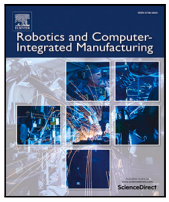
Robotics and Computer–Integrated Manufacturing 81 (2023) 102517

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim



Review

A review on reinforcement learning for contact-rich robotic manipulation tasks

Íñigo Elguea-Aguinaco [a,b,*], Antonio Serrano-Muñoz [b], Dimitrios Chrysostomou [c], Ibai Inziarte-Hidalgo [a,d], Simon Bøgh [c], Nestor Arana-Arexolaleiba [b,c]

[a] Research & Development Department, Electrotécnica Alavesa S.L., 1010 Vitoria-Gasteiz, Spain
[b] Robotics and Automation Electronics and Computer Science Department, University of Mondragon, 20500 Mondragon, Spain
[c] Department of Materials and Production, Aalborg University, 9220 Aalborg East, Denmark
[d] Automation Department, Montajes Mantenimiento y Automatismos Eléctricos Navarra S.L., 31195 Aizoain, Spain

ARTICLE INFO

Keywords:
Reinforcement learning
Contact-rich manipulation
Industrial manipulators
Rigid object manipulation
Deformable object manipulation

ABSTRACT

Research and application of reinforcement learning in robotics for contact-rich manipulation tasks have exploded in recent years. Its ability to cope with unstructured environments and accomplish hard-to-engineer behaviors has led reinforcement learning agents to be increasingly applied in real-life scenarios. However, there is still a long way ahead for reinforcement learning to become a core element in industrial applications. This paper examines the landscape of reinforcement learning and reviews advances in its application in contact-rich tasks from 2017 to the present. The analysis investigates the main research for the most commonly selected tasks for testing reinforcement learning algorithms in both rigid and deformable object manipulation. Additionally, the trends around reinforcement learning associated with serial manipulators are explored as well as the various technological challenges that this machine learning control technique currently presents. Lastly, based on the state-of-the-art and the commonalities among the studies, a framework relating the main concepts of reinforcement learning in contact-rich manipulation tasks is proposed. The final goal of this review is to support the robotics community in future development of systems commanded by reinforcement learning, discuss the main challenges of this technology and suggest future research directions in the domain.

1. Introduction

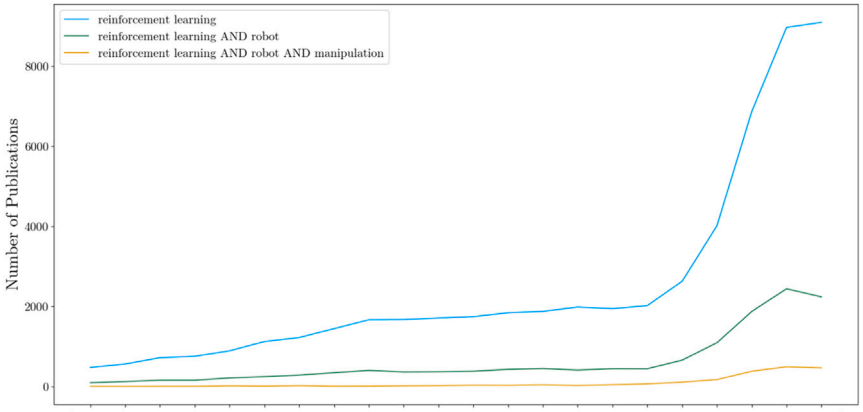
The embrace of Industry 4.0 brought about a paradigm shift in many domains, leading multiple companies to bet on the automation and digitization of their manufacturing processes through robots and artificial intelligence (AI) techniques [1]. Now, at the ten-year mark since the introduction of this production model, far from stagnating, its overall market value is expected to double in the next five years [2]. Robotics will be one of the markets that is projected to record one of the highest growth, specifically the collaborative robotics segment. This growth will be driven by multiple drivers, such as the increasing adoption of automation in the end-use industry and high-precision work, reaching USD 1.71 billion by the end of 2022 [3]. These robots, like conventional robots, although the latter to a lesser extent, are typically intended for complex handling tasks, also known as contact-rich tasks, where they are occasionally required to have adaptive capabilities not always attainable through conventional control. Specifically, according to [4–6], a contact-rich manipulation task is any task that involves close interaction between the robot and its environment and comprises complex, high-dimensional and even nonlinear contact dynamics. These tasks are characterized by contact situations such as sliding, sticking or obstacle-constrained motion.

In this sense, intelligent control, in particular machine learning (ML), emerges as an alternative to control a dynamic and flexible system in which controllers learn directly from examples, data and experience, thus enhancing the decision-making ability of robots when faced with complexity, variability, and uncertainty. Namely, there are three fundamental branches in ML: supervised learning, where the system is trained with labeled data to predict categories in new or test data; unsupervised learning, where the system aims to detect features that make data similar to each other; and reinforcement learning (RL), where the system learns decisions from the experience of interacting with the environment [7].

RL methods are a promising approach, as they hold the promise of solving control tasks in complex unconstructed environments [8].

* Corresponding author at: Research & Development Department, Electrotécnica Alavesa S.L., 1010 Vitoria-Gasteiz, Spain.
E-mail address: ielguea@aldakin.com (Í. Elguea-Aguinaco).

https://doi.org/10.1016/j.rcim.2022.102517
Received 26 July 2022; Received in revised form 3 November 2022; Accepted 18 December 2022
Available online 27 December 2022



	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
reinforcement learning	476	562	720	755	891	1124	1224	1446	1669	1675	1713	1746	1845	1877	1986	1945	2021	2628	4014	6866	8969	9095
reinforcement learning AND robot	97	126	159	159	217	248	287	348	403	367	372	382	429	450	411	446	444	658	1094	1875	2440	2240
reinforcement learning AND robot AND manipulation	6	5	7	8	18	12	22	10	12	18	24	36	33	45	28	47	68	111	176	383	492	465

Fig. 1. Publications per year on reinforcement learning, robotics and manipulation in Scopus.

Indeed, they allow agents to learn through interaction with their surroundings and, ideally, to generalize the learned behavior to new, unseen scenarios.

These methods have gained increasing interest due to their promising results in areas such as playing video games [9,10] and games [11–13], fluid dynamics [14], autonomous ground [15,16] and air navigation [17,18], recommender systems [19,20], energy management [21,22], Internet of Things (IoT) [23], natural language processing [24], healthcare [25,26], Industry 4.0 [27], pick and place [28], grasping [29] and robotic manipulators [30,31].

Notwithstanding, although several research papers have been published analyzing the capabilities of RL in robotic manipulation, to the authors' knowledge, there is no work that provides a comprehensive review in this area. Therefore, this paper strives to address this gap and aims to review the most relevant and up-to-date work on the application of RL for the execution of contact-rich tasks in the last lustrum. The contributions of the paper are:

- An analysis of the current status and application of RL in contact-rich tasks over the last five years.
- An insight on the main current trends around the application of RL in contact-rich tasks, as well as the main gaps.
- A framework for establishing a relationship among RL engineering concepts for different contact-rich manipulation tasks.

The content of the paper is organized as follows. Section 2 contains a description of the methodology to identify and select relevant papers. In Section 3, a theoretical background on RL is provided for the understanding of the state-of-the-art analysis in the field of robotic contact-rich manipulation. Section 4 highlights the main contact-rich tasks where RL is applied in both rigid and deformable object manipulation, and describes the reviewed papers in their respective subsection. Section 5 identifies the main trends and gaps in contact-rich tasks through RL based on the reviewed literature. Section 6 provides an outline of the main headings to consider for future research, providing points for discussion and reflection and listing the research gaps identified. Lastly, Section 7 concludes with a summary of the knowledge gained.

2. Search methodology

The relevance of RL relies on its adaptability in a highly changing and unstructured environment compared to more conventional or other AI control techniques. Therefore, its applicability in robotic manipulators has been a hot topic recently. This can be seen in Fig. 1, which shows the number of scientific publications on this field in the multidisciplinary database Scopus,[1] from the beginning of the century to the present day. By using the keywords "reinforcement learning" AND "robot" AND "manipulation", one can observe that from 2017 onwards, there has been a significant and progressive increase in the number of publications, reaching its peak in 2021. This milestone is marked by the successful integration of deep neural networks (deep learning) in RL algorithms, which enabled researchers to deal with hitherto intractable complex problems. Notwithstanding, to the authors' knowledge, there is currently no analysis that compiles the most relevant works in this domain and highlights its current trends and challenges, which might be beneficial for this field. Consequently, the goal of this review is to provide an overview of the main studies employing RL in contact-rich manipulation tasks, as well as an analysis of where the trends in this field are heading. For this aim, a broad literature review is executed, and the content of more than 150 papers in related areas is researched and reviewed. A summary of the chosen search criteria can be found in Table 1.

Initially, a search was conducted in multidisciplinary databases, namely Scopus, Google Scholar,[2] Web of Science,[3] and Engineering Village[4] for the period from 2017 to 2022. Several search terms related to the application context were used, including "Reinforcement Learning" AND "Robot" AND "Manipulation", "Reinforcement Learning" AND "Robot" AND "Contact", and "Reinforcement Learning" AND "Robot" AND "Control". The selection of these terms was based on the argument that they must include RL as a control technique, a robotic mechanism, and a relation to contact tasks, although these may be diverse.

At the same time, studies that, although relevant to the field of RL, were not suitable for the scope of this review had to be excluded. In particular, the authors decided to exclude non-English studies and those whose use case the authors did not strictly consider a contact-rich task according to the definition provided in Section 1. This included tasks such as pick and place [32,33], grasping [34,35], and lifting [36] and tilting [37] once the object is already attached to the manipulator.

The period of this search was selected to cover January 2017 to July 2022. The beginning of this period was chosen because according to Fig. 1, approximately from this year onwards was when the application of RL in robotics started to gain greater emphasis.

[1] http://www.scopus.com/
[2] https://scholar.google.es/
[3] https://www.webofscience.com/
[4] https://www.engineeringvillage.com/

Table 1
Overview of the various review criteria applied during the search process for relevant literature.

Search criteria	Description
Search terms	"Reinforcement Learning" AND "Robot" AND "Manipulation", "Contact", "Control"
Time period	January 2017–July 2022
Publication type	Peer-reviewed academic conference paper and journal articles
Exclusion criteria	**Description**
Language	Non-English
Contextual	Non-contact-rich tasks

3. Background on reinforcement learning

RL [7] is a type of ML in which an agent learns to interact with its environment with the goal of maximizing the rewards it receives in the long run. This interaction learning problem is formalized through the ideas of dynamical systems theory, used to describe the behavior of complex systems that evolve over time and which are generally modeled as a Markov Decision Process (MDP).

The MDP is a process that defines sequential decision-making as a semi-random and agent-dependent pathway. It also specifies that decisions made and executed (actions) influence not only immediate rewards, but also subsequent situations (states) through future rewards. Thus, MDPs are made up of three elements: the state the agent is in, the action the agent takes, and the agent's ultimate goal. In their simplest forms, the elements are usually represented by the following tuple:

$$[S, A, P(s'|s,a), R(s,s',a), \gamma] \tag{1}$$

where S is the set of possible states of the agent and A the set of actions. $P(s'|s,a)$ is the probability of transition to a future state s', when the agent is in state s and applies action a. $R(s,s',a)$ is the reward that the agent expects to obtain when it transits from state s to state s', and is calculated through the reward function. Finally, γ is the discount factor of the reward function. Thus, for each time step, the agent will select an action, and the environment will respond to this action, on the one hand, by presenting a new situation to the agent and, on the other hand, by returning a reward, the numerical value that the agent will try to maximize. Fig. 2 shows the basic MDP scheme underlying the decision process of any RL agent. This process can be defined through the following sequence:

$$s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, r_3 \ldots \tag{2}$$

A policy $\pi(a|s)$ is a mapping between the states and the probabilities of selecting each possible action. Likewise, the total amount of reward that an agent expects to accrue in the future, starting from a particular state and following a particular policy is called value function.

3.1. Reinforcement learning algorithms taxonomy

Although it is difficult to make a standardized classification of RL algorithms due to their wide modularity, many current studies opt to divide them into model-based and model-free algorithms. The latter, in turn, are divided into value-based, policy-based and actor–critic algorithms (see Fig. 3). The main difference between model-based and model-free algorithms is the use of a model of the interactions between the agent and its environment. In the case of model-based algorithms, the transition dynamics of the environment are known, so this model can be used for the derivation of rewards and next state. However, in the case of model-free algorithms, these dynamics are unknown, so rewards and actions are derived from the trial-and-error interaction that the agent performs with its environment during learning.

While the scientific community is giving much attention to model-free algorithms owing to their ease of implementation, model-based algorithms can significantly reduce the number of iterations required for the algorithm to converge. Nevertheless, it is necessary to know the exact dynamics of the environment. Table 2 summarizes the advantages and disadvantages of both methods.

3.1.1. Value-based algorithms

Value-based algorithms do not store any explicit policy, but use temporal difference learning to compute the value function of each state or state–action pair until its values converge. This enables reducing the variance in the estimates of the expected returns, although this implies a computationally expensive optimization procedure. The optimal policy can then be derived directly from the value function by acting greedily (selecting the action with the best value) on the computed function.

3.1.2. Policy-based algorithms

Unlike value-based algorithms, policy-based algorithms explicitly construct the policy that will map each state to the corresponding best action and keep this parameterized function in memory during learning. Thus, these algorithms update the policy without considering the estimates of the value function, which allows them to generate a

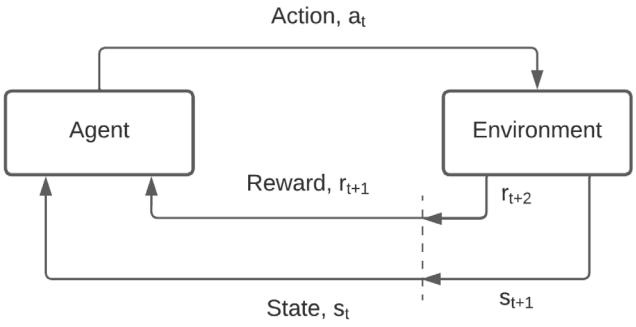


Fig. 2. RL scheme [7].

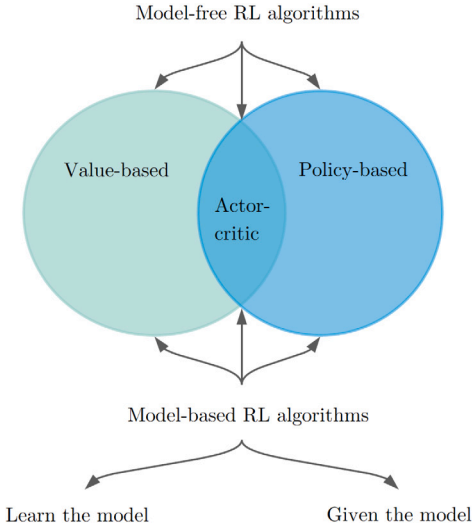


Fig. 3. Types of RL algorithms.

Table 2
Advantages and disadvantages of model-based and model-free algorithms.

Method	Advantage	Disadvantage
Model-based algorithms	- Sample efficiency. - Reduction in the number of interactions between the agent and its environment.	- Dependence on transition models. - Accurate knowledge of transition dynamics.
Model-free algorithms	- No prior knowledge of transitions. - Ease of implementation.	- Poor sample efficiency.

continuous spectrum of actions, albeit with a high variance. Also, these algorithms can be obtained through gradient-based or gradient-free parameter estimation methods.

3.1.3. Actor–critic algorithms

Actor–critic algorithms combine the advantages of value-based and policy-based methods. These algorithms are promising options for learning approximations of both the policy function and the value function, where the "actor" is a reference to the learned policy and the "critic" refers to the value function.

The learning is on-policy, while the parameterized actor estimates continuous actions without the need for optimization over a value function, the critic provides the actor with low variance knowledge about performance. More precisely, the critic's estimation of expected performance allows the actor to update with lower variance gradients, thus speeding up the learning process.

4. Contact-rich manipulation tasks

In contact-rich manipulation tasks, the dynamics of the interaction between the manipulator, the object to be manipulated, and the environment determine the final outcome of the task. This manipulation can be performed with both rigid and deformable objects. The following is a descriptive section that seeks to identify the main contributions and shortcomings of the studies reviewed on this subject with both types of manipulable objects.

4.1. Rigid object manipulation tasks

The effective resolution of complex handling tasks in an unstructured or highly variable environment remains a field of study. Current research focuses mainly on assembly and insertion tasks [38,39], but includes many other applications [40]. RL methods have shown reliable performance to uncertainties, leading more and more researchers to target on learning manipulation skills. In the following, current studies on robotic rigid object contact-rich manipulation tasks are grouped and summarized according to the main application scenarios.

4.1.1. Assembly and insertion tasks

Assembly defines the sequential aggregation of parts and subassemblies resulting into functional products and can include any number of a vast array of robotic technology throughout the process to achieve efficiency, productivity and cost-effectiveness goals. Assembly is one of the most common use cases worldwide. In fact, the global assembly automation market is expected to grow by around 30% over the next five years [41]. It is, therefore, reasonable to assume that it is the most widely addressed use case within RL. Among the most common applications are peg-in-hole tasks and eclectic connector bonding. However, the dimensions of the clearances with respect to robot precision, the pose uncertainty of the peg-hole, and the uncertain and complex contact dynamics in each assembly, among others, mean not all studies deal with the same concerns. Within the assembly, studies emphasize performance improvement and the search for sample efficiency and generalization capability (see Fig. 4).

The performance of RL policies has been one of the most extensively revised lines of research in the last five years. Yet, performance can be targeted towards different purposes, among others, precision, stability, safety, execution time or robustness.

Bearing precision in mind, Inoue et al. [42] and Wu et al. [43] calculated and iteratively improved the values of the Q-value function to correct positional and angular errors to achieve assemblies with an inter-part tolerance of micrometers and to adapt to small pose misalignment, respectively. However, in both cases, the use of discrete actions reduced the potential of the policies. More recent studies such as [44–46], used continuous actions for assembly tasks with wooden and tolerance-prone parts, millimeter or even micrometer tolerances respectively, this time with promising results. Nevertheless, [46] concluded that contact-stability was still a requirement for further analysis.

Contact-stability was addressed in [47,48]. In [47], the authors developed a framework in which a model-based algorithm computed a trajectory, and a model-free algorithm learned manipulation skills by solving potential stability problems caused by stiffness and force/torque feedback. In [48], the authors used multimodal information, namely, vision and force, to represent the agent state and obtain smoother insertion strategies. However, both inputs were deemed equally useful throughout the assembly, which reduced the performance of the approach when the visual guidance suffered occlusions. In contrast, Khader et al. [49] analyzed stability by considering that any state trajectory should be bounded and tend to the target position required by the task. For this purpose, they shaped the exploration of the RL agent with a Lyapunov function. This constrained exploration, in turn, guaranteed a safe and predictable behavior of the manipulator.

Specifically, safety in the agent's decision making, mainly during training, is a factor that the researchers seek to ensure. For instance, in [50–52], the authors constrained the robot's displacement and velocity motion commands, as well as the contact forces on the end-effector. In case any action did not meet the safety conditions, the manipulator did not execute any action in that time step. In contrast, Li et al. [53] trained an RL agent on a digital twin that was subsequently employed

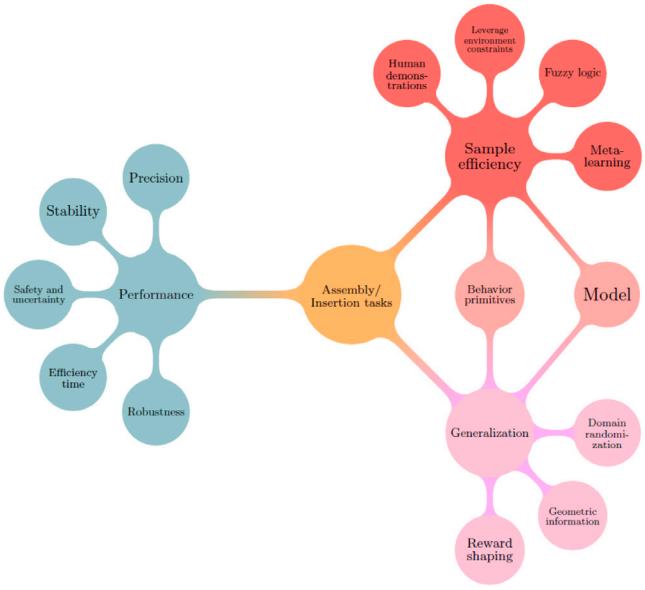


Fig. 4. Main themes addressed by assembly and insertion tasks studies.

to oversee the assembly. This approach allowed the authors not only to provide safety protection while performing the task but also to balance precision and efficiency during training and prevent the performance drops in reality in high-contact tasks.

Wang et al. [54], meanwhile, focused on assembly times. They argued that grip position can directly impact assembly efficiency and corroborated how process times could be reduced.

On the other hand, Kulkarni et al. [55], Ennen et al. [56] and Wirnshofer et al. [57] focused on improving robustness. In [55], the authors combined RL along with conventional control strategies to obtain robust policies under small uncertainties in the environment. In [56], taking Guided Policy Search as a starting point, the authors proposed a new representation of policies that executed stable actions despite being outside the distribution of trajectories explored in training through motor reflex generation. This was possible as long as the input states could be reliably encoded in the same latent state distribution that had been explored during training. In [57], the authors dealt with uncertainty in manipulation tasks with partial observability. To do so, they relied on a belief about the state of the system using a particle filter. The belief representation gave rise to a discrete set of motion controllers among which the RL agent switched between to perform the manipulation task.

However, many of these studies focused on improving the performance of RL policies in assembly reported long training periods [58]. This is a recurrent drawback mainly in those high-dimensional tasks where large exploration is required. Therefore, many studies also put their efforts into mitigating this limitation and gaining greater sample efficiency [59,60].

One of the most explored lines of research in the last years to increase sampling efficiency has been the combination of RL with human demonstrations [61,62]. While early papers required tens [63] or even a hundred [64] demonstrations, more recent papers can compute trajectories even from a single demonstration [65]. [66–68] employed human demonstrations in combination with residual RL [69] to, from an initial trajectory, improve exploration performance and finely tune the force-sensitive policy. Nonetheless, these approaches were highly dependent on the geometry of the trajectories provided [70], which could reduce their generalizability to other tasks. By contrast, Jin et al. [71] used expert knowledge to perform offline counterfactual predictions that, together with online observations, allowed predicting future states based on the action the agent could take. This approach guided exploration toward states in which the agent acquired sparse reward. However, the authors only considered task-relevant spatial information which, as in previous papers, could also reduce generalization ability. Similarly, analogous to human demonstrations, Hoppe et al. [72] proposed an approximate trajectory optimization approach for exploration based on the upper confidence bound of the advantage function. This approach rapidly found sample trajectories that sped up the learning process.

Other studies, oppositely, attempt to leverage physical or environmental constraints to increase sample efficiency [73]. For instance, Hamaya et al. [74] took advantage of environmental constraints and soft robot capabilities to learn a non-linear model that they used to acquire skills in reduced dimensionality peg-in-hole tasks. Simonic et al. [75] mapped the information obtained during disassembly tasks to assembly tasks. To do so, they implemented a hierarchical RL algorithm and a graph representation under the criterion that assembly is the reverse of disassembly broken down into multiple stages. However, this assumption was largely limited to standardized disassemblies, where the state of the end-of-life (EoL) product remained undamaged.

In a different line of research, but also aiming to improve sampling efficiency, Xu et al. [76] combined RL with fuzzy logic. Specifically, they applied a flexible fuzzy reward system that considered more comprehensive task factors than hand-engineered rewards. This reward stabilized the training process and accelerated agent convergence. However, they did not obtain positive results in generalization to other environments, and even the mismatch between the contact forces estimated by the simulator and the actual forces prevented the authors from applying the simulation training results in reality. Years later, the same authors used fuzzy logic-driven variable time-scale prediction to map the predicted environment with the controller impedance parameters [77]. Notwithstanding, in the vast majority of these scenarios, a bottleneck is produced as the generalization capacity is reduced with the improvement of the sample efficiency.

Indeed, generalization ability, as well as the deployment of the policy from simulation to reality, are two of the major challenges reported around RL at present and, as such, there are many assembly-related studies that focus their concern directly on this topic. Both concepts, generalization and the deployment of the policy learned into reality, are highly related, as greater generalization capability leads to robust policies that are less susceptible to the simulation-reality gap.

In this sense, a widely used approach in multiple studies is domain randomization. This technique allows to uniformly randomize a distribution of real data in predefined ranges in each training episode to obtain more robust policies. Depending on the components of the simulator randomized, two broad methods of domain randomization may be distinguished: visual randomization [78] and dynamic randomization [79]. Visual randomization targets to provide sufficient simulated variability of visual parameters, such as object location and their estimation in space or illumination. Dynamic randomization can help to acquire a robust control policy by randomizing various physical parameters in the simulator such as object dimensions, friction coefficients, or the damping coefficients of the robot joints, among others. However, too much randomization of the environment can hinder agent learning and lead to suboptimal policies. In [80], the authors proposed to leverage geometric information to guide RL learning and a neural network that, from that learned solution, exploited the knowledge to other task configurations. But this required CAD files of the elements to be manipulated, which are not always available in all contact-rich tasks. On the other hand, Lee et al. [81] used supervised learning to learn a multimodal representation with vision and haptic inputs that favored generalization around the variation of different objects.

Reward shaping can also take a key role in generalization. Wu et al. [82], for example, trained an encoder–decoder network with visual and haptic inputs to provide dense rewards autonomously according to task progress. This self-supervised way of providing rewards yielded greater rewards per episode than those received by hand-engineered rewards. However, this approach was only intended for monotonic tasks. Leyendecker et al. [83], in turn, proposed a reward-curriculum learning approach that, in combination with domain randomization, dynamically adjusted the reward function according to the agent's learning performance. This system allowed the authors to obtain a robust controller and force-sensitive adaptive motion trajectories. Other studies, conversely, sought to avoid hand-crafted rewards and to obtain these functions directly from human demonstrations. This is the case of [84], where inverse RL was used to acquire the reward functions. Although the gap between simulation and reality remained, the proposed approach outperformed behavior cloning and could be used to generate new policies by optimizing the learned reward function for different task configurations.

Lastly, there are articles that seek to address sample efficiency and generalization simultaneously. To this end, many researchers employ models. Zhao et al. [85], for example, employed a Gaussian Process stochastic model to learn environment dynamics, speeding up the learning process, while ensuring exploration efficiency. Subsequently, this model was used to improve the estimation of the target value and generate virtual data to argue the transition samples. Tanaka et al. [86] presented a model-based method that used a collection of dynamics models from the source environment to learn the state transition dynamics of the target environment. This allowed them to approximate the target state transition dynamics without knowing the exact dynamic parameters of each environment. Another approach

proposed to overcome the lack of generalization was [87]. In this case, the authors employed a dynamics model that relied solely on force–torque feedback, sampled from multiple poses. This model was then enriched by generating numerous offline synthetic trajectories based on the grid-sampled feedback itself, and trained on a data set that contained a mixture of pegs and holes with different traits.

More recent studies, in contrast, point to the combination of dynamic movement primitives (DMPs) [88] with RL and learning in the task space as a method to improve generalization. DMPs allow the contact-rich task to be divided into low-dimensional, easily adaptable sub-tasks, which also increase training efficiency. RL can then be used to adjust these trajectories and explore in the task space, which can improve policy robustness in insertion tasks [89,90]. Following this line of research, RL can also be employed to learn and select parameterized actions, hybrid actions consisting of a set of discrete primitives with continuous primitive parameters, resulting in control policies that are flexible, efficient, and robust to simulation-reality transfer [91].

Other studies, instead, use sample efficiency and generalization approaches simultaneously to improve both. These investigations point, for instance, to combining RL with meta-learning [92], also known as meta-RL. Meta-learning enhances sample efficiency, as it enables explicit learning of a latent structure over a family of tasks that can later be easily adapted to the target task. Studies such as [93,94] employed meta-learning with an RL agent to learn insertion tasks. In addition, during training, they used domain randomization to foster the agent's generalization capabilities. Zhao et al. [95], in turn, used meta-learning to learn an RL policy from offline data. Subsequently, they employed direct online finetuning to also improve the agent's generalization capabilities.

Performance, data efficiency and generalization are the three cornerstones of RL that are currently being pursued for enhancement within assembly. While performance can be approached from different perspectives, clearly assisted learning and/or the use of other control techniques alongside RL seems to play a key role in improving learning efficiency or agents' ability to generalize to new environments. However, at times, this seems to generate a bottleneck that compromises either data efficiency or generalization. In spite of this, in recent years, there has been a growing trend to jointly enhance both matters.

4.1.2. Disassembly tasks

Responsible treatment of EoL products can include reusing, recycling, or remanufacturing. These processes can be environmentally and economically beneficial since waste is minimized while valuable components and materials are recovered. Remanufacturing, in particular, is an emerging field due to its contribution to the growth of greener manufacturing industries. One of the main procedures required for the remanufacturing and treatment of any product at the end of its useful life is disassembly. This procedure involves the removal and segregation of parts, pieces, or materials desired for future repair and maintenance of the new product.

Currently, most of the applications related with disassembly in RL are focused on Waste Electrical and Electronic Equipment (WEEE). The growing impact of WEEE has underlined the relevance of the circular economy and thus the importance of disassembly. However, the lack of technology for remanufacturing and knowledge about the high variability among some products causes there are still few studies that employ RL as part of this segregation process.

Kristensen at al. [96] proposed a framework in which they used the Q-learning algorithm to train and test agents in robotic unscrewing tasks. In turn, Herold et al. [97] proposed strategies that allowed the separation of fixed components into a slot. For this, the authors identified that adjusting the robot's position end-effector proportionally to the measured forces and including oscillating motion could be a suitable solution for the assignment. But they executed the task by performing predefined actions, which made it difficult to generalize to other scenarios. In this sense, Serrano-Muñoz et al. [98] specifically analyzed the generalization capability of two actor–critic algorithms, namely DDPG and TD3, in contact-rich disassembly tasks. For this purpose, they randomized both the rotation and the position of a peg embedded on a base which the robot had to extract, obtaining promising results in real world.

Indeed, generalization capability might play a key role in disassembly since, unlike in assembly, the condition of an EoL product can be highly heterogeneous. Therefore, disassembly through RL should consider agents that can deal with the physical uncertainties associated with the product condition, considering the large variety within one product category, and complexities in process planning and operation.

4.1.3. Polishing and grinding tasks

Polishing and grinding are two highly employed machining processes. While the former employs free abrasives to smooth surfaces with almost no material removal, the latter uses fixed abrasives for machining with higher dimensional accuracy and less roughness that machining by stock removal. Both tasks are applied in the manufacture of components in multiple fields, ranging from industrial to medical applications [99]. In the face of increasingly higher machining requirements and more complex polishing tasks, the use of robots for both tasks is on the rise [100,101]. One of the first studies to use RL within this scenario was [102]. The authors trained a Q-learning algorithm to optimize trajectories for both polishing and grinding tasks. However, they did not consider contact uncertainty or instability. In fact, one of the challenges that often arises within these tasks is how to mitigate the vibration that occurs between the robot tool and the workpiece due to the low stiffness of some manipulators and that results in low process quality.

Therefore, recent studies base their research on achieving contact-stability. Zhang et al. [103] first used a pressure model to compensate and stabilize the initial normal contact force between the robot and the workpiece. Subsequently, they used a model-based RL agent to obtain the displacement offset parameters to maintain a constant force throughout the grinding process. In contrast, Ding et al. [104] focused only on polishing. The authors selected impedance control to manage the dynamic relationship between the force and positional deflection of the robot and used RL to dynamically adjust the stiffness and damping parameters to ensure the stability of the system. However, they concluded that their approach lacked generalization capability and that its application to other components polishing should be explored.

4.1.4. Stacking and unstacking tasks

The loading and transfer of goods is a recurring activity in companies worldwide. However, when the human worker lifts continuously heavy loads, the likelihood of musculoskeletal injury becomes higher. In this sense, palletizing or unstacking robots are a solution that many companies are starting to implement.

Although at first glance stacking and unstacking tasks may appear similar to pick and place tasks, and often are, the authors have seen fit to treat them as distinct activities. Unlike pick and place tasks that are based on hand–object contacts, stacking and unstacking tasks involve object–object interactions. This implies that the robot controller does not focus solely on performing a pick and release with the gripper, but must consider contacts that may or may not enable the desired interaction. This include, for example, how one object is aligned or snapped relative to another to avoid a collapse or even modifying stiffness and damping parameters in unstacking tasks with sticky object–object contacts. In these cases, for instance, the authors may leverage information about contact forces that they estimate using tactile sensors of the robot.

Within stacking tasks, Cabi et al. [105] combined expert knowledge with reward sketching for cube piling. For this purpose, they first generated a database with teleoperation demonstration trajectories that were subsequently leveraged to reduce exploration and facilitate reward learning by providing examples of successful behaviors. Using

the available database, and the human-defined reward function, the robot was able to learn a new manipulation task. However, this implied a human-in-the-loop continuously during training. Besides, the reward sketch used was not universal, making the agent's generalization ability highly dependent on other strategies. Belousov et al. [106] also employed human demonstrations, rather than starting from scratch, to collect a larger data set in exploration. In this case, the authors adjusted the placement of building modules through tactile feedback. Although their results showed a path to promising research, inconsistencies between simulation and reality, as well as in the simulation of vision-based tactile sensors itself, led the authors to conclude that the most encouraging approach would be to perform learning directly in reality.

This learning straight in reality was conducted in [107], where the authors proposed an inner/outer loop impedance control method based on an RL algorithm for unstacking rubbers with time-varying adhesion forces. The required impedance was applied by controlling the inner/outer loop impedance with time-delay estimation, which could correct the modeling error and compensate the nonlinear dynamics term to improve the system efficiency. RL was utilized to optimize the online impedance parameters, which improved the accuracy and robustness in an unstructured dynamic environment.

4.1.5. Door/drawers opening tasks

Door or drawers opening is a practice that has been increasingly studied in RL. When the robot grasps the handle, the kinematic chain is closed, and the robot must be able to push or pull the door or drawer to open it. The complexity of these tasks lies in the motion constraint of the robotic arm, where the controller needs to find a feasible trajectory within the available workspace. While some studies tackle this task once the knob is already grasped, other research, on the other hand, addresses the task from the approach and grasping of the door handle. In both cases, studies focus primarily on generalization and improving learning speed.

The first studies to consider generalization were based on guided policy search [108]. This method employs a set of optimized local policies to subsequently train a global policy that is generalized across cases. In this configuration, RL is used to train the simple local policies. Under this approach, whereas in [109], the authors proposed a global policy sampling scheme to sample new instances of the task and increase the diversity of the training data to improve the generalization capability of the agent, in [110], the same authors employed distributed and asynchronous policy learning. In [111], on the other hand, the authors proposed a structured search exploiting the physical constraints of the environment. In this way, the underlying controller generated motions along the defined directions admissible by the physical constraints of the task; that is, it applied forces to the unconstrained degrees of freedom of the robot. However, in all studies, the IP^2 algorithm was applied, which aims to simultaneously optimize the reference trajectory and the gain schedule as early as possible. This may result in slow response speed due to its overemphasis on gain reduction while completing the task [112].

In contrast, Gu et al. [113] focused on improving learning times. They accomplished this by parallelizing learning across multiple robots and gathered updates from each of the policies asynchronously. More recent studies, such as Englert et al. [114] and Lin et al. [115], addressed generalization and sample efficiency to reduce training times, namely through human demonstrations, simultaneously. The former used a single demonstration together with the analytic motion optimization and RL to create constrained motion optimization representation that enabled them to generalize the skill to different initial states. The latter also used human demonstrations, albeit in this case to train primitive actions that were subsequently aligned in a logical sequence for adaption to a door opening.

Moreover, in the latter three studies, the authors considered safety during learning by constraining the action space of the manipulator in case the next predicted reachable state deviated significantly from the trajectory or subtrajectory being executed at the time. In high-dimensional continuous problems, such as doors and drawers opening tasks, besides the need to shorten training times and increase generalization capability, there seems to be a concern among the authors to ensure safe learning. This is because a large agent action space could cause damage to both the environment and the robot itself.

4.1.6. Pushing tasks

Pushing is an essential motion primitive in the repertoire of any robot. It is crucial for the efficient manipulation of objects under uncertainty or even to position the object in a suitable configuration prior to grasping. Yet, while humans are capable of executing manipulations deftly and smoothly and transferring these manipulation behaviors to different objects, this task for robots is more challenging. The difficulty arises from the unknown and nonlinear dynamics, when the robot must perform gentle manipulation on objects of different sizes, shapes, and textures [116]. Therefore, studies related to object pushing focused on safe handling and generalization. All reviewed papers focus additionally on planar pushing, where the agent pushes the object on the horizontal support plane while gravity acts along the vertical.

For instance, Lin et al. [117] adopted a common force–torque sensor and touch sensors and demonstrated that in object pushing tasks, a simple force-based reward and corrective feedback action could improve the safety effectively, and significantly reduce the impact and abnormal behaviors. Nevertheless, the authors simplified the tactile data through Boolean values, which resulted in corrective feedback only being provided when a certain threshold was exceeded, which could lead to sub-optimal policies. Additionally, the performance of the policy was greatly reduced when deployed in the real world, where abnormal movements occurred due to illumination and visual detection. Huang et al. [118] obtained better results in reality than in [117]. Instead, they introduced both a penalty for impact forces and a curiosity-focused positive intrinsic signal to encourage exploration in the reward function. Although the agent's performance was highly dependent on the choice of the curiosity focus, the approach encouraged smooth interaction with the environment.

These studies suggest a concern for performing soft contact-rich tasks. In this sense, the definition of reward functions may play a key role, and may enhance gentle manipulation by penalizing large contact forces.

Cong et al. [119], in turn, improved the generalization capability of their agent through a model based on a variational autoencoder that extracted task-relevant information from visual inputs into a latent space. Thus, the authors forced the agent to heed useful state information, yielding a robust policy for pushing tasks with multiple and random object shapes and sizes.

4.1.7. Multiple tasks

Occasionally, researchers do not focus on a single use case, but evaluate their policies in multiple scenarios. In these situations, it is essential to find common points or connections among the different skills, so that the control policy is easily transferable across tasks [120]. This subsection gathers these studies.

As in assembly and insertion tasks, here, researchers also highlight the need to improve the performance of policies, reduce their training times and extend their generalization capability. But although similar approaches are often adopted to address these concerns, these strategies are not always the same (see Fig. 5).

Akinola et al. [121] focused on improving agent precision in insertion and stacking tasks. Faced with the potential occlusions that may arise with a single camera, the authors proposed an approach combining multiple uncalibrated static cameras that, without the need for explicit 3D representations, obtained low error rates in these precision-based tasks.

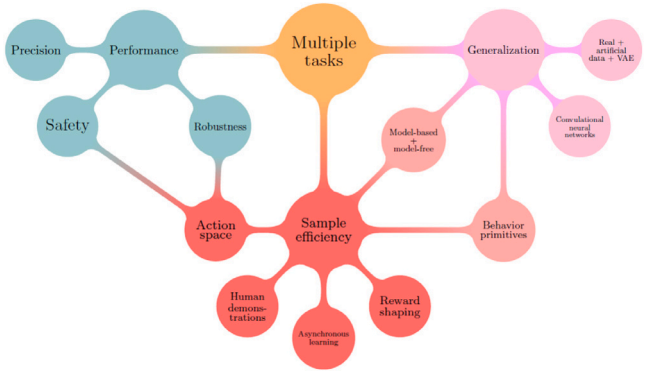


Fig. 5. Main themes addressed by multiple tasks studies.

Agent performance, in particular safety and robustness, were also addressed in [122], although indirectly. In this case, the authors focused chiefly on improving sample efficiency. To this end, they underlined the relevance of action space choice. They proposed a variable impedance control in the end-effector space and argued how this control matched the task characteristics and simplified exploration, and improved robustness to perturbations by exploiting the constraints and contacts of tasks such as door opening and surface wiping. Inspired by this study, in a more recent research, many of the same authors proposed an encoder–decoder model for learning contact-rich tasks in a latent action space [123]. Subsequently, the latent actions of the policy were mapped back to the original action space. By maintaining dynamic consistency, the agent explored more easily in the latent space, resulting in faster convergence in the original action space. In addition, the authors proposed both an online and offline variant of the method, where the action representation was learned through expert policy experiences.

In contrast, other studies used human demonstrations to improve sample efficiency. This is the case of [124], in which a high-dimensional multi-fingered robot was used in tasks such as in-hand manipulation, door opening or tool use. Specifically, the authors proposed to incorporate demonstrations into policy gradient methods, which resulted in learning the tasks with no need for reward shaping. Although they did not deploy the policy in a real environment, this study was further continued in [125]. This time, the authors transferred the policy into a real multi-fingered robot and evaluated it in valve rotation and door opening tasks. Balakuntala et al. [126] also used expert knowledge with sparse rewards in contact-rich tasks, namely writing, surface wiping and peeling.

Reward shaping for learning contact-rich tasks is non-trivial, and yet it is the primary determinant of agent performance. Consequently, there are already authors who choose to simplify reward engineering and apply sparse rewards by leveraging initial trajectories, e.g., through human demonstrations. Although reward functions are usually linked to specific tasks, the application of sparse rewards according to target states can simplify the formulation of the task.

Conversely, other authors opt for a combination of dense and sparse rewards. Vulin et al. [127], for example, used a dense intrinsic reward to guide exploration and a sparse extrinsic reward to finalize task attainment. In order to encourage exploration and improve sample efficiency, the intrinsic reward was based on the interaction forces between the robot and the objects to be manipulated. Prioritizing states and trajectories with higher contact forces by using a contact-priority repetition buffer, physical interaction was incentivized. However, the force threshold needed to be high enough to prevent the intrinsic reward from being false due to sensor noise. This may limit the application of the approach to tasks requiring soft contact.

Other approaches to speed up learning times and reduce sample complexity focus on asynchronous learning. This is the case of Zhang et al. [128], who worked on policy regularization by interleaving policy learning and model learning.

Generalization is another concern covered by studies that evaluate their approaches in multiple testing scenarios. Guo et al. [129] used a convolutional neural network to reduce the dimensionality of the image input information and extract the essential feature points of the task to obtain a robust policy for different poses in insertion and book placement tasks. However, they only assessed their policy in simulation. Kim et al. [130], on the other hand, considered not only the generalization but also this simulation-reality gap that may occur when deploying the policy on a real robot. To prevent this from happening, the authors employed a generative model that converted real images into simulation images. In this way, the simulation results were easily transferable to reality. For generalization and multiple tasks learning, the authors also employed a variational encoder–decoder to extract the latent vectors containing key information about the state.

Lastly, there is some research that attempts to improve sample efficiency and generalization capacity simultaneously. This is the case of Chebotar et al. [131], who integrated fast model-based updates into a model-free framework of path integral policy improvement to perform corrective updates in the face of unknown dynamics of the environment. However, the proposed technique required restoring the environment to consistent, non-random initial states. This, in turn, called for further improvements of its generalization capability. In contrast, Nasiriany et al. [132] leveraged a library of behavior primitives to create a hierarchical framework that selected a primitive type and its corresponding parameter. This approach also helped the authors generate task sketches, boosting sample efficiency and generalization through the transfer of control policies from a source task to a semantically similar task variant, reducing learning by up to five times compared to learning from scratch.

Performance, sample efficiency, and generalization are once again the most recurrent themes. However, the way of addressing these issues expands with respect to the approaches analyzed in Section 4.1.1. Among others, the adoption of state–action latent spaces for both sample efficiency improvement and generalization or the use of real data during simulation learning for a subsequent zero-shot transfer are two of the most noteworthy ideas.

4.1.8. Other tasks

This subsection encompasses all those contact-rich tasks studies whose testing scenarios do not fall into any of the previously defined categories. Despite the heterogeneity of the use cases, all studies analyze performance-related issues, namely safety and uncertainty and contact-stability.

Kuo et al. [133] focused on alleviating the risk of causing damage to the environment by intense or unexpected contacts. To obtain safe contact manipulation, they presented a model-based algorithm that associated a probabilistic predictive control with model uncertainty in mixing and scooping tasks. In this way, the agent's actions were adjusted according to the learning progress and were limited in those states where there was greater uncertainty. However, the evaluations were performed in a restricted action space, so the authors concluded that, in larger workspaces, their approach might penalize learning efficiency for exploring with caution. In [134], by contrast, instead of modeling the environment and working in those states with low uncertainty, the authors analyzed how the controller configuration influences task performance with contact uncertainties. To this end, a policy based on a model-free algorithm was proposed that allowed joint control of impedance and desired position in joint space. The simultaneous control of both variables made the policy robust, mainly to contact uncertainties such as friction, stiffness and contact location. Luo et al. [135] also proposed an impedance-based control strategy that combined a Q-learning agent to optimize online stiffness and damping parameters with a Maxwell stress model to deal with time-varying contact forces in the stripping of molten metal surface. Compared to

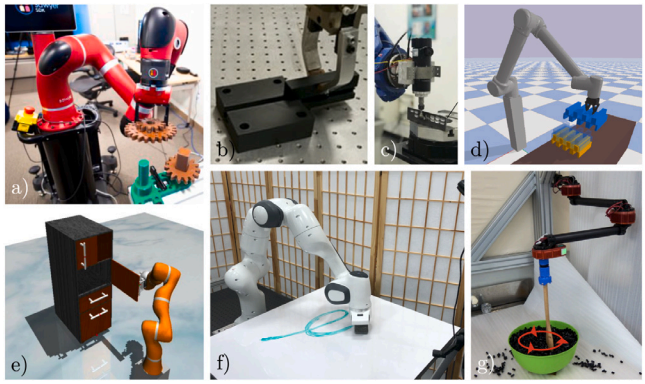


Fig. 6. Rigid object manipulation tasks: (a) assembly [45]; (b) disassembly [98]; grinding [103]; (d) stacking [106]; (e) door opening [111]; (f) surface sliding [122]; (g) mixing [133].

the traditional constant impedance control model, the usage of the intelligent agent provided a greater adaptive capability to the process, although the authors concluded that more comparative experiments should be performed.

In line with [134,135], although RL provides the ability to adapt to varying dynamics of the environment, in order for robots to perform interaction tasks with human-like dexterity and obtain characteristics such as stability and/or robustness to contact uncertainty, control methods that enable the manipulator to have compliant behavior must be employed. Compliant robot behavior can be achieved by passive mechanical compliance built into the manipulator, or by active compliance control implemented in the servo control loop, e.g., hybrid force/position or impedance control [136]. In this regard, Anand et al. [137] focused on comparing two control strategies, namely variable impedance control and hybrid force/motion control, when a dynamics model is available in surface sliding tasks. For the former, strategies were learned for the parameters of an adaption law. For the latter, the framework was used to learn strategies directly for its damping and stiffness parameters. While faster convergence of the desired force was obtained with variable impedance control, a significant improvement in force tracking error was obtained with hybrid force/motion control.

Clearly, the choice of control method is key to performing a successful contact-rich task. This choice also determines which actions the RL agent should perform.

Fig. 6 shows some examples of the mentioned rigid object manipulation scenarios, and Table 3 collects the main features of all reviewed papers on rigid object contact-rich manipulation.

4.2. Deformable object manipulation tasks

In the case of handling deformable objects, unlike rigid pieces, there is no clear manipulation state representation. The shape of deformable objects varies along and between trajectories, and is challenging to characterize due to their high-dimensional configuration spaces.

Sánchez et al. [138] proposed an object classification according to its geometry. Specifically, they defined four types: (1) uniparametric objects called linear objects; (2) biparametric objects with no compression strength, (3) biparametric objects that present a large strain, called planar objects; and (4) triparametric objects, also known as solid or volumetric.

Despite the application of RL in deformable object manipulation, there is scant heterogeneity among the use cases. Therefore, all reviewed studies have been divided into rope folding tasks, directly related to uniparametric objects; clothing/fabrics folding tasks, dealing with biparametric/planar objects without compressive strength; tensioning/cutting tasks, where medical application in biparametric objects with large deformation stand out; and volumetric object manipulation tasks, where testing scenarios are more diverse.

4.2.1. Rope folding tasks

Rope folding is one of the basic tasks in deformable object manipulation. Although it is a two-dimensional task where linear deformable materials are generally employed, the complexity of achieving specific rope configurations makes it an appealing task for researchers. Despite its similarity to pick and place tasks, its technical challenge lies in its high-dimensionality, where the movements made by the robot at one end of the rope may affect other pieces of the rope that had already reached their target.

As in rigid object manipulation, both sample efficiency and generalization are again the main concerns within this domain. Han et al. [139], for instance, proposed a sample-efficient model-based RL approach to place a linear deformable rope from an initial to a target position in a two-dimensional workspace using a dual-arm robot. For this purpose, they modeled the rope configurations with a stochastic model. In spite of its successful validation in reality, the dynamics of these objects are often complex, and despite being linear, different materials and even environmental factors can render their modeling challenging. Consequently, many other research studies chose to address this use case through model-free approaches [140]. Lin et al. [141] improved sample efficiency without requiring explicit state estimation or access to the ground-truth state of the environment. They proposed a reward function based on a goal image classification. In addition, to further speed up the convergence of the algorithm, they balanced the positive and negative rewards received by the agent and filtered out those transitions that the authors considered to have a higher chance of being false negatives.

Attending to generalization capability, Laezza and Karayiannidis [142] formulated a generalizable shape control problem for materials with elastoplastic properties using the rope curvature state representation. However, the approach was simplistic as it restricted the motion of the robot gripper along a straight line.

Other studies, on the opposite, tried to address sample efficiency, generalization and the deployment of the learned policy into reality, simultaneously. Indeed, maintaining performance despite the simulation-reality gap emerges as a major concern among researchers [140,142]. This is because physical properties and parameters are not easily modelable in the simulation engine. Inspired by studies that focused on action space to improve sample efficiency, Wu et al. [143] used a conditional action space through which, by learning a placing policy from scratch, they were able to learn also a picking policy by finding a pick-up point that maximized the object placement value and sped up agent convergence in rope and clothes folding tasks. In addition, they demonstrated that by using simple domain randomization, the policy could be transferred from a simulator to a real robot.

4.2.2. Clothing and fabrics folding tasks

Accurate manipulation of fabric or garment parts is particularly challenging due to their many degrees of freedom and the underlying properties that affect their dynamics. These challenges can affect different perspectives of RL, such as performance, and the ability to learn and generalize to other environments or even overcome the simulation-reality gap. Therefore, once again, studies related to fabric and garment manipulation focus on improving these aspects.

Petrik et al. [144] approached performance from the accuracy point of view. Specifically, they focused on fabric strip folding using feedback-based control. However, they used a simulation model to match the behavior of real fabric strips, which limited the application of their approach to other fabrics with different properties. In addition, the approach required the manipulated object to be fully visible. This is a condition demanded by many studies, as they base their state

Table 3
RL properties of each study in rigid object manipulation.

Task	Reference	Year	Control method	RL taxonomy	Algorithm	Learning	Observation space								Action space							Reward	Simulation/real-world
							image	joint force/torque	external force/torque	joint angle	joint velocity	end-effector pose/pos	end-effector velocity	others	joint force/torque	applied force/torque	joint angle	joint velocity	pose/pos (Cartesian)	stiffness/damping	others		
Assembly/ insertion	[42]	2017	P/F	VB	Q-learning (RNN)	S			X			X				X			X			S	R
	[64]	2017	V, C/I	AC	DDPG	A		X		X	X			X				X				B	B
	[58]	2018	C/I	AC	DDPG	S		X						X					X	X		B	R
	[80]	2018	PV	MB	iLQG	A				X	X	X	X				X	X			X	D	B
	[43]	2019	P	VB	DQN	A			X					X					X			D	R
	[45]	2019	P/F	MB	iLQG	S				X	X	X	X		X							S	R
	[47]	2019	V	MB-AC	Guided DDPG	S			X			X				X						D	B
	[50]	2019	P	AC	DDPG	S		X	X	X		X					X					B	R
	[56]	2019	F	MB	GPS	S				X	X				X							D	B
	[63]	2019	V	AC	DDPG	A	X	X		X	X	X						X				S	B
	[72]	2019	V	AC	DDPG	A								X							X	D	B
	[75]	2019	C/I	VB	Hierarchical SARSA	S								X					X			S	B
	[76]	2019	F	MB-AC	Model-driven DDPG	A			X			X							X			B	B
	[81]	2019	C/I	AC	TRPO	S	X	X							X							B	B
	[87]	2019	P	MB	MPC	S			X										X			D	R
	[38]	2020	P/F	AC	DDPG	S			X			X				X					X	S	S
	[39]	2020	A	VB	Q-learning	S			X			X								X		S	R
	[51]	2020	P/F, A	AC	SAC	S						X		X		X			X			D	B
	[57]	2020	C/I	MB-VB	MLE DQN	S				X	X				X							D	B
	[61]	2020	F	AC	DDPG	A		X		X							X	X			X	D	R
	[62]	2020	C/I	AC	SAC, TD3	A	X		X			X							X			B	R
	[71]	2020	V	AC	SAC	A		X		X				X				X				B	B
	[73]	2020	P	AC	A3C	S	X												X			B	B
	[74]	2020	V	MB	PILCO	S						X	X						X		X	D	B
	[77]	2020	C/I	AC	DQN, DDPG	A			X			X		X		X			X			B	R
	[78]	2020	C/I	AC	SAC	A			X			X	X	X					X			D	B
	[83]	2020	P	AC	PPO	S	X		X	X				X					X			D	B
	[89]	2020	C/I	AC	PPO	S	X		X			X		X					X			B	B
	[85]	2020	C/I	MB-AC	MB SAC	S			X					X		X						B	R
	[93]	2020	P, C/I	AC	PEARL	A	X		X			X							X			S	B
	[44]	2021	V	AC	DDPG	A			X			X									X	D	B
	[46]	2021	C/I	AC	SAC	S			X			X		X					X	X		D	B
	[48]	2021	C/I	VB	NAF	S	X		X										X			S	S
	[49]	2021	C/I	PB	CEM	S				X	X				X							D	B
	[54]	2021	C/I	VB	DQN	S	X		X			X									X	S	S
	[55]	2021	C/I	AC	TD3	S			X			X		X					X			B	R
	[59]	2021	C/I	AC	DDPG	S			X			X	X		X				X			D	S
	[60]	2021	C/I	VB	Q-learning	S			X							X			X			B	R
	[65]	2021	C/I	MB-AC	MB DDPG	A			X					X					X			D	B
	[66]	2021	F	MB-VB	DDQN	A			X							X						S	R

(continued on next page)

Table 3 (*continued*).

Task	Reference	Year	Control method	RL taxonomy	Algorithm	Learning	Observation space								Action space							Reward	Simulation/real-world
							image	joint force/torque	external force/torque	joint angle	joint velocity	end-effector pose/pos	end-effector velocity	others	joint force/torque	applied force/torque	joint angle	joint velocity	pose/pos (Cartesian)	stiffness/damping	others		
	[67]	2021	P/F	AC	SAC	A								X					X		X	D	B
	[68]	2021	P	VB	NAF	A			X										X			B	R
	[70]	2021	P/F	AC	SAC	A								X					X		X	D	B
	[79]	2021	P	AC	PPO	S			X					X					X			B	B
	[82]	2021	F	AC	SAC	A	X		X						X				X			D	S
	[84]	2021	C/I	-	Adversarial IRL	A								X		X				X		D	B
	[86]	2021	V	MB	-	S						X		X							X	D	B
	[94]	2021	C/I	MB	-	A			X			X				X			X			-	B
	[52]	2022	C/I	AC	DDPG	S						X				X						B	B
	[53]	2022	P	AC	DDPG	S	X		X	X				X					X			D	B
	[90]	2022	C/I	AC	SAC	A			X					X							X	B	B
	[91]	2022	V	VB	TS-MP-DQN	S			X				X								X	B	B
	[95]	2022	C/I	AC	DDPGfD, AWAC	A			X			X	X						X			S	R
Disassembly	[96]	2019	P	VB	Q-learning	S			X			X		X					X		X	B	S
	[97]	2020	C/I	VB	-	S																-	–
	[98]	2021	P	AC	DDPG, TD3	S			X			X		X					X			B	B
Polishing/ grinding	[102]	2018	P	VB	Q-learning	S													X			D	S
	[103]	2020	F	MB	-	S								X							X	D	B
	[104]	2022	C/I	-	-	S			X					X						X		D	B
Stacking/ unstacking	[105]	2020	V	AC	D4PG	A	X		X										X			D	R
	[107]	2021	C/I	AC	NAC	S				X	X									X		D	R
	[106]	2022	PV	AC	TD3	A			X	X				X				X			X	D	B
Doors/ drawers opening	[109]	2017	F, V	PB	PI^2	A	X								X			X				D	R
	[110]	2017	F, V	PB	PI^2	A				X	X	X	X		X			X				D	B
	[111]	2017	C/I	PB	ICL and PI^2	S				X					X							D	B
	[113]	2017	V	PB	Asynchronous NAF	S				X	X	X	X	X				X				B	B
	[114]	2018	P	MB	CORL	A			X					X			X					D	R
	[115]	2022	P	PB	DAPG	A			X			X		X							X	R	B
Pushing	[117]	2019	F	AC	DDPG	S			X			X		X		X						S	B
	[118]	2019	P	AC	D4PG	S			X	X	X						X					D	B
	[119]	2022	V	AC	SAC	S	X														X	B	B
Multiple	[124]	2017	P	PC, AC	NPG, DAPG, DDPG	A				X				X			X					B	S
	[131]	2017	P	MB-PB	PILQR	S				X	X	X	X	X			X					D	B
	[129]	2018	P	MB	LQG	S	X			X	X	X	X				X					D	S
	[122]	2019	C/I	AC	PPO	S	X					X	X	X	X							B	B
	[125]	2019	P	PB	DAPG	A				X				X			X					D	B
	[128]	2019	F	MB MB-AC	ME-MPO, ME-TRPO, ME-PPO	S				X	X	X				X						D	B
	[121]	2020	P	VB	QT-Opt	A	X							X					X		X	B	S
	[123]	2021	F, P	AC	SAC	A				X		X		X	X		X					D	S

(*continued on next page*)

Table 3 (*continued*).

Task	Reference	Year	Control method	RL taxonomy	Algorithm	Learning	Observation space								Action space							Reward	Simulation/real-world
							image	joint force/torque	external force/torque	joint angle	joint velocity	end-effector pose/pos	end-effector velocity	others	joint force/torque	applied force/torque	joint angle	joint velocity	pose/pos (Cartesian)	stiffness/damping	others		
	[126]	2021	C/I	AC	SAC	A	X	X	X	X					X		X					B	R
	[127]	2021	F	AC	DDPG	S			X	X	X			X	X							S	S
	[130]	2022	P	AC	SAC	S	X					X	X						X			B	B
	[132]	2022	P	AC	SAC	S						X		X					X		X	D	B
	[134]	2020	C/I	AC	DDPG	S				X	X				X		X				X	D	B
Other	[133]	2021	V	MB	pMPC	S								X				X				D	B
	[135]	2021	C/I	V	Q-learning	S								X						X		D	S
	[137]	2022	C/I, F/P	MB	PILCO	S			X			X	X							X		-	B

Control methods: position control (P), velocity control (V), force control (F), hybrid/parallel position/force control (P/F), compliance/impedance control (C/I), admittance control (A). **RL taxonomy**: model-based (MB), value-based (VB), policy-based (PB), actor–critic (AC). **Learning**: self learning (S), assistive learning (A). **Reward**: sparse (S), dense (D), both (B). **Simulation/real-world**: simulation (S), real-world (R), both (B).

representation or reward on visuomotor servoassistance or location-based point chain [145]. Both are guides with strategic gripping and pulling points that provide visual cues, but their performance is conditioned by possible occlusions. This can hamper learning due to the high dimensionality and multiple configurations of the parts to be manipulated, leading to sub-optimal policies.

In order to avoid dependence on visual inputs and improve sample efficiency, Verleysen et al. [146] integrated tactile sensor cells into a textile part to use the obtained signals as a reward when the sensors detected whether the part was bent. However, this implied the need to integrate the sensors into the part, which in certain use cases may not be feasible. Amadio et al. [147], instead, proposed another approach in which they represented the motion of a robotic arm as the symmetrization of the primitive motion of another serial manipulator, thus reducing the number of parameters. This offered rapid convergence during the learning process, but limited its scope to straightforward tasks.

In fact, generalization and the deployment of the policy learned into reality are among the main concerns encountered by researchers when it comes to the manipulation of fabrics and garments [148]. For example, studies such as [149] showed some errors during fabric grasping and even inaccurate wrinkling movements after the application of domain randomization and transfer to the real world.

Ebert et al. [150] proposed a model-based framework for sensory prediction, particularly visual, that attempted to generalize to tasks never seen in the real world through three phases: unsupervised data collection, training of the predictive model, and planning-based control through the model. However, planning was only effective on a short-term basis. In addition, as in previous cases, the system needed all objects to be visible during task execution. Hoque et al. [151] extended the prior framework by adding support for depth detection. Thus, they reduced the long data collection time of the base framework and increased the execution horizon of the task. Although their proposal to use RGBD data resulted in significant improvements in the success rate, the performance in reality was limited, mainly due to the difference in dynamics between the simulated and real environments. In contrast, Zhou et al. [152] succeeded in training an offline policy in a latent action space, which not only generalized well within the data set they used for learning, but provided robust generalization in out-of-distribution actions when the Q function generalized without significant extrapolation error.

Nevertheless, generalization and transfer of policies to the real world remain a research topic within cloth and garment manipulation. While visuomotor servoassistance may provide support to train the policies, the sole use of visual information seems to generate a bottleneck that risks sample efficiency and even limits the applicability of the policies in a real robot.

4.2.3. Tensioning and cutting tasks

Tensioning and cutting tasks of biparametric elements are mainly bounded in healthcare domain, and emulate tensioning and cutting of human tissues or gauze. Surgical scissors are one of the most effective tools for cutting soft and deformable tissues. These tissues are highly nonlinear and, for cutting, generally need to be tensioned. However, the direction and magnitude of the cutting forces vary as the cut proceeds, so these forces must be adapted to improve reliability and accuracy of the cut [153]. Therefore, the studies reviewed emulate hypothetical intraoperative situations, where many them focus on accuracy.

This is the case of Thananjeyan et al. [154], who trained an agent for two-dimensional surgical tensioning and cutting using a finite element tissue simulator. Although they obtained accurate and sensitive results, the simulation-reality gap between the robot and tool models used in learning and the actual ones resulted in occasional entanglement and even tissue deformation during cutting. In addition, the policy only chose a fixed pinch point during the entire cutting process, regardless of the complexity of the cutting pattern. This limited the applicability of the approach when complex cutting patterns were required. Taking this study as a starting point, Nguyen et al. [155,156] proposed an autonomous multiple pinch point tension planner for surgical soft tissue cutting tasks. Their results improved accuracy, but the number of tension directions was constrained to four, which could also limit its application in real-world use cases.

Meanwhile, other studies focused on improving sample efficiency and selecting useful features of tensioning and cutting tasks through human demonstrations. Shin et al. [157], for example, presented a control framework that combined model-based RL with learning from demonstration to understand the dynamics and automate the soft tissue manipulation task. Krishnan et al. [158], on the other hand, focused on

segmenting gauze tensioning and cutting tasks using a Da Vinci robot into shorter subtasks by combining the exploration and demonstration paradigms and thereby assigning local reward functions that favored algorithm convergence. Pedram et al. [159], lastly, used an approximate linear Q-learning method in which human knowledge contributed to selecting useful, albeit simple, tissue manipulation features, while the algorithm learned to perform optimal actions and accomplished the task. However, they all concluded that their work should be further extended to less constrained three-dimensional action spaces with higher dimensionality.

4.2.4. Volumetric object manipulation tasks

Studies on deformable volumetric objects are tested in specific and diverse scenarios. Despite this heterogeneity in the use cases, all studies aim to improve the performance of the policy, either from the precision, efficiency or physical interaction point of view.

Luo et al. [160] were among the first to employ RL to address an industrial challenge with deformable objects. The authors specifically focused on the precision aspect. They developed a policy search framework covering robotic assembly combining rigid and deformable parts. As a result, they succeeded in proving a position- and velocity-controlled robot with haptic feedback to insert a rigid peg into a non-linear deformable part with a hole. However, the system lacked a vision system, and its performance declined when the peg was not close to the hole.

Gonnochenko et al. [161], on the other hand, focused on how grasping could affect the efficiency of the process in unloading bags from a cart and placing them in a given order on a table. The bags had varying shapes, moving centers of gravity and different weights. Although the authors cared mainly about the design of a specific gripper, they used RL to identify the best configuration of the end-effector to maximize gripping success.

Meanwhile, in [162,163], the authors focused on improving the performance and optimizing the interaction between the robot and a supple environment whose dynamics were unknown. The former addressed the challenge of shaping an elastoplastic mass by using a novel elastic end-effector to roll the dough in different lengths through an RL frame in which the agent iteratively improved the transition model by scanning to compensate for ill-defined models. The latter, on the other hand, employed a full state–space equation that considered both the desired trajectory of the robot, commanded by a Q-learning agent, and the dynamics of the environment and position parameters to solve the control problem.

Fig. 7 shows some examples of the mentioned deformable object manipulation scenarios, and Table 4 collects the main features of all reviewed papers on deformable object contact-rich manipulation.

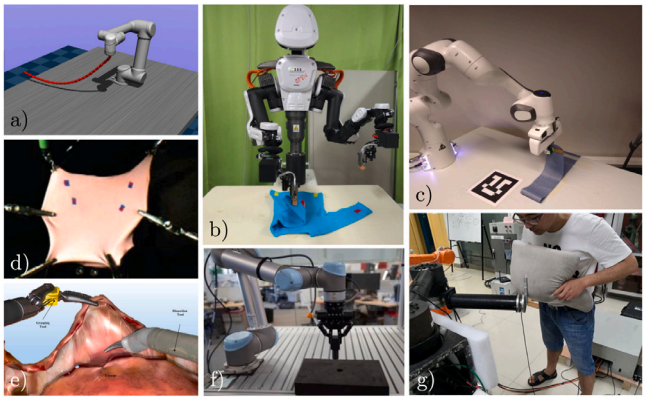


Fig. 7. Deformable object manipulation tasks: (a) rope folding [140]; (b) clothing folding [145]; (c) fabrics folding [144]; (d) tissue tensioning [157]; (e) tissue cutting [159], (f) peg-in-hole [160], (g) cushion touching [163].

5. Analysis

Current RL studies on contact-rich manipulation tasks point to multiple avenues of research. Overall, these topics are framed around improving policy performance, which in turn may encompass different domains, enriching sample efficiency to reduce learning times, or increasing generalization ability and reducing the simulation-reality gap. Yet, these lines of research do not receive equal emphasis. The following are those trends as well as possible open challenges that, for the authors, could be the most relevant both from a theoretical and practical standpoint.

5.1. Testing scenarios

Assembly and insertion tasks are currently the main use cases commonly used to evaluate the performance of RL policies in rigid object manipulation. This can be clearly seen in the diagram in Fig. 8a (86 papers considered), where 56% of the papers reviewed are related to this topic. Note that there are more case studies than reviewed papers as there are articles that cover multiple tasks.

Within assembly, the peg-in-hole task is the benchmark case study, being generalizable to daily or even industrial activities, from plugs, and USB connectors to car refueling nozzles, and clearly continues to present challenges in robot manipulation. This task is generally divided into the search and insertion phases, the latter being where the robot must align the axis of the peg with that of the hole and insert the peg to the desired depth. Commonly, when the robot holds the peg away from the hole, the RL agent's observation is based on two types of data, namely vision and force. Vision data contains the relative pose between the peg and the hole, although its accuracy may be subject to possible occlusions in the environment. Force, on the other hand, directly indicates the magnitude of contact during assembly, although for this to happen, contact must take place. In this type of task, an RL agent needs to have both pieces of information available. However, depending on the task phase, the inputs' weight should be modified, giving greater importance to the contact forces when the assembly process reaches a blind spot.

The robustness of current policies to difficult-to-model contact forces also remains an open field of study in assembly for researchers. Unlike tasks such as door or drawer opening, whose challenges are based on finding the ideal axis configuration within a constrained workspace, or manipulating a translation joint so that the next joint unlocks and whose observations are more linked to the state of the robot joints, contact forces are also a common challenge in grinding, polishing, wiping or pushing tasks. Effects such as chatter [164] or non-constant contacts still often prevent policy deployment in real industrial environments. In these cases, it is convenient to use model-free algorithms and work on designing rewards that foster smooth and stable contacts. Although the use of complex models for tasks such as grinding or polishing can provide better results, the RL agent would be limited to machining components with the same properties as the model.

This quandary once again highlights the need to address the generalization capability of RL agents, which becomes even more relevant in tasks such as disassembly, where a product may have multiple models, and even their physical state at the end of life will rarely be similar, or in those applications where the RL agent must be able to perform multiple tasks. In these latter applications, the semantics and identifying commonalities between the different tasks are fundamental. Therefore, enriching the agent's observations with multiple sources of information can be the first step. Moreover, contextual meta-RL also seems to offer promising results by facilitating fast task adaptation from a few samples in dynamic environments [165]. However, obtaining

Table 4
RL properties of each study in deformable object manipulation.

Object type	Task	Reference	Year	Control method	RL taxonomy	Algorithm	Learning	Observation space								Action space						Reward	Simulation/real-world
								point chain (position)	image	joint position	end-effector pose/pos	end-effector velocity	force/torque	target position	others	joint angle	pose/pos (Cartesian)	gripping	pulling	force/torque	others		
Uniparametric	Rope folding	[139]	2017	P	MB	PILCO	S	X			X				X		X					D	R
		[140]	2019	V	AC	PPO, DDPG, IMPALA, SAC	S	X						X							X	D	S
		[141]	2019	P	AC	DDPG	S		X								X					S	S
		[142]	2020	V	AC	DDPG, TD3, SAC	S	X					X								X	D	S
Uniparametric Biparametric (no compression strength)	Rope folding Clothing/ fabrics folding	[143]	2020	P	AC	SAC	S	X	X									X	X			D	B
Biparametric (no compression strength)	Clothing/ fabrics folding	[149]	2018	V	AC	Modified DDPG	S		X	X	X							X			X	S	B
		[150]	2018	P	MB	MPC	S		X									X	X		X	D	R
		[144]	2019	P	MB	Black DROPS	S				X		X		X		X				X	B	B
		[147]	2019	P	PB	REPS	A								X	X						D	B
		[145]	2020	P	PB	DPN, DDPN	S		X									X	X			D	R
		[146]	2020	P	VB	Fitted Q-learning	S			X					X	X						S	R
		[148]	2020	V	AC	DDPG	A	X			X	X		X	X			X			X	S	S
		[151]	2020	P	MB AC	Imitation learning, Visuospatial Foresight, DDPG	S/A		X									X	X			B	B
		[152]	2020	P	AC	BCQ, TD3	S				X		X		X		X					B	R
Biparametric (large strain)	Tissue tensioning	[155]	2019	P	AC	TRPO	S	X							X		X					S	S
		[157]	2019	P	MB	MPC	A	X	X		X				X		X					D	S
		[159]	2020	P	VB	Q-learning	A	X									X					D	S
	Tissue cutting	[154]	2017	P	AC	TRPO	S	X							X		X					S	B
	Tissue cutting/ tensioning	[156]	2019	P	AC	TRPO	S	X							X		X					S	S
	Gauze cutting/ tensioning	[158]	2019	P	VB	Q-learning	A		X		X		X				X					D	R
Triparametric	Peg-in-hole	[160]	2018	F	MB	Mirror descent GPS	S			X	X	X	X		X					X		D	R
	Bag manipulation	[161]	2020	P	AC	SAC, PPO	S								X		X					D	B
	Dough rolling	[162]	2021	P	MB	-	S								X						X	D	R
	Cushion touching	[163]	2021	P	VB	LQR, Q-learning	S				X		X				X					D	B

Control methods: position control (P), velocity control (V), force control (F), hybrid/parallel position/force control (P/F), compliance/impedance control (C/I), admittance control (A). **RL taxonomy**: model-based (MB), value-based (VB), policy-based (PB), actor–critic (AC). **Learning**: self learning (S), assistive learning (A). **Reward**: sparse (S), dense (D), both (B). **Simulation/real-world**: simulation (S), real-world (R), both (B).

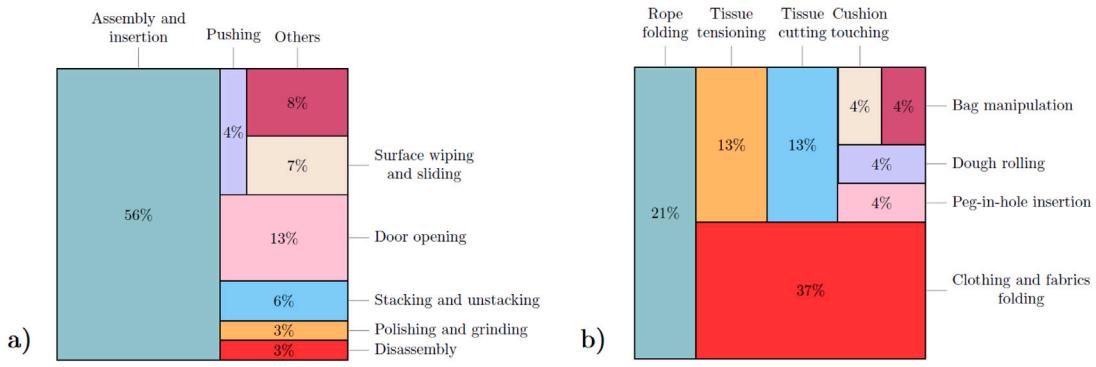


Fig. 8. (a) Rigid object manipulation tasks; (b) deformable object manipulation tasks.

rich, dynamic representations for fast adaptation beyond simple benchmark problems is not a simple task. As task complexity increases, the similarities among tasks could be reduced, thus modifying the network representations needed to solve each task. In this regard, a recent study advocates the introduction of a neuromodulated network to increase the ability to encode rich and flexible dynamic representations and, consequently, be able to modify its policy [166].

Attending to the manipulation of deformable objects, more than one-third of the studies deal with textile manipulation (see Fig. 8b, 24 papers considered) and are knowledge-based approaches [167]. Through the relationship between robot manipulation and garment shape, the authors focus on achieving a target fabric configuration by employing either vision directly or a strategic guide of gripping and pulling points, called a location-based point chain. Still, this information does not always avoid one of the major challenges of manipulating deformable objects. The state representation is too complex due to the countless degrees of freedom of the parts, and in tasks such as clothing or fabric folding; the observation can also be partial. Therefore, a typical approach is to model these tasks as continuous state partially observable MDPs (POMDPs), where the agent makes decisions based on belief states.

5.2. Reinforcement learning algorithms

According to the taxonomy of RL algorithms described in Section 3.1, the most predominant type of algorithms in the reviewed papers are model-free (see Table 5). Although some approaches employ models for learning, a large share of authors report hardship in generating accurate models due to changing dynamics of the environment and hardly modelable collision forces in rigid object manipulation in contact-rich tasks. For instance, in peg-in-hole tasks, a slight misalignment can cause high friction between parts and even affect the task completion. In general, potential friction and jamming are difficult to model. This can be extrapolated to other handling tasks where the contact forces between the robot and the object to be manipulated are not linear, making it unlikely to find optimal policies with model-based methods. Therefore, the vast majority of researchers resort to model-free algorithms.

Within the model-free category, value-based and actor–critic algorithms prevail. However, while the former tends to use discrete action spaces, which limits their application in high-dimensional problems, the latter seems to be the most efficient when dealing with contact-rich tasks on rigid objects. Among them, DDPG [168], SAC [169] and PPO [170] are the most employed algorithms (see Fig. 9a, 53 actor–critic algorithms considered). Whereas PPO is an on-policy algorithm that updates the policy, ensuring that the deviation from the previous policy is relatively small, DDPG and SAC are off-policy algorithms, that use a replay buffer memory to store experiences and reuse the most valuable information for efficient training. DDPG is a deterministic algorithm that uses deep function approximators to learn the policy and estimate the value function in continuous, high-dimensional action spaces. SAC, on the other hand, aims to optimize the maximum entropy together with the discounted long term return. The maximum entropy helps to enhance the exploration where it is needed.

As far as contact-rich manipulation of deformable objects is concerned, although many approaches employ models for learning the manipulation of deformable linear objects [139], or for simulating the folding of garments and fabrics [144], actor–critic algorithms are still the most commonly implemented. In this case, DDPG and SAC are again the most employed algorithms in the studies, while TRPO [171], which optimizes the policy based on the KL divergence between the old and the updated parameters and present in those papers dealing with the manipulation of biparametric objects in the healthcare domain, emerges as the third one (see Fig. 9b, 18 actor–critic algorithms considered).

5.3. Safety

"How do we formulate safety specifications to incorporate them into RL, and how do we ensure that these specifications are robustly satisfied throughout exploration" are the two main questions formulated by Ray et al. [172]. Generally speaking, safety is focused on exploring the unknown space safely, while robust control concentrates on guaranteeing the stability of a system for pre-specified bounded disturbances.

[173,174] are two comprehensive surveys addressing controller safety. Both surveys consider data-based (like RL) and model-based control theories. Data-based approaches try to use data to manage uncertainties and reduce the conservatism of the safe controller.

Brunke et al. [173] defined three safety levels. (1) Safety Level 1 promotes safety and robustness in RL, but does not guarantee hard safety constraints. (2) Safety Level 2 learns uncertain dynamics to improve performance safely. At this level, there are no hard safety guarantees, but safety issues probability can be estimated. Typically, prior knowledge is used, and uncertain dynamics are learned from the data. (3) Finally, Safety Level 3 provides safety certificates to the

Table 5
Types of algorithms employed in reviewed studies.

Type of object manipulation	Model-based	Model-based & model-free			Model-free		
		Value-based	Policy-based	Actor–critic	Value-based	Policy-based	Actor–critic
Rigid	18	2	1	5	15	7	44
Deformable	7	0	0	0	4	3	11

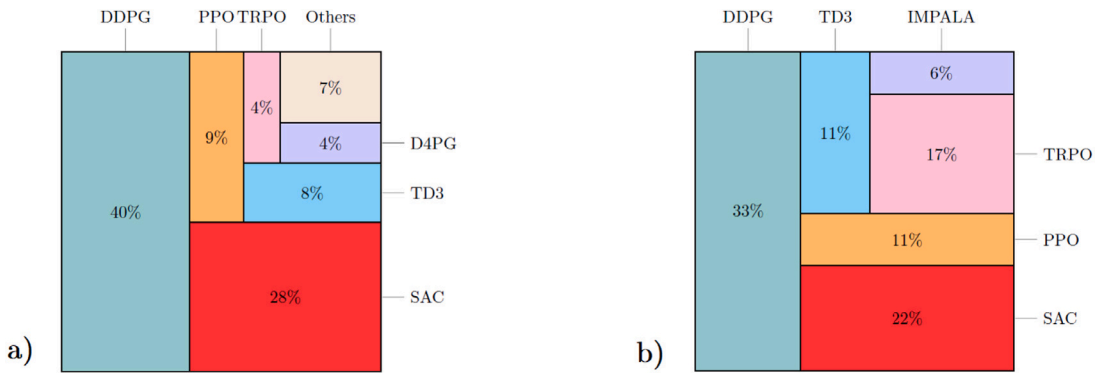


Fig. 9. Actor–critic algorithms employed in (a) rigid object manipulation tasks and, (b) deformable object manipulation tasks.

controller that does not consider safety constraints, which can be done, i.e., modifying the controller output.

Among all the papers reviewed, around 45% of the articles concerning rigid object manipulation consider safety-related strategies, while including research dealing with deformable object manipulation, it is slightly more than 38%. These safety strategies are approached from five main perspectives (see Fig. 10).

- 12% of the reviewed papers establish an episode termination reward. Generally, this reward is linked to possible collisions. In case of a collision, the agent receives a penalty, and the training episode is terminated [42,86] (Safety Level 1).
- 22% of the works perform the exploration at reduced velocity. In this approach, collisions are allowed without risk [53,84] (Safety Level 1).
- Besides, one paper that uses an uncertainty model was found [133]. This model helps to explore cautiously when uncertainty is high (Safety Level 2).
- Lastly, almost 100% of the papers that mention safety-related strategies use a restricted action space, either position (46%) or force (59%). [52,78], for instance, constrained the agent choices for force and position control parameters by imposing upper and lower limits. Thus, if an action exceeds the allowed threshold, the action is not executed, and the next action is waited for (Safety Level 3).

For measuring research progress on safe RL, in [172], the authors presented the safety benchmark suite Safety Gym, a new slate of high-dimensional continuous control environments. In the same vein, Brunke et al. [173] proposed a framework called Safe Control Gym. Both

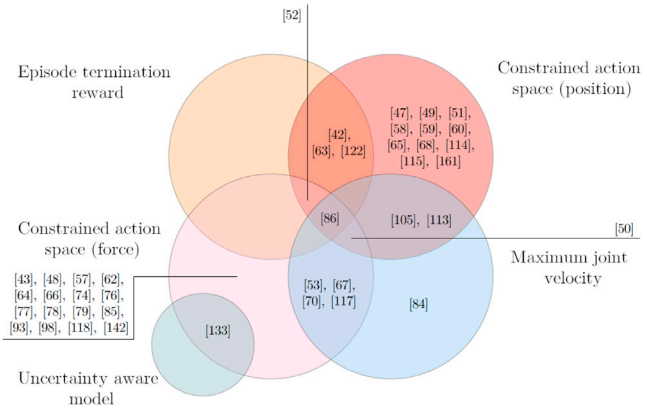


Fig. 10. Venn diagram of safety strategies employed in RL for contact-rich manipulation tasks.

frameworks are based on the most used Open Gym interface. However, their use has not been identified among the reviewed articles.

5.4. Assistive learning

RL is characterized by lengthy learning times. Training can take minutes to hours or days, even for relatively simple applications. The learning time is mainly influenced by the agent's exploration during training to augment existing knowledge through actions and interactions with the environment. A factor that can reduce the learning time is the available initial knowledge. In this sense, an increasingly noticeable trend is to provide prior knowledge to the agent to avoid learning from scratch.

Prior knowledge injection can be performed through human demonstrations. Specifically, about 35% of the papers reviewed rely on human demonstration-assisted learning. Just as humans learn through a combination of imitation and experience gained by interacting with the environment, robots can learn in the same way using the so-call apprenticeship learning [31]. In this regard, Braun et al. [175] highlight three approaches to provide a demonstration: (1) human motions can be recorded performing the task; (2) kinesthetic guidance can be provided by a human directly guiding the robot; (3) a human can provide a demonstration by telemanipulation the robot.

Notwithstanding, human demonstration injection is not the only existing approach to prove prior knowledge of the task to the agent. For instance, other studies leverage the initial movement plans provided by previously learned policies [123]. Similarly, in recent years, meta-learning has also begun to be used along with RL for contact-rich manipulation tasks. For instance, Yu et al. [176] created an open-source simulated benchmark to train meta-RL agents in multiple robotic manipulation tasks. Instead of considering each task independently, data from previous tasks are employed to acquire a learning procedure that can be later adapted to new tasks. In this way, the agent learns the underlying principles of correlated tasks [177]. Lastly, there are occasions when the design of a reward function may be too complex, lack high task knowledge, not be easily generalizable, and not be sufficient to evaluate a sequence of actions. Many studies harness the human experience in the process to avoid the reward engineering effort and incorporate it as part of the agent's learning. Inverse RL [84] or interactive RL [178] are approaches to extract the reward functions from the human demonstration or communicate with the human to improve its learning speed, respectively. Other approaches include, for example, cooperative inverse RL [179], where the human teaches the robot its reward function and both try to maximize it, or active reward learning [180], where the reward function is actively learned from the expert while the policy is refined.

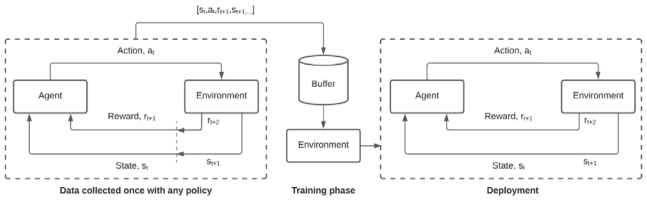


Fig. 11. Offline RL scheme [181].

5.5. Offline learning

Most of the reviewed articles are oriented toward online RL learning. This online RL approach allows the agent to interact with its environment and leverage the collected experience, either immediately or through a replay buffer, to update its policy. However, this approach has limitations, such as prohibitive times for exploration and sufficient data collection, narrow distributions of states that make the policy susceptible to small changes, or even the execution of potentially hazardous actions in case of training the policy directly in reality.

These limitations are not as common in supervised learning. The offline RL formulation resembles this type of learning since the agent no longer has the ability to interact with the environment, but rather receives a static data set of transitions. Through this data, the agent must learn the best policy [181] (see Fig. 11).

Offline data can be collected from a reference controller, or even through human demonstrations. Moreover, not all of these behaviors need to be correct, unlike imitation learning methods. All this makes it possible to generate and employ large and diverse data sets that represent multiple real-world situations.

However, offline RL also presents certain shortcomings and open challenges at present. The first and most obvious is that exploration cannot be improved. Therefore, the data set must capture states with high rewards. Another limitation is that imperfect uncertainty sets may result in overly conservative estimates, which hinder learning, or overly lax estimates, which result in exploitation of actions outside the distribution. Given the possibility that the agent would pursue a course of actions different from that seen in the data, counterfactual queries, as in [71], of "what if" type should be answered. If the policy is intended to perform optimally outside of the behavior seen in the data set, possibly, actions that are somehow different should be executed. Unfortunately, this strains the capabilities of many current ML tools, which are designed around the assumption that the data are independent and identically distributed [181].

5.6. Reward shaping

The RL formulation represents goals through rewards. Rewards can be provided only at the end of the episode (sparse rewards) [78,93], or they can be issued at each time step t (dense rewards) [44,67]. Dense rewards provide intermediate feedback to the agent, indicating how good the action taken in the previous time step was towards the final goal. This intermediate signal is essential in the definition of some problems, in particular when considering long experience streams. Without this intermediate feedback, in problems with large exploration spaces, learning would not be feasible. Some studies also combine dense and sparse rewards during learning, providing both intermediate feedback and an episode termination reward based on the fulfillment of pre-established conditions [64,68].

Rewards can be given directly [43], can formulate seemingly fuzzy goals [76], vary based on the agent's learning rate [83], or be defined through demonstrations [84] or even by a human-in-the-loop, providing online feedback through clicks [178]. However, reward shaping goes further. Rewards can express multiple goals and, as such, can be employed for different purposes in learning. Silver et al. [182] hypothesized that *"intelligence, and its associated abilities, can be understood as subserving the maximization of reward by an agent acting in its environment"*. This suggests that an agent that maximizes reward to achieve its goal might implicitly produce intelligence-associated abilities. These associated skills could be orthogonal to the agent's primary goal and directed towards multiple other pragmatic goals of the agent's intelligence, such as generalization or imitation.

5.7. Generalization and simulation-reality gap

Robot learning training requires a large number of training episodes and exploration of the environment. Poor exploration of the environment can lead to sub-optimal policies that produce uncertainty in the newness of states not known to the agent that may jeopardize both the entirety of the robot and its surroundings. In addition, during the learning itself, the agent's exploratory learning may give rise to certain potentially hazardous or unexpected robot behaviors, making it almost unfeasible to train the robot directly in reality. For this reason, RL algorithms have typically been developed in simulation environments.

However, the inherent mismatches between simulation and reality in many simulation tools have led to the limited deployment of policies in reality at times. For instance, almost 20% of the reviewed papers did not transfer their control policy to a real robot. To reduce the gap between simulation and reality, more realistic simulations and/or more robust policies are needed. According to [183], while the choice of the simulation environment, identifying the physical parameters of the robot through system identification, or domain adaptation are considered to obtain the most realistic simulations possible, current approaches to obtain robust policies include the introduction of perturbations in the environment [184] or domain randomization [185,186]. However, it is not always easy or possible to fully model the environment, so the emergence of novel experiences not considered in the simulation once the agent has been deployed in the real world [187] is another aspect to be taken into account. In this sense, some approaches focus on continuous learning, combining artificial and real data [188,189]. Thus, while simulation learning reduces the exploration space and increases safety, real data guarantees the convergence of the algorithm.

5.8. Control methods

Much of the research reviewed uses motion-controlled robots (position or velocity control), mainly in deformable object manipulation (see Fig. 12), where most objects do not present any resistance to manipulation.

$$q_{error} = q_d - q \tag{3}$$

$$\tau = f(q_{error}) \tag{4}$$

where q_{error} is the difference between the desired motion command q_d and the actual motion q, f is the command law, and τ is the motor torque. However, motion control does not seem to be an optimal control strategy when manipulating rigid objects with high contact dynamics. This is because any contact with the environment is considered a disturbance of the controller [51].

In contrast, force control helps to regulate the contact force. Force regulation is especially well suited in rigid object manipulations where the robot needs to consider object resistance and thus be more adaptive to the dynamics of the objects.

However, occasionally, it is not enough to control only the force, but the motion also needs to be directed. In this sense, hybrid motion-force control considers both motion and force variables. The motion and force are controlled as two independent variables. Because of the duality principle, both variables cannot be controlled simultaneously

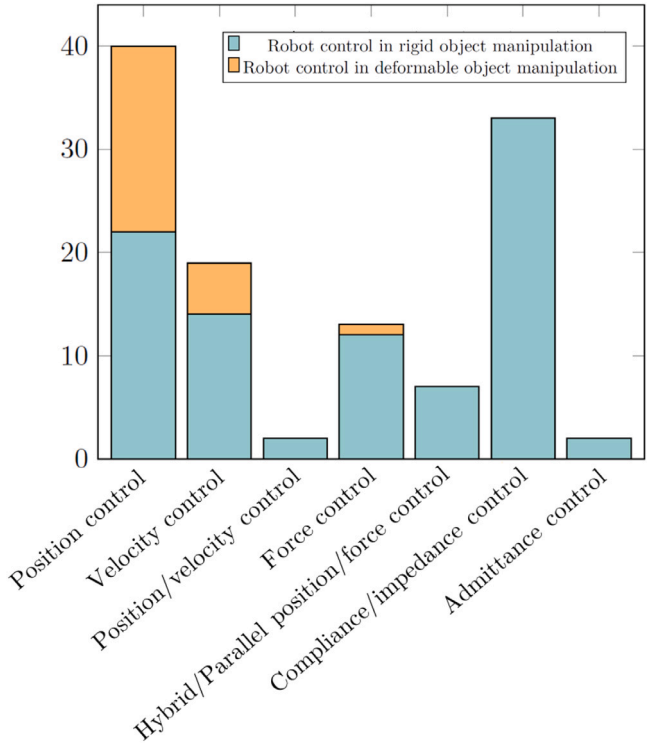


Fig. 12. Robot control methods.

in the same direction. So, for each direction, one variable should be selected.

$$[S]q_q \tag{5}$$

$$[I - S]f_d \tag{6}$$

where S is the motion and force selection matrix.

Other approaches like impedance and admittance control [190] provide a solution to overcome position uncertainties and avoid large impact forces simultaneously, which can damage the robot. Impedance control maps the motion q and/or velocity $\dot{q}$ (motion) deviation into the force as defined in Eq. (7). In contrast, admittance control maps the forces/torque signal into motion.

$$\tau_{ext} = K(q_d - q) + D(\dot{q}_d - \dot{q}) + M(q)(\ddot{q}_d - \ddot{q}) \tag{7}$$

where τ_{ext} is the extra torque applied to the motor, K is the stiffness matrix, D is the damping matrix, and M is the mass matrix.

Impedance control is the most commonly applied control method for contact-rich manipulation tasks on rigid objects (see Fig. 12). Impedance control based on precise control of joint torque is a powerful approach for robots to achieve high performance compliance control [191], thereby rendering it as a suitable control method for this type of tasks. In addition, multiple studies, such as [89,122], advocate performing impedance control in the task space which, in turn, can improve sample efficiency during training.

6. Discussion

Research and application of RL in robotics for contact-rich manipulation tasks have exploded over the past few years. This article reviews the literature and state-of-the-art methods in RL for manipulation tasks via serial manipulators for the period 2017–2022, based on the current technological advances and trends around this ML technique.

Despite the heterogeneity of the use cases in the reviewed articles, the research converges in the way the challenges are addressed. These approaches enable to establish connections among the different sections analyzed throughout this article for research around RL applied to contact-rich manipulation tasks (see Fig. 13). The green nodes represent aspects related to design choices, but do not involve a study on RL per se. The orange nodes are the main lines of research within RL. Lastly, the pink nodes can be linked either to a design choice of the researchers or as part of the RL investigation, since their study may influence one of the main lines of research. This might be the case, for instance, of reward shaping, hybrid action space, or the adaptation of observations to operate in latent action space.

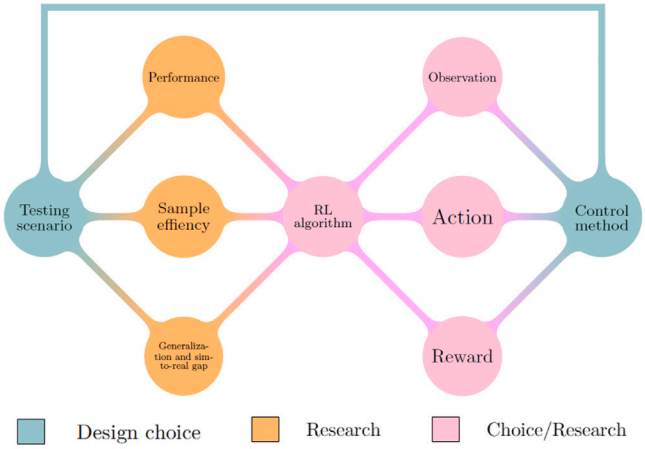


Fig. 13. RL engineering concepts relationship for contact-rich manipulation tasks.

6.1. Research trends

Currently, research on RL in contact-rich manipulation tasks relies on three pillars, namely, performance, sample efficiency and generalization and the simulation-reality gap. All three represent technological challenges to be overcome in order to foster the application of RL to more realistic industrial and/or healthcare scenarios. Performance encompasses different domains, e.g., accuracy [42], robustness [56], contact-stability [103], or safety [133], and generally depends on the needs that researchers seek to address. In contrast, sample efficiency and generalization and the simulation-reality gap are inherent challenges of the technology that many researchers are trying to tackle. Sample efficiency is often managed through prior knowledge input via human demonstrations [70], whereby initial trajectories are generated. For generalization, there are multiple approaches. Among them, domain randomization [78], adding perturbations to the environment [79] or combining real and artificial data [130] are some of the most widely employed. The choice of the simulator for learning can also play a key role for the deployment of the policy into a real robot. Of all the reviewed papers around rigid object manipulation, 75% of them use some simulation environment to learn a control policy, while, as far as papers dealing with deformable object manipulation are concerned, two out of three do so. Aspects such as sensor support, physics engines, or rendering quality should be considered for realistic simulations [192]. Although some research does not specify which simulator is used to train the RL agent, in rigid object manipulation mainly MuJoCo[5] (40%) [49,57], Gazebo[6] (≈15%) [51,96] and PyBullet[7] (≈11%) [73,134] stand out, while in deformable object manipulation so does MuJoCo (≈38%) [140,143], followed by finite element simulators

[5] https://mujoco.org/
[6] https://gazebosim.org/home
[7] https://pybullet.org/wordpress/

(≈19%) [154,155] and CHAI3D[8] (≈13%) [157,159]. MuJoCo is clearly the most commonly used choice among researchers, as it is a cross platform simulation engine that allows to simulate both rigid and deformable objects, has RGBD and force sensors in its portfolio, enables randomization of textures and friction of rendered objects and since its purchase by DeepMind in 2021, it has become freely available and open source. By contrast, it does not have multiple physics engines as Gazebo does, and still offers limitations in terms of realism [79]. In this sense, simulators such as NVIDIA Isaac Sim[9] are increasingly closing the gap. In addition, NVIDIA has also recently released its Isaac Gym simulator,[10] specifically for RL agent training. Although it is still at an early stage, there are already studies that have begun to use it [82]. Among its strengths is the potential to launch multiple environments simultaneously with small variations in their settings, thus being able to speed up the gathering of new experiences through exploration. Although sample efficiency and generalization seemed to generate a bottleneck that jeopardized one or the other a few years ago, more recent studies are trying to improve both simultaneously. This simulator seems likely to favor the achievement of both challenges, the overcoming of which would not only lead to more powerful RL-based solutions but could also lead to the application of this control technique to other manipulation tasks not yet explored.

6.2. Researchers' design choice

Another key aspect within the research of contact-rich manipulation through RL is to select the scenario in which the RL algorithm is to be tested or applied. While in the more applicative studies, it is generally the use case that determines whether RL is required and, therefore, the line of research to be followed [107,135], in theoretical studies, the authors tend to choose the use case as a pure evaluation scenario to test their research [49,124]. In any case, it was identified that most of the current research is focused on assembly. A logical explanation for this trend is that the assembly automation market is one of the most powerful markets in the industry, valued at more than USD 15 billion at the beginning of the decade and expected to grow at a compound annual growth rate (CAGR) of 6.5% until 2026. This growth will be underpinned by demand for digitization and plant automation for greater equipment efficiency and process accuracy [41].

Researchers can also select the control method to command the robot. The RL can be used directly as a controller or be used along with a conventional controller to adjust its parameters according to the environment. Although the latter is an alternative used in many studies [45,55], the employment of RL independently prevails. Also, although many studies use position-controlled robots with force/torque sensors coupled to measure the contact forces between the robot and the environment, the main trend in rigid object manipulation is the application of impedance control. This is because impedance control based on joint torque servo allows for compliance control in industrial applications by mapping the deviation of motion and/or velocity onto force. In deformable object handling tasks, on the other hand, position and velocity control methods are predominant. Clearly, the use of impedance control in applications such as rope, clothing or fabric folding is not practical since there are no contact forces. However, its application in tensioning and cutting tasks or in the handling of deformable volumetric objects is also null.

6.3. Points open to research or design choice

Once the most appropriate control method for the specific manipulation task has been identified, the properties of the RL are defined, i.e., the observations, actions and rewards. The control method selected to command the robot will condition the definition of the observations and actions. Likewise, the selection of observations and actions and the way rewards are provided can also be considered as part of the lines of research, as they could improve both the sample efficiency [122,123] and the generalization [83,127]. Tables 3 and 4 collect which observation and action spaces are used in both rigid and deformable object manipulation tasks. In rigid object contact-rich manipulation tasks, end-effector pose and external forces/torque are the most reported observations. The latter is due to the fact that the vast majority of the robots employed in the studies are not equipped with internal force/torque sensors, so information about the dynamics must be acquired through external sensors. As far as the actions are concerned, a large part of the researchers choose to act on the pose of the end-effector. This approach is compatible with impedance control, but the authors modify the position of the robot instead of parameters such as damping or stiffness. Moreover, unlike in the robot's joint action space where each action requires approximating the Jacobian of each pose, and whose erroneous modeling could further prevent the transfer of learning to reality, the position or force commands in Cartesian space can be sent to the robot's internal controller, which uses the internally encoded Jacobian to determine the joint torques. Thus, the specification of actions can improve robustness and even accelerate the learning rate by improving sample efficiency.

While the end-effector pose as the most employed action space is also applicable to the manipulation of deformable objects, the observation space is mainly represented from images. In fact, except for [146,160], none of the studies that focus on the manipulation of deformable objects use force-related information to perform the task. In contrast, many of them only employ a point cloud as information that relates the state of the initial points with respect to those that are considered the target states. Although this approach can be effective in dealing with rope or garment manipulation, the simulation of these objects is challenging due to their high dimensionality, and the performance of the policy may drop upon deployment in the real world due to the simulation-reality gap.

Considering rewards, despite being the key element to enhance the agent's learning, it is still a matter of study. In general, most researchers choose to provide dense feedback signals to guide robot learning. However, hand-engineered rewards are time consuming, and their potential ill-defined nature may lead to suboptimal policies. Other researchers opt instead for sparse rewards that they combine with techniques such as hindsight experience replay [193], which allows learning from rewards that are sparse and binary.

Once the MDP is formalized, the type of RL agent is decided. Despite the higher sample efficiency provided by model-based algorithms, the need for accurate knowledge of transition dynamics remains an obstacle among researchers to apply this type of algorithm. Among the model-free algorithms, actor–critic algorithms have been found to be the first choice. One potential explanation for the use of such algorithms is that they learn approximations of both the policy and the value function, which makes them suitable for high-dimensional manipulation tasks.

7. Concluding remarks

RL is a ML control technique that has been gaining widespread relevance across multiple industries recently. In this paper, the latest research related to RL in the context of robotics is discussed. Specifically, it presents an overview of the existing literature on contact-rich manipulation tasks and provides a high-level analysis of the main trends. Yet, this approach is far from reaching its potential and also raises a number of challenges that hinder its deployment in realistic applications.

The main lines of research are linked to the technological challenges of RL. Although this control technique has progressed over the

[8] https://www.chai3d.org/
[9] https://developer.nvidia.com/isaac-sim
[10] https://developer.nvidia.com/isaac-gym

years thanks to the use of neural networks, aspects such as policy performance, sample efficiency, and generalization and the simulation-reality gap still remain bottlenecks to the application of RL in more complex and realistic use cases. Indeed, the hard transfer of results to industrial environments due to high uncertainty is also a major industrial challenge. Nevertheless, recent studies point to the simultaneous improvement of sample efficiency and generalization.

All industrial environments pose unique challenges that may constitute interesting dimensions for future research. However, the main research paradigm would be to move towards defining high-level goals that the robot could achieve, while maintaining its performance in reality, regardless of the dimensionality and exploration space it interacts with.

CRediT authorship contribution statement

Íñigo Elguea-Aguinaco: Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Visualization. **Antonio Serrano-Muñoz:** Methodology, Writing – review & editing. **Dimitrios Chrysostomou:** Writing – review & editing, Visualization. **Ibai Inziarte-Hidalgo:** Visualization. **Simon Bøgh:** Writing – review & editing, Supervision. **Nestor Arana-Arexolaleiba:** Conceptualization, Writing – review & editing, Visualization, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Inigo Elguea-Aguinaco reports financial support was provided by Basque Government Department of Economic Development, Sustainability and Environment. Ibai Inziarte-Hidalgo reports financial support was provided by Navarra Government. Dimitrios Chrysostomou reports financial support was provided by European Commission Horizon 2020. Nestor Arana-Arexolaleiba reports financial support was provided by European Commission Horizon 2020.

Data availability

The authors do not have permission to share data.

Funding

This work has been partially funded by the Basque Government Department of Economic Development, Sustainability and Environment, Spain through the Bikaintek 2020 program, by the FEDER, Spain 2014–2020 Operational Program (GA 0011-1365-2020-000285) “Célula robótica para el pulido de piezas estructurales de metal duro del sector aeronáutico - ARISTARCO” and by the H2020-WIDESPREAD (GA 857061) “Networking for Research and Development of Human Interactive and Sensitive Robotics Taking Advantage of Additive Manufacturing - R2P2”.

References

[1] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, M. Hoffmann, Industry 4.0, Bus. Inform. Syst. Eng. 6 (4) (2014) 239–242.
[2] Industry 4.0 - Market study by Global Industry Analysts, Inc., 2022, https://www.strategyr.com/market-report-industry-4.0-forecasts-global-industry-analysts-inc.asp. (Accessed on 14 July 2022).
[3] Collaborative Robots Market is reaching a valuation of US$ 8.65 billion by 2029, 2022, https://www.globenewswire.com/news-release/2022/02/04/2379352/0/en/Collaborative-Robots-Market-is-reaching-a-valuation-of-US-8-65-Billion-by-2029-Comprehensive-Research-Report-by-FMI.html. (Accessed on 08 June 2022).
[4] S. Levine, P. Abbeel, Learning neural network policies with guided policy search under unknown dynamics, Adv. Neural Inf. Process. Syst. 27 (2014).
[5] F. Wirnshofer, P.S. Schmitt, P. Meister, G.v. Wichert, W. Burgard, State estimation in contact-rich manipulation, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 3790–3796.
[6] P. Khader, H. Yin, D. Kragic, Probabilistic model learning and long-term prediction forcontact-rich manipulation tasks, 2019, CoRR abs/1909.04915.
[7] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.
[8] Deep reinforcement learning for the control of robotic manipulation: A focussed mini-review, Robotics 10 (2021) 22, http://dx.doi.org/10.3390/robotics10010022, URL https://www.mdpi.com/2218-6581/10/1/22.
[9] G.N. Yannakakis, J. Togelius, Artificial Intelligence and Games, Vol. 2, Springer, 2018.
[10] O. Vinyals, I. Babuschkin, J. Chung, M. Mathieu, M. Jaderberg, W.M. Czarnecki, A. Dudzik, A. Huang, P. Georgiev, R. Powell, et al., Alphastar: Mastering the real-time strategy game starcraft II, DeepMind Blog 2 (2019).
[11] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of Go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489.
[12] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al., Mastering the game of Go without human knowledge, Nature 550 (7676) (2017) 354–359.
[13] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al., A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play, Science 362 (6419) (2018) 1140–1144.
[14] J. Viquerat, P. Meliga, E. Hachem, A review on deep reinforcement learning for fluid mechanics: An update, 2021, http://dx.doi.org/10.48550/ARXIV.2107.12206, arXiv, URL https://arxiv.org/abs/2107.12206.
[15] S. Lange, M. Riedmiller, A. Voigtländer, Autonomous reinforcement learning on raw visual input data in a real world application, in: The 2012 International Joint Conference on Neural Networks, IJCNN, IEEE, 2012, pp. 1–8.
[16] D. Li, D. Zhao, Q. Zhang, Y. Chen, Reinforcement learning and deep learning based lateral control for autonomous driving [application notes], IEEE Comput. Intell. Mag. 14 (2) (2019) 83–98.
[17] A.Y. Ng, H.J. Kim, M.I. Jordan, S. Sastry, S. Ballianda, Autonomous helicopter flight via reinforcement learning, in: NIPS, Vol. 16, Citeseer, 2003.
[18] J. de Lope, et al., Learning autonomous helicopter flight with evolutionary reinforcement learning, in: International Conference on Computer Aided Systems Theory, Springer, 2009, pp. 75–82.
[19] Y. Lin, Y. Liu, F. Lin, L. Zou, P. Wu, W. Zeng, H. Chen, C. Miao, A survey on reinforcement learning for recommender systems, 2021, http://dx.doi.org/10.48550/ARXIV.2109.10665, arXiv, URL https://arxiv.org/abs/2109.10665.
[20] M.M. Afsar, T. Crump, B. Far, Reinforcement learning based recommender systems: A survey, 2021, http://dx.doi.org/10.48550/ARXIV.2101.06286, arXiv, URL https://arxiv.org/abs/2101.06286.
[21] L. Yu, S. Qin, M. Zhang, C. Shen, T. Jiang, X. Guan, A review of deep reinforcement learning for smart building energy management, IEEE Internet Things J. 8 (15) (2021) 12046–12063, http://dx.doi.org/10.1109/jiot.2021.3078462.
[22] T. Yang, L. Zhao, W. Li, A.Y. Zomaya, Reinforcement learning in sustainable energy and electric systems: A survey, Annu. Rev. Control 49 (2020) 145–163, http://dx.doi.org/10.1016/j.arcontrol.2020.03.001, URL https://www.sciencedirect.com/science/article/pii/S1367578820300079.
[23] L. Lei, Y. Tan, K. Zheng, S. Liu, K. Zhang, X. Shen, Deep reinforcement learning for autonomous Internet of Things: Model, applications and challenges, IEEE Commun. Surv. Tutor. 22 (3) (2020) 1722–1760, http://dx.doi.org/10.1109/COMST.2020.2988367.
[24] V. Uc-Cetina, N. Navarro-Guerrero, A. Martin-Gonzalez, C. Weber, S. Wermter, Survey on reinforcement learning for language processing, 2021, http://dx.doi.org/10.48550/ARXIV.2104.05565, arXiv, URL https://arxiv.org/abs/2104.05565.
[25] C. Yu, J. Liu, S. Nemati, G. Yin, Reinforcement learning in healthcare: A survey, ACM Comput. Surv. 55 (1) (2021) http://dx.doi.org/10.1145/3477600.
[26] A.A. Abdellatif, N. Mhaisen, Z. Chkirbene, A. Mohamed, A. Erbad, M. Guizani, Reinforcement learning for intelligent healthcare systems: A comprehensive survey, 2021, http://dx.doi.org/10.48550/ARXIV.2108.04087, arXiv, URL https://arxiv.org/abs/2108.04087.
[27] T. Kegyes, Z. Süle, J. Abonyi, M. Andrea, The applicability of reinforcement learning methods in the development of industry 4.0 applications, Complex 2021 (2021) http://dx.doi.org/10.1155/2021/7179374.
[28] A. Lobbezoo, Y. Qian, H.-J. Kwon, Reinforcement learning for pick and place operations in robotics: A survey, Robotics 10 (3) (2021) http://dx.doi.org/10.3390/robotics10030105, URL https://www.mdpi.com/2218-6581/10/3/105.
[29] K. Kleeberger, R. Bormann, W. Kraus, M.F. Huber, A survey on learning-based robotic grasping, Curr. Robot. Rep. (2020).
[30] B.D. Argall, S. Chernova, M. Veloso, B. Browning, A survey of robot learning from demonstration, Robot. Auton. Syst. 57 (5) (2009) 469–483.
[31] J. Kober, J.A. Bagnell, J. Peters, Reinforcement learning in robotics: A survey, Int. J. Robot. Res. 32 (11) (2013) 1238–1274.

[32] R. Jeong, J. Kay, F. Romano, T. Lampe, T. Rothorl, A. Abdolmaleki, T. Erez, Y. Tassa, F. Nori, Modelling generalized forces with reinforcement learning for sim-to-real transfer, 2019, arXiv preprint arXiv:1910.09471.

[33] A. Franceschetti, E. Tosello, N. Castaman, S. Ghidoni, Robotic arm control and task training through deep reinforcement learning, in: International Conference on Intelligent Autonomous Systems, Springer, 2022, pp. 532–550.

[34] H. Zhang, F. Wang, J. Wang, B. Cui, Robot grasping method optimization using improved deep deterministic policy gradient algorithm of deep reinforcement learning, Rev. Sci. Instrum. 92 (2) (2021) 025114.

[35] L. Lu, M. Zhang, D. He, Q. Gu, D. Gong, L. Fu, A method of robot grasping based on reinforcement learning, J. Phys.: Conf. Ser. 2216 (1) (2022) 012026.

[36] L. Roveda, J. Maskani, P. Franceschi, A. Abdi, F. Braghin, L.M. Tosatti, N. Pedrocchi, Model-based reinforcement learning variable impedance control for human-robot collaboration, J. Intell. Robot. Syst. 100 (2) (2020) 417–433.

[37] A. Perrusquía, W. Yu, Robot position/force control in unknown environment using hybrid reinforcement learning, Cybern. Syst. 51 (4) (2020) 542–560.

[38] A. Lämmle, T. König, M. El-Shamouty, M.F. Huber, Skill-based programming of force-controlled assembly tasks using deep reinforcement learning, Procedia CIRP 93 (2020) 1061–1066.

[39] M. Oikawa, K. Kutsuzawa, S. Sakaino, T. Tsuji, Assembly robots with optimized control stiffness through reinforcement learning, 2020, arXiv preprint arXiv:2002.12207.

[40] P. Shukla, M. Pegu, G. Nandi, Development of behavior based robot manipulation using actor-critic architecture, in: 2021 8th International Conference on Signal Processing and Integrated Networks, SPIN, IEEE, 2021, pp. 469–474.

[41] Assembly automation market | Market size, share & forecast analysis, 2022, https://www.stratviewresearch.com/1900/assembly-automation-market.html. (Accessed on 08 July 2022).

[42] T. Inoue, G. De Magistris, A. Munawar, T. Yokoya, R. Tachibana, Deep reinforcement learning for high precision assembly tasks, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2017, pp. 819–825.

[43] X. Wu, D. Zhang, F. Qin, D. Xu, Deep reinforcement learning of robotic precision insertion skill accelerated by demonstrations, in: 2019 IEEE 15th International Conference on Automation Science and Engineering, CASE, IEEE, 2019, pp. 1651–1656.

[44] A.A. Apolinarska, M. Pacher, H. Li, N. Cote, R. Pastrana, F. Gramazio, M. Kohler, Robotic assembly of timber joints using reinforcement learning, Autom. Constr. 125 (2021) 103569.

[45] J. Luo, E. Solowjow, C. Wen, J.A. Ojea, A.M. Agogino, A. Tamar, P. Abbeel, Reinforcement learning on variable impedance controller for high-precision robotic assembly, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 3080–3087.

[46] Y.-G. Kim, M. Na, J.-B. Song, Reinforcement learning-based sim-to-real impedance parameter tuning for robotic assembly, in: 2021 21st International Conference on Control, Automation and Systems, ICCAS, IEEE, 2021, pp. 833–836.

[47] Y. Fan, J. Luo, M. Tomizuka, A learning framework for high precision industrial assembly, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 811–817.

[48] A. Li, R. Liu, X. Yang, Y. Lou, Reinforcement learning strategy based on multimodal representations for high-precision assembly tasks, in: International Conference on Intelligent Robotics and Applications, Springer, 2021, pp. 56–66.

[49] S.A. Khader, H. Yin, P. Falco, D. Kragic, Stability-guaranteed reinforcement learning for contact-rich manipulation, IEEE Robot. Autom. Lett. 6 (1) (2020) 1–8.

[50] F. Li, Q. Jiang, W. Quan, R. Song, Y. Li, Manipulation skill acquisition for robotic assembly using deep reinforcement learning, in: 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM, IEEE, 2019, pp. 13–18.

[51] C.C. Beltran-Hernandez, D. Petit, I.G. Ramirez-Alpizar, T. Nishi, S. Kikuchi, T. Matsubara, K. Harada, Learning force control for contact-rich manipulation tasks with rigid position-controlled robots, IEEE Robot. Autom. Lett. 5 (4) (2020) 5709–5716.

[52] X. Li, J. Xiao, W. Zhao, H. Liu, G. Wang, Multiple peg-in-hole compliant assembly based on a learning-accelerated deep deterministic policy gradient strategy, Ind. Robot: Int. J. Robot. Res. Appl. (2021).

[53] J. Li, D. Pang, Y. Zheng, X. Guan, X. Le, A flexible manufacturing assembly system with deep reinforcement learning, Control Eng. Pract. 118 (2022) 104957.

[54] Y. Wang, S. Zhu, Q. Zhang, R. Zhou, R. Dou, H. Sun, Q. Yao, M. Xu, Y. Zhang, A visual grasping strategy for improving assembly efficiency based on deep reinforcement learning, J. Sensors 2021 (2021).

[55] P. Kulkarni, J. Kober, R. Babuška, C. Della Santina, Learning assembly tasks in a few minutes by combining impedance control and residual recurrent reinforcement learning, Adv. Intell. Syst. (2021) 2100095.

[56] P. Ennen, P. Bresenitz, R. Vossen, F. Hees, Learning robust manipulation skills with guided policy search via generative motor reflexes, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 7851–7857.

[57] F. Wirnshofer, P.S. Schmitt, G. von Wichert, W. Burgard, Controlling contact-rich manipulation under partial observability, in: Robotics: Science and Systems, 2020.

[58] T. Ren, Y. Dong, D. Wu, K. Chen, Learning-based variable compliance control for robotic assembly, J. Mech. Robot. 10 (6) (2018) 061008.

[59] C. Wang, C. Lin, B. Liu, C. Su, P. Xu, L. Xie, Deep reinforcement learning with shaping exploration space for robotic assembly, in: 2021 3rd International Symposium on Robotics & Intelligent Manufacturing Technology, ISRIMT, IEEE, 2021, pp. 345–351.

[60] Y. Shi, Z. Chen, H. Liu, S. Riedel, C. Gao, Q. Feng, J. Deng, J. Zhang, Proactive action visual residual reinforcement learning for contact-rich tasks using a torque-controlled robot, in: 2021 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2021, pp. 765–771.

[61] Y.-L. Kim, K.-H. Ahn, J.-B. Song, Reinforcement learning based on movement primitives for contact tasks, Robot. Comput.-Integr. Manuf. 62 (2020) 101863.

[62] G. Schoettler, A. Nair, J. Luo, S. Bahl, J.A. Ojea, E. Solowjow, S. Levine, Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2020, pp. 5548–5555.

[63] M. Vecerik, O. Sushkov, D. Barker, T. Rothörl, T. Hester, J. Scholz, A practical approach to insertion with variable socket position using deep reinforcement learning, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 754–760.

[64] M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, M. Riedmiller, Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards, 2017, arXiv preprint arXiv:1707.08817.

[65] Y. Li, D. Xu, Skill learning for robotic insertion based on one-shot demonstration and reinforcement learning, Int. J. Autom. Comput. 18 (3) (2021) 457–467.

[66] Y. Shi, Z. Chen, Y. Wu, D. Henkel, S. Riedel, H. Liu, Q. Feng, J. Zhang, Combining learning from demonstration with learning by exploration to facilitate contact-rich tasks, 2021, arXiv preprint arXiv:2103.05904.

[67] Y. Wang, C.C. Beltran-Hernandez, W. Wan, K. Harada, Robotic imitation of human assembly skills using hybrid trajectory and force learning, 2021, arXiv preprint arXiv:2103.05912.

[68] Y. Ma, D. Xu, F. Qin, Efficient insertion control for precision assembly based on demonstration learning and reinforcement learning, IEEE Trans. Ind. Inform. 17 (7) (2020) 4492–4502.

[69] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J.A. Ojea, E. Solowjow, S. Levine, Residual reinforcement learning for robot control, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 6023–6029.

[70] Y. Wang, C.C. Beltran-Hernandez, W. Wan, K. Harada, Hybrid trajectory and force learning of complex assembly tasks: A combined learning framework, IEEE Access 9 (2021) 60175–60186.

[71] J. Jin, D. Graves, C. Haigh, J. Luo, M. Jagersand, Offline learning of counterfactual perception as prediction for real-world robotic reinforcement learning, 2020, arXiv preprint arXiv:2011.05857.

[72] S. Hoppe, Z. Lou, D. Hennes, M. Toussaint, Planning approximate exploration trajectories for model-free reinforcement learning in contact-rich manipulation, IEEE Robot. Autom. Lett. 4 (4) (2019) 4042–4047.

[73] L. Shao, T. Migimatsu, J. Bohg, Learning to scaffold the development of robotic manipulation skills, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 5671–5677.

[74] M. Hamaya, R. Lee, K. Tanaka, F. von Drigalski, C. Nakashima, Y. Shibata, Y. Ijiri, Learning robotic assembly tasks with lower dimensional systems by leveraging physical softness and environmental constraints, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 7747–7753.

[75] M. Simonič, L. Žlajpah, A. Ude, B. Nemec, Autonomous learning of assembly tasks from the corresponding disassembly tasks, in: 2019 IEEE-RAS 19th International Conference on Humanoid Robots, Humanoids, IEEE, 2019, pp. 230–236.

[76] J. Xu, Z. Hou, W. Wang, B. Xu, K. Zhang, K. Chen, Feedback deep deterministic policy gradient with fuzzy reward for robotic multiple peg-in-hole assembly tasks, IEEE Trans. Ind. Inform. 15 (3) (2018) 1658–1667.

[77] Z. Hou, Z. Li, C. Hsu, K. Zhang, J. Xu, Fuzzy logic-driven variable time-scale prediction-based reinforcement learning for robotic multiple peg-in-hole assembly, IEEE Trans. Autom. Sci. Eng. (2020).

[78] C.C. Beltran-Hernandez, D. Petit, I.G. Ramirez-Alpizar, K. Harada, Variable compliance control for robotic peg-in-hole assembly: A deep-reinforcement-learning approach, Appl. Sci. 10 (19) (2020) 6923.

[79] M. Hebecker, J. Lambrecht, M. Schmitz, Towards real-world force-sensitive robotic assembly through deep reinforcement learning in simulations, in: 2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM, IEEE, 2021, pp. 1045–1051.

[80] G. Thomas, M. Chien, A. Tamar, J.A. Ojea, P. Abbeel, Learning robotic assembly from CAD, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2018, pp. 3524–3531.

[81] M.A. Lee, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, J. Bohg, Making sense of vision and touch: Learning multimodal representations for contact-rich tasks, IEEE Trans. Robot. 36 (3) (2020) 582–596.

[82] Z. Wu, W. Lian, V. Unhelkar, M. Tomizuka, S. Schaal, Learning dense rewards for contact-rich manipulation tasks, in: 2021 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2021, pp. 6214–6221.

[83] L. Leyendecker, M. Schmitz, H.A. Zhou, V. Samsonov, M. Rittstieg, D. Lütticke, Deep reinforcement learning for robotic control in high-dexterity assembly tasks-A reward curriculum approach, in: 2021 Fifth IEEE International Conference on Robotic Computing, IRC, IEEE, 2021, pp. 35–42.

[84] X. Zhang, L. Sun, Z. Kuang, M. Tomizuka, Learning variable impedance control via inverse reinforcement learning for force-related tasks, IEEE Robot. Autom. Lett. 6 (2) (2021) 2225–2232.

[85] X. Zhao, H. Zhao, P. Chen, H. Ding, Model accelerated reinforcement learning for high precision robotic assembly, Int. J. Intell. Robot. Appl. 4 (2) (2020) 202–216.

[86] K. Tanaka, R. Yonetani, M. Hamaya, R. Lee, F. von Drigalski, Y. Ijiri, Trans-AM: Transfer learning by aggregating dynamics models for soft robotic assembly, in: 2021 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2021, pp. 4627–4633.

[87] J. Ding, C. Wang, C. Lu, Transferable force-torque dynamics model for peg-in-hole task, 2019, arXiv preprint arXiv:1912.00260.

[88] S. Schaal, Dynamic movement primitives-a framework for motor control in humans and humanoid robotics, in: Adaptive Motion of Animals and Machines, Springer, 2006, pp. 261–280.

[89] O. Spector, M. Zacksenhouse, Deep reinforcement learning for contact-rich skills using compliant movement primitives, 2020, arXiv preprint arXiv:2008.13223.

[90] T. Davchev, K.S. Luck, M. Burke, F. Meier, S. Schaal, S. Ramamoorthy, Residual learning from demonstration: Adapting DMPs for contact-rich manipulation, IEEE Robot. Autom. Lett. 7 (2) (2022) 4488–4495.

[91] X. Zhang, S. Jin, C. Wang, X. Zhu, M. Tomizuka, Learning insertion primitives with discrete-continuous hybrid action space for robotic assembly tasks, in: 2022 International Conference on Robotics and Automation, ICRA, IEEE, 2022, pp. 9881–9887.

[92] J. Vanschoren, Meta-learning, in: Automated Machine Learning, Springer, Cham, 2019, pp. 35–61.

[93] G. Schoettler, A. Nair, J.A. Ojea, S. Levine, E. Solowjow, Meta-reinforcement learning for robotic industrial insertion tasks, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2020, pp. 9728–9735.

[94] D. Liu, X. Zhang, Y. Du, D. Gao, M. Wang, M. Cong, Industrial insert robotic assembly based on model-based meta-reinforcement learning, in: 2021 IEEE International Conference on Robotics and Biomimetics, ROBIO, IEEE, 2021, pp. 1508–1512.

[95] T.Z. Zhao, J. Luo, O. Sushkov, R. Pevceviciute, N. Heess, J. Scholz, S. Schaal, S. Levine, Offline meta-reinforcement learning for industrial insertion, in: 2022 International Conference on Robotics and Automation, ICRA, IEEE, 2022, pp. 6386–6393.

[96] C.B. Kristensen, F.A. Sørensen, H.B. Nielsen, M.S. Andersen, S.P. Bendtsen, S. Bøgh, Towards a robot simulation framework for e-waste disassembly using reinforcement learning, Procedia Manuf. 38 (2019) 225–232.

[97] R. Herold, Y. Wang, D. Pham, J. Huang, C. Ji, S. Su, Using active adjustment and compliance in robotic disassembly, in: Industry 4.0–Shaping the Future of the Digital World, CRC Press, 2020, pp. 101–105.

[98] A. Serrano-Muñoz, N. Arana-Arexolaleiba, D. Chrysostomou, S. Bøgh, Learning and generalising object extraction skill for contact-rich disassembly tasks: An introductory study, Int. J. Adv. Manuf. Technol. (2021) 1–13.

[99] Z.-W. Zhong, Advanced polishing, grinding and finishing processes for various manufacturing applications: A review, Mater. Manuf. Process. 35 (12) (2020) 1279–1303.

[100] D. Zhu, X. Feng, X. Xu, Z. Yang, W. Li, S. Yan, H. Ding, Robotic grinding of complex components: A step towards efficient and intelligent machining–challenges, solutions, and applications, Robot. Comput.-Integr. Manuf. 65 (2020) 101908.

[101] J. Li, T. Zhang, X. Liu, Y. Guan, D. Wang, A survey of robotic polishing, in: 2018 IEEE International Conference on Robotics and Biomimetics, ROBIO, IEEE, 2018, pp. 2125–2132.

[102] G.A. Odesanmia, I. Iqbalb, B. Jiec, Z. Congd, J. Wange, L.M. Liuf, Q learning based trajectory generation for robotic grinding and polishing, in: 2018 International Symposium on Advances in Abrasive Technology, ISAAT2018, 2018.

[103] T. Zhang, M. Xiao, Y. Zou, J. Xiao, Robotic constant-force grinding control with a press-and-release model and model-based reinforcement learning, Int. J. Adv. Manuf. Technol. 106 (1) (2020) 589–602.

[104] Y. Ding, J. Zhao, X. Min, Impedance control and parameter optimization of surface polishing robot based on reinforcement learning, Proc. Inst. Mech. Eng. B (2022) 09544054221100004.

[105] S. Cabi, S.G. Colmenarejo, A. Novikov, K. Konyushkova, S. Reed, R. Jeong, K. Zolna, Y. Aytar, D. Budden, M. Vecerik, et al., Scaling data-driven robotics with reward sketching and batch reinforcement learning, 2019, arXiv preprint arXiv:1909.12200.

[106] B. Belousov, B. Wibranek, J. Schneider, T. Schneider, G. Chalvatzaki, J. Peters, O. Tessmann, Robotic architectural assembly with tactile skills: Simulation and optimization, Autom. Constr. 133 (2022) 104006.

[107] L. Liang, Y. Chen, L. Liao, H. Sun, Y. Liu, A novel impedance control method of rubber unstacking robot dealing with unpredictable and time-variable adhesion force, Robot. Comput.-Integr. Manuf. 67 (2021) 102038.

[108] S. Levine, V. Koltun, Guided policy search, in: International Conference on Machine Learning, PMLR, 2013, pp. 1–9.

[109] Y. Chebotar, M. Kalakrishnan, A. Yahya, A. Li, S. Schaal, S. Levine, Path integral guided policy search, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2017, pp. 3381–3388.

[110] A. Yahya, A. Li, M. Kalakrishnan, Y. Chebotar, S. Levine, Collective robot reinforcement learning with distributed asynchronous guided policy search, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2017, pp. 79–86.

[111] B. Nemec, L. Žlajpah, A. Ude, Door opening by joining reinforcement learning and intelligent control, in: 2017 18th International Conference on Advanced Robotics, ICAR, IEEE, 2017, pp. 222–228.

[112] Y. Hou, H. Xu, J. Luo, Y. Lei, J. Xu, H.-T. Zhang, Variable impedance control of manipulator based on DQN, in: International Conference on Intelligent Robotics and Applications, Springer, 2020, pp. 296–307.

[113] S. Gu, E. Holly, T. Lillicrap, S. Levine, Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2017, pp. 3389–3396.

[114] P. Englert, M. Toussaint, Learning manipulation skills from a single demonstration, Int. J. Robot. Res. 37 (1) (2018) 137–154.

[115] N. Lin, Y. Li, K. Tang, Y. Zhu, X. Zhang, R. Wang, J. Ji, X. Chen, X. Zhang, Manipulation planning from demonstration via goal-conditioned prior action primitive decomposition and alignment, IEEE Robot. Autom. Lett. 7 (2) (2022) 1387–1394.

[116] J. Stüber, C. Zito, R. Stolkin, Let's push things forward: A survey on robot pushing, Front. Robot. AI (2020) 8.

[117] N. Lin, L. Zhang, Y. Chen, Y. Zhu, R. Chen, P. Wu, X. Chen, Reinforcement learning for robotic safe control with force sensing, in: 2019 WRC Symposium on Advanced Robotics and Automation, WRC SARA, IEEE, 2019, pp. 148–153.

[118] S.H. Huang, M. Zambelli, J. Kay, M.F. Martins, Y. Tassa, P.M. Pilarski, R. Hadsell, Learning gentle object manipulation with curiosity-driven deep reinforcement learning, 2019, arXiv preprint arXiv:1903.08542.

[119] L. Cong, H. Liang, P. Ruppel, Y. Shi, M. Görner, N. Hendrich, J. Zhang, Reinforcement learning with vision-proprioception model for robot planar pushing, Front. Neurorobot. 16 (2022).

[120] K. Hausman, J.T. Springenberg, Z. Wang, N. Heess, M. Riedmiller, Learning an embedding space for transferable robot skills, in: International Conference on Learning Representations, 2018.

[121] I. Akinola, J. Varley, D. Kalashnikov, Learning precise 3D manipulation from multiple uncalibrated cameras, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 4616–4622.

[122] R. Martín-Martín, M.A. Lee, R. Gardner, S. Savarese, J. Bohg, A. Garg, Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2019, pp. 1010–1017.

[123] A. Allshire, R. Martín-Martín, C. Lin, S. Manuel, S. Savarese, A. Garg, Laser: Learning a latent action space for efficient reinforcement learning, in: 2021 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2021, pp. 6650–6656.

[124] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, S. Levine, Learning complex dexterous manipulation with deep reinforcement learning and demonstrations, 2017, arXiv preprint arXiv:1709.10087.

[125] H. Zhu, A. Gupta, A. Rajeswaran, S. Levine, V. Kumar, Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 3651–3657.

[126] M.V. Balakuntala, U. Kaur, X. Ma, J. Wachs, R.M. Voyles, Learning multimodal contact-rich skills from demonstrations without reward engineering, 2021, arXiv preprint arXiv:2103.01296.

[127] N. Vulin, S. Christen, S. Stevšić, O. Hilliges, Improved learning of robot manipulation tasks via tactile intrinsic motivation, IEEE Robot. Autom. Lett. 6 (2) (2021) 2194–2201.

[128] Y. Zhang, I. Clavera, B. Tsai, P. Abbeel, Asynchronous methods for model-based reinforcement learning, 2019, arXiv preprint arXiv:1910.12453.

[129] W. Guo, C. Wang, Y. Fu, F. Zha, Deep reinforcement learning algorithm for object placement tasks with manipulator, in: 2018 IEEE International Conference on Intelligence and Safety for Robotics, ISR, IEEE, 2018, pp. 608–613.

[130] S. Kim, H. Jo, J.-B. Song, Object manipulation system based on image-based reinforcement learning, Intell. Serv. Robot. (2022) 1–7.

[131] Y. Chebotar, K. Hausman, M. Zhang, G. Sukhatme, S. Schaal, S. Levine, Combining model-based and model-free updates for trajectory-centric reinforcement learning, in: International Conference on Machine Learning, PMLR, 2017, pp. 703–711.

[132] S. Nasiriany, H. Liu, Y. Zhu, Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks, in: 2022 International Conference on Robotics and Automation, ICRA, IEEE, 2022, pp. 7477–7484.
[133] C.-Y. Kuo, A. Schaarschmidt, Y. Cui, T. Asfour, T. Matsubara, Uncertainty-aware contact-safe model-based reinforcement learning, IEEE Robot. Autom. Lett. 6 (2) (2021) 3918–3925.
[134] M. Bogdanovic, M. Khadiv, L. Righetti, Learning variable impedance control for contact sensitive tasks, IEEE Robot. Autom. Lett. 5 (4) (2020) 6129–6136.
[135] Y. Luo, D. Xu, J. Zhu, Y. Lei, Impedance control of slag removal robot based on Q-learning, in: 2021 China Automation Congress, CAC, IEEE, 2021, pp. 1338–1343.
[136] M. Schumacher, J. Wojtusch, P. Beckerle, O. von Stryk, An introductory review of active compliant control, Robot. Auton. Syst. 119 (2019) 185–200.
[137] A.S. Anand, M.H. Myrestrand, J.T. Gravdahl, Evaluation of variable impedance and hybrid force/motion controllers for learning force tracking skills, in: 2022 IEEE/SICE International Symposium on System Integration, SII, IEEE, 2022, pp. 83–89.
[138] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, Y. Mezouar, Robotic manipulation and sensing of deformable objects in domestic and industrial applications: A survey, Int. J. Robot. Res. 37 (7) (2018) 688–716.
[139] H. Han, G. Paul, T. Matsubara, Model-based reinforcement learning approach for deformable linear object manipulation, in: 2017 13th IEEE Conference on Automation Science and Engineering, CASE, IEEE, 2017, pp. 750–755.
[140] M. Bednarek, K. Walas, Comparative assessment of reinforcement learning algorithms in the taskof robotic manipulation of deformable linear objects, in: 2019 4th International Conference on Robotics and Automation Engineering, ICRAE, IEEE, 2019, pp. 173–177.
[141] X. Lin, H.S. Baweja, D. Held, Reinforcement learning without ground-truth state, 2019, arXiv preprint arXiv:1905.07866.
[142] R. Laezza, Y. Karayiannidis, Shape control of elastoplastic deformable linear objects through reinforcement learning.
[143] Y. Wu, W. Yan, T. Kurutach, L. Pinto, P. Abbeel, Learning to manipulate deformable objects without demonstrations, 2019, arXiv preprint arXiv:1910.13439.
[144] V. Petrík, V. Kyrki, Feedback-based fabric strip folding, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2019, pp. 773–778.
[145] Y. Tsurumine, Y. Cui, E. Uchibe, T. Matsubara, Deep reinforcement learning with smooth policy update: Application to robotic cloth manipulation, Robot. Auton. Syst. 112 (2019) 72–83.
[146] A. Verleysen, T. Holvoet, R. Proesmans, C. Den Haese, et al., Simpler learning of robotic manipulation of clothing by utilizing DIY smart textile technology, Appl. Sci. 10 (12) (2020) 4088.
[147] F. Amadio, A. Colomé, C. Torras, Exploiting symmetries in reinforcement learning of bimanual robotic tasks, IEEE Robot. Autom. Lett. 4 (2) (2019) 1838–1845.
[148] R. Jangir, G. Alenyà, C. Torras, Dynamic cloth manipulation with deep reinforcement learning, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 4630–4636.
[149] J. Matas, S. James, A.J. Davison, Sim-to-real reinforcement learning for deformable object manipulation, in: Conference on Robot Learning, PMLR, 2018, pp. 734–743.
[150] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, S. Levine, Visual foresight: Model-based deep reinforcement learning for vision-based robotic control, 2018, arXiv preprint arXiv:1812.00568.
[151] R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A.K. Tanwani, N. Jamali, K. Yamane, S. Iba, K. Goldberg, Visuospatial foresight for multi-step, multi-task fabric manipulation, 2020, arXiv preprint arXiv:2003.09044.
[152] W. Zhou, S. Bajracharya, D. Held, PLAS: Latent action space for offline reinforcement learning, 2020, arXiv preprint arXiv:2011.07213.
[153] A. Murali, S. Sen, B. Kehoe, A. Garg, S. McFarland, S. Patil, W.D. Boyd, S. Lim, P. Abbeel, K. Goldberg, Learning by observation for surgical subtasks: Multilateral cutting of 3D viscoelastic and 2D orthotropic tissue phantoms, in: 2015 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2015, pp. 1202–1209.
[154] B. Thananjeyan, A. Garg, S. Krishnan, C. Chen, L. Miller, K. Goldberg, Multilateral surgical pattern cutting in 2D orthotropic gauze with deep reinforcement learning policies for tensioning, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2017, pp. 2371–2378.
[155] T. Nguyen, N.D. Nguyen, F. Bello, S. Nahavandi, A new tensioning method using deep reinforcement learning for surgical pattern cutting, in: 2019 IEEE International Conference on Industrial Technology, ICIT, IEEE, 2019, pp. 1339–1344.
[156] N.D. Nguyen, T. Nguyen, S. Nahavandi, A. Bhatti, G. Guest, Manipulating soft tissues by deep reinforcement learning for autonomous robotic surgery, in: 2019 IEEE International Systems Conference, SysCon, IEEE, 2019, pp. 1–7.
[157] C. Shin, P.W. Ferguson, S.A. Pedram, J. Ma, E.P. Dutson, J. Rosen, Autonomous tissue manipulation via surgical robot using learning based model predictive control, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 3875–3881.
[158] S. Krishnan, A. Garg, R. Liaw, B. Thananjeyan, L. Miller, F.T. Pokorny, K. Goldberg, SWIRL: A sequential windowed inverse reinforcement learning algorithm for robot tasks with delayed rewards, Int. J. Robot. Res. 38 (2–3) (2019) 126–145.
[159] S.A. Pedram, P.W. Ferguson, C. Shin, A. Mehta, E.P. Dutson, F. Alambeigi, J. Rosen, Toward synergic learning for autonomous manipulation of deformable tissues via surgical robots: An approximate Q-learning approach, in: 2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics, BioRob, IEEE, 2020, pp. 878–884.
[160] J. Luo, E. Solowjow, C. Wen, J.A. Ojea, A.M. Agogino, Deep reinforcement learning for robotic assembly of mixed deformable and rigid objects, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2018, pp. 2062–2069.
[161] A. Gonnochenko, A. Semochkin, D. Egorov, D. Statovoy, S. Zabihifar, A. Postnikov, E. Seliverstova, A. Zaidi, J. Stemmler, K. Limkrailassiri, Coinbot: Intelligent robotic coin bag manipulation using deep reinforcement learning and machine teaching, 2020, arXiv preprint arXiv:2012.01356.
[162] C. Matl, R. Bajcsy, Deformable elasto-plastic object shaping using an elastic hand and model-based reinforcement learning, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2021, pp. 3955–3962.
[163] X. Liu, S.S. Ge, F. Zhao, X. Mei, Optimized interaction control for robot manipulator interacting with flexible environment, IEEE/ASME Trans. Mechatronics 26 (6) (2020) 2888–2898.
[164] Y. Altintas, M. Weck, Chatter stability of metal cutting and grinding, CIRP Ann. 53 (2) (2004) 619–642.
[165] R. Strudel, A. Pashevich, I. Kalevatykh, I. Laptev, J. Sivic, C. Schmid, Learning to combine primitive skills: A step towards versatile robotic manipulation, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 4637–4643.
[166] E. Ben-Iwhiwhu, J. Dick, N.A. Ketz, P.K. Pilly, A. Soltoggio, Context meta-reinforcement learning via neuromodulation, Neural Netw. 152 (2022) 70–79.
[167] D. Tanaka, S. Arnold, K. Yamazaki, Emd net: An encode–manipulate–decode network for cloth manipulation, IEEE Robot. Autom. Lett. 3 (3) (2018) 1771–1778.
[168] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, 2015, arXiv preprint arXiv:1509.02971.
[169] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in: International Conference on Machine Learning, PMLR, 2018, pp. 1861–1870.
[170] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, 2017, arXiv preprint arXiv:1707.06347.
[171] J. Schulman, S. Levine, P. Abbeel, M. Jordan, P. Moritz, Trust region policy optimization, in: International Conference on Machine Learning, PMLR, 2015, pp. 1889–1897.
[172] A. Ray, J. Achiam, D. Amodei, Benchmarking safe exploration in deep reinforcement learning, 2019.
[173] L. Brunke, M. Greeff, A.W. Hall, Z. Yuan, S. Zhou, J. Panerati, A.P. Schoellig, Safe learning in robotics: From learning-based control to safe reinforcement learning, 2022, arXiv abs/2108.06266.
[174] J. García, F. Fernández, A comprehensive survey on safe reinforcement learning, J. Mach. Learn. Res. 16 (1) (2015) 1437–1480.
[175] M. Braun, S. Wrede, Incorporation of expert knowledge for learning robotic assembly tasks, in: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation, Vol. 1, ETFA, IEEE, 2020, pp. 1594–1601.
[176] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, S. Levine, Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning, in: Conference on Robot Learning, PMLR, 2020, pp. 1094–1100.
[177] Z. Ding, H. Dong, Challenges of reinforcement learning, in: Deep Reinforcement Learning, Springer, 2020, pp. 249–272.
[178] S.C. Akkaladevi, M. Plasch, S. Maddukuri, C. Eitzinger, A. Pichler, B. Rinner, Toward an interactive reinforcement based learning framework for human robot collaborative assembly processes, Front. Robot. AI 5 (2018) 126.
[179] D. Hadfield-Menell, S.J. Russell, P. Abbeel, A. Dragan, Cooperative inverse reinforcement learning, Adv. Neural Inf. Process. Syst. 29 (2016) 3909–3917.
[180] C. Daniel, M. Viering, J. Metz, O. Kroemer, J. Peters, Active reward learning, in: Robotics: Science and Systems, Vol. 98, 2014.
[181] S. Levine, A. Kumar, G. Tucker, J. Fu, Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020, arXiv preprint arXiv:2005.01643.
[182] D. Silver, S. Singh, D. Precup, R.S. Sutton, Reward is enough, Artificial Intelligence 299 (2021) 103535.
[183] W. Zhao, J.P. Queralta, T. Westerlund, Sim-to-real transfer in deep reinforcement learning for robotics: A survey, in: 2020 IEEE Symposium Series on Computational Intelligence, SSCI, IEEE, 2020, pp. 737–744.

[184] L. Pinto, J. Davidson, R. Sukthankar, A. Gupta, Robust adversarial reinforcement learning, in: International Conference on Machine Learning, PMLR, 2017, pp. 2817–2826.

[185] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, P. Abbeel, Domain randomization for transferring deep neural networks from simulation to the real world, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2017, pp. 23–30.

[186] Z. Ding, Y.-Y. Tsai, W.W. Lee, B. Huang, Sim-to-real transfer for robotic manipulation with tactile sensory, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2021, pp. 6778–6785.

[187] G. Kahn, A. Villaflor, V. Pong, P. Abbeel, S. Levine, Uncertainty-aware reinforcement learning for collision avoidance, 2017, arXiv preprint arXiv:1702.01182.

[188] G. Kalweit, J. Boedecker, Uncertainty-driven imagination for continuous deep reinforcement learning, in: Conference on Robot Learning, PMLR, 2017, pp. 195–206.

[189] K. Kang, S. Belkhale, G. Kahn, P. Abbeel, S. Levine, Generalization through simulation: Integrating simulated and real data into deep reinforcement learning for vision-based autonomous flight, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 6008–6014.

[190] F.J. Abu-Dakka, M. Saveriano, Variable impedance control and learning – A review, 2020, http://dx.doi.org/10.48550/ARXIV.2010.06246, arXiv, URL https://arxiv.org/abs/2010.06246.

[191] Y. Dong, T. Ren, D. Wu, K. Chen, Compliance control for robot manipulation in contact with a varied environment based on a new joint torque controller, J. Intell. Robot. Syst. 99 (1) (2020) 79–90.

[192] J. Collins, S. Chand, A. Vanderkop, D. Howard, A review of physics simulators for robotic applications, IEEE Access 9 (2021) 51416–51431.

[193] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, W. Zaremba, Hindsight experience replay, 2017, arXiv preprint arXiv:1707.01495.