



HAL
open science

Deep reinforcement learning for the control of robotic manipulation: a focussed mini-review

Rongrong Liu, Florent Nageotte, Philippe Zanne, Michel de Mathelin, Birgitta Dresp

► To cite this version:

Rongrong Liu, Florent Nageotte, Philippe Zanne, Michel de Mathelin, Birgitta Dresp. Deep reinforcement learning for the control of robotic manipulation: a focussed mini-review. *Robotics*, 2021, 10 (1), pp.22. 10.3390/robotics10010022 . hal-03140078

HAL Id: hal-03140078

<https://hal.science/hal-03140078v1>

Submitted on 31 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Article

Deep reinforcement learning for the control of robotic manipulation: a focussed mini-review

Rongrong Liu ^{1,*} , Florent Nageotte ¹, Philippe Zanne ¹ Michel de Mathelin ¹ and Birgitta Dresp-Langley ^{2,*}

¹ ICube Lab Robotics Department Strasbourg University UMR 7357 CNRS; rong.rong.liu@unistra.fr; Nageotte@unistra.fr; zanne.philippe@unistra.fr; demathelin@unistra.fr

² ICube Lab UMR 7357 Centre National de la Recherche Scientifique CNRS; birgitta.dresp@unistra.fr

* Correspondence: birgitta.dresp@unistra.fr; rong.rong.liu@unistra.fr

Version January 18, 2021 submitted to Robotics

Abstract: Deep learning has provided new ways of manipulating, processing and analyzing data. It sometimes may achieve results comparable to, or surpassing human expert performance, and has become a source of inspiration in the era of artificial intelligence. Another subfield of machine learning named reinforcement learning, tries to find an optimal behavior strategy through interactions with the environment. Combining deep learning and reinforcement learning permits resolving critical issues relative to the dimensionality and scalability of data in tasks with sparse reward signals, such as robotic manipulation and control tasks, that neither method permits resolving when applied on its own. In this paper, we present recent significant progress of deep reinforcement learning algorithms, which try to tackle the problems for the application in the domain of robotic manipulation control, such as sample efficiency and generalization. Despite of these continuous improvements, currently the challenges of learning robust and versatile manipulation skills for robots with deep reinforcement learning are still far from being resolved for real world applications.

Keywords: Deep learning; Artificial intelligence; Machine learning; Reinforcement learning; Deep reinforcement learning; Robotic manipulation control; Sample efficiency; Generalization

1. Introduction

Robots are originally designed to assist or replace humans by performing repetitive and/or dangerous tasks which humans prefer not to do, or are unable to do because of physical limitations imposed by extreme environments. Such would include the limited accessibility of narrow, long pipes underground, anatomical locations in specific minimally invasive surgery procedure, objects at the bottom of the sea, for example. With the continuous developments in mechanics, sensing technology [1], intelligent control and other modern technologies, robots have improved autonomy capabilities and are more dexterous. Nowadays, commercial and industrial robots are in widespread use with lower long-term cost and greater accuracy and reliability, in the fields like manufacturing, assembly, packing, transport, surgery, earth and space exploration, etc.

There are a variety of different types of robots available, which can be grouped into several categories depending on their movement, Degrees of Freedom (DoF), and function. Articulated robots, are among the most common robots used today. They look like a human arm and that is why they are also called robotic arm or manipulator arm [2]. In some context, a robotic arm may also refer to a part of a more complex robot. A robotic arm can be described as a chain of links that are moved by joints which are actuated by motors. We will start from a brief explanation of these mechanical components of a typical robotic manipulator [3] [4]. Figure 1 shows the schematic diagram of a simple two-joint robotic arm mounted on a stationary base on the floor .

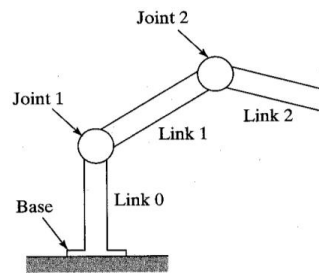


Figure 1. Simplified schematic diagram of mechanical components of a two-joint robotic arm

- 33 • Joints are similar to joints in the human body, which provide relative motion between two parts
34 of the body. In robotic field, each joint is a different axis and provides an additional DoF of
35 controlled relative motion between adjacent links, as shown in Figure 1. In nearly all cases, the
36 number of degrees of freedom is equal to the number of joints [5].
- 37 • An end-effector, as the name suggests, is an accessory device or tool which is attached at the
38 end of the chain, and actually accomplishes the intended task. The simplest end-effector is the
39 gripper, which is capable of opening and closing for gripping objects, but it can also be designed
40 as a screw driver, a brush, a water jet, a thruster, or any mechanical device, according to different
41 applications. An end-effector can also be called a robotic hand.
- 42 • Links are the rigid or nearly rigid components that connect either the base, joints or end effector,
43 and bear the load of the chain.
- 44 • An actuator is a device that converts electrical, hydraulic, or pneumatic energy into robot motion.

45 Currently the control sequence of a robotic manipulator is mainly achieved by solving inverse
46 kinematic equations to move or position the end effector with respect to the fixed frame of reference [6]
47 [7]. The information is stored in memory by a programmable logic controller for fixed robotic tasks [8].
48 Robots can be controlled in open-loop or with an exteroceptive feedback. The open-loop control does
49 not have external sensors or environment sensing capability, but heavily relies on highly structured
50 environments that are very sensitively calibrated. If a component is slightly shifted, the control system
51 may have to stop and to be recalibrated. Under this strategy, the robot arm performs by following a
52 series of positions in memory, and moving to them at various times in their programming sequence.
53 In some more advanced robotic systems, exteroceptive feedback control is employed, through the
54 use of monitoring sensors, force sensors, even vision or depth sensors, that continually monitor the
55 robot's axes or end-effector, and associated components for position and velocity. The feedback is then
56 compared to information stored to update the actuator command so as to achieve the desired robot
57 behavior. Either auxiliary computers or embedded microprocessors are needed to perform interface
58 with these associated sensors and the required computational functions. These two traditional control
59 scenarios are both heavily dependent on hardware-based solutions. For example, conveyor belts and
60 shaker tables, are commonly used in order to constrain the situation physically.

61 With the advancements in modern technologies in artificial intelligence, such as deep learning,
62 and recent developments in robotics and mechanics, both the research and industrial communities
63 have been seeking more software based control solutions using low-cost sensors, which has less
64 requirements for the operating environment and calibration. The key is to make minimal but effective
65 hardware choices and focus on robust algorithms and software. Instead of hard-coding directions
66 to coordinate all the joints, the control policy could be obtained by learning and then be updated
67 accordingly. Deep Reinforcement Learning (DRL) is among the most promising algorithms for this
68 purpose because no predefined training dataset is required, which ideally suits robotic manipulation
69 and control tasks, as illustrated in Table 1. A reinforcement learning approach might use input from a
70 robotic arm experiment, with different sequences of movements, or input from simulation models.
71 Either type of dynamically generated experiential data can be collected, and used to train a Deep
72 Neural Network (DNN) by iteratively updating specific policy parameters of a control policy network.

Table 1. Comparison between traditional control and DRL based control expectation

	traditional control	DRL based control expectation
control solution	hardware based	software based
monitoring sensor	expensive	low-cost
environment requirement	structured	unstructured situations
hardware calibration	sensitive to calibration	tolerate to calibration
control algorithm	hand coding required	data driven

73 This review paper tries to provide a brief and self-contained review of DRL in the research of
 74 robotic manipulation control. We will start with a brief introduction of deep learning, reinforcement
 75 learning, and DRL in the second section. The recent progress of robotic manipulation control with the
 76 DRL based methods will be then discussed in the third section. What need to mention here is that,
 77 we can not cover all the brilliant algorithms in detail in a short paper. For the algorithms mentioned
 78 here, one still need to refer to those original papers for the detailed information. Finally, we follow the
 79 discussion and present other real-world challenges of utilizing DRL in robotic manipulation control in
 80 forth section, with a conclusion of our work in the last section.

81 2. Deep reinforcement learning

82 In this part, we will start from deep learning and reinforcement learning, to better illustrate their
 83 combination version, DRL.

84 2.1. Deep learning

85 Deep learning is quite popular in the family of machine learning, with its outstanding performance
 86 in a variety of domains, not only in classical computer vision tasks, but also in in many other
 87 practical applications, to just name a few, natural language processing, social network filtering,
 88 machine translation, bioinformatics, material inspection and board games, where these deep-learning
 89 based methods have produced results comparable to, and in some cases surpassing human expert
 90 performance. Deep learning has changed the way we process, analyze and manipulate data.

91 The adjective "deep" in deep learning comes from the use of multiple layers in the network. Figure
 92 2 demonstrates a simple deep learning architecture with basic fully connected strategy. A general
 93 comparison is conducted in Table 2 between deep learning and traditional machine learning. With
 94 deep learning, the raw data like images are directly fed into a deep neural network multiple layers
 95 that progressively extract higher-level features, while with traditional machine learning, the relevant
 96 features of input data are manually extracted by experts. Besides, deep learning often requires a large
 97 amount of data to reach optimal results, thus it's also computationally intensive, accordingly.

Table 2. Comparison between traditional machine learning and deep learning

	traditional machine learning	deep learning
dataset requirement	performs well with small dataset	requires large dataset
accuracy	accuracy plateaus	excellent performance potential
feature extraction	selected manually	learned automatically
algorithm structure	simple model	multi-layer model
model training time	quick to train a model	computationally intensive
hardware requirement	works with not powerful hardware	high-performance computer

98 Deep models can be interpreted as artificial neural networks with deep structures. The idea
 99 of artificial neural networks is not something new, which can date back to 1940s [9]. In the
 100 following years, the research community witnessed many important milestones with perceptrons [10],
 101 backpropagation algorithm [11] [12], Rectified Linear Unit, or ReLU [13], Max-pooling [14], dropout
 102 [15], batch normalization [16], etc. It is all these continuous algorithmic improvements, together with
 103 the emergence of large-scale training data and the fast development of high performance parallel

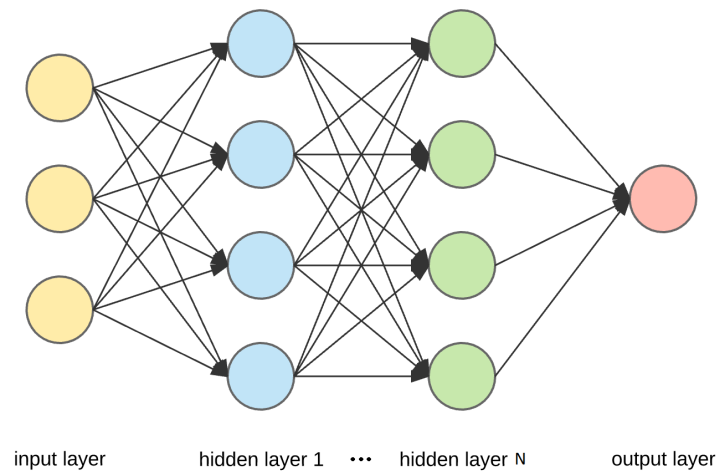


Figure 2. A simple deep learning architecture

104 **computing systems**, such as Graphics Processing Units (GPUs) that allow deep learning to prosper
 105 nowadays [17].

106 The first great success for deep learning is based on convolutional neural network for classification
 107 in 2012 [18]. It applies hundreds of thousands data-label pairs iteratively to train the parameters with
 108 loss computation and backpropagation. Although this technique has been improved continuously
 109 and rapidly since it took off, and is now one of the most popular deep learning structures, it is not
 110 quite suitable for robotic manipulation control, as it is too time-consuming to obtain large number of
 111 images of joints angles with labeled data to train the model. Indeed, there are some researches using
 112 convolutional neural network to learn the motor torques needed to control the robot with raw RGB
 113 video images [19]. However, a more promising and interesting idea is using DRL, as we will discuss
 114 hereafter.

115 2.2. Reinforcement learning

116 Reinforcement learning [20] is a subfield of machine learning, concerned with how to find
 117 an optimal behavior strategy to maximize the outcome through trial and error dynamically and
 118 autonomously, which is quite similar with the intelligence of human and animals, as the general
 119 definition of intelligence is the ability to perceive or infer information, and to retain it as knowledge
 120 to be applied towards adaptive behaviors in the environment. This autonomous self-teaching
 121 methodology is actively studied in many domains, like game theory, control theory, operations
 122 research, information theory, system optimization, recommendation system and statistics [21].

123 Figure 3 illustrates the universal model of reinforcement learning, which is biologically plausible,
 124 as it is inspired by human learning through punishment or reward due to state changes in the
 125 environment, which are either favorable (reinforcing) to certain behaviors/actions, or unfavourable
 126 (suppressing). Natural reinforcement learning is driven by the evolutionary pressure of optimal
 127 behavioral adaptation to environmental constraints.

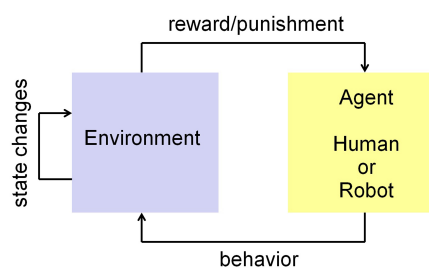


Figure 3. Universal model of reinforcement learning

When an agent is in a state, it chooses an action according to its current policy and then it receives a reward from the environment for executing that action. By learning from this reward, it transitions to a new state, chooses a new action and then iterates through this process. In order to make it even easier to understand, one can compare reinforcement learning to the structure of how we play a video game, in which the character, namely the agent, engages in a series of trials, or actions, to obtain the highest score, which is reward.

Reinforcement learning is different from supervised learning, where a training set of labeled examples is available. In interactive problems like robot control domain using reinforcement learning, it is often impractical to obtain examples of desired behavior that are both correct and representative of all the situations in which the agent has to act. Instead of labels, we get rewards which in general are weaker signals. Reinforcement learning is not a kind of unsupervised learning, which is typically about finding structure hidden in collections of unlabeled data. In reinforcement learning the agent has to learn to behave in the environment based only on those sparse and time-delayed rewards, instead of trying to find hidden structure. Therefore, reinforcement learning can be considered as a third machine learning paradigm, alongside supervised learning and unsupervised learning and perhaps other future paradigms as well [22].

2.3. Deep reinforcement learning

As the name suggests, DRL emerges from reinforcement learning and deep learning, and can be regarded as the bridge between conventional machine learning and true artificial intelligence, as illustrated in Figure 4. It combines both the technique of giving rewards based on actions from reinforcement learning, and the idea of using a neural network for learning feature representations from deep learning. Traditional reinforcement learning is limited to domains with simple state representations, while DRL makes it possible for agents to make decisions from high-dimensional and unstructured input data [23] using neural networks to represent policies. In the past few years, research in DRL has been highly active with a significant amount of progress, along with the rising interest in deep learning.

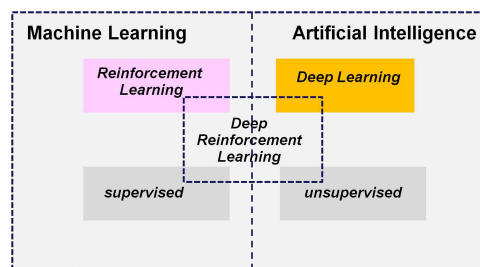


Figure 4. Deep reinforcement learning

Firstly, DRL has gained a lot of attraction, especially due to its well-known achievement in games. Beginning around 2013, DeepMind showed impressive learning results in Atari video games at human level proficiency with no hand coded features using unprocessed pixels for input [24] [25], which can be regarded as the creation of this subfield. Another milestone was in 2016 when AlphaGo [26] first beat a human professional player of Go, which is a game from ancient China. This computer program was improved by AlphaZero [27] in 2017, together with its efficiency also in chess and shogi. In 2019, Pluribus [28] showed its success over top human professionals in multiplayer poker, and OpenAI Five [29] beat the previous world champions in a Dota 2 demonstration match.

Apart from the field of games, it has large potential in other domains, including but not limited to, robotics [30], natural language processing [31], computer vision [32], transportation [33], finance [34] and healthcare [35]. Many exciting breakthroughs of this research have been published by both of giant companies, which include Google Brain, DeepMind and Facebook, and top academic labs such

166 as in Berkeley, Stanford and Carnegie Mellon University, together with some independent non-profit
167 research organizations like openAI and some other industrially focused companies.

168 The most commonly used DRL algorithms can be categorized in value-based methods, policy
169 gradient methods and model-based methods. The value-based methods construct a value function
170 for defining a policy, which is based on the Q-learning algorithm [36] using the Bellman equation [37]
171 and its variant, the fitted Q-learning [38] [39]. The Deep Q-Network (DQN) algorithm used with great
172 success in [25] is the representative of this class, followed by various extensions, such as double DQN
173 [40], Distributional DQN [41] [42], etc. A combination of these improvements has been studied in [43]
174 with a state-of-the-art performance on the Atari 2600 benchmark, both in terms of data efficiency and
175 final performance.

176 However, the DQN-based approaches are limited to problems with discrete and low-dimensional
177 action spaces, and deterministic policies, while policy gradient methods are able to work with
178 continuous action spaces and can also represent stochastic policies. Thanks to variants of stochastic
179 gradient ascent with respect to the policy parameters, policy gradient methods are developed to
180 find a neural network parameterized policy to maximize the expected cumulative reward [44]. Like
181 other policy-based methods, policy gradient methods typically require an estimate of a value function
182 for the current policy and a sample efficient approach is to use an actor-critic architecture that can
183 work with off-policy data. The Deep Deterministic Policy Gradient (DDPG) algorithm [45] [46] is a
184 representation of this type of methods. There are also some researchers working on combining policy
185 gradient methods with Q-learning [47].

186 Both value-based and policy-based methods do not make use of any model of the environment
187 and are also called model-free methods, which limits their sample efficiency. On the contrary, in
188 the model-based methods, a model of the environment is either explicitly given or learned from
189 experience by the function approximators [48] [49] in conjunction with a planning algorithm. In order
190 to obtain advantages from both sides, there are many researches available integrating model-free and
191 model-based elements [50] [51] [52], which are among the key areas for the future development of
192 DRL algorithms [53].

193 3. Deep reinforcement learning in robotic manipulation control

194 In this section, the recent progress of DRL in the domain of robotic manipulation control will be
195 discussed. Two of the most important challenges here concern sample efficiency and generalization.
196 The goal of DRL in the context of robotic manipulation control is to train a deep policy neural network,
197 like in Figure 2, to detect the optimal sequence of commands for accomplishing the task. As illustrated
198 in Figure 5, the input is the current state, which can include the angles of joints of the manipulator,
199 position of the end effector, and their derivative information, like velocity and acceleration. Moreover,
200 the current pose of target objects can also be counted in the current state, together with the state
201 of corresponding sensors if there are some equipped in the environment. The output of this policy
202 network is an action indicating control commands to be implemented to each actuator, such as torques
203 or velocity commands. When the robotic manipulator accomplishes a task, a positive reward will
204 be generated. With these delayed and weak signals, the algorithm is expected to find out the most
205 successful control strategy for the robotic manipulation.

206 3.1. Sample efficiency

207 As we know, in supervised deep learning, a training set of input-output pairs are fed to neural
208 networks to construct an approximation that maps an input to an output [54] [55]. This learned target
209 function can then be used for labeling new examples when a test set is given. The study of sample
210 efficiency for supervised deep learning tries to answer the question of how large a training set is
211 required in order to learn a good approximation to the target concept [56]. Accordingly, for DRL in
212 robotic manipulation control, the study of sample efficiency discusses how much data need to be
213 collected in order to build an optimal policy to accomplish the designed task.

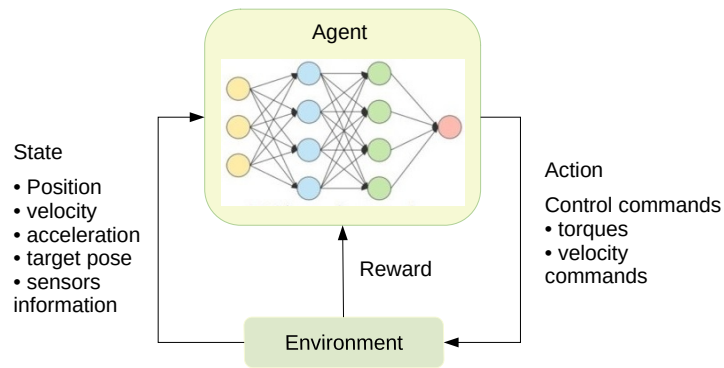


Figure 5. A schematic diagram of robotic manipulation control using DRL

214 The sample efficiency for DRL is considerably more challenging than that for supervised learning
 215 for various reasons [57]. First, the agent can not receive a training set provided by the environment
 216 unilaterally, but information which is determined by both the actions it takes and dynamics of the
 217 environment. Second, although the agent desires to maximize the long-term reward, the agent can
 218 only observe the immediate reward. Additionally, there is no clear boundary between training and test
 219 phases. The time the agent spends trying to improve the policy often comes at the expense of utilizing
 220 this policy, which is often referred to as the exploration-exploitation trade-off [53].

221 Since gathering experiences by interacting with the environment for robots is relatively expensive,
 222 a number of approaches have been proposed in the literature to address sample efficient learning. For
 223 example, in [58], which is the first demonstration of using DRL on a robot, trajectory optimization
 224 techniques and policy search methods with neural networks were applied to achieve reasonable sample
 225 efficient learning. A range of dynamic manipulation behaviors were learned, such as stacking large
 226 Lego blocks at a range of locations, threading wooden rings onto a tight-fitting peg, screwing caps
 227 onto different bottles, assembling a toy airplane by inserting the wheels into a slot. Videos and other
 228 supplementary materials can be found at [http : // rll.berkeley.edu / icra2015gps / index.htm](http://rll.berkeley.edu/icra2015gps/index.htm). Millimeter
 229 level precision can be achieved with dozens of examples by this algorithm, but the knowledge of the
 230 explicit state of the world at training time is required to enable sample efficiency.

231 In [59], a novel technique called Hindsight Experience Replay (HER) was proposed. Each episode
 232 was replayed but with a different goal than the one the agent was trying to achieve. With this clever
 233 strategy for augmenting samples, the policy for the pick-and-place task, which was learned using
 234 only sparse binary rewards, performed well on the physical robot without any finetuning. The video
 235 presenting their experiments is available at [https : // goo.gl / SMrQnI](https://goo.gl/SMrQnI). But this algorithm relies
 236 on special value function approximators, which might not be trivially applicable to every problem.
 237 Besides, this technique can not be extended well with the use of reward shaping.

238 Some other researchers try to achieve sample efficient learning through demonstrations in
 239 imitation learning [60] [61], where a mentor provides demonstrations to replace the random neural
 240 network initialization. [62] was an extension of DDPG algorithm [46] for tasks with sparse rewards,
 241 where both demonstrations and actual interactions were used to fill a replay buffer. Experiments of four
 242 simulation tasks and a real robot clip insertion problem were conducted. A video demonstrating the
 243 performance can be viewed at [https : // www.youtube.com / watch?v = WGJwLfeVN9w](https://www.youtube.com/watch?v=WGJwLfeVN9w). However,
 244 the object location and the explicit states of joints, such as position and velocity, must be provided to
 245 move from simulation to real world, which limits its application to high-dimensional data.

246 Based on the work of generative adversarial imitation learning in [63], [64] used Generative
 247 Adversarial Networks (GANs) [65] to generate an additional training data to solve sample complexity
 248 problem, by proposing a multi-modal imitation learning framework that was able to handle
 249 unstructured demonstrations of different skills. The performance of this framework was evaluated in
 250 simulation for several tasks, such as reacher and gripper-pusher. The video of simulated experiments

251 is available at <http://sites.google.com/view/nips17intentiongan>. Like most GANs techniques, it is
252 quite difficult to train and many samples are required.

253 3.2. Generalization

254 Generalization, refers to the capacity to use previous knowledge from a source environment to
255 achieve a good performance in a target environment. It is widely seen as a step necessary to produce
256 artificial intelligence that behaves similarly to humans. Generalization may improve the characteristics
257 of learning the target task by increasing the starting reward on the target domain, the rate of learning
258 for the target task, and the maximum reward achievable [66].

259 In [67], Google proposed a method to learn hand-eye coordination for robot grasp task. In the
260 experiment phrase, they collected about 800,000 grasp attempts over two months from multiple robots
261 operating simultaneously, and then used these data to train a controller that work across robots. These
262 identical uncalibrated robots had differences in camera placement, gripper wear or tear. Besides, a
263 second robotic platform with eight robots collected a dataset consisting of over 900,000 grasp attempts,
264 which was used to test transfer between robots. The results of transfer experiment illustrated that
265 data from different robots can be combined to learn more reliable and effective grasping. One can
266 refer to the video at https://youtu.be/cXaic_k80uM for supplementary results. In contrast to many
267 prior methods, there is no simulation data or explicit representation, but an end-to-end training
268 directly from image pixels to gripper motion in task space by learning just from this high-dimensional
269 representation. Despite of its attractive success, this method still can not obtain satisfactory accuracy
270 for real application, let alone it is very hardware and data intensive.

271 To develop generalization capacities, some researchers turn to meta learning [68], which is also
272 known as learning to learn. The goal of meta learning is to train a model on a variety of learning tasks,
273 such that it can solve new learning tasks using only a small number of training samples [69]. In [70],
274 meta learning was combined with aforementioned imitation learning in order to learn to perform tasks
275 quickly and efficiently in complex unstructured environments. The approach was evaluated on planar
276 reaching and pushing tasks in simulation, and visual placing tasks on a real robot, where the goal is
277 to learn to place a new object into a new container from a single demonstration. The video results
278 are available at <https://sites.google.com/view/one-shot-imitation>. The proposed meta-imitation
279 learning method allows a robot to acquire new skills from just a single visual demonstration, but the
280 accuracy needs to be further improved.

281 There are also many other researches available to tackle other challenges in this domain. For
282 example, no matter whether a control policy is learned directly for a specific task, or transferred from
283 previous tasks, another important but understudied question is how well will the policy performs,
284 namely policy evaluation problem. A behavior policy search algorithm was proposed in [71] for more
285 efficiently estimating the performance of learned policies.

286 4. Discussion

287 Although algorithms of robotic manipulation control using DRL have been emerging in large
288 numbers in the past few years, some even with demonstration videos showing how an experimental
289 robotic manipulator accomplishes a task with the policy learned, as we have illustrated above, the
290 challenges of learning robust and versatile manipulation skills for robots with DRL are still far from
291 being resolved satisfactorily for real world application.

292 Currently, robotic manipulation control with DRL may be suited to fault tolerant tasks, like
293 picking up and placing objects, where a disaster will not be caused if the operation fails occasionally. It
294 is quite attractive in situations, where there is enough variation that the explicit modeling algorithm
295 does not work. Potential applications can be found in warehouse automation to replace human pickers
296 for objects of different size and shape, clothes and textiles manufacturing, where cloth is difficult to
297 manipulate by nature, and food preparation industry, where, for example, every chicken nugget looks
298 different, and it is not going to matter terribly if a single chicken nugget is destroyed.

299 However, even in this kind of applications, DRL based methods are not widely used in real world
 300 robotic manipulation. The reasons are multiple, including the two concerns we have discussed in
 301 the previous section, sample efficiency and generation, where more progress is still required, as both
 302 gathering experiences by interacting with the environment and collecting expert demonstrations for
 303 imitation learning are expensive procedures, especially in situations where robots are heavy, rigid and
 304 brittle, and it will cost too much if the robot is damaged in exploration. Another very important issue
 305 is safety guarantee. Not like simulation tasks, we need to be very careful that learning algorithms are
 306 safe, reliable and predictable in real scenarios, especially if we move to other applications that require
 307 safe and correct behaviors with high confidence, such as surgery or household robots taking care of
 308 the elder or the disabled. There are also other challenges including but not limited to the algorithm
 309 explainability, the learning speed, high-performance computational equipment requirements.

310 5. Conclusions

311 The scalability of DRL, discussed and illustrated here, is well-suited for high-dimensional data
 312 problems in a variety of domains. In this paper, we have presented a brief review of the potential
 313 of DRL for policy detection in robotic manipulation control and discussed the current research and
 314 development status of real world applications. Through a joint development of deep learning and
 315 reinforcement learning, with inspiration from other machine learning methods like imitation learning,
 316 GANs, or meta learning, new algorithmic solutions can emerge, and are still needed, to meet challenges
 317 in robotic manipulation control for practical applications.

318 **Author Contributions:** conceptualization, R.L. and B.D.-L.; methodology, R.L. and B.D.-L.; investigation, R.L.,
 319 F.N., P.Z. and B.D.-L.; resources, R.L., F.N., P.Z. and B.D.-L.; writing–original draft preparation, R.L.; writing–review
 320 and editing, F.N., P.Z. and B.D.-L.; supervision, B.D.-L.; project administration, B.D.-L. and M.d.M.; funding
 321 acquisition, B.D.-L. and M.d.M.

322 **Funding:** This research work is part of a project funded by the University of Strasbourg’s Initiative D’EXcellence
 323 (IDEX).

324 **Acknowledgments:** The support of the CNRS is gratefully acknowledged.

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

326 Abbreviations

327 The following abbreviations are used in this manuscript:

DoF	Degrees of Freedom
DRL	Deep Reinforcement Learning
DNN	Deep Neural Network
328 DQN	Deep Q-Network
DDPG	Deep Deterministic Policy Gradient
HER	Hindsight Experience Replay
GANs	Generative Adversarial Networks

329 References

- 330 1. Dresch-Langley B, Nageotte F, Zanne P, et al. Correlating grip force signals from multiple sensors highlights
 331 prehensile control strategies in a complex task-user system. *Bioengineering*. 2020, 7(4): 143.
- 332 2. Eranki V K P, Reddy Gurudu R. Design and Structural Analysis of a Robotic Arm. 2017.
- 333 3. Christ R D, Wernli R L. Manipulators. *The ROV Manual*. 2014, 503–534.
- 334 4. Ivanescu M. Control. *Mechanical Engineer’s Handbook*. 2001, 611–714.
- 335 5. Savatekar R D, Dum A A. Design of control system for articulated robot using leap motion sensor. *Int Res J*
 336 *Eng Technol (IRJET)*. 2016, 3, 1407–1417.
- 337 6. Wei H, Bu Y, Zhu Z. Robotic arm controlling based on a spiking neural circuit and synaptic plasticity.
 338 *Biomedical Signal Processing and Control*. 2020, 55, 101640.

- 339 7. Božek P, Al Akkad M A, Blištan P, et al. Navigation control and stability investigation of a mobile robot
340 based on a hexacopter equipped with an integrated manipulator. *International Journal of Advanced Robotic
341 Systems*. **2017**, *14*(6).
- 342 8. Safdar B. Theory of Robotics Arm Control with PLC. *Saimaa University of Applied Sciences*. **2015**.
- 343 9. Pitts W, McCulloch W S. How we know universals the perception of auditory and visual forms. *The Bulletin
344 of mathematical biophysics*. **1947**, *9*(3), 127–147.
- 345 10. Rosenblatt F. Perceptron simulation experiments. *Proceedings of the IRE*. **1960**, *48*(3), 301–309.
- 346 11. Rumelhart D E, Hinton G E, Williams R J. Learning internal representations by error propagation. *California
347 Univ San Diego La Jolla Inst for Cognitive Science*. **1985**.
- 348 12. LeCun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition. *Neural
349 computation*. **1989**, *1*(4), 541–551.
- 350 13. Jarrett K, Kavukcuoglu K, Ranzato M A, et al. What is the best multi-stage architecture for object recognition.
351 *IEEE 12th international conference on computer vision*. **2009**, 2146–2153.
- 352 14. Ciresan D C, Meier U, Masci J, et al. Flexible, high performance convolutional neural networks for image
353 classification. *Twenty-second international joint conference on artificial intelligence*. **2011**.
- 354 15. Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of
355 feature detectors. *arXiv preprint*. **2012**, arXiv:1207.0580.
- 356 16. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate
357 shift. *arXiv preprint*. **2015**, arXiv:1502.03167.
- 358 17. Liu R. Multispectral images-based background subtraction using Codebook and deep learning approaches.
359 *Université Bourgogne Franche-Comté*. **2020**.
- 360 18. Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks.
361 *Advances in neural information processing systems*. **2012**, 1097–1105.
- 362 19. Levine S, Finn C, Darrell T, et al. End-to-end training of deep visuomotor policies. *The Journal of Machine
363 Learning Research*. **2016**, 1334–1373.
- 364 20. Kaelbling L P, Littman M L, Moore A W. Reinforcement learning: A survey. *Journal of artificial intelligence
365 research*. **1996**, *4*, 237–285.
- 366 21. Kober J, Bagnell J A, Peters J. Reinforcement learning in robotics: A survey. *The International Journal of
367 Robotics Research*. **2013**, *32*(11), 1238–1274.
- 368 22. Sutton R S, Barto A G. Reinforcement learning: An introduction. *MIT press*. **2018**.
- 369 23. Dresch-Langley B, Ekseth O K, Fesl J, et al. Occam's Razor for Big Data? On detecting quality in large
370 unstructured datasets. *Applied Sciences*. **2019**, *9*(15), 3065.
- 371 24. Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning. *arXiv preprint*. **2013**,
372 arXiv:1312.5602.
- 373 25. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. *nature*.
374 **2015**, *518*(7540), 529–533.
- 375 26. Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree
376 search. *nature*. **2016**, *529*(7587), 484–489.
- 377 27. Silver D, Hubert T, Schrittwieser J, et al. Mastering chess and shogi by self-play with a general reinforcement
378 learning algorithm. *arXiv preprint*. **2017**, arXiv:1712.01815.
- 379 28. Brown N, Sandholm T. Superhuman AI for multiplayer poker. *Science* **2019**, *365*(6456), 885–890.
- 380 29. Berner C, Brockman G, Chan B, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint*
381 **2019**, arXiv:1912.06680.
- 382 30. Gu S, Holly E, Lillicrap T, et al. Deep reinforcement learning for robotic manipulation with asynchronous
383 off-policy updates. *IEEE international conference on robotics and automation (ICRA)*. **2017**, 3389–3396.
- 384 31. Sharma A R, Kaushik P. Literature survey of statistical, deep and reinforcement learning in natural language
385 processing. *International Conference on Computing, Communication and Automation (ICCCA)*. **2017**, 350–354.
- 386 32. Yun S, Choi J, Yoo Y, et al. Action-decision networks for visual tracking with deep reinforcement learning.
387 *Proceedings of the IEEE conference on computer vision and pattern recognition*. **2017**, 2711–2720.
- 388 33. Farazi N P, Ahamed T, Barua L, et al. Deep Reinforcement Learning and Transportation Research: A
389 Comprehensive Review. *arXiv preprint*. **2015**, arXiv:2010.06187.
- 390 34. Mosavi A, Ghamisi P, Faghan Y, et al. Comprehensive Review of Deep Reinforcement Learning Methods
391 and Applications in Economics. *arXiv preprint*. **2015**, arXiv:2004.01509.

- 392 35. Liu Y, Logan B, Liu N, et al. Deep reinforcement learning for dynamic treatment regimes on medical registry
393 data. *IEEE International Conference on Healthcare Informatics (ICHI)*. **2017**, 380–385.
- 394 36. Watkins C J C H, Dayan P. Q-learning. *Machine learning*. **1992**, 8(3-4), 279–292.
- 395 37. Bellman R E, Dreyfus S E. Applied dynamic programming. *Princeton university press*. **2015**.
- 396 38. Gordon G J. Stable fitted reinforcement learning. *Advances in neural information processing systems*. **1995**, 8,
397 1052–1058.
- 398 39. Riedmiller M. Neural fitted Q iteration—first experiences with a data efficient neural reinforcement learning
399 method. *European Conference on Machine Learning*. **2005**, 317–328.
- 400 40. Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-Learning. *Proceedings of the Thirtieth
401 AAAI Conference on Artificial Intelligence*. **2016**, 2094–2100.
- 402 41. Bellemare M G, Dabney W, Munos R. A distributional perspective on reinforcement learning. *Proceedings of
403 the 34th International Conference on Machine Learning*. **2017**, 449–458.
- 404 42. Dabney W, Rowland M, Bellemare M G, et al. Distributional reinforcement learning with quantile regression.
405 *In Proceedings of the 32th AAAI Conference on Artificial Intelligence*. **2018**.
- 406 43. Hessel M, Modayil J, Van Hasselt H, et al. Rainbow: Combining improvements in deep reinforcement
407 learning. *arXiv preprint*. **2017**, arXiv:1710.02298.
- 408 44. Salimans T, Ho J, Chen X, et al. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv
409 preprint*. **2017**, arXiv:1703.03864.
- 410 45. Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms. *In Proceedings of the 31st
411 International Conference on International Conference on Machine Learning*. **2014**, 387–395.
- 412 46. Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning. *arXiv preprint*.
413 **2015**, arXiv:1509.02971.
- 414 47. O’Donoghue B, Munos R, Kavukcuoglu K, et al. Combining policy gradient and Q-learning. *arXiv preprint
415 2016*, arXiv:1611.01626.
- 416 48. Oh J, Guo X, Lee H, et al. Action-conditional video prediction using deep networks in atari games. *Advances
417 in neural information processing systems*. **2015**, 2863–2871.
- 418 49. Nagabandi A, Kahn G, Fearing R S, et al. Neural network dynamics for model-based deep reinforcement
419 learning with model-free fine-tuning. *IEEE International Conference on Robotics and Automation (ICRA)*. **2018**,
420 7559–7566.
- 421 50. Silver D, Hasselt H, Hessel M, et al. The predictron: End-to-end learning and planning. *International
422 Conference on Machine Learning*. **2017**, 3191–3199.
- 423 51. Tamar A, Wu Y, Thomas G, et al. Value iteration networks. *Advances in neural information processing systems*.
424 **2016**, 2154–2162.
- 425 52. François-Lavet V, Bengio Y, Precup D, et al. Combined reinforcement learning via abstract representations.
426 *Proceedings of the AAAI Conference on Artificial Intelligence*. **2019**, 3582–3589.
- 427 53. François-Lavet V, Henderson P, Islam R, et al. An introduction to deep reinforcement learning. *arXiv preprint*.
428 **2018**, arXiv:1811.12560.
- 429 54. Wandeto J M, Drespe-Langley B. The quantization error in a Self-Organizing Map as a contrast and colour
430 specific indicator of single-pixel change in large random patterns. *Neural Networks*. **2019**, 119, 273–285.
- 431 55. Drespe-Langley B, Wandeto J M. Pixel precise unsupervised detection of viral particle proliferation in cellular
432 imaging data. *Informatics in medicine unlocked*. **2020**, 20: 100433.
- 433 56. Anthony M, Bartlett P L. Neural network learning: Theoretical foundations. *cambridge university press*. **2009**.
- 434 57. Kakade S M. On the sample complexity of reinforcement learning. *University of London*. **2003**.
- 435 58. Sergey L, Wagener N, Abbeel P. Learning contact-rich manipulation skills with guided policy search.
436 *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*. **2015**, 26–30.
- 437 59. Andrychowicz M, Wolski F, Ray A, et al. Hindsight experience replay. *Advances in neural information
438 processing systems*. **2017**, 30, 5048–5058.
- 439 60. Tai L, Zhang J, Liu M, et al. A survey of deep network solutions for learning control in robotics: From
440 reinforcement to imitation. *arXiv preprint*. **2016**, arXiv:1612.07139.
- 441 61. Andrew J. Bagnell. An invitation to imitation. Technical report, Robotics Institute, *Carnegie Mellon University*.
442 **2015**.
- 443 62. Vecerik M, Hester T, Scholz J, et al. Leveraging demonstrations for deep reinforcement learning on robotics
444 problems with sparse rewards. *arXiv preprint*. **2017**, arXiv:1707.08817.

- 445 63. Ho J, Ermon S. Generative adversarial imitation learning. *Advances in neural information processing systems*.
446 **2016**, 4565–4573.
- 447 64. Hausman K, Chebotar Y, Schaal S, et al. Multi-modal imitation learning from unstructured demonstrations
448 using generative adversarial nets. *Advances in Neural Information Processing Systems*. **2017**, 1235–1245.
- 449 65. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. *Advances in neural information*
450 *processing systems*. **2014**, 2672–2680.
- 451 66. Spector B, Belongie S. Sample-efficient reinforcement learning through transfer and architectural priors.
452 *arXiv preprint*. **2018**, arXiv:1801.02268.
- 453 67. Levine S, Pastor P, Krizhevsky A, et al. Learning hand-eye coordination for robotic grasping with deep
454 learning and large-scale data collection. *The International Journal of Robotics Research*. **2018**, 37(4-5), 421–436.
- 455 68. Thrun S, Pratt L. Learning to learn. *Springer Science & Business Media*. **2012**.
- 456 69. Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint*.
457 **2017**, arXiv:1703.03400.
- 458 70. Finn C, Yu T, Zhang T, et al. One-shot visual imitation learning via meta-learning. *arXiv preprint*. **2017**,
459 arXiv:1709.04905.
- 460 71. Hanna J P, Thomas P S, Stone P, et al. Data-efficient policy evaluation through behavior policy search. *arXiv*
461 *preprint*. **2017**, arXiv:1706.03469.

462 © 2021 by the authors. Submitted to *Robotics* for possible open access publication under the terms and conditions
463 of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).