* **2017**, arXiv:1709.05706.
81. Bruce, J.; Sünderhauf, N.; Mirowski, P.; Hadsell, R.; Milford, M. One-Shot Reinforcement Learning for Robot Navigation with Interactive Replay. *arXiv* **2017**, arXiv:1711.10137.
82. Chablot, D.S.; Parisotto, E.; Salakhutdinov, R. Active Neural Localization. *arXiv* **2018**, arXiv:1801.08214.
83. Savinov, N.; Dosovitskiy, A.; Koltun, V. Semi-Parametric Topological Memory for Navigation. *arXiv* **2018**, arXiv:1803.00653.
84. Heess, N.; Sriram, S.; Lemmon, J.; Merel, J.; Wayne, G.; Tassa, Y.; Erez, T.; Wang, Z.; Eslami, A.; Riedmiller, M.; et al. Emergence of Locomotion Behaviours in Rich Environments. *arXiv* **2017**, arXiv:1707.02286.
85. Calisti, M.; Giorelli, M.; Levy, G.; Mazzolai, B.; Hochner, B.; Laschi, C.; Dario, P. An octopus-bioinspired solution to movement and manipulation for soft robots. *Bioinspir. Biomim.* **2011**, *6*, 036002. [[CrossRef](#)] [[PubMed](#)]
86. Martinez, R.V.; Branch, J.L.; Fish, C.R.; Jin, L.; Shepherd, R.F.; Nunes, R.M.D.; Suo, Z.; Whitesides, G.M. Robotic tentacles with three-dimensional mobility based on flexible elastomers. *Adv. Mater.* **2013**, *25*, 205–212. [[CrossRef](#)] [[PubMed](#)]
87. Caldera, S. Review of Deep Learning Methods in Robotic Grasp Detection. *Multimodal Technol. Interact.* **2018**, *2*, 57. [[CrossRef](#)]
88. Zhou, J.; Chen, S.; Wang, Z. A Soft-Robotic Gripper with Enhanced Object Adaptation and Grasping Reliability. *IEEE Robot. Autom. Lett.* **2017**, *2*, 2287–2293. [[CrossRef](#)]
89. Finn, C.; Tan, X.Y.; Duan, Y.; Darrell, T.; Levine, S.; Abbeel, P. Deep Spatial Autoencoders for Visuomotor Learning. *arXiv* **2015**, arXiv:1509.06113.
90. Tzeng, E.; Devin, C.; Hoffman, J.; Finn, C.; Peng, X.; Levine, S.; Saenko, K.; Darrell, T. Towards Adapting Deep Visuomotor Representations from Simulated to Real Environments. *arXiv* **2015**, arXiv:1511.07111v3.
91. Fu, J.; Levine, S.; Abbeel, P. One-shot learning of manipulation skills with online dynamics adaptation and neural network priors. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4019–4026.
92. Kumar, V.; Todorov, E.; Levine, S. Optimal control with learned local models: Application to dexterous manipulation. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), New York, NY, USA, 16–20 May 2016; pp. 378–383.
93. Gupta, A.; Eppner, C.; Levine, S.; Abbeel, P. Learning dexterous manipulation for a soft robotic hand from human demonstrations. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 3786–3793.

94. Popov, I.; Heess, N.; Lillicrap, T.; Hafner, R.; Barth-Maron, G.; Vecerik, M.; Lampe, T.; Tassa, Y.; Erez, T.; Riedmiller, M. Data-Efficient Deep Reinforcement Learning for Dexterous manipulation. *arXiv* **2017**, arXiv:1704.03073.
95. Prituja, A.; Banerjee, H.; Ren, H. Electromagnetically Enhanced Soft and Flexible Bend Sensor: A Quantitative Analysis with Different Cores. *IEEE Sens. J.* **2018**, *18*, 3580–3589. [[CrossRef](#)]
96. Sun, J.Y.; Zhao, X.; Illeperuma, W.R.; Chaudhuri, O.; Oh, K.H.; Mooney, D.J.; Vlassak, J.J.; Suo, Z. Highly stretchable and tough hydrogels. *Nature* **2012**, *489*, 133–136. [[CrossRef](#)]
97. Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep Domain Confusion: Maximizing for Domain Invariance. *arXiv* **2014**, arXiv:1412.3474.
98. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems (NIPS 2014), Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
99. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**, arXiv:1511.06434.
100. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein gan. *arXiv* **2017**, arXiv:1701.07875.
101. Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.Y.; Isola, P.; Saenko, K.; Efros, A.A.; Darrell, T. Cycada: Cycle-Consistent Adversarial Domain Adaptation. *arXiv* **2017**, arXiv:1711.03213.
102. Doersch, C. Tutorial on Variational Autoencoders. *arXiv* **2016**, arXiv:1606.05908v2.
103. Szabó, A.; Hu, Q.; Portenier, T.; Zwicker, M.; Favaro, P. Challenges in Disentangling Independent Factors of Variation. *arXiv* **2017**, arXiv:1711.02245v1.
104. Mathieu, M.; Zhao, J.J.; Sprechmann, P.; Ramesh, A.; LeCun, Y. Disentangling factors of variation in deep representations using adversarial training. In Proceedings of the NIPS 2016, Barcelona, Spain, 5–10 December 2016.
105. Bousmalis, K.; Irpan, A.; Wohlhart, P.; Bai, Y.; Kelcey, M.; Kalakrishnan, M.; Downs, L.; Ibarz, J.; Pastor, P.; Konolige, K.; et al. Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping. *arXiv* **2017**, arXiv:1709.07857.
106. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 23–30.
107. Peng, X.B.; Andrychowicz, M.; Zaremba, W.; Abbeel, P. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. *arXiv* **2017**, arXiv:1710.06537.
108. Rusu, A.A.; Vecerik, M.; Rothörl, T.; Heess, N.; Pascanu, R.; Hadsell, R. Sim-to-Real Robot Learning from Pixels with Progressive Nets. *arXiv* **2016**, arXiv:1610.04286.
109. Zhang, J.; Tai, L.; Xiong, Y.; Liu, M.; Boedecker, J.; Burgard, W. Vr Goggles for Robots: Real-to-Sim Domain Adaptation for Visual Control. *arXiv* **2018**, arXiv:1802.00265.
110. Ruder, M.; Dosovitskiy, A.; Brox, T. Artistic style transfer for videos and spherical images. *Int. J. Comput. Vis.* **2018**, *126*, 1199–1219. [[CrossRef](#)]
111. Koenig, N.P.; Howard, A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. *IROS. Citeseer* **2004**, *4*, 2149–2154.
112. Maddern, W.; Pascoe, G.; Linegar, C.; Newman, P. 1 year, 1000 km: The Oxford RobotCar dataset. *Int. J. Robot. Res.* **2017**, *36*, 3–15. [[CrossRef](#)]
113. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. *arXiv* **2017**, arXiv:1711.03938.
114. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected Crfs. *arXiv* **2014**, arXiv:1412.7062.
115. Yang, L.; Liang, X.; Xing, E. Unsupervised Real-to-Virtual Domain Unification for End-to-End Highway Driving. *arXiv* **2018**, arXiv:1801.03458.
116. Uesugi, K.; Shimizu, K.; Akiyama, Y.; Hoshino, T.; Iwabuchi, K.; Morishima, K. Contractile performance and controllability of insect muscle-powered bioactuator with different stimulation strategies for soft robotics. *Soft Robot.* **2016**, *3*, 13–22. [[CrossRef](#)]
117. Niiyama, R.; Sun, X.; Sung, C.; An, B.; Rus, D.; Kim, S. Pouch Motors: Printable Soft Actuators Integrated with Computational Design. *Soft Robot.* **2015**, *2*, 59–70. [[CrossRef](#)]

118. Gul, J.Z.; Sajid, M.; Rehman, M.M.; Siddiqui, G.U.; Shah, I.; Kim, K.C.; Lee, J.W.; Choi, K.H. 3D printing for soft robotics—A review. *Sci. Technol. Adv. Mater.* **2018**, *19*, 243–262. [[CrossRef](#)] [[PubMed](#)]
119. Umedachi, T.; Vikas, V.; Trimmer, B. Highly deformable 3-D printed soft robot generating inching and crawling locomotions with variable friction legs. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 4590–4595.
120. Mutlu, R.; Tawk, C.; Alici, G.; Sariyildiz, E. A 3D printed monolithic soft gripper with adjustable stiffness. In Proceedings of the IECON 2017—43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 29 October–1 November 2017; pp. 6235–6240.
121. Lu, N.; Hyeong Kim, D. Flexible and Stretchable Electronics Paving the Way for Soft Robotics. *Soft Robot.* **2014**, *1*, 53–62 [[CrossRef](#)]
122. Rohmer, E.; Singh, S.P.; Freese, M. V-REP: A versatile and scalable robot simulation framework. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–8 November 2013; pp. 1321–1326.
123. Shah, S.; Dey, D.; Lovett, C.; Kapoor, A. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*; Springer: Berlin, Germany, 2018; pp. 621–635.
124. Pan, X.; You, Y.; Wang, Z.; Lu, C. Virtual to Real Reinforcement Learning for Autonomous Driving. *arXiv* **2017**, arXiv:1704.03952.
125. Savva, M.; Chang, A.X.; Dosovitskiy, A.; Funkhouser, T.; Koltun, V. MINOS: Multimodal Indoor Simulator for Navigation in Complex Environments. *arXiv* **2017**, arXiv:1712.03931.
126. Wu, Y.; Wu, Y.; Gkioxari, G.; Tian, Y. Building Generalizable Agents with a Realistic and Rich 3D Environment. *arXiv* **2018**, arXiv:1801.02209.
127. Coevoet, E.; Bieze, T.M.; Largilliere, F.; Zhang, Z.; Thieffry, M.; Sanz-Lopez, M.; Carrez, B.; Marchal, D.; Goury, O.; Dequidt, J.; et al. Software toolkit for modeling, simulation, and control of soft robots. *Adv. Robot.* **2017**, *31*, 1208–1224. [[CrossRef](#)]
128. Duriez, C.; Coevoet, E.; Largilliere, F.; Bieze, T.M.; Zhang, Z.; Sanz-Lopez, M.; Carrez, B.; Marchal, D.; Goury, O.; Dequidt, J. Framework for online simulation of soft robots with optimization-based inverse model. In Proceedings of the 2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN), San Francisco, CA, USA, 13–16 December 2016; pp. 111–118.
129. Olaya, J.; Pintor, N.; Avilés, O.F.; Chaparro, J. Analysis of 3 RPS Robotic Platform Motion in SimScape and MATLAB GUI Environment. *Int. J. Appl. Eng. Res.* **2017**, *12*, 1460–1468.
130. Yekutieli, Y.; Sagiv-Zohar, R.; Aharonov, R.; Engel, Y.; Hochner, B.; Flash, T. Dynamic model of the octopus arm. I. Biomechanics of the octopus reaching movement. *J. Neurophysiol.* **2005**, *94*, 1443–1458. [[CrossRef](#)] [[PubMed](#)]
131. Ratliff, N.D.; Bagnell, J.A.; Srinivasa, S.S. Imitation learning for locomotion and manipulation. In Proceedings of the 2007 7th IEEE-RAS International Conference on Humanoid Robots, Pittsburgh, PA, USA, 29 November–1 December 2007; pp. 392–397.
132. Langsfeld, J.D.; Kaipa, K.N.; Gentili, R.J.; Reggia, J.A.; Gupta, S.K. Towards Imitation Learning of Dynamic Manipulation Tasks: A Framework to Learn from Failures. Available online: <https://pdfs.semanticscholar.org/5e1a/d502aeb5a800f458390ad1a13478d0fbd39b.pdf> (accessed on 18 January 2019).
133. Ross, S.; Gordon, G.; Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 627–635.
134. 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.
135. Tai, L.; Li, S.; Liu, M. A deep-network solution towards model-less obstacle avoidance. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 2759–2764.
136. Giusti, A.; Guzzi, J.; Ciresan, D.C.; He, F.L.; Rodríguez, J.P.; Fontana, F.; Faessler, M.; Forster, C.; Schmidhuber, J.; Di Caro, G.; et al. A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots. *IEEE Robot. Autom. Lett.* **2016**, *1*, 661–667. [[CrossRef](#)]
137. Codevilla, F.; Müller, M.; Dosovitskiy, A.; López, A.; Koltun, V. End-to-End Driving via Conditional Imitation Learning. *arXiv* **2017**, arXiv:1710.02410.

138. Duan, Y.; Andrychowicz, M.; Stadie, B.C.; Ho, J.; Schneider, J.; Sutskever, I.; Abbeel, P.; Zaremba, W. One-Shot Imitation Learning. In Proceedings of the NIPS, Long Beach, CA, USA, 4–9 December 2017.
139. Finn, C.; Yu, T.; Zhang, T.; Abbeel, P.; Levine, S. One-Shot Visual Imitation Learning via Meta-Learning. *arXiv* **2017**, arXiv:1709.04905.
140. Finn, C.; Abbeel, P.; Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *arXiv* **2017**, arXiv:1703.03400.
141. Eitel, A.; Hauff, N.; Burgard, W. Learning to Singulate Objects Using a Push Proposal Network. *arXiv* **2017**, arXiv:1707.08101.
142. Ziebart, B.D.; Maas, A.L.; Bagnell, J.A.; Dey, A.K. Maximum Entropy Inverse Reinforcement Learning. In Proceedings of the AAAI, Chicago, IL, USA, 13–17 July 2008; Volume 8, pp. 1433–1438.
143. 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, 16–20 May 2016; pp. 2889–2895.
144. Pfeiffer, M.; Schwesinger, U.; Sommer, H.; Galceran, E.; Siegwart, R. Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 2096–2101.
145. Kretzschmar, H.; Spies, M.; Sprunk, C.; Burgard, W. Socially compliant mobile robot navigation via inverse reinforcement learning. *Int. J. Robot. Res.* **2016**, *35*, 1289–1307. [[CrossRef](#)]
146. Wulfmeier, M.; Ondruska, P.; Posner, I. Maximum Entropy Deep Inverse Reinforcement Learning. *arXiv* **2015**, arXiv:1507.04888.
147. Ho, J.; Ermon, S. Generative adversarial imitation learning. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 4565–4573.
148. Baram, N.; Ansel, O.; Mannor, S. Model-Based Adversarial Imitation Learning. *arXiv* **2016**, arXiv:1612.02179.
149. 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 in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5320–5329.
150. Li, Y.; Song, J.; Ermon, S. Inferring the Latent Structure of Human Decision-Making from Raw Visual Inputs. *arXiv* **2017**, arXiv:1604.07316.
151. Tai, L.; Zhang, J.; Liu, M.; Burgard, W. Socially-Compliant Navigation through Raw Depth Inputs with Generative Adversarial Imitation Learning. *arXiv* **2017**, arXiv:1710.02543.
152. Stadie, B.C.; Abbeel, P.; Sutskever, I. Third-Person Imitation Learning. *arXiv* **2017**, arXiv:1703.01703.
153. DelPreto, J.; MacCurdy, R.B.; Rus, D. Exploration of underwater life with an acoustically controlled soft robotic fish. *Sci. Robot.* **2018**, *3*. [[CrossRef](#)]
154. Katzschmann, R.K.; Marchese, A.D.; Rus, D. Hydraulic Autonomous Soft Robotic Fish for 3D Swimming. In Proceedings of the ISER, Marrakech and Essaouira, Morocco, 15–18 June 2014.
155. Katzschmann, R.K.; de Maille, A.; Dorhout, D.L.; Rus, D. Cyclic hydraulic actuation for soft robotic devices. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 3048–3055.
156. DelPreto, J.; Katzschmann, R.K.; MacCurdy, R.B.; Rus, D. A Compact Acoustic Communication Module for Remote Control Underwater. In Proceedings of the WUWNet, Washington, DC, USA, 22–24 October 2015.
157. Marchese, A.D.; Onal, C.D.; Rus, D. Towards a Self-contained Soft Robotic Fish: On-Board Pressure Generation and Embedded Electro-permanent Magnet Valves. In Proceedings of the ISER, Quebec City, QC, Canada, 17–21 June 2012.
158. Narang, Y.S.; Degirmenci, A.; Vlassak, J.J.; Howe, R.D. Transforming the Dynamic Response of Robotic Structures and Systems Through Laminar Jamming. *IEEE Robot. Autom. Lett.* **2018**, *3*, 688–695. [[CrossRef](#)]
159. Narang, Y.S.; Vlassak, J.J.; Howe, R.D. Mechanically Versatile Soft Machines Through Laminar Jamming. *Adv. Funct. Mater.* **2017**, *28*, 1707136. [[CrossRef](#)]
160. Kim, T.; Yoon, S.J.; Park, Y.L. Soft Inflatable Sensing Modules for Safe and Interactive Robots. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3216–3223. [[CrossRef](#)]
161. Qi, R.; Lam, T.L.; Xu, Y. Mechanical design and implementation of a soft inflatable robot arm for safe human-robot interaction. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 3490–3495.

162. Zeng, H.; Wani, O.M.; Wasylczyk, P.; Priimagi, A. Light-Driven, Caterpillar-Inspired Miniature Inching Robot. *Macromol. Rapid Commun.* **2018**, *39*, 1700224. [[CrossRef](#)]
163. Banerjee, H.; Ren, H. Optimizing double-network hydrogel for biomedical soft robots. *Soft Robot.* **2017**, *4*, 191–201. [[CrossRef](#)]
164. Henderson, P.; Islam, R.; Bachman, P.; Pineau, J.; Precup, D.; Meger, D. Deep Reinforcement Learning that Matters. *arXiv* **2017**, arXiv:1709.06560.
165. Vecerík, M.; Hester, T.; Scholz, J.; Wang, F.; Pietquin, O.; Piot, B.; Heess, N.; Rothörl, T.; Lampe, T.; Riedmiller, M.A. Leveraging Demonstrations for Deep Reinforcement Learning on Robotics Problems with Sparse Rewards. *arXiv* **2017**, arXiv:1707.08817v1.
166. Nair, A.; McGrew, B.; Andrychowicz, M.; Zaremba, W.; Abbeel, P. Overcoming Exploration in Reinforcement Learning with Demonstrations. *arXiv* **2017**, arXiv:1709.10089.
167. Gao, Y.; Lin, J.; Yu, F.; Levine, S.; Darrell, T. Reinforcement Learning from Imperfect Demonstrations. *arXiv* **2018**, arXiv:1802.05313.
168. Zhu, Y.; Wang, Z.; Merel, J.; Rusu, A.; Erez, T.; Cabi, S.; Tunyasuvunakool, S.; Kramár, J.; Hadsell, R.; de Freitas, N.; et al. Reinforcement and Imitation Learning for Diverse Visuomotor Skills. *arXiv* **2018**, arXiv:1802.09564.
169. Nichol, A.; Schulman, J. Reptile: A Scalable Metalearning Algorithm. *arXiv* **2018**, arXiv:1803.02999.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).