Wiley
International Journal of Intelligent Systems
Volume 2024, Article ID 1636497, 29 pages
https://doi.org/10.1155/int/1636497

WILEY

Review Article

A Review on Reinforcement Learning for Motion Planning of Robotic Manipulators

Íñigo Elguea-Aguinaco[1,2], **Ibai Inziarte-Hidalgo**[3], **Simon Bøgh**[4] and **Nestor Arana-Arexolaleiba**[2,4]

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

Correspondence should be addressed to Íñigo Elguea-Aguinaco; ielguea@aldakin.com

Received 18 March 2024; Revised 7 November 2024; Accepted 9 December 2024

Academic Editor: Mohamadreza (Mohammad) Khosravi

Effective motion planning is an indispensable prerequisite for the optimal performance of robotic manipulators in any task. In this regard, the research and application of reinforcement learning in robotic manipulators for motion planning have gained great relevance in recent years. The ability of reinforcement learning agents to adapt to variable environments, especially those featuring dynamic obstacles, has propelled their increasing application in this domain. Notwithstanding, a clear need remains for a resource that critically examines the progress, challenges, and future directions of this machine learning control technique in motion planning. This article undertakes a comprehensive review of the landscape of reinforcement learning, offering a retrospective analysis of its application in motion planning from 2018 to the present. The exploration extends to the trends associated with reinforcement learning in the context of serial manipulators and motion planning, as well as the various technological challenges currently presented by this machine learning control technique. The overarching objective of this review is to serve as a valuable resource for the robotics community, facilitating the ongoing development of systems controlled by reinforcement learning. By delving into the primary challenges intrinsic to this technology, the review seeks to enhance the understanding of reinforcement learning's role in motion planning and provides insights that may suggest future research directions in this domain.

1. Introduction

In recent years, the integration of advanced sensing, data processing, and decision-making technologies has significantly enhanced the motion planning capabilities of robotic systems [1, 2]. These capabilities are particularly useful in mobile robots or robotic manipulators operating in human–robot interaction (HRI) environments. In such settings, robots must determine a feasible sequence of joint configurations at each time, enabling the robot to transition from an initial to a target position while avoiding potential obstacles along the way.

The standard motion planning module typically comprises two key components: the global motion planner and the local motion planner. The global motion planner generates kinodynamically feasible and executable trajectories using structured prior map information. Meanwhile, the local motion planner assists the robot in making real-time motion decisions within dynamic local environments [3].

Conventional artificial intelligence (AI) techniques for robotic motion planning, such as artificial potential fields (APFs) [4] or rapidly exploring random trees (RRTs) [5], have undergone extensive research and have demonstrated utility across various settings. However, these techniques are typically map-based, wherein the global and local planners function independently and require separate and hierarchical configuration. Such features pose challenges in adapting traditional motion planners to unstructured,

complex, and dynamic environments. For instance, APFs may encounter challenges in scenarios where attractive and repulsive forces are comparable, leading to convergence issues in local minima. On the other hand, RRTs are less efficient in executing rapid, reactive, and dynamically adaptive collision avoidance actions [6]. Consequently, investigating map-free motion planners that offer generalization, robustness, and adaptability is of significant importance.

In this regard, intelligent control employing machine learning (ML) arises as an alternative for controlling dynamic and flexible systems. Controllers operating on ML principles learn directly from examples, data, and experience, enhancing robots' decision-making capabilities when facing variable and unpredictable environments. In particular, reinforcement learning (RL) methods are a promising approach, as they hold the promise of solving control tasks in complex unconstructed environments [7]. Indeed, they allow agents to learn through interaction with their surroundings and, ideally, to generalize the learned behavior to new, unseen scenarios. However, despite its rapid evaluation and ongoing research [8], the full realization of this control technique remains a work in progress. In complex and variable environments, the learning process can be time-consuming, incurring computational overhead, as the agent is compelled to explore diverse state–action spaces to discern an optimal policy.

Nevertheless, with the development of parallel computing capacity, namely, graphics processing units (GPUs), and the implementation of deep learning within this domain, these methods have gained increasing interest due to their promising results in the motion planning of mobile robots [9–12] or in the navigation of autonomous unmanned aerial vehicles (UAVs), such as drones [13, 14]. RL-based planners enable end-to-end planning, eliminating the need for the complex hierarchical multilevel framework traditionally used to couple global and local planners. By unifying these planning components, RL-based approaches optimize the state update policy through iterative improvements driven by feedback from the environment. Tailored rewards structures and training paradigms can be developed to align with specific task objectives. These characteristics make RL-based motion planners well-suited for deployment in unstructured and dynamic environments, where real-time mapping is particularly challenging. However, while there exist papers that offer partial reviews of research about RL in the context of motion planning for robotic manipulators [15], to the authors' knowledge, there is no comprehensive review specifically focused on this domain. Thus, this paper seeks to fill this void and aims to review the most relevant and up-to-date work on the application of RL for motion planning of robotic manipulators over the past 5 years. The contributions of the paper are as follows:

- A comprehensive analysis of RL's current status and application in motion planning for robotic manipulators over the last lustrum, categorized by testing scenarios and identifying the RL properties in each study.
- An insight on RL's main current trends and challenges faced in motion planning for robotic manipulators, including commonly employed algorithms, performance and safety considerations, sample efficiency, generalization, and the simulation-reality gap.

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 motion planning for robotic manipulators. Section 4 briefly explains some fundamental concepts around motion planning. Section 5 describes the reviewed papers. Section 6 identifies the main trends and challenges in motion planning for robotic manipulators through RL based on the reviewed literature. Lastly, Section 7 concludes with a summary of the knowledge gained.

2. Search Methodology

The relevance of RL is underscored by its responsiveness and adaptability in dynamic environments, distinguishing it from other AI robotic control techniques. Consequently, its application in motion planning has emerged as a focal point of recent research. This trend is illustrated in Figure 1, depicting the volume of scholarly publications within the multidisciplinary Scopus[1] database from the onset of the last decade to the present. Employing the keywords "motion planning" OR "path planning" OR "trajectory planning" OR "collision avoidance" AND "reinforcement learning," a discernible upsurge in publications is evident from 2017 onward, reaching its peak in 2023. This surge is particularly associated with the incorporation of deep neural networks into RL algorithms, enabling the handling of more intricate environments. Despite this, to the best of the authors' knowledge, there is a dearth of analyses consolidating the foremost contributions in this domain concerning robotic manipulators. Such an analysis, outlining the current trends and challenges, could provide valuable insights for the field. Consequently, this review aims to furnish an overview of key studies utilizing RL in motion planning for robotic manipulators, coupled with an analysis of the prevailing trends and future directions in this domain. A summary of the chosen search criteria can be found in Table 1.

Initially, a search was conducted across multidisciplinary databases, specifically Scopus, Google Scholar[2], and Web of Science[3], covering the timeframe from 2018 to 2023. Various search terms relevant to the application context were employed, including "motion planning" OR "path planning" OR "trajectory planning" OR "collision avoidance" AND "reinforcement learning" AND "robot manipulator" OR "robot arm." The selection of these terms was guided by the rationale that research papers should establish a connection to motion planning, RL as a control technique, and a robotic arm.

Simultaneously, studies pertinent to the realm of RL but divergent from the scope of this review were systematically excluded. Notably, studies not in the English language and those not involving (rigid) serial manipulators were omitted.

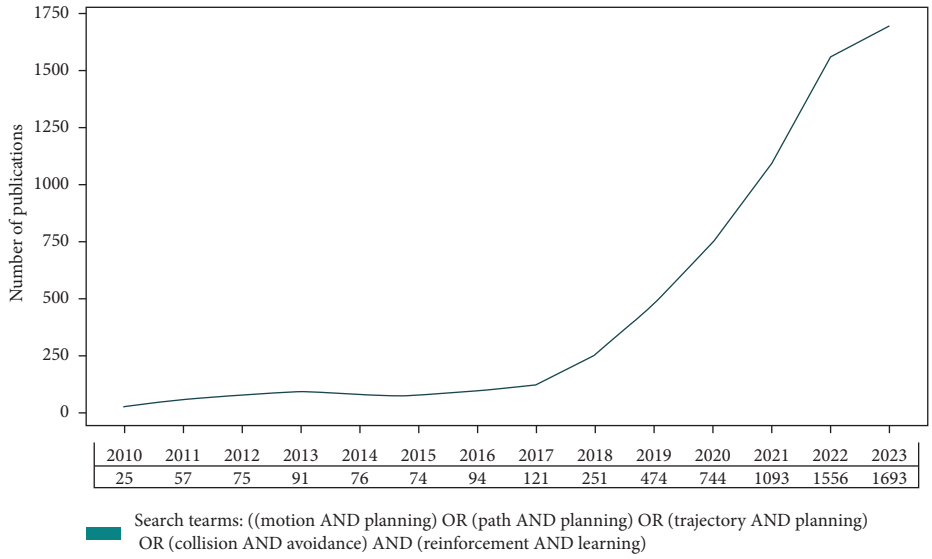

Figure 1: Publications per year on motion planning, path planning, trajectory planning, collision avoidance, and reinforcement learning in Scopus.

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

Search criteria	**Description**
Search terms	((Motion AND planning) OR (path AND planning) OR (trajectory AND planning) OR (collision AND avoidance)) AND (reinforcement AND learning) AND ((robot AND manipulator) OR (robot AND arm))
Time period	January 2018–October 2023
Publication type	Peer-reviewed academic conference papers and journal articles
Exclusion criteria	**Description**
Language	Non-English
Contextual	Nonrobotic or soft robotic manipulators

The search duration was delimited from January 2018 to October 2023. The initiation of this timeframe was determined based on the discernible emphasis on the application of RL in the motion planning of robotic manipulators, as indicated in Figure 1, approximately from 2018 onward. This increase is attributed to the implementation of neural networks into RL algorithms, resulting in the emergence of deep RL. This advancement enabled researchers to address previously intractable, more complex environments.

3. Background on RL

RL [16] is a type of ML in which an agent learns to interact with its environment to maximize rewards over time. This interaction is modeled as a Markov decision process (MDP).

The MDP is a process that defines sequential decision-making as a semirandom and agent-dependent pathway. Think of it as a sequence of steps where the agent is in a particular state, takes an action, and receives a reward based on the outcome of that action. For instance, imagine a robot navigating a 3×3 grid world (see Figure 2(a)). The robot can move up, down, left, or right. The robot's state is its current position on the grid world, and its actions are the directions it can move in. Each grid is assigned a reward or penalty based on the robot's movement and its proximity to the target location. Upon transitioning to a new state, the environment provides a corresponding numerical value for the grid the robot is in. In formal terms, an MDP is represented by the following tuple equation:

$$[S, A, P(s_{t+1}|s_t, a_t),\ R(s_t, s_{t+1}, a_t), \gamma], \tag{1}$$

where S is the set of possible states of the agent and A is the set of actions. $P(s_{t+1}|s_t, a_t)$ is the probability of transition to a future state s_{t+1} when the agent is in state s_t and applies action a_t. $R(s_t, s_{t+1}, a_t)$ is the reward that the agent expects to obtain when it transits from state s_t to state s_{t+1}, 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. Figure 2(b) shows the basic MDP scheme underlying the decision process of any RL agent. This process can be defined through the following sequence equation:

$$s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, \ldots \tag{2}$$

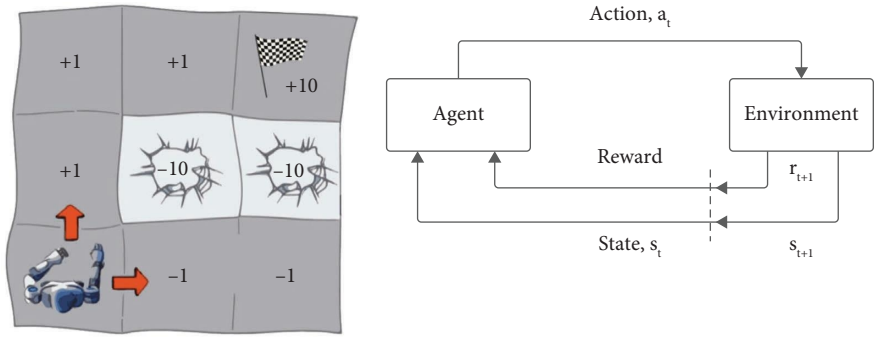

Figure 2: (a) An example of a robot moving in a grid world environment; (b) RL scheme [16].

A policy $\pi(a|s)$ is the strategy the agent uses to decide which action to take in each state. Likewise, the value function tells the agent how valuable a state is, based on the future rewards it can expect if it follows a particular policy.

3.1. RL Algorithms Taxonomy. Although it is difficult to make a standardized classification of RL algorithms due to their wide modularity, many current studies divide them into model-based and model-free algorithms:

- Model-based algorithm. The agent has access to a model of the environment, meaning it knows how its actions affect the environment. These algorithms are more efficient but require detailed knowledge of the environment's dynamics.
- Model-free algorithms. These algorithms rely on trial-and-error interactions with the environment, without needing to know how the environment works.

Table 2 summarizes the advantages and disadvantages of both methods.

Model-free algorithms can be further divided into three categories:

- Value-based algorithms. These algorithms estimate the value of each state or state–action pair. The agent selects actions based on these values. For instance, in the robot example, the agent would compute how good each grid is in terms of helping it reach the goal.
- Policy-based algorithms. The algorithms directly learn the best action for each state without using value estimates. The agent memorizes a policy that tells it which action to take in each state.
- Actor–critic algorithms. These algorithms combine the strengths of value-based and policy-based approaches. The actor learns the policy, while the critic estimates the action's quality.

4. Motion Planning, Path Planning, and Trajectory Planning

Motion planning is one of the integral components of the high-level planning and navigation module. This component allows the robot to move safely from an initial to a target position while considering collision avoidance with static or dynamic obstacles in the environment. Implementing a rapid online motion planning algorithm is vital, particularly in scenarios demanding safe collaboration between robots and humans [17].

Motion planning encompasses both path planning and trajectory planning. Path planning typically corresponds to global motion planning, as it involves generating a collision-free path based on geometry, disregarding the dynamics and mobility limits of the robot. As a purely geometrical concern, it does not consider specific temporal laws. Therefore, path planning is responsible for addressing geometric constraints, such as limitations in joint configurations and obstacle avoidance. Global motion planning focuses on finding this high-level route, guiding the robot from start to goal across the entire environment.

In contrast, trajectory planning aligns with local motion planning, as it entails assigning a temporal law to the geometric trajectory. This means that the trajectory corresponds to the robot's configuration at each moment. Consequently, the output of the trajectory planner must be temporally scaled to generate a feasible trajectory. In this manner, trajectory planning manages kinodynamic variables, including joint velocities, acceleration, torque, and time derivatives of joint angles [15]. Local motion planning ensures that the robot can follow the path generated by the global planning while adhering to dynamic constraints, making real-time adjustments in the presence of obstacles or changes in the environment.

In the majority of scenarios, path planning precedes trajectory planning; nevertheless, these two phases are not inherently separate. Path planning and trajectory planning are concurrently addressed in experiments involving via-points, where both the initial and final positions are specified [18]. This review covers motion planning as a whole.

5. Motion Planning Through RL for Robotic Manipulators

In scholarly discourse, RL finds application independently or in conjunction with other AI control methodologies for motion planning in robotic manipulators. The subsequent

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.

section seeks to delineate the principal contributions and limitations discerned within the reviewed studies. The section discusses separately those standalone RL applications and the instances where RL is combined with other AI control techniques, particularly APFs and RRTs. Note that the classification of the examined studies is determined by the task on which the studies are evaluated and their primary contribution(s). Specifically, the classification according to the evaluation scenarios is outlined as follows:

- Via-point tasks (Sections 5.1.1, 5.2.1.1, 5.2.2.1). Via-point tasks delineate scenarios where the robotic manipulator, starting from an initial position, is required to attain a predetermined endpoint.
- Pick-up, pick-and-place, and welding tasks (Sections 5.1.2, 5.2.1.2). Although pick-up, pick-and-place, and welding tasks share similarities with via-point experiments as point-to-point tasks, they are delineated as distinct activities in this analysis. The rationale behind this distinction lies in the additional complexity inherent in these tasks. In contrast to via-point experiments, these activities necessitate not only reaching the target point but also achieving a specific posture with the robot's end-effector to accomplish the task.
- Contact-rich manipulation tasks (Section 5.1.3). According to [19–21], a contact-rich manipulation task is defined as any task involving close interaction between the robot and its environment, characterized by complex, high-dimensional, and possibly nonlinear contact dynamics. These tasks typically entail contact situations such as sliding, sticking, or motion constrained by obstacles.
- Multiple tasks (Sections 5.1.4, 5.2.2.2, 5.2.3.1). In certain instances, researchers adopt a broader perspective by evaluating their policies across multiple scenarios rather than focusing on a single use case. This section categorizes studies wherein the proposed approach is assessed across various tasks.
- Other tasks (Section 5.1.5). This section encompasses studies whose evaluation scenarios do not fall into any of the categories defined earlier.

5.1. RL-Based Control Approaches

5.1.1. Via-Point Tasks. These tasks can be accomplished with either single- [22] or dual-arm [23] robot manipulators. In this category of tasks, studies commonly focus on improving the performance or safety of the motion planning policy [24, 25], the search for sample efficiency during training [26], and the generalization capability of the RL agents [27] (see Figure 3).

5.1.1.1. Performance and Safety. The performance and safety of RL policies have been a focal point in motion planning research over the past 5 years. However, the assessment of these themes can be directed toward various objectives. While specific studies adopt a more social perspective, exemplified by research on energy management in robotics to minimize industrial electricity consumption [28], the broader body of performance and safety-related investigations predominantly centers on ensuring smooth trajectories and collision avoidance.

Indeed, in HRI environments, the perceived safety and the safety of the human collaborator are two aspects that should be considered. Perceived safety in this context refers to the user's subjective assessment of the potential danger and their comfort level during interactions with a robot [29]. Consequently, numerous studies emphasize generating smooth and legible trajectories in motion planning to enhance the acceptance of the robot by the human collaborator.

Some methodologies adhere to traditional deep RL paradigms, concentrating on enhancing both trajectory smoothness and sample efficiency concurrently [30, 31] through techniques such as hindsight experience replay (HER) [32] or decaying episode mechanisms. HER addresses sparse reward challenges by retrospectively redefining failed task attempts as successful, facilitating the agent to learn from failures and thereby enhancing sample efficiency during training. On the other hand, the decaying episode mechanism is a way to dynamically adjust the step number within an episode during training based on the training accuracy reaching a certain threshold. This adjustment aims to bring about a new stable state in the training process without compromising the agent's performance in trajectory planning. Conversely, other investigations specifically target the enhancement of user comfort and task efficiency in addressing this concern within HRI scenarios. One notable investigation is by Yang et al. [33], wherein biomechanical attributes of human arm motion were integrated into robot motion planning to achieve a humanoid movement. The authors employed a motion capture system for collecting human arm movement data, extracted pertinent features from the human arm, and formulated reward functions for training through RL. However, despite achieving fluid and humanoid movements, the authors constrained the arm motion compared to real-world capabilities. In contrast, Zhao et al. [34] concentrated on enhancing the legibility of the robot's motions to facilitate the human collaborator in identifying the robot's intentions. Their approach involved a policy network serving as a motion planner and a recurrent neural network emulating the human perception of the

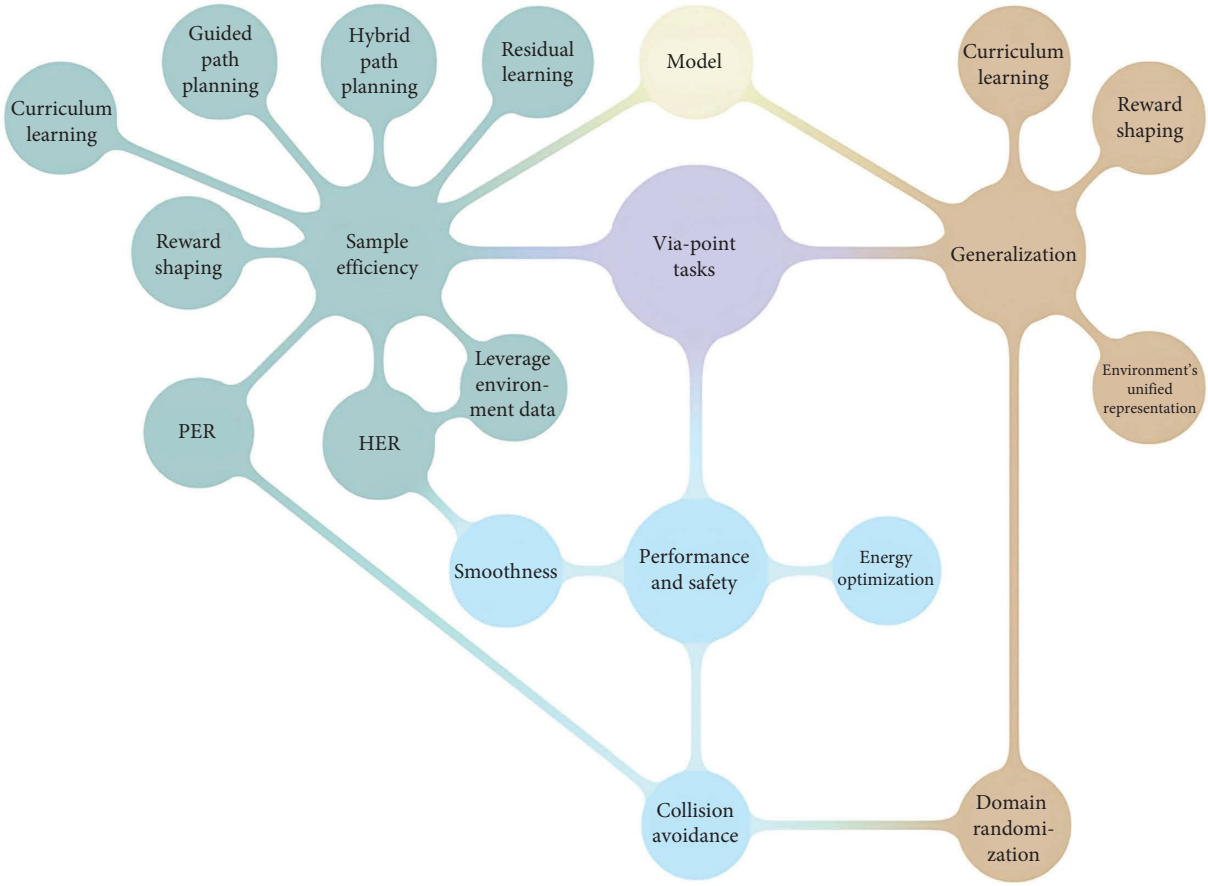

Figure 3: Main themes addressed by RL in via-point tasks.

robot's movement predictability. Lastly, they evaluated the efficacy of their method through participant assessments of the robot's legibility, learnability, and naturalness. Although this approach might excel in user comfort by ensuring that the robot's intentions are clear and predictable, the focus on legibility may come at the expense of trajectory smoothness, as the robot's motions may appear more mechanical compared to the fluid movements achieved in [33]. However, this trade-off makes this last method more effective in dynamic and HRI environments, where quick understanding of the robot's actions is crucial.

Notwithstanding, ensuring safety extends beyond perceived safety alone. The domain of motion planning predominantly tackles safety concerns through strategies focused on collision avoidance [35, 36].

In studies utilizing a single robotic arm, a subset of research assesses their methodologies using static obstacles [37–39] or with linear movements within a single plane [40]. Alternatively, some investigations opted for evaluating their approaches in scenarios featuring obstacles with unpredictable, random movements [41, 42]. These types of obstacles have the benefit of reflecting more realistic workplace scenarios. El-Shamouty et al. [43] proposed a framework that translated HRI tasks and safety requirements onto RL settings, wherein human motion simulated for collision avoidance was entirely stochastic. Conversely, Sangiovanni et al. [44] devised a real-time model-free collision avoidance strategy catering to robotic tasks featuring an unforeseeable obstacle invading the robot's workspace. However, they only considered the terminal element of the robot. Subsequently, this research was expanded to encompass a more comprehensive scenario, endowing the system with self-configuring capabilities, albeit presuming ideal internal control for the robot, which may not always be realistic in practical scenarios [45]. Nonetheless, the addition of self-configuring capabilities allowed the system to autonomously adjust its configuration more complex and dynamic environments, providing greater flexibility and robustness. This improvement made the system more effective in handling changing conditions, thus enhancing its applicability to a broader range of tasks. While these studies enhanced realism by incorporating continuous motion of obstacles or humans within an HRI environment, the reliance on random movements may still present limitations, inadequately capturing the full spectrum of potentially hazardous scenarios. To overcome this constraint, in [46, 47], the dynamic obstacle in the environment was characterized by real movements represented an operator's 3D point cloud. However, in the initial study, the negative reward assigned to the agent was applied only after surpassing a specific threshold. This method might extend training duration and potentially impact convergence if the

target remained unreachable during exploration. Therefore, in the subsequent study, the authors included a new source of reward to guide learning. Thus, they proposed a hybrid approach combining so-called extrinsic and intrinsic rewards. The intrinsic reward underwent updates via policy gradient methods, optimizing the extrinsic function to concurrently diminish the risk of collision and enhance process productivity.

On the other hand, some investigations delve into the intricacies of motion planning with dual-arm robots, emphasizing the avoidance of self-collisions and collisions with the surroundings. For instance, Salmaninejad et al. [48] focused on collision avoidance between two robotic arms. However, their methodology underwent testing solely in a two-dimensional space, where only one of the arms acquired collision avoidance capabilities while the other moved freely. In contrast, more recent investigations, such as [49], expanded the scope to encompass more complex three-dimensional scenarios. Here, the authors employed a long short-term memory (LSTM) network to forecast the position of dynamic obstacles in the environment. The network predicted the moving obstacle's position across multiple future sampling times by utilizing current and past position information, subsequently employed to train an RL agent. Additionally, the two manipulators were conceptualized as a single virtual manipulator to simplify spatial configuration and facilitate learning. As a counterpoint to this study, Wong et al. [50] advocated for individual control of each arm in a dual-arm robot through two distinct agents, one for each arm. This strategy conferred greater flexibility, obviating the need for preplanned trajectories for both arms before executing an action. Nevertheless, in such approaches, training from scratch can engender prolonged training periods. Consequently, various collision avoidance methodologies for both single-arm [51] and dual-arm robots [52] integrate strategies such as HER or prioritized experience replay (PER) [53] to enhance sample efficiency concurrently. This latter experience replay assigns higher priority to experiences that lead to large temporal differences and samples them more frequently during the learning process.

5.1.1.2. Sample Efficiency. Similar to performance and safety improvement, sample efficiency seems to be another highly researched aspect in RL-based motion planning. While other domains, such as robotic manipulation [54, 55], commonly address this concern through human demonstrations in the context of via-point experiments, although the endpoint is known, the computed trajectory may vary based on encountered obstacles. This constraint may restrict the utility of human demonstrations for acquiring initial trajectories. Human demonstrations typically yield geometry-dependent trajectories, potentially constraining the agent's capacity to avoid collisions with obstacles that intersect with said trajectory. Consequently, in the realm of motion planning, researchers propose alternative strategies.

Two distinct yet comparably aligned strategies are presented by Shen et al. [56] and Akinola, Wang, and Allen [57], guided path planning and residual RL, respectively. The former introduced a position-based servo method, wherein the robot advanced toward the target point, and the RL policy was activated upon detecting a potential collision, adjusting the robot's joints accordingly. The latter, in contrast, combined a low-dimensional policy to address the target-reaching task with residual learning to formulate a definitive policy that avoided obstacles while reaching the target. Therefore, the output of the residual policy was integrated with the output of the base policy. In both instances, the authors achieved a reduction in sampling requirements due to the trajectories established initially, diminishing the necessity for extensive exploration. However, the latter method may offer superior sample efficiency due to the residual learning framework, which reuses the base policy for the primary task while refining behavior during obstacle avoidance. This allows for more efficient use of samples, as the system builds on existing knowledge rather than learning everything from scratch. In contrast, in [56], guided path planning is effective but may require more samples in dynamic environments, as the RL policy is only activated during potential collisions, limiting its ability to reuse prior knowledge.

Abdi, Adhikari, and Park [58] also adopted a bifurcated strategy to address the sample efficiency problem in motion planning, presenting a hybrid approach. In the initial phase of the task, termed the "active approach," the authors devised a policy capable of determining a sequence of elementary actions, such as up, down, right, and left. By simplifying the problem, they reduced complexity and expedited the learning process. Once this policy was established, the subsequent phase, denoted the "passive approach," involved acquiring the necessary angles for the identified actions. This second network did not necessitate repeated training; instead, it could be trained once and subsequently applied. Nevertheless, given the simplicity of the actions, the authors could only evaluate the approach in two-dimensional grid world tasks. A year later, the authors expanded their research to three-dimensional tasks to overcome these limitations, integrating real-time object detection and localization for real-world applications [59].

Alternative methodologies, including curriculum learning [60] and reward shaping [61], have also proven effective in enhancing training efficiency and curtailing convergence times. Notably, several research studies have delved into applying reward shaping as a key strategy in this context. Innovations in dense rewards functions, as proposed by articles such as [62–64], showcased accelerated learning, quicker convergence, and improved exploration strategies. Although reward functions varied, all research studies shared a common thread by primarily mitigating the challenges associated with exploration blindness.

Other studies, on the other hand, attempt to leverage environmental data to increase sample efficiency. For instance, Bhuiyan et al. [65] employed observations solely based on virtual laser scanning. This approach enabled them to gather extensive environmental information, rendering their method less susceptible to environmental complexities and surpassing other state-of-the-art sampling-based

planners. On the other hand, as seen in [49], in [66], the authors reconfigured the configuration space of a multiarm manipulator, treating it as a single-arm virtual manipulator. Despite the increased task dimensionality, they utilized a single RL agent to control all robotic arms. Additionally, the incorporation of HER further enhanced sample efficiency. This research was subsequently expanded a year later, evaluating the methodology not only with static obstacles but also with dynamic obstacles following a periodic linear trajectory [67].

Lastly, some articles seek to address sample efficiency through models. This approach is exemplified by [68, 69]. In both studies, the integration of model predictive control (MPC) [70] with RL was employed to enhance sample acquisition with a heightened success rate. MPC, operating under constraint control, exclusively supplied training samples for the neural network during the training phase to ensure the safe operation of the manipulator. Utilizing these high-success-rate samples expedites the training process, diminishing early-stage failures in training and enabling the RL agent to formulate an appropriate policy based on input observations. Nevertheless, in [69], a noticeable constraint on generalization capability was acknowledged, underscoring the need for future investigations to address this issue. Ku et al. [71] introduced a hybrid data-model-driven algorithm to overcome this limitation. This approach combined particle swarm optimization (PSO) [72] with an actor–critic algorithm, utilizing PSO initially to optimize model parameters and subsequently pretraining the agent before starting to interact with the environment. This methodology exhibited adaptability in coping with environmental variations. It is worth noting, however, that nearly all studies focusing on enhancing sample efficiency undergo evaluation in collision avoidance tasks featuring static obstacles or linear movements.

Despite the potential exhibited by many of the methodologies in the reviewed studies, their generalizability to dynamic environments and real-world contexts raises significant considerations.

5.1.1.3. Generalization. Indeed, the current challenges in RL include the ability to generalize and the deployment of learned policies from simulation to reality. Therefore, there are motion planning-related studies that focus their concern directly on this topic. The concepts of generalization and policy deployment are intricately connected, with enhanced generalization capability contributing to developing robust policies that are susceptible to the simulation-reality gap. In this sense, while some researchers also opt for curriculum learning to facilitate this sim-to-real transfer and align robot behavior with simulation outcomes [73], others advocate the utilization of binary rewards at each time step, coupled with diversity rewards, to obtain robust policies [74]. Curriculum learning provides strong generalization by employing a progressive learning approach; however, it can be time-intensive to design and less adaptable to novel tasks not encountered during training. In contrast, using binary and diverse rewards may enhance the robustness of the policy, though this approach might compromise precise control during task execution.

Alternatively, Zhang et al. [75] proposed a unified representation of obstacles and targets, aiming to capture the underlying dynamics of the environment. Their approach utilized 3D bounding boxes to represent obstacles and target objects in both virtual and real-world settings, ensuring independence from the geometry and appearance of the objects. This design choice enabled generalization to unseen objects and scenarios.

5.1.1.4. Initial Findings. Undoubtedly, enhancing performance, safety, sample efficiency, and generalization is the primary focus within the realm of RL applied to motion planning in via-point tasks. While these aspects are addressed through different strategies, some of these strategies seem to exhibit interconnected influences on multiple aspects. Nonetheless, challenges arise where enhancing one aspect may inadvertently pose a bottleneck to others. At times, for instance, the enhancement of sample efficiency may compromise the generalizability of the RL agent. Notwithstanding such challenges, a discernible trend in recent years involves an increasing inclination toward simultaneously addressing multiple aspects, recognizing the interdependencies among performance, sample efficiency, and generalization. Illustratively, certain studies leverage models that facilitate the concurrent enhancement of both sample efficiency and generalizability.

5.1.2. Pick-Up, Pick-and-Place, and Welding Tasks. As in via-point tasks, the exploration of performance and safety, sample efficiency, and generalization constitutes key research areas in these tasks (see Figure 4).

5.1.2.1. Performance and Safety. Some studies concentrate on computing smooth trajectories [76, 77], while others endeavor to enhance the robustness of their methodologies by employing RL as a high-level decision-maker and seamlessly switching between conventional controllers [78, 79]. However, the predominant emphasis within this domain centers on ensuring safety [80].

Wu et al. [81] and Heaton and Givigi [82] focused on safety considerations, particularly in collision avoidance during picking-up and pick-and-place tasks, respectively. The former specifically addressed HRI environments, emphasizing the significance of preventing collisions between a robot larger than a human and the worker to avoid potentially severe injuries. On the other hand, the latter evaluated their approach using a dual-arm robot engaged in a tower-building task, where considerations involved avoiding collisions with both the robot itself and the surrounding environment during the execution of picking and placing tasks. Notably, while both approaches were posited as applicable in dynamic environments, neither underwent assessment with moving obstacles. This specific evaluation was undertaken in [83]. Here, the authors proposed a real-time collision-free trajectory planner,

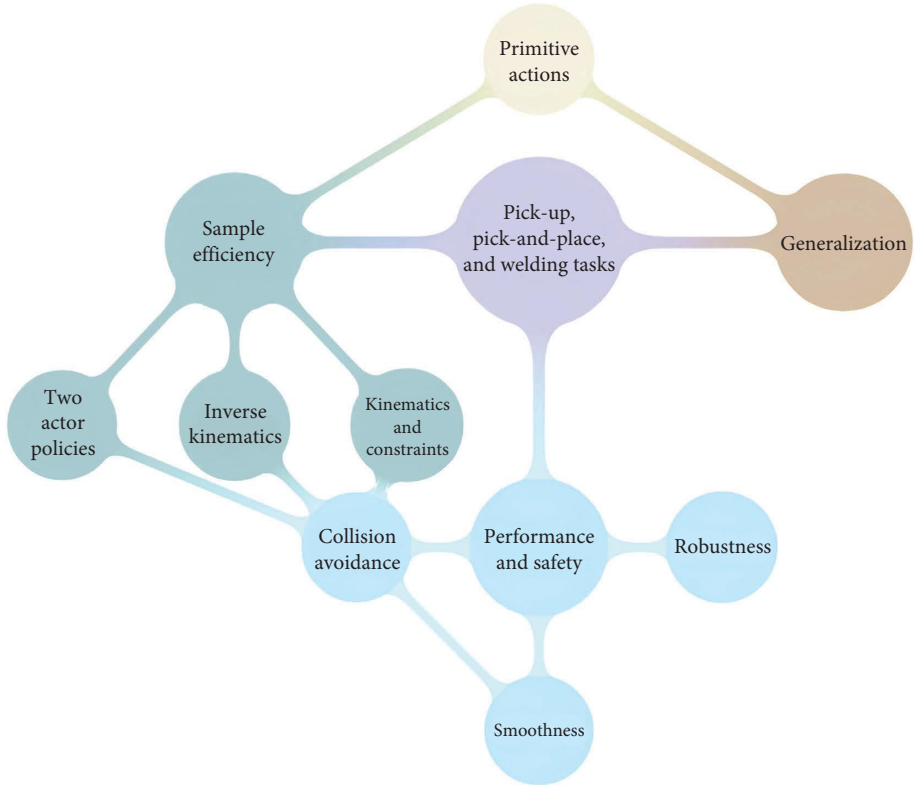

Figure 4: Main themes addressed by RL in pick-up, pick-and-place, and welding tasks.

assessed within an HRI scenario, with human arm positions tracked through the OpenPifPaf human pose estimation method [84]. Despite the encouraging results, their approach lacked evaluation in a real-world environment, where potential latency in skeleton tracking might impact real-time responsiveness to obstacles. This same constraint was encountered by Nicola and Ghidoni [85]. In this study, the authors concurrently addressed safety and trajectory smoothness to augment worker trust within an HRI environment. Their methodology defined each action as the concatenation of parametric subtrajectories derived from polynomial functions. This design choice afforded the authors the flexibility to influence the grade continuity and smoothness of the final trajectory by controlling the degree of the polynomial function and to provide a means to regulate not only the acceleration but also higher order time derivatives of position, such as jerk or snap. Nevertheless, a recurring observation in many of the mentioned studies is the compromise in sample efficiency due to the substantial exploration demands inherent in these tasks.

5.1.2.2. Sample Efficiency. In this context, certain studies tackle both safety and sample efficiency concurrently [86]. For instance, Hu et al. [87] focused on collision avoidance within welding tasks. The authors utilized two action–network structures to enhance sample efficiency in this case. Maintaining identical structures, the main actor–network and the subactor network differed solely in the state space they employed, with the latter designated for guided search on the main network. This structure enhanced sample efficiency by narrowing the exploration space, although it may require significant computational resources to manage both networks. Within a similar use case, Zhong, Wang, and Cheng [88], in turn, incorporated an inverse kinematics module to offer prior knowledge, thereby diminishing the need for extensive exploration in the state–action space. Likewise, to prevent excessive exploitation arising from this prior knowledge based on inverse kinematics, they dynamically adjusted the impact of the inverse kinematics-based action by introducing a gain module. This method might provide better sample efficiency than [87] through the inverse kinematics module, which accelerated learning by reducing the exploration space. Additionally, the gain module helped balance exploration and exploitation, avoiding overreliance on prior knowledge. Nevertheless, in such methodologies or when providing human demonstrations, it is worth considering that initial trajectories may exhibit significant dependence on the provided initial geometry, potentially jeopardizing the agent's generalization capability [89]. Hence, akin to via-point tasks, the concern for generalization ability extends to point-to-point motion planning tasks such as pick-and-place.

5.1.2.3. Generalization. Strudel et al. [90] proposed a hierarchical approach that targeted both sample efficiency and generalization. Starting from simple primitive actions learned through synthetic demonstrated trajectories using behavioral cloning, the RL agent executed high-level actions

by combining these low-level skills. This approach enhanced sample efficiency by constraining exploration to sequences with a limited number of primitive actions, while its versatility and robustness to diverse environmental perturbations were bolstered by not necessitating full-task demonstrations.

5.1.2.4. Initial Findings. Similar to via-point tasks, the central themes of concern in point-to-point tasks encompass performance and safety, sample efficiency, and generalization. However, the strategies to tackle these challenges extend beyond those discussed in Section 5.1.1. Notably, incorporating prior knowledge that minimizes the initial exploration without unduly constraining the agent's generalization capacity could be one of the most prominent ideas.

5.1.3. Contact-Rich Manipulation Tasks. In the realm of motion planning, while there are studies evaluating its applications in healthcare contexts [91], the literature predominantly features investigations targeting industrial environments. These studies commonly seek a trade-off between safety and performance in remanufacturing tasks. Some studies also consider aspects such as generalization (see Figure 5).

5.1.3.1. Performance and Safety. Veraamani and Muthuswamy [92] introduced a hybrid multirobot system for sheet metal milling. The system comprised a serial manipulator responsible for the machining operation, with its end-effector serving as a tool, and two fixtureless assembly platforms providing support from beneath the sheet metal. Employing inverse kinematics and prioritized RL to control the serial manipulator robot and the two robots acting as swarm robotic fixtures, respectively, the authors proposed a hierarchical-based decentralized offline planner. This planner effectively coordinated all three robots, calculating optimal collision-free trajectories during machining. However, the increased number of agents could lead to higher computational complexity. The approach ensured precise support and improved machining quality, but its reliance on static task planning may limit adaptability in dynamic environments. This approach was refined to extend its control strategy to drilling processes in a subsequent study [93], introducing a revised five-step locomotion strategy that enhanced path planning and reduced detours, yet the offline nature of the planning remained a constraint for real-time applications. The refined strategy successfully handled various drilling patterns while maintaining optimal coordination, though it may still face challenges in irregular environments.

In [94], in turn, the authors directed their attention to disassembly within an HRI environment. They operated under the assumption that the human collaborator could be positioned either to the right or left of the robot and might traverse from one side to the other. The robot was tasked with extracting a peg assembled on a base to the opposite side of the human; a maneuver devised to avert potential collisions. To tackle the challenge of generalization, the authors introduced randomization elements, such as varying the rotation of the base or adjusting the friction between the two objects. However, the task performed served as a rudimentary proof of concept. The robot started with the peg already grasped, and the disassembly process involved only a minor translation, rendering it a somewhat simplistic demonstration.

5.1.3.2. Initial Findings. It is noteworthy that numerous manufacturing tasks not only require carrying the robotic manipulator from one point to another but also intricate robotic manipulation. In this context, concurrently addressing the environment's safety to prevent potential collisions and optimizing task performance can prove particularly advantageous, especially in environments where robots assume a pivotal role.

5.1.4. Multiple Tasks. Occasionally, a control policy undergoing slight variations may behave correctly across different settings or tasks. In such scenarios, it becomes imperative to train policies that exhibit a high generalizability, facilitating seamless transferability across diverse tasks (refer to Figure 6).

5.1.4.1. Generalization. One potential strategy to achieve this is the employment of sparse rewards, fostering the development of policies less susceptible to performance variations resulting from minor disparities among environments. However, the absence of explicit guidance during the learning process may lead to the emergence of suboptimal policies. To address this challenge, Al-Gabalawy [95] employed HER to enhance the sample efficiency of his approach, which he subsequently assessed in the context of FetchSlideball and FetchToss tasks. Both tasks involved a robotic arm to move an object to a designated point, potentially beyond the arm's immediate reach. The sole distinction between the two environments lays using a ball instead of a cylinder, accompanied by an increased distance to the goal. Nevertheless, the study outcomes were deemed less satisfactory, with the author concluding that HER encounters difficulties in tasks characterized by extensive goal distances and intricate solutions.

Still, Liu et al. [96] also leveraged HER as a mechanism to address the challenges associated with sparse rewards. This particular investigation focused on cooperative tasks involving reaching, pushing, and picking and placing with a dual-arm robot, all while ensuring collision avoidance between the two arms. Notably, each arm was autonomously controlled by an independent agent, prompting the adoption of a centralized training framework with decentralized execution. To facilitate this, the authors introduced a multiarm actor–critic algorithm. In this algorithm, the critic component could access supplementary information regarding other agents' policies, while the actor component exclusively acquired local information. Despite evaluating tasks

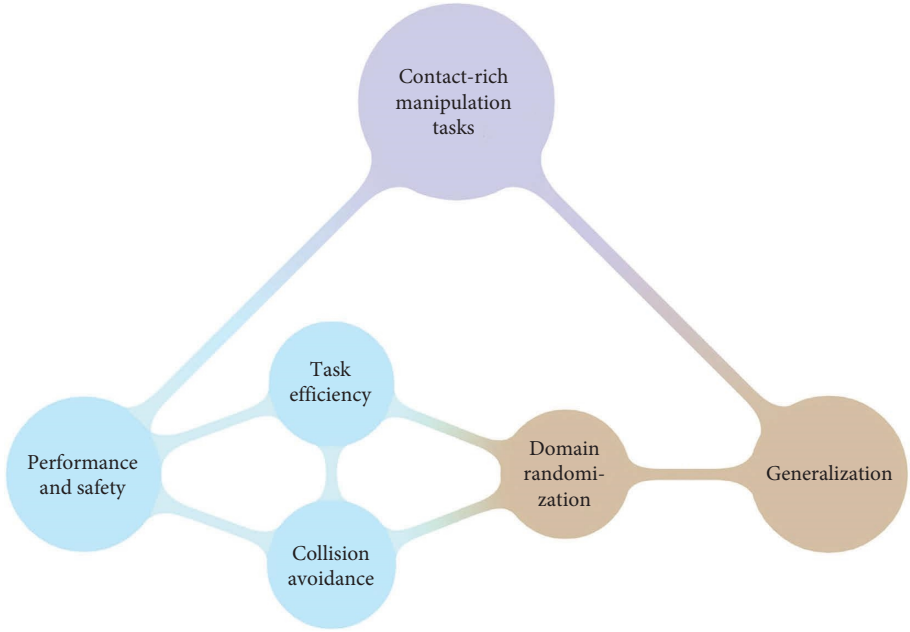

Figure 5: Main themes addressed by RL in contact-rich tasks.

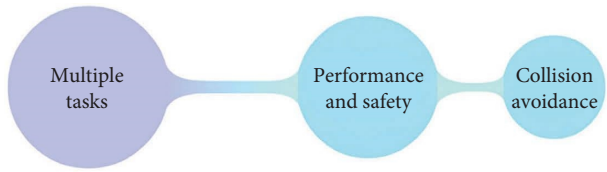

Figure 6: Main themes addressed by RL across multiple tasks.

considered relatively straightforward, as opposed to [95], the authors achieved optimal results, even when deploying their approach in reality.

5.1.4.2. Initial Findings. Evidently, the employment of sparse rewards represents sparse rewards represent a viable strategy for acquiring generalized policies conducive to seamless transferability across analogous tasks. However, their usage is often accompanied by HER. This integrated approach serves the twofold purpose of enhancing sample efficiency during the learning phase and mitigating the risk of developing suboptimal policies.

5.1.5. Other Tasks. Up to this point, all studies assessed their approaches to tasks falling within the predefined labels. Nonetheless, there might exist studies whose application scenarios cannot be readily classified into these established categories. Notably, there is only one study with a use case distinct from those mentioned previously, focusing on navigating a robot through a duct, commonly referred to as the duck-enter task. In a sense, this task consists of guiding the robot from an initial point to a target position within the confines of a duct, presenting challenges owing to the restricted space and complex interior (see Figure 7). Likewise, the skills developed for such operations could potentially find application in more prevalent manufacturing scenarios like paint spraying or welding.

5.1.5.1. Performance and Safety. Addressing the constraints of tight spaces, Hua et al. [97] concentrated on RL safety, specifically emphasizing robot safety achieved through collision avoidance. Therefore, their proposed trajectory planner decomposed the duct entry task into two subtasks: a reachability task managed by the inverse kinematics of the redundant robot end-effector and an obstacle avoidance task treated as an RL-based self-motion optimization problem. Notwithstanding, despite the favorable outcomes achieved, the authors underscored the need to address limitations associated with generalization.

5.1.5.2. Initial Findings. Despite the lack of literature addressing motion planning tasks that deviate slightly from those expounded upon in preceding sections, a recurring challenge is discernible. Frequently, obtaining a policy that performs well in a specific environment poses limitations on its applicability in environments exhibiting small variations. Achieving a trade-off between performance and agent generalizability becomes essential in such scenarios. To

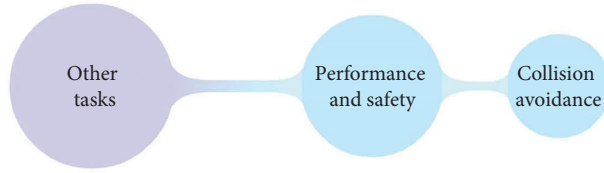

Figure 7: Main themes addressed by RL in other tasks.

address this challenge, domain randomization stands out as a commonly employed technique.

Figure 8 presents illustrative examples of the evaluation scenarios discussed, while Table 3 provides a comprehensive overview of the key features of the motion planning articles reviewed employing RL.

5.2. Combination of RL With Other AI Control Techniques

5.2.1. RL With APFs. APFs model the robot's motion by defining attractive and repulsive forces within the environment, directing the agent toward a target position while steering clear of obstacles [98]. Within the literature, there exists research that integrates this technique with RL, aligning with the primary themes of performance and safety, as well as sample efficiency. In the following, the subsequent review delves into the studies identified in the literature that combine RL with APFs according to the type of task in which they assess their approach, alongside their principal contributions.

5.2.1.1. Via-Point Tasks. APFs struggle to avoid collisions near the target due to potential similarities that might occur between attractive and repulsive forces, leading the agent into local minima. The combination of RL and APFs is intended to improve this aspect along with sample efficiency (refer to Figure 9).

5.2.1.1.1. Performance and Safety. Li, Gong, and Yu [99] proposed a hybrid approach combining RL and APFs to overcome the limitation of the latter control technique in collision avoidance. Initially, APFs were employed to perceive the environment and create a collision-free trajectory. When the manipulator entered the effective collision radius of a dynamic obstacle or a collision event occurred, the so-called distance or force enhancement factors were activated to move away from the obstacle or impact site, respectively. To ensure the functionality of the method, both factors were integrated into the learning reward, representing the distance between the robot and the obstacle and the collision force. However, although the study did not explicitly emphasize sample efficiency, RL can also be employed as a plausible strategy to improve the convergence times of APFs. Indeed, APF method's output contains many trajectory nodes, which reduces the computational speed of the robotic inverse kinematics analysis.

5.2.1.1.2. Sample Efficiency. As an illustrative case, Fang and Liang [100] utilized RL to filter unnecessary trajectory nodes, reducing the computational load of robotic inverse kinematics analysis during path planning, improving efficiency, and reducing energy consumption.

Thus far, the two previously mentioned methodologies employed APFs as the principal approach for motion planning, utilizing RL as a supplementary tool for refining trajectories computed by the former. However, these control techniques can be reversed. Zheng et al. [101] prioritized RL as the primary method for trajectory planning in dynamic scenarios, employing APFs to formulate reward functions. In this context, APFs were integrated to facilitate real-time obstacle avoidance by configuring the obstacle as an exclusion potential reward function and the target position as an attraction potential reward function. This strategy effectively minimized invalid exploration, thereby enhancing sample efficiency. Although utilizing APFs as the primary approach, with RL refining trajectories, can enhance sample efficiency and reduce computational cost in static or semidynamic environments, APFs may restrict the policy's adaptability in more complex, dynamic scenarios, where local minima can create bottlenecks. Conversely, employing RL as the primary method can improve sample efficiency in dynamic environments by minimizing invalid exploration through the guidance of APF-based reward functions. Nonetheless, the authors noted a reduction in generalizability due to the reward function capturing only scenario-specific features.

5.2.1.1.3. Initial Findings. In summary, the synergistic use of APFs and RL holds promise for enhancing sample efficiency and reducing algorithm convergence times. However, akin to other methods, the combined use of these control techniques may create a bottleneck that challenges the generalization capability of the agent.

5.2.1.2. Pick-and-Place Tasks. The combination of RL with APFs can also be harnessed to enhance other aspects of an agent's performance, including accuracy (refer to Figure 10).

5.2.1.2.1. Performance and Safety. In [102], RL was applied to motion planning across large ranges, while APF was only employed as an attractive potential field for local position adjustments in pick-and-place tasks. Thus, the switch between motion planning algorithms occurred when the distance between the effector and the target position fell below a defined threshold. Notably, the authors employed a simplified APF algorithm, omitting the establishment of repulsive potential fields. This simplification obviated the need to select control points on the manipulator's links and calculate distances between these points and obstacles. By sidestepping these complexities, the authors averted local

FIGURE 8: Motion planning evaluation scenarios: (a) and (b) via-point tasks [44, 51]; (c) pick-up tasks [90], (d) and (e) pick-and-place tasks [78, 89]; (f) disassembly tasks [94]; (g) duck-enter tasks [97].

minima arising from similar attractive and repulsive forces, thereby facilitating high-accuracy motion planning.

5.2.1.2.2. Initial Findings. Integrating RL with APFs may find utility in tasks demanding high precision. In situations where RL could yield noisy movements, mainly when the manipulator is near the target point, APFs, with their inherent attraction capability, excel in executing tasks with enhanced accuracy.

5.2.2. RL and RRTs. In addition to studies that integrate RL with APFs for motion planning, a body of literature explores the combination of RL with RRTs. This latter control technique employs probabilistic algorithms to iteratively expand a tree structure from a randomly sampled configuration space, thereby generating feasible paths [103]. These studies also address performance and safety, sample efficiency, and generalization. The ensuing section comprehensively examines these papers, according to their use case and respective contributions.

5.2.2.1. Via-Point Tasks. In via-point tasks, current research encompasses the three primary lines of investigation, namely, performance and safety, sample efficiency, and generalization, often addressing them collectively (see Figure 11). Improving the performance and safety of policies can be targeted at manifold aspects.

5.2.2.1.1. Performance and Safety. Jing et al. [104] were pioneers in integrating RRTs and RL in surface inspection applications, specifically targeting cycle time reduction. Their methodology involved employing RRTs to randomly sample viewpoints surrounding the target object, followed by applying an online RL-based tree search planning

Table 3: RL properties of each study in the motion planning (only RL). The studies are classified according to their evaluation task and sorted by publication date.

Task	References	Year	Control method	RL taxonomy	Algorithm	Learning	Observation space							Action space				Reward	Simulation/ real-world
							Joint angle	Joint velocity	End-effector pose/pos	External force/torque	Target pose/pos	Obstacle pose	Others	Joint angle	Joint velocity	Pose/pos (cartesian)	Others		
Via-point	[24]	2018	P	VB	DQN	S							X	X			X	B	S
	[44]	2018	V	VB	NAF	S	X	X	X	X	X		X		X			D	S
	[61]	2018	P	—	Metropolis	S							X			X		D	B
	[35]	2019	P	VB	Q-learning	S	X							X				S	S
	[46]	2019	V	AC	DDPG	S	X			X	X				X		X	D	S
	[62]	2019	T	AC	DDPO DDPG	S			X		X	X					X	D	S
	[30]	2020	P	AC	A3C TD3+HER	S	X				X			X				S	B
	[34]	2020	V	AC	PPO	S	X		X	X			X		X			D	S
	[37]	2020	P	VB	Fitted SGTPD	S	X										X	D	S
	[43]	2020	P	AC	DDPG + HER DDPG + HRA	S			X	X	X	X				X		D	S
	[45]	2020	V	VB	NAF + SBL	S	X	X	X	X	X		X		X			D	S
	[97]	2020	P	VB	Q-learning	S			X					X				B	S
	[66]	2020	P	AC	SAC + HER	S	X							X			X	S	B
	[22]	2021	P	AC	DDPG	S	X				X		X	X				B	S
	[25]	2021	P	—	—	S			X	X	X		X			X	X	B	S
	[33]	2021	P	AC	DDPG + HER	S	X				X		X	X				D	B
	[47]	2021	V	AC	IRDDPG	S	X			X	X				X			D	S
	[50]	2021	P	AC	SAC	S	X				X	X	X	X			X	B	B
	[51]	2021	V	AC	DDPG + HER	A	X	X	X				X		X			D	B
	[56]	2021	V	AC	SAC	S	X			X					X			D	S
	[57]	2021	P	VB	Q-learning + HER	S	X				X		X	X				D	S
	[58]	2021	P	VB	Q-learning	S					X		X	X				B	B
	[60]	2021	P	AC	PPO	S	X		X	X				X				B	S
	[67]	2021	P	AC	SAC + HER	S	X							X			X	S	B
	[68]	2021	—	VB	DQN	S												—	S
	[23]	2022	P	AC	DDPG	S	X				X		X	X				B	S
	[27]	2022	P	VB	Q-learning	S					X		X			X		B	B
	[28]	2022	T	AC	TD3	S	X	X			X	X					X	D	S
	[36]	2022	P	AC	SAC	S		X					X				X	B	S
	[38]	2022	P	VB	Q-learning DQN SARSA DDQN	S					X	X	X				X	B	B
	[39]	2022	P	AC	PPO	S	X				X		X	X				D	B
	[40]	2022	P	VB	DQN	S	X		X		X	X				X		B	S
	[41]	2022	V	VB	NAF	S	X	X	X	X	X				X			D	S
	[42]	2022	P	AC	DDPG SAC	S			X		X		X			X		D	S
	[49]	2022	P	AC	SAC	S	X				X	X		X				S	B
	[59]	2022	P	VB	Q-learning	S					X		X	X				B	B
	[63]	2022	P	AC	PPO	S			X		X	X				X		B	S
	[69]	2022	V	VB	Q-learning	S	X		X								X	D	S
	[71]	2022	P	MB-AC	P-SAC	S	X				X	X	X	X				D	S
	[73]	2022	P	AC	PPO	S	X	X	X		X		X	X				B	B
	[75]	2022	P	AC	SAC	S	X				X	X		X				B	B
	[26]	2023	P	VB	Q-learning	S	X					X	X	X				D	S
	[31]	2023	V	AC	PPO	S			X		X		X		X			B	B
	[64]	2023	P	AC	SAC DDPG	S			X		X		X			X		D	S
	[65]	2023	P	AC	PPO	S	X		X		X		X			X		B	S
	[74]	2023	P	AC	SAC + HER	S	X		X					X				D	S

Table 3: Continued.

Task	References	Year	Control method	RL taxonomy	Algorithm	Learning	Observation space							Action space				Reward	Simulation/real-world
							Joint angle	Joint velocity	End-effector pose/pos	External force/torque	Target pose/pos	Obstacle pose	Others	Joint angle	Joint velocity	Pose/pos (cartesian)	Others		
Point-to-point																			
Pick up	[80]	2018	P	AC	DDPG	S	X				X	X		X				D	S
	[90]	2020	V	AC	PPO	A							X				X	S	B
	[81]	2023	P	AC	DDPG	S					X		X	X				B	B
	[83]	2023	P	AC	SAC	S	X		X		X	X	X	X				D	S
Pick and place	[78]	2019	P, A	VB	DQN	S	X	X						X			X	D	B
	[76]	2020	P	AC	PPO	A	X						X	X				D	B
	[77]	2021	P	AC	—	S	X		X		X		X	X				B	B
	[79]	2021	V	VB	Q-learning	S	X		X	X	X				X			S	S
	[85]	2021	P	AC	TD3	S	X	X			X	X	X	X				D	S
	[82]	2023	P	AC	SAC	S			X		X	X				X		B	S
	[89]	2023	C/I	—	IRL (waypoint trajectory generator)	A	X	X			X			X	X			D	B
Welding	[88]	2021	V	AC	DDPG	A	X		X		X	X			X			B	S
	[86]	2022	A	—	—	S	X	X					X				X	B	S
	[87]	2022	P	AC	SDAC	S	X		X		X	X	X	X				B	S
Contact-rich manipulation	[92]	2021	P	VB	SARSA	S						X	X				X	D	B
	[91]	2022	—	AC	DDPG + HER	—												—	S
	[93]	2022	P	VB	SARSA PPO	S						X	X				X	D	B
	[94]	2022	P, C/I	AC	SAC DDPG	S			X	X		X	X			X		B	B
Multiple	[95]	2021	P	AC	DDPG + HER	S			X		X					X		S	S
	[96]	2021	P	AC	MADDPG + HER	S			X		X		X			X	X	S	B
Other	[97]	2020	P	AC	DDPG	S	X				X		X	X			X	S	S

Note: Control methods: position control (P), velocity control (V), torque control (T), compliance/impedance control (C/I). 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).

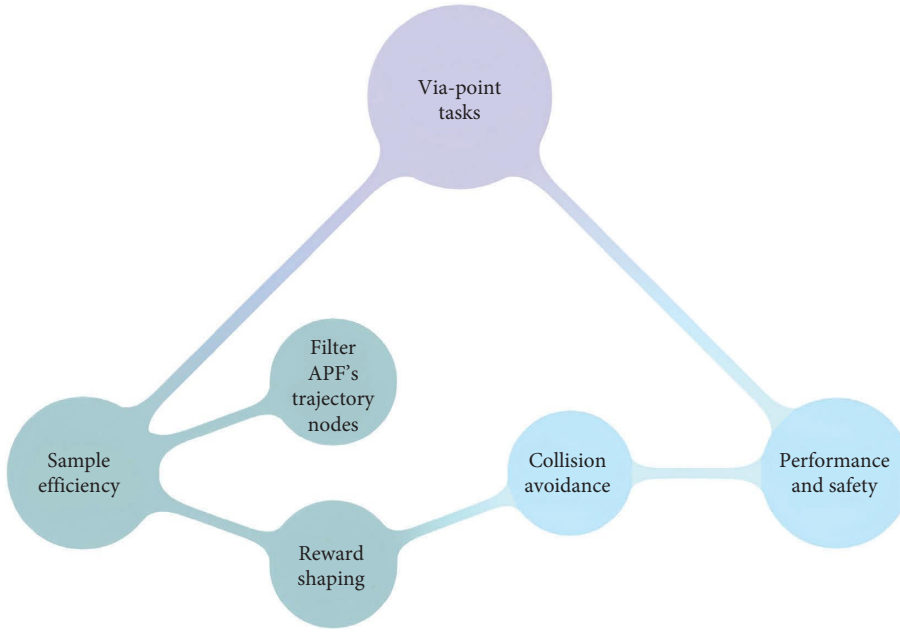

Figure 9: Main themes addressed by RL in via-point tasks in combination with APFs.

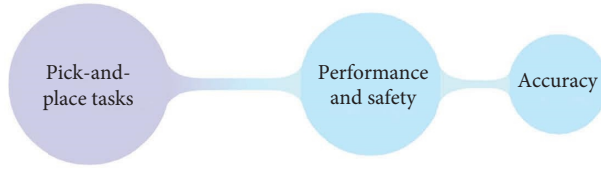

Figure 10: Main themes addressed by RL in pick-and-place tasks in combination with APFs.

algorithm. This RL algorithm determined the optimal viewpoint for guiding the robot to fulfill surface coverage requirements. The implementation resulted in an average cycle time reduction of 24%. Despite this success, the authors acknowledged limitations in sample efficiency.

Yet, the combination of RRTs and RL presents a promising strategy to overcome the sample efficiency challenges inherent in RL. Studies such as [105, 106] concurrently tackled sample efficiency and other performance aspects such as safety and energy optimization, respectively. Notably, optimal RRT (RTT*) was employed in both cases. In [105], RTT* facilitated trajectory exploration for multiple robot arms, while Q-learning was utilized to ensure collision avoidance between these arms. This combination of control strategies empowered each robot arm with the flexibility to adapt to unforeseen circumstances through RL while guaranteeing optimality via graph search. Similarly, in [106], RTT* capabilities were harnessed for initial random sampling. However, in this case, the authors leveraged the RL agent's replay buffer to optimize the policy based on accumulated exploration experiences progressively. While none of the papers explicitly focused on enhancing sample efficiency, the method utilized in [105] has the potential to lead to faster convergence. However, it may prove less effective in optimizing energy usage and adapting to unknown environments. In turn, [106] offers improved long-term sample efficiency by integrating RL with traditional path planning, allowing for energy-efficient solutions through policy optimization. However, this method may require higher computational resources during initial learning phases due to the complexity of energy optimization across multiple dimensions.

5.2.2.1.2. Sample Efficiency and Generalization. In contrast, alternative investigations sought to simultaneously address sample efficiency, generalizability, and policies' robustness to mitigate the simulation-reality gap. Zhou et al. [107] exemplified this approach by introducing a motion planning strategy rooted in residual RL. Their methodology was initiated with an initial policy derived from RTT*, which served as both a directional guide and a means to simplify the convergence challenges of the RL algorithm. The RL component supplemented the initial policy by providing key elements such as self-adaptation and generalization. Notably, it could avoid complex obstacle geometries that were not considered in the initial policy. In turn, Zhang, Guo, and Bai [108] utilized RRTs to compute heuristic reward functions employed in training an RL agent. These reward functions

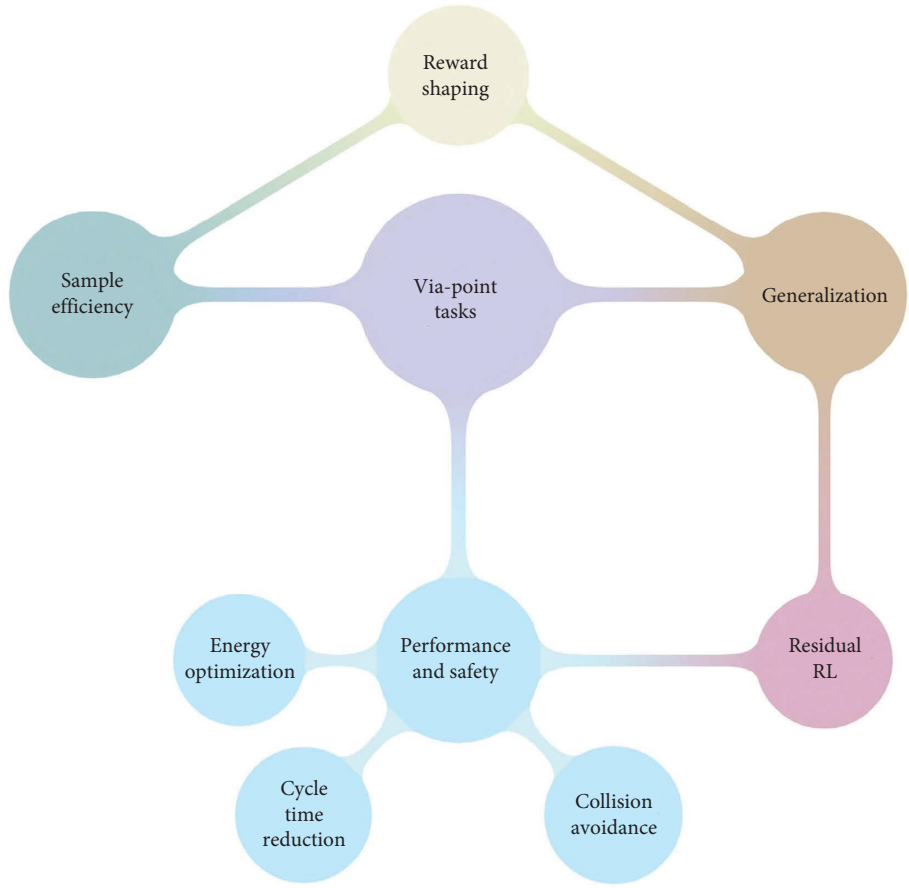

Figure 11: Main themes addressed by RL in via-point tasks in combination with RRTs.

not only expedited the convergence speed during training but also yielded a robust policy capable of zero-shot deployment in a real-world environment.

5.2.2.1.3. Initial Findings. The minimal computational requirements of RRTs for trajectory generation in an environment can be harnessed to diminish the exploration times needed by an RL agent during the learning process. Following this, RL can be employed to fine-tune these trajectories or acquire more generalized motion patterns.

5.2.2.2. Multiple Tasks. Within the literature, several studies that combine RL with RRTs also assess their methodologies across diverse tasks. While some tasks involve point-to-point actions, such as picking-up objects, many are entwined with contact-rich manipulation tasks, such as pushing an object on a surface or assembly processes. These investigations typically employ RL to enhance performance or generalization, while RRTs are tasked with improving sample efficiency (see Figure 12).

5.2.2.2.1. Performance and Safety. An illustrative instance is the work by Yamada et al. [109], where the emphasis lies on manufacturing performance and sample efficiency. The authors proposed a framework coupling an actor–critic algorithm with an RRT-based motion planner. In this setup, while the motion planner performed large joint displacements and explored obstructed environments with collision-free trajectories, the RL policy handled fine manipulations within the agent's action space. Nonetheless, the authors outlined the translation of their outcomes into real-world applications as a future research direction.

5.2.2.2.2. Generalization. Indeed, sample efficiency and robustness to the simulation-reality gap were two of the main concerns of the authors in [110]. Employing a methodology akin to [109], the authors utilize RRTs to generate a reference path, deemed computationally more efficient than generating a path through RL from scratch. Subsequently, the policies were parameterized with goal locations to enhance generalization, allowing the agent to be trained for multiple goals concurrently.

5.2.2.2.3. Initial Findings. As stated before, similar to integrating RL with APFs, the combination of RL with RRTs holds the promise of enhancing the sample efficiency of motion planning algorithms. RRTs can generate a reference trajectory with lower computational demands compared to

Figure 12: Main themes addressed by RL across multiple tasks in combination with RRTs.

RL. Following this, RL can be utilized to refine the initial trajectory or execute various types of manipulation tasks.

5.2.3. RL and Multiple Control Techniques. RL, APFs, RRTs, and various other control techniques for motion planning possess distinct advantages and shortcomings contingent upon the specific robotic context. The efficacy of these control strategies varies, implying that optimal performance hinges on selecting the most suitable control strategy for a given circumstance.

5.2.3.1. Multiple Tasks. At times, RL may operate as a discrete agent, triggering low-level controls contingent on the task requirements and environment conditions (see Figure 13).

5.2.3.1.1. Performance and Safety. Sacchi et al. [111] developed a high-level decision-making RL agent collision avoidance capability and the computational expense associated with different control strategies. Among three alternative motion planning strategies, APFs, convex optimization, and joint space position control, this agent could dynamically choose the most appropriate strategy based on the encountered circumstances in via-point, pick-and-place, and welding tasks. This dynamic strategy selection mechanism aimed to optimize the execution of a given task by adapting to the specific demands of the robot's environment.

5.2.3.1.2. Initial Findings. On occasions, employing a singular control method may not be optimal, especially when the same method is intended for diverse tasks or scenarios. In such instances, implementing a high-level controller to discern the most suitable low-level control method proves to be a valuable proposition.

Table 4 provides a comprehensive overview of the key features of the motion planning articles reviewed employing RL with other control techniques.

6. Analysis

6.1. Testing Scenarios. Experiments involving via-points [44] or point-to-point tasks such as pick-up [81], pick-and-place [85], or welding [88] represent predominant use cases for evaluating the effectiveness of RL policies in the motion planning of robotic manipulators. The prevalence of these tasks is evident in the schematic diagram presented in Figure 14, which encompasses the 79 articles reviewed, revealing that around 88% of these studies are associated with one of these specified tasks. Note that there are more use cases than reviewed papers, as there are works that cover multiple tasks.

Among these tasks, the via-point experiment stands out as the most frequently employed use case. In such experiments, the robotic manipulator initiates from an initial position and, at each time instant, endeavors to compute a feasible trajectory until it reaches a designated target point, systematically steering clear of potential collisions with its surroundings. In such instances, the RL agent's observations often include the robot's joint configuration, the final target's location, the spatial arrangement of obstacles within the environment [50], or the relative position between the robot and the obstacles [23].

While these tasks are instrumental for assessing the viability of the proposed RL-based motion planning approach, it is imperative to contemplate the ultimate role of the robotic manipulator in practical scenarios. As automation and decision-making capabilities in robotics advance, robots are increasingly tasked with performing intricate manipulation assignments [112]. In such cases, the learned policy must be capable of sequential decision-making by composing simpler behaviors. For instance, in HRI assembly or disassembly operations, the robot must adeptly calculate a collision-free path, avoiding collisions with humans and the surrounding environment while, at the same time, maintaining workflow efficiency through the execution of the designated manipulation tasks. However, RL faces challenges with temporally extended tasks. Furthermore, within workspaces characterized by a significant degree of variability stemming from the unpredictability inherent in human behavior, the efficacy of a manipulation task might be contingent upon the person's location. Could RL, therefore, be utilized to automatically generate the necessary abstractions for both motion planning and subsequent tasks? Approaches such as goal-conditioned RL [113] or hierarchical RL [114] offer a way for the agent to focus on which states to reach rather than how to reach them. In this sense, recent studies have demonstrated the effectiveness of RL in concurrently managing manipulation and collision avoidance tasks [94]. These methods hold significant potential, particularly for policy implementation in industrially relevant environments. In this regard, there exists a promising avenue for further research aimed at imbuing manipulator robots with heightened cognitive abilities in such multifaceted tasks.

6.2. RL Algorithms. Following the taxonomy of RL algorithms described in Section 3.1, all the reviewed papers utilize model-free algorithms. This is grounded in the fact that most of these papers assess the effectiveness of their

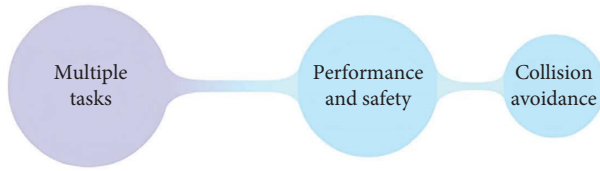

Figure 13: Main themes addressed by RL across multiple tasks in combination with other control techniques.

approaches in dynamic environments featuring moving obstacles, where the modeling of transition dynamics is unfeasible (see Table 5. Note that there are fewer algorithms than articles reviewed as some studies do not specify the algorithm employed).

Within the model-free algorithms category, value-based and actor–critic algorithms prevail. Nonetheless, the former often operate within discrete action spaces. Notably, a significant portion of works employing value-based algorithms evaluate their approaches in two-dimensional grid world scenarios, where the robot end-effector must be moved between grid positions [27, 38, 58]. Consequently, actor–critic algorithms emerge as the preferred option for researchers tackling challenges in more realistic environments. Among these algorithms, SAC [115], DDPG [116], and PPO [117] constitute the most frequently adopted choices, as illustrated in Figure 15 (58 actor–critic algorithms considered). While PPO is an on-policy algorithm, adjusting the policy to maintain a relatively minor deviation from the preceding one, SAC and DDPG are off-policy algorithms. These two algorithms employ a replay buffer memory to archive experiences, leveraging the most valuable information for efficient training. SAC strives to optimize both the maximum entropy and the discounted long-term return. The integration of maximum entropy aids in augmenting exploration where it is deemed essential. Conversely, DDPG is characterized as a deterministic algorithm using deep function approximators to learn the policy and estimate the value function within continuous, high-dimensional action spaces.

6.3. Safety. Safety in RL represents one of the main bottlenecks to the safe and productive use of robots in real manufacturing or healthcare contexts. This safety concern is primarily associated with the training phase, where the robot must adhere to specific safety requirements during exploration. Unlike scenarios such as video games, where an RL agent operates within the same state space in which it was trained, this cannot be guaranteed in robotic applications. Robots may encounter unforeseen perturbations that push them beyond their training space, leading to an entrance into an unknown state not accounted for during training. Consequently, safety considerations in motion planning extend to both the learning phase [118, 119] and the postdeployment phase [120, 121].

Concerning safety during the learning phase, Brunke et al. [119] established a framework with three safety levels. Safety Level 1 promotes safety and robustness in RL but does not assure strict adherence to safety constraints. One approach to achieving this is by introducing a penalty when undesired collisions with the environment occur during the learning process [45]. Safety Level 2 aims to enhance performance safely by learning uncertain dynamics. While there are no rigid safety guarantees at this level, it allows for estimating the likelihood of safety issues, often leveraging prior knowledge. For instance, Park et al. [49] utilized an LSTM network to predict the positions of dynamic obstacles in the robot's environment. Lastly, Safety Level 3 involves providing safety certificates to a controller that does not inherently consider safety constraints, potentially achieved through modifications to the controller output. This level is less prevalent in motion planning as it necessitates a stringent constraint model to guide RL agent learning in highly variable environments. How can these safety assurances, therefore, be integrated into the RL agent's training process? The emergence of shielded RL presents a promising avenue for safe exploration. Shielded RL involves the integration of an external safety mechanism, or "shield," which oversees the learning process of an RL agent. The shield's role is to scrutinize the agent's actions and intervene when it detects potentially unsafe behaviors, thereby preventing the agent from executing actions that might lead to undesirable states. In such instances, the shield proposes an alternative action to the one suggested by the agent, ensuring the agent transitions to a safe state (see Figure 16). While predominantly assessed in discrete environments, exemplified by grid world scenarios [122], researchers have already started its application in continuous HRI environments for collision avoidance in conjunction with formal verification methods. For instance, Thumm and Althoff [123] proposed a high-frequency formal verification safety shield for HRI settings. This shield continuously sampled the full range of actions available to the agent and verified the safety of each action during its execution at a high frequency. As a result, the shield was able to stop the robot before any potential collision with a human, while still allowing the RL agent a high degree of freedom in its movements. Nonetheless, the success rate in certain experiment was below 70%, indicating that further research is needed to advance these hybrid technologies.

On the other hand, despite the robot undergoing training in a safe space, its deployment may expose it to states beyond the training space, triggered by unforeseen external perturbations. In such scenarios, methodologies like shielded RL may not adequately account for unexpected disturbances in dynamic environments, where the shielding mechanism's efficacy might be limited. Furthermore, their application in expansive, continuous action spaces remains a topic requiring thorough investigation. In this context,

Table 4: RL properties of each study in the motion planning (RL with other control techniques). The studies are classified according to their evaluation task and sorted by publication date.

Task		References	Year	Control method	RL taxonomy	Algorithm	Learning	Observation space							Action space				Reward	Simulation/ real-world
								Joint angle	Joint velocity	End-effector pose/pos	External force/ torque	Target pose/ pos	Obstacle pose	Others	Joint angle	Joint velocity	Pose/pos (cartesian)	Others		
APF	Via-point	[99]	2021	P	—	RL-APFM	S	X			X	X	X		X				D	B
		[100]	2021	P	VB	Q-learning	S							X				X	—	B
		[101]	2023	V	AC	DDPG TD3 SAC	S	X		X				X		X			D	B
	Pick and place	[102]	2023	P	AC	SAC	S	X		X		X	X		X				B	B
RRT	Via-point	[104]	2018	P	—	e-greedy FTS	S	X						X				X	—	S
		[107]	2021	P	AC	PPO	S	X		X	X			X	X				B	S
		[105]	2022	P	VB	Q-learning	S							X			X		D	S
		[106]	2022	P	AC	TAC	A							X				X	D	B
	Multiple	[108]	2022	P	AC	SAC	A	X	X	X		X			X				B	B
		[110]	2019	V	AC	TD3	S	X	X							X			D	B
		[109]	2020	P	AC	SAC	S	X	X	X	X			X	X				S	S
Multiple	Multiple	[111]	2021	P	VB	DQN	S	X				X	X	X				X	B	S

Note: Control methods: position control (P), velocity control (V), torque control (T), compliance/impedance control (C/I). 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).

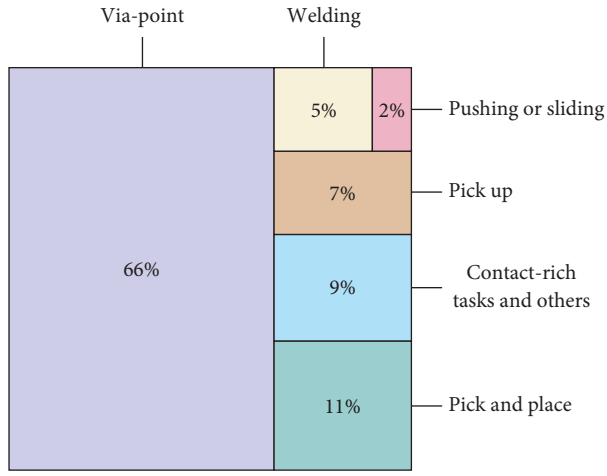

Figure 14: Motion planning tasks.

Table 5: Types of algorithms employed in reviewed studies.

Model-based and model-free	Model-free		
Actor–critic	Value-based	Policy-based	Actor–critic
1	23	0	50

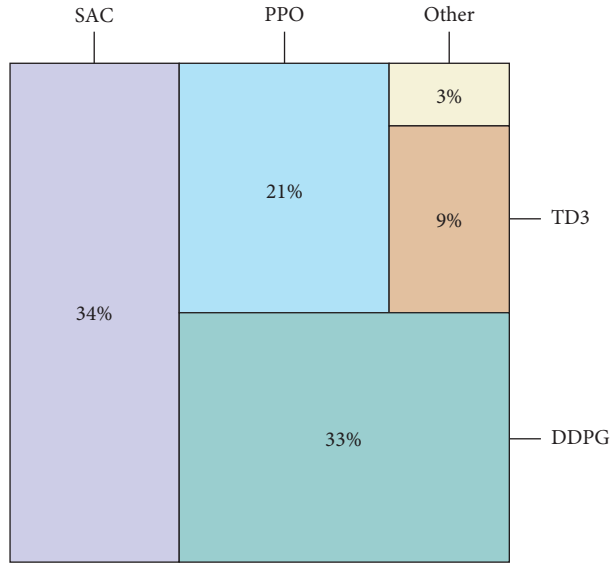

Figure 15: Actor–critic algorithms employed in motion planning tasks.

alternative research [124] proposes an approach akin to shielded RL, specifically designed for situations where the RL policy has already been deployed on a physical robot, self-stabilization [125]. This concept enables the system to autonomously return to a valid state, mitigating the absence of centralized control.

Moreover, a significant portion of the reviewed literature on motion planning focuses on HRI environments [43, 46]. In these contexts, ensuring human safety is paramount. Consequently, standards such as ISO 10218-1/2 [126, 127] and the technical report ISO/TS 15066 [128] delineate safety specifications aimed at mitigating potential hazards in HRI workspaces. To achieve this, four fundamental operational modes for using collaborative robots are defined, namely, safety-rated monitored stop, hand guiding, speed and separation monitoring, and power and force limiting. However, when an RL algorithm functions as a motion planner, computing collision-free trajectories, it is worth noting that stringent safety constraints may not always be guaranteed, leading to the possibility of occasional collisions. In scenarios where the promotion of workflows without safety-induced production halts is desired, the robot should be

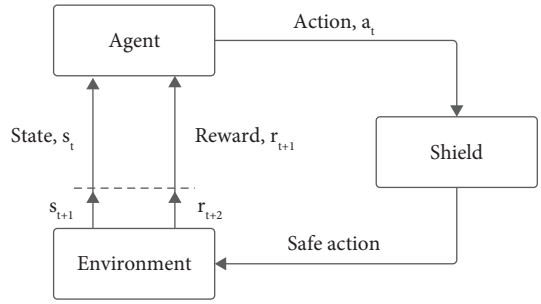

Figure 16: Shielded RL scheme.

equipped with a power and force-limiting module. This module serves to diminish risks following an inadvertent collision with a human collaborator.

Lastly, beyond ensuring safety, it is equally important to address the perceived safety by users interacting with the robot. Thus, it is not enough for robot trajectories to be merely safe, and they should compute smooth trajectories, allowing users to experience trust, comfort, and a sense of control [129].

6.4. Reward Shaping. The RL formulation represents goals through rewards. Rewards can be provided only at the end of the episode, resulting in sparse rewards [30, 79], or they can be issued at each time step t, known as dense rewards [41, 52]. Dense rewards offer intermediate feedback to the agent, indicating the quality of the action taken in the previous time step toward achieving the final goal. This intermediate signal is particularly crucial in problems with long experience streams, as it facilitates learning in exploration-intensive spaces. For instance, Peng et al. [64] designed three reward functions, namely, the posture reward, the stride reward, and the stage incentive mechanism. The posture reward was proposed to reduce the blindness of exploration and accelerate the learning process by modeling the distance and direction constraints of the robot. The stride reward aimed to enhance the stability of learning by considering both distance and movement distance of joint constraints. Lastly, the stage incentive mechanism was divided into the hard- and soft-stage incentive rewards that combined the two previous rewards. The hard-stage incentive reward segmented the robot's workspace into a fast-approach zone and a slow adjustable area, applying the posture and stride rewards, respectively. Meanwhile, the soft-stage incentive reward continuously adjusted the combination of both rewards, resulting in faster convergence, improved stability, and increased robustness.

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 [24, 59]. In [94], the authors utilized a dense reward function to guide the robot through disassembly and collision avoidance tasks. Upon successful completion of the extraction or upon reaching an undesired state, the RL agent was either granted a high reward or incurred a penalty, after which the training episode was reset to its initial state.

In motion planning, dense rewards are frequently linked to the Euclidean distance between the robot and the target point, exhibiting an inversely proportional relationship where a smaller Euclidean distance yields a higher positive reward, and an increased distance results in a corresponding penalty. The efficacy of dense rewards in enhancing sample efficiency has been substantiated in motion planning literature [62, 63]. Nevertheless, formulating these functions can be intricate, prompting some researchers to opt for sparse rewards during learning. While sparse rewards contribute to the development of more robust policies with enhanced generalizability, they carry the risk of some agents failing to reach the target point during training and getting trapped in local minima. Consequently, contemporary approaches increasingly integrate sparse rewards with methodologies such as HER [66, 67].

6.5. Sample Efficiency. RL is characterized by prolonged learning times, spanning from minutes to hours or even days, particularly for applications of moderate complexity. The agent's exploration predominantly influences the temporal aspect of learning during training, wherein actions and interactions with the environment contribute to the accumulation of knowledge.

In contrast to certain domains where leveraging prior knowledge, such as human demonstrations, can effectively reduce training times by bypassing the need to learn from scratch, this approach encounters challenges in motion planning. The difficulty arises from the geometric dependency of initial trajectories provided by demonstrations, posing limitations on generalization capability. This geometric dependency becomes particularly critical in scenarios involving moving obstacles, as it may lead to collisions, jeopardizing the safety of the robot and its surroundings.

Presently, addressing sample efficiency in motion planning involves diverse perspectives, including curriculum learning [60], reward shaping [64], or employing models [68]. However, an emerging research direction involves the integration of RL with more conventional AI techniques in motion planning [100, 110]. Combining APFs or RRTs with RL offers a synergistic approach that capitalizes on the strengths of both paradigms, enhancing sample efficiency in robotic motion planning. APFs provide a computationally efficient means of generating collision-free paths by modeling attractive and repulsive forces, guiding

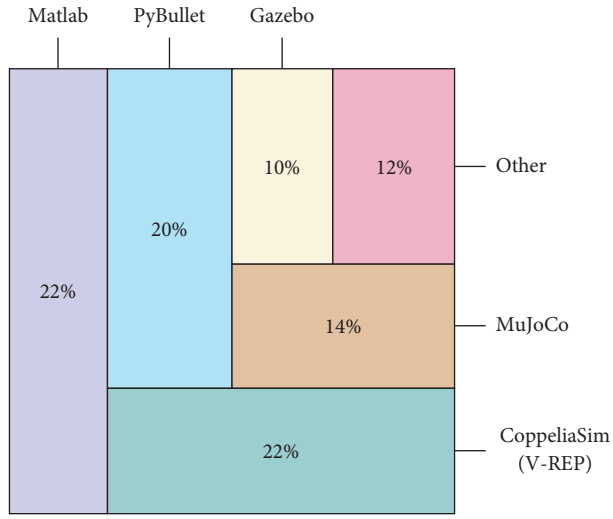

Figure 17: Simulators employed in motion planning tasks.

the robot toward its goal while avoiding obstacles. RRTs, on the other hand, excel at exploring the configuration space and efficiently sampling feasible trajectories. RL may subsequently be employed to refine the initial reference trajectories posited by APFs or RRTs, thereby mitigating the inherent limitations of these methodologies and generating policies that outperform the results that would be provided by these techniques individually. An example of the benefits arising from the combination of both techniques is demonstrated in [105]. In this study, the authors integrated RTT* with a value-based RL algorithm to address a collision-free trajectory planning problem involving multiple robot arms. Initially, each robot arm searched for a trajectory to the target posture using RTT*, followed by the application of Q-learning to prevent collisions between the arms. The results revealed a reduction in trajectory planning time by more than 20% compared to using RTT* alone. Moreover, this hybrid approach also outperformed the use of RL alone. In [107], the authors combined RTT* with an actor–critic RL algorithm, which not only accelerated convergence but also achieved a higher average reward compared to the RL algorithm alone.

Lastly, although none of the reviewed articles employed this method, interactive RL offers another promising avenue to enhance sample efficiency. This approach can involve either an artificial or human supervisor. An artificial supervisor, often another RL policy, creates the so-called teacher–student framework. In such cases, the teacher policy can provide corrective guidance to the student policy, narrowing the exploration space and accelerating learning. The student policy can potentially outperform the teacher, balancing guidance with independent exploration to maintain generalization [130]. Alternatively, if a human is included in the apprenticeship loop, they could apply their cognitive skills and life experience models to provide both corrective actions and evaluative feedback, effectively replacing rewards and penalties coming from the environment [131].

6.6. Generalization and Simulation-Reality Gap. The training process of robot learning necessitates a substantial number of training episodes and a thorough exploration of the environment. Insufficient exploration can result in suboptimal policies that introduce uncertainty when encountering unfamiliar states, jeopardizing both the robot and its surroundings. Furthermore, during the learning phase, the agent's exploratory behavior may exhibit potentially hazardous or unexpected robot actions, rendering direct robot training in real-world settings nearly impractical. Consequently, RL algorithms have conventionally been developed and evaluated within simulation environments. Indeed, unlike in other RL fields, all reviewed articles learn in simulation when it comes to motion planning. However, the inherent mismatches between simulation and reality have occasionally hindered the implementation of policies in real-world settings. Remarkably, nearly 60% of the reviewed papers did not transfer their control policy to a real robot.

Numerous investigations concentrating on enhancing sample efficiency [89, 97, 101] underscored the need to augment the generalization capabilities of their approaches in forthcoming research. Conversely, employing static obstacles or obstacles featuring random or linear motions within a singular plane might underrepresent hazardous scenarios, thereby constraining the agent's generalizability. Other approaches rely on specific movements replicating those of a human collaborator [51, 94]. However, focusing exclusively on realistic user movements may also underrepresent unforeseen human behaviors. Therefore, to what extent could combining random obstacles and realistic human movements improve collision avoidance and motion planning? Incorporating both types of obstacles during training could enhance the policy's generalization capabilities. Nevertheless, for the moment, introducing perturbations in the environment [110] or randomizing the domain [41, 90] represents potential strategies in addressing this limitation and fostering improved generalizability of the agent, consequently yielding more robust policies.

On the other hand, selecting an appropriate simulator for learning purposes significantly influences the successful deployment of policies onto physical robots. Aspects such as sensor support or physics engines should be considered for realistic simulations [132]. Although the specific simulators utilized by certain research studies are not always explicitly mentioned, it is noteworthy that Matlab[4] [35, 71], CoppeliaSim[5] [88, 111], and PyBullet[6] [57, 87] are the simulators most commonly used by researchers to train RL policies around motion planning (refer to Figure 17, 69 papers considered).

CoppeliaSim and PyBullet are widely utilized simulators in various robotics domains due to their onboard sensors like RGBD, LiDAR, and force, along with inverse kinematics capabilities. However, both exhibit certain limitations concerning realism, as indicated by [132]. In this regard, simulators such as NVIDIA Isaac Sim[7] are emerging as viable alternatives that aim to bridge this gap by offering improved capabilities. Furthermore, NVIDIA has also introduced its Isaac Gym simulator[8], specifically designed for training RL agents, which has gained initial attention from researchers [73]. Notably, Isaac Gym enables the simultaneous launch of multiple environments with minor configuration variations, facilitating the expedited acquisition of new experiences through exploration. Recent studies actively address the challenge of enhancing sample efficiency and generalizability, which were previously seen as potential bottlenecks. The ongoing advancements in this direction, particularly with the aid of the aforementioned simulator, have the potential to not only yield more robust RL-based solutions but also extend the applicability of this control technique to currently unexplored scenarios.

7. Conclusions

This paper presents a comprehensive review of recent advancements in RL within robotics, with a specific emphasis on motion planning. It provides an in-depth overview of current literature and a high-level analysis of prevailing trends and unresolved challenges in this domain.

From an applied perspective, this review identifies two primary trends in researchers' choices for evaluating their methodologies. Predominantly, studies conduct evaluations using via-point experiments, which involve computing feasible trajectories from an initial point to a target point, alongside actor–critic algorithms. Within this type of algorithms, the SAC, DDPG, and PPO agents are particularly prominent.

On a theoretical level, RL research in motion planning tasks prioritizes three main objectives: enhancing policy performance, improving sample efficiency during training, and expanding policy generalization capabilities. In terms of performance improvement, a major challenge lies in ensuring the safety of learned trajectories. Current studies largely implement Safety Levels 1 (soft safety constraints) and 2 (probabilistic safety constraints), which do not incorporate stringent safety constraints. Future research should therefore shift toward Safety Level 3 (hard safety constraints) approaches, which can provide safety certifications. For sample efficiency enhancement, a prevailing trend is to integrate RL with other more sample-efficient AI techniques, such as APFs or RRTs. Lastly, many studies underscore the need to improve generalization and, consequently, addressing the gap between simulation and real-world application. Notably, over half of the reviewed works do not test their approaches in real environments. Future efforts should prioritize the development of realistic simulation environments coupled with techniques like domain randomization, enabling policies to maintain both performance and safety when deployed in real-world scenarios despite minor environmental variations.

Nonetheless, amid the diverse lines of ongoing research, the main research paradigm emphasizes the definition of high-level goals for robots. This paradigm seeks to enable the achievement of these goals while upholding performance in real-world scenarios, irrespective of the dimensionality and exploration space in which the robot operates.

Data Availability Statement

The authors have nothing to report.

Conflicts of Interest

The authors declare no conflicts of interest.

Funding

This work has been partially funded by the Basque Government Department of Economic Development, Sustainability and Environment through the Bikaintek 2020 program and by the H2020-WIDESPREAD (GA 857061) "Networking for Research and Development of Human Interactive and Sensitive Robotics Taking Advantage of Additive Manufacturing—R2P2."

Endnotes

[1]http://www.scopus.com/.

[2]https://scholar.google.es/.

[3]https://www.webofscience.com/.

[4]https://www.mathworks.com/help/robotics/robot-simulation.html.

[5]https://www.coppeliarobotics.com/.

[6]https://pybullet.org/wordpress/.

[7]https://developer.nvidia.com/isaac-sim.

[8]https://developer.nvidia.com/isaac-gym.

References

[1] J. Wang, T. Zhang, N. Ma, et al., "A Survey of Learning-Based Robot Motion Planning," *IET Cyber-Systems and Robotics* 3, no. 4 (2021): 302–314, https://doi.org/10.1049/csy2.12020.

[2] C. Zhou, B. Huang, and P. Fränti, "A Review of Motion Planning Algorithms for Intelligent Robots," *Journal of Intelligent Manufacturing* 33, no. 2 (2022): 387–424, https://doi.org/10.1007/s10845-021-01867-z.

[3] L. Dong, Z. He, C. Song, and C. Sun, "A Review of Mobile Robot Motion Planning Methods: from Classical Motion Planning Workflows to Reinforcement Learning-Based Architectures," *Journal of Systems Engineering and Electronics* 34, no. 2 (2023): 439–459, https://doi.org/10.23919/jsee.2023.000051.

[4] H. C. Lin, C. Liu, Y. Fan, and M. Tomizuka, "Real-time Collision Avoidance Algorithm on Industrial Manipulators," *2017 IEEE Conference on Control Technology and Applications (CCTA)* 2017 (2017): 1294–1299, https://doi.org/10.1109/CCTA.2017.8062637.

[5] M. Bonilla, L. Pallottino, and A. Bicchi, "Noninteracting Constrained Motion Planning and Control for Robot Manipulators," *2017 IEEE International Conference on Robotics and Automation (ICRA)* (2017): 4038–4043, https://doi.org/10.1109/ICRA.2017.7989463.

[6] Y. Wang, X. Ye, Y. Yang, and W. Zhang, "Collision-free Trajectory Planning in Human-Robot Interaction through Hand Movement Prediction from Vision," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)* (2017), 305–310, https://doi.org/10.1109/HUMANOIDS.2017.8246890.

[7] R. Liu, F. Nageotte, P. Zanne, M. de Mathelin, and B. Dresp-Langley, "Deep Reinforcement Learning for the Control of Robotic Manipulation: A Focussed Mini-Review," *Robotics* 10, no. 1 (2021): 22–13, https://doi.org/10.3390/robotics10010022.

[8] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement Learning in Robotics: A Survey," *The International Journal of Robotics Research* 32, no. 11 (2013): 1238–1274, https://doi.org/10.1177/0278364913495721.

[9] H. Jiang, H. Wang, W.-Y. Yau, and K.-W. Wan, "A Brief Survey: Deep Reinforcement Learning in Mobile Robot Navigation," in *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)* (2020), https://doi.org/10.1109/ICIEA48937.2020.9248288.

[10] H. Sun, W. Zhang, R. Yu, and Y. Zhang, "Motion Planning for Mobile Robots - Focusing on Deep Reinforcement Learning: A Systematic Review," *IEEE Access* 9 (2021): 69061–69081, https://doi.org/10.1109/ACCESS.2021.3076530.

[11] K. Zhu and T. Zhang, "Deep Reinforcement Learning Based Mobile Robot Navigation: A Review," *Tsinghua Science and Technology* 26, no. 5 (2021): 674–691, https://doi.org/10.26599/TST.2021.9010012.

[12] K. Almazrouei, I. Kamel, and T. Rabie, "Dynamic Obstacle Avoidance and Path Planning through Reinforcement Learning," *Applied Sciences* 13, no. 14 (2023): 8174, https://doi.org/10.3390/app13148174.

[13] A. T. Azar, A. Koubaa, N. Ali Mohamed, et al., "Drone Deep Reinforcement Learning: A Review," *Electronics* 10, no. 9 (2021): 999–1030, https://doi.org/10.3390/electronics10090999.

[14] F. AlMahamid and K. Grolinger, "Autonomous Unmanned Aerial Vehicle Navigation Using Reinforcement Learning: A Systematic Review," *Engineering Applications of Artificial Intelligence* 115 (2022): 105321, https://doi.org/10.1016/j.engappai.2022.105321.

[15] M. G. Tamizi, M. Yaghoubi, and H. Najjaran, "A Review of Recent Trend in Motion Planning of Industrial Robots," *International Journal of Intelligent Robotics and Applications* 7, no. 2 (2023): 253–274, https://doi.org/10.1007/s41315-023-00274-2.

[16] R. S. Sutton and A. G. Barto, "Reinforcement Learning. An Introduction," *Second Edi* (2018).

[17] J. Pan and D. Manocha, "Efficient Configuration Space Construction and Optimization for Motion Planning," *Engineering* 1, no. 1 (2015): 046–057, https://doi.org/10.15302/J-ENG-2015009.

[18] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Path Planning and Trajectory Planning Algorithms: A General Overview," *Springer* 29 (2015).

[19] S. Levine and P. Abbeel, "Learning Neural Network Policies with Guided Policy Search under Unknown Dynamics," *Advances in Neural Information Processing Systems* 2 (2014): 1071–1079.

[20] F. Wirnshofer, P. S. Schmitt, P. Meister, G. Wichert, and W. Burgard, *State Estimation in Contact-Rich Manipulation* (2019).

[21] P. Khader, H. Yin, and D. Kragic, "Probabilistic Model Learning and Long-Term Prediction Forcontact-Rich Manipulation Tasks," *CoRR* 1 (2019).

[22] J. Weber and M. Schmidt, "An Improved Approach for Inverse Kinematics and Motion Planning of an Industrial Robot Manipulator with Reinforcement Learning," in *2021 Fifth IEEE International Conference on Robotic Computing (IRC)* (2021), 10–17, https://doi.org/10.1109/IRC52146.2021.00009.

[23] L. L. Liu, E. L. Chen, Z. G. Gao, and Y. Wang, "Research on Motion Planning of Seven Degree of Freedom Manipulator Based on DDPG," *Lecture Notes in Electrical Engineering* 484 (2019): 356–367, https://doi.org/10.1007/978-981-13-2375-1_44.

[24] E. W. Staley, K. D. Katyal, and P. Burlina, "DRL Based Intelligent Joint Manipulator and Viewing Camera Control for Reaching Tasks and Environments with Obstacles and Occluders," *2018 International Joint Conference on Neural Networks (IJCNN)* 2018 (2018): 1–7, https://doi.org/10.1109/IJCNN.2018.8489273.

[25] Q. Li, J. Nie, H. Wang, X. Lu, and S. Song, "Manipulator Motion Planning Based on Actor-Critic Reinforcement Learning," *2021 40th Chinese Control Conference (CCC)* 2021 (2021): 4248–4254, https://doi.org/10.23919/CCC52363.2021.9550010.

[26] G. Xinlan, L. Tao, and Z. Jian, "Trajectory Planning and Obstacle Avoidance Behavior of Manipulator Based on Q-Learning Algorithm," in *2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT)* (2023), 1235–1241, https://doi.org/10.1109/ICCECT57938.2023.10140814.

[27] F. M. Ribeiro and V. H. Pinto, "Reinforcement Learning Techniques Applied to the Motion Planning of a Robotic Manipulator," in *2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)* (2022), 173–178, https://doi.org/10.1109/ICARSC55462.2022.9784814.

[28] T. Shen, X. Liu, Y. Dong, and Y. Yuan, "Energy-Efficient Motion Planning and Control for Robotic Arms via Deep Reinforcement Learning," *2022 34th Chinese Control and Decision Conference (CCDC)* (2022): 5502–5507, https://doi.org/10.1109/CCDC55256.2022.10033563.

[29] C. Bartneck, D. Kulić, E. Croft, and S. Zoghbi, "Measurement Instruments for the Anthropomorphism, Animacy, Likeability, Perceived Intelligence, and Perceived Safety of Robots," *International Journal of Social Robotics* 1, no. 1 (2009): 71–81, https://doi.org/10.1007/s12369-008-0001-3.

[30] M. S. Kim, D. K. Han, J. H. Park, and J. S. Kim, "Motion Planning of Robot Manipulators for a Smoother Path Using a Twin Delayed Deep Deterministic Policy Gradient with Hindsight Experience Replay," *Applied Sciences* 10, no. 2 (2020): 575, https://doi.org/10.3390/app10020575.

[31] S. Zhang, Q. Xia, M. Chen, and S. Cheng, "Multi-Objective Optimal Trajectory Planning for Robotic Arms Using Deep

Reinforcement Learning," *Sensors* 23, no. 13 (2023): 5974–6015, https://doi.org/10.3390/s23135974.

[32] M. Andrychowicz, F. Wolski, A. Ray, et al., "Hindsight Experience Replay," *Advances in Neural Information Processing Systems* 2017 (2017): 5049–5059.

[33] A. Yang, Y. Chen, W. Naeem, M. Fei, and L. Chen, "Humanoid Motion Planning of Robotic Arm Based on Human Arm Action Feature and Reinforcement Learning," *Mechatronics* 78 (2021): 102630, https://doi.org/10.1016/j.mechatronics.2021.102630.

[34] X. Zhao, T. Fan, D. Wang, Z. Hu, T. Han, and J. Pan, "An Actor-Critic Approach for Legible Robot Motion Planner," *2020 IEEE International Conference on Robotics and Automation (ICRA)* (2020): 5949–5955, https://doi.org/10.1109/ICRA40945.2020.9197102.

[35] M. Ji, L. Zhang, and S. Wang, "A Path Planning Approach Based on Q-Learning for Robot Arm," *2019 3rd International Conference on Robotics and Automation Sciences (ICRAS)* 2019 (2019): 15–19, https://doi.org/10.1109/ICRAS.2019.8809005.

[36] Z. Huang, G. Chen, Y. Shen, Y. Liu, H. You, and T. Li, "LNOA: A Real-Time Obstacle Avoidance Motion Planning Method for Redundant Manipulator Based on Reinforcement Learning," in *2022 International Conference on Service Robotics (ICoSR)* (2022), 1–6, https://doi.org/10.1109/ICoSR57188.2022.00019.

[37] T. Jurgenson, O. Avner, E. Groshev, and A. Tamar, "Sub-goal Trees – A Framework for Goal-Based Reinforcement Learning," *Proc. 37th Int. Conf. Mach. Learn.* (2020): 5020–5030.

[38] A. Singh, M. Shakeel, V. Kalaichelvi, and R. Karthikeyan, "A Vision-Based Bio-Inspired Reinforcement Learning Algorithms for Manipulator Obstacle Avoidance," *Electronics* 11, no. 21 (2022): 3636, https://doi.org/10.3390/electronics11213636.

[39] J. C. Kiemel and T. Kroger, "Learning Collision-free and Torque-Limited Robot Trajectories Based on Alternative Safe Behaviors," *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)* 2022 (2022): 223–230, https://doi.org/10.1109/Humanoids53995.2022.10000077.

[40] X. Cheng and S. Liu, "Dynamic Obstacle Avoidance Algorithm for Robot Arm Based on Deep Reinforcement Learning," in *2022 IEEE 11th Data Driven Control and Learning Systems Conference (DDCLS)* (2022), 1136–1141, https://doi.org/10.1109/DDCLS55054.2022.9858561.

[41] X. Zhu, Y. Liang, H. Sun, X. Wang, and B. Ren, "Robot Obstacle Avoidance System Using Deep Reinforcement Learning," *Industrial Robot: The International Journal of Robotics Research and Application* 49, no. 2 (2022): 301–310, https://doi.org/10.1108/IR-06-2021-0127.

[42] L. Chen, Z. Jiang, L. Cheng, A. C. Knoll, and M. Zhou, "Deep Reinforcement Learning Based Trajectory Planning under Uncertain Constraints," *Frontiers in Neurorobotics* 16 (2022): 883562–883610, https://doi.org/10.3389/fnbot.2022.883562.

[43] M. El-Shamouty, X. Wu, S. Yang, M. Albus, and M. F. Huber, "Towards Safe Human-Robot Collaboration Using Deep Reinforcement Learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (2020), 4899–4905, https://doi.org/10.1109/ICRA40945.2020.9196924.

[44] B. Sangiovanni, A. Rendiniello, G. P. Incremona, A. Ferrara, and M. Piastra, "Deep Reinforcement Learning for Collision Avoidance of Robotic Manipulators," in *2018 European Control Conference (ECC)* (2018), 2063–2068, https://doi.org/10.23919/ECC.2018.8550363.

[45] B. Sangiovanni, G. P. Incremona, M. Piastra, and A. Ferrara, "Self-Configuring Robot Path Planning with Obstacle Avoidance via Deep Reinforcement Learning," *IEEE Control Systems Letters* 5, no. 2 (2021): 397–402, https://doi.org/10.1109/LCSYS.2020.3002852.

[46] B. Xiong, Q. Liu, B. Yao, Z. Liu, and Z. Zhou, "Deep Reinforcement Learning-Based Safe Interaction for Industrial Human-Robot Collaboration," in *49th International Conference on Computers & Industrial Engineering* (2019), 1–13.

[47] Q. Liu, Z. Liu, B. Xiong, W. Xu, and Y. Liu, "Deep Reinforcement Learning-Based Safe Interaction for Industrial Human-Robot Collaboration Using Intrinsic Reward Function," *Advanced Engineering Informatics* 49 (2021): 101360, https://doi.org/10.1016/j.aei.2021.101360.

[48] M. Amir Salmaninejad, S. Zilles, and R. V. Mayorga, "Motion Path Planning of Two Robot Arms in a Common Workspace," *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* 2020 (2020): 45–51, https://doi.org/10.1109/SMC42975.2020.9283018.

[49] K. W. Park, M. S. Kim, J. S. Kim, and J. H. Park, "Path Planning for Multi-Arm Manipulators Using Soft Actor-Critic Algorithm with Position Prediction of Moving Obstacles via LSTM," *Applied Sciences* 12, no. 19 (2022): 9837, https://doi.org/10.3390/app12199837.

[50] C. C. Wong, S. Y. Chien, H. M. Feng, and H. Aoyama, "Motion Planning for Dual-Arm Robot Based on Soft Actor-Critic," *IEEE Access* 9 (2021): 26871–26885, https://doi.org/10.1109/ACCESS.2021.3056903.

[51] X. Zhao, T. Fan, Y. Li, Y. Zheng, and J. Pan, "An Efficient and Responsive Robot Motion Controller for Safe Human-Robot Collaboration," *IEEE Robotics and Automation Letters* 6, no. 3 (2021): 6068–6075, https://doi.org/10.1109/LRA.2021.3088091.

[52] P. Chen, J. Pei, W. Lu, and M. Li, "A Deep Reinforcement Learning Based Method for Real-Time Path Planning and Dynamic Obstacle Avoidance," *Neurocomputing* 497 (2022): 64–75, https://doi.org/10.1016/j.neucom.2022.05.006.

[53] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized Experience Replay" (2015).

[54] Y. L. Kim, K. H. Ahn, and J. B. Song, "Reinforcement Learning Based on Movement Primitives for Contact Tasks," *Robotics and Computer-Integrated Manufacturing* 62 (2020): 101863, https://doi.org/10.1016/j.rcim.2019.101863.

[55] Y. Wang, C. C. Beltran-Hernandez, W. Wan, and K. Harada, "Hybrid Trajectory and Force Learning of Complex Assembly Tasks: A Combined Learning Framework," *IEEE Access* 9 (2021): 60175–60186, https://doi.org/10.1109/access.2021.3073711.

[56] Y. Shen, Q. Jia, Z. Huang, R. Wang, and G. Chen, "Guided Deep Reinforcement Learning for Path Planning of Robotic Manipulators," in *Cognitive Systems and Signal Processing* (2021), 301–308.

[57] I. Akinola, Z. Wang, and P. Allen, "CLAMGen: Closed-Loop Arm Motion Generation via Multi-View Vision-Based RL," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2021), 2376–2382, https://doi.org/10.1109/IROS51168.2021.9636369.

[58] A. Abdi, D. Adhikari, and J. H. Park, "A Novel Hybrid Path Planning Method Based on Q-Learning and Neural Network for Robot Arm," *Applied Sciences* 11, no. 15 (2021): 6770, https://doi.org/10.3390/app11156770.

[59] A. Abdi, M. H. Ranjbar, and J. H. Park, "Computer Vision-Based Path Planning for Robot Arms in Three-Dimensional Workspaces Using Q-Learning and Neural Networks," *Sensors* 22, no. 5 (2022): 1697, https://doi.org/10.3390/s22051697.

[60] D. Zhou, R. Jia, and H. Yao, "Robotic Arm Motion Planning Based on Curriculum Reinforcement Learning," in *2021 6th*

International Conference on Control and Robotics Engineering (ICCRE) (2021), 44–49, https://doi.org/10.1109/ICCRE51898.2021.9435700.

[61] E. Ratner, D. Hadfield-Menell, and A. D. Dragan, "Simplifying Reward Design through Divide-and-Conquer," *Robotics: Science and Systems XIV* (2018): https://doi.org/10.15607/RSS.2018.XIV.048.

[62] J. Xie, Z. Shao, Y. Li, Y. Guan, and J. Tan, "Deep Reinforcement Learning with Optimized Reward Functions for Robotic Trajectory Planning," *IEEE Access* 7 (2019): 105669–105679, https://doi.org/10.1109/ACCESS.2019.2932257.

[63] S. Yang and Q. Wang, "Robotic Arm Motion Planning with Autonomous Obstacle Avoidance Based on Deep Reinforcement Learning," *2022 41st Chinese Control Conference (CCC)* 2022 (2022): 3692–3697, https://doi.org/10.23919/CCC55666.2022.9902722.

[64] G. Peng, J. Yang, X. Li, and M. O. Khyam, "Deep Reinforcement Learning with a Stage Incentive Mechanism of Dense Reward for Robotic Trajectory Planning," *IEEE Transactions on Systems, Man, and Cybernetics* 53, no. 6 (2023): 3566–3573, https://doi.org/10.1109/TSMC.2022.3228901.

[65] T. Bhuiyan, L. Kastner, Y. Hu, B. Kutschank, and J. Lambrecht, "Deep-Reinforcement-Learning-Based Path Planning for Industrial Robots Using Distance Sensors as Observation," in *2023 8th International Conference on Control and Robotics Engineering (ICCRE)* (2023), 204–210, https://doi.org/10.1109/ICCRE57112.2023.10155608.

[66] E. Prianto, M. Kim, J. H. Park, J. H. Bae, and J. S. Kim, "Path Planning for Multi-Arm Manipulators Using Deep Reinforcement Learning: Soft Actor–Critic with Hindsight Experience Replay," *Sensors* 20, no. 20 (2020): 5911–5923, https://doi.org/10.3390/s20205911.

[67] E. Prianto, J. H. Park, J. H. Bae, and J. S. Kim, "Deep Reinforcement Learning-Based Path Planning for Multi-Arm Manipulators with Periodically Moving Obstacles," *Applied Sciences* 11, no. 6 (2021): 2587, https://doi.org/10.3390/app11062587.

[68] D. Qiao, Z. Zhong, H. Zhang, and Y. Zhao, "Trajectory Planning of Manipulator Based on DQN Algorithm Guided by MPC Sampling," in *2021 3rd International Symposium on Robotics & Intelligent Manufacturing Technology (ISRIMT)* (2021), 319–323, https://doi.org/10.1109/ISRIMT53730.2021.9597010.

[69] A. Baselizadeh, W. Khaksar, and J. Torresen, "Motion Planning and Obstacle Avoidance for Robot Manipulators Using Model Predictive Control-Based Reinforcement Learning," *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* 2022 (2022): 1584–1591, https://doi.org/10.1109/SMC53654.2022.9945504.

[70] C. E. Garcia, D. M. Prett, and M. Morari, "Model Predictive Control: Theory and Practice - A Survey," *Automatica* 25, no. 3 (1989): 335–348, https://doi.org/10.1016/0005-1098(89)90002-2.

[71] T. Ku, J. Li, J. Liu, Y. Lin, and X. Liu, "A Motion Planning Algorithm for Live Working Manipulator Integrating PSO and Reinforcement Learning Driven by Model and Data," *Frontiers in Energy Research* 10 (2022): 1–11, https://doi.org/10.3389/fenrg.2022.957869.

[72] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks* 4 (1995): 1942–1948, https://doi.org/10.1109/icnn.1995.488968.

[73] L. Vahrens, D. D. Álvarez, U. Berger, and S. Bøgh, "Learning Task-independent Joint Control for Robotic Manipulators with Reinforcement Learning and Curriculum Learning," in *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA 2022)* (2022), 1250–1257, https://doi.org/10.1109/ICMLA55696.2022.00201.

[74] H. C. Yao, C. K. Ho, and C. T. King, "Learning Diverse and Efficient Goal-Reaching Policies for Robot Motion Planning," *2023 IEEE 32nd International Symposium on Industrial Electronics (ISIE)* 2023 (2023): 1–7, https://doi.org/10.1109/ISIE51358.2023.10227978.

[75] T. Zhang, K. Zhang, J. Lin, W. Y. G. Louie, and H. Huang, "Sim2real Learning of Obstacle Avoidance for Robotic Manipulators in Uncertain Environments," *IEEE Robotics and Automation Letters* 7, no. 1 (2022): 65–72, https://doi.org/10.1109/LRA.2021.3116700.

[76] K. Kamali, I. A. Bonev, and C. Desrosiers, "Real-time Motion Planning for Robotic Teleoperation Using Dynamic-Goal Deep Reinforcement Learning," in *2020 17th Conference on Computer and Robot Vision (CRV)* (2020), 182–189, https://doi.org/10.1109/CRV50864.2020.00032.

[77] W. Liu, H. Niu, M. N. Mahyuddin, G. Herrmann, and J. Carrasco, "A Model-free Deep Reinforcement Learning Approach for Robotic Manipulators Path Planning," *2021 21st International Conference on Control, Automation and Systems (ICCAS)* 2021 (2021): 512–517, https://doi.org/10.23919/ICCAS52745.2021.9649802.

[78] P. S. Schmitt, F. Wirnshofer, K. M. Wurm, G. V. Wichert, and W. Burgard, "Planning Reactive Manipulation in Dynamic Environments," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2019): 136–143, https://doi.org/10.1109/IROS40897.2019.8968452.

[79] G. Golluccio, D. Di Vito, A. Marino, A. Bria, and G. Antonelli, "Task-motion Planning via Tree-Based Q-Learning Approach for Robotic Object Displacement in Cluttered Spaces," in *Proceedings of the 18th International Conference on Informatics in Control, Automation and Robotics* (2021), 130–137, https://doi.org/10.5220/0010542601300137.

[80] S. Wen, J. Chen, S. Wang, H. Zhang, and X. Hu, "Path Planning of Humanoid Arm Based on Deep Deterministic Policy Gradient," *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (2018): 1755–1760, https://doi.org/10.1109/ROBIO.2018.8665248.

[81] X. Wu, L. Yi, M. Klar, M. Hussong, M. Glatt, and J. C. Aurich, "Intelligent Robotic Arm Path Planning (IRAP2) Framework to Improve Work Safety in Human-Robot Collaboration (HRC) Workspace Using Deep Deterministic Policy Gradient (DDPG) Algorithm," *Lecture Notes in Mechanical Engineering* (2023): 179–187, https://doi.org/10.1007/978-3-031-18326-3_18.

[82] J. Heaton and S. Givigi, "A Deep Reinforcement Learning Solution for the Low Level Motion Control of a Robot Manipulator System," in *2023 IEEE International Systems Conference (SysCon)* (2023), https://doi.org/10.1109/SysCon53073.2023.10131174.

[83] S. Xie, L. Gong, Z. Chen, and B. Chen, "Simulation of Real-Time Collision-free Path Planning Method with Deep Policy Network in Human-Robot Interaction Scenario," in *2023 International Conference on Advanced Robotics and Mechatronics (ICARM)* (2023), 360–365, https://doi.org/10.1109/ICARM58088.2023.10218854.

[84] S. Kreiss, L. Bertoni, and A. Alahi, "PifPaf: Composite Fields for Human Pose Estimation," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2019 (2019): 11969–11978, https://doi.org/10.1109/CVPR.2019.01225.

[85] G. Nicola and S. Ghidoni, "Deep Reinforcement Learning for Motion Planning in Human Robot Cooperative Scenarios," in *2021 26th IEEE International Conference on Emerging*

Technologies and Factory Automation (ETFA) (2021), https://doi.org/10.1109/ETFA45728.2021.9613505.

[86] Y. Xu, T. Wang, C. Chen, and B. Hu, "Learning Collision-Freed Trajectory of Welding Manipulator Based on Safe Reinforcement Learning," *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)* 2022 (2022): 836–841, https://doi.org/10.1109/CASE49997.2022.9926592.

[87] B. Hu, T. Wang, C. Chen, Y. Xu, and L. Cheng, "Collision-free Path Planning for Welding Manipulator via Deep Reinforcement Learning," *2022 27th International Conference on Automation and Computing (ICAC)* 2022 (2022): 1–6, https://doi.org/10.1109/ICAC55051.2022.9911177.

[88] J. Zhong, T. Wang, and L. Cheng, "Collision-free Path Planning for Welding Manipulator via Hybrid Algorithm of Deep Reinforcement Learning and Inverse Kinematics," *Complex & Intelligent Systems* 8, no. 3 (2022): 1899–1912, https://doi.org/10.1007/s40747-021-00366-1.

[89] A. Avaei, L. van der Spaa, L. Peternel, and J. Kober, "An Incremental Inverse Reinforcement Learning Approach for Motion Planning with Separated Path and Velocity Preferences," *Robotics* 12, no. 2 (2023): 61–22, https://doi.org/10.3390/robotics12020061.

[90] R. Strudel, A. Pashevich, I. Kalevatykh, I. Laptev, J. Sivic, and C. Schmid, "Learning to Combine Primitive Skills: A Step towards Versatile Robotic Manipulation," in *Proceedings of International Conference on Robotics and Automation* (2020), 4637–4643, https://doi.org/10.1109/ICRA40945.2020.9196619.

[91] Q. Hou, C. Gu, X. Wang, Y. Zhang, and P. Zhao, "Dynamic Trajectory Planning of a 7-DOF Surgical Robot Based on HER-DDPG Algorithm," *ASME International Mechanical Engineering Congress and Exposition* 85598 (2021): V005T05A064.

[92] S. Veeramani and S. Muthuswamy, "Hybrid Type Multi-Robot Path Planning of a Serial Manipulator and SwarmItFIX Robots in Sheet Metal Milling Process," *Complex & Intelligent Systems* 8, no. 4 (2022): 2937–2954, https://doi.org/10.1007/s40747-021-00499-3.

[93] S. Veeramani, S. Muthuswamy, and R. Setchi, "Coordination and Path Planning of a Heterogeneous Multi-Robot System for Sheet Metal Drilling," *Procedia Computer Science* 207, no. Kes (2022): 2335–2344, https://doi.org/10.1016/j.procs.2022.09.292.

[94] Í. Elguea-Aguinaco, A. Serrano-Muñoz, D. Chrysostomou, I. Inziarte-Hidalgo, S. Bøgh, and N. Arana-Arexolaleiba, "Goal-Conditioned Reinforcement Learning within a Human-Robot Disassembly Environment," *Applied Sciences* 12, no. 22 (2022): 11610, https://doi.org/10.3390/app122211610.

[95] M. Al-Gabalawy, "Path Planning of Robotic Arm Based on Deep Reinforcement Learning Algorithm," *Advanced Control for Applications* 4, no. 1 (2022): e79, https://doi.org/10.1002/adc2.79.

[96] L. Liu, Q. Liu, Y. Song, B. Pang, X. Yuan, and Q. Xu, "A Collaborative Control Method of Dual-Arm Robots Based on Deep Reinforcement Learning," *Applied Sciences* 11, no. 4 (2021): 1816–16, https://doi.org/10.3390/app11041816.

[97] X. Hua, G. Wang, J. Xu, and K. Chen, "Reinforcement Learning-Based Collision-free Path Planner for Redundant Robot in Narrow Duct," *Journal of Intelligent Manufacturing* 32, no. 2 (2021): 471–482, https://doi.org/10.1007/s10845-020-01582-1.

[98] C. W. Warren, "Global Path Planning Using Artificial Potential Fields," in *IEEE International Conference on Robotics and Automation* (1989), 316–317.

[99] H. Li, D. Gong, and J. Yu, "An Obstacles Avoidance Method for Serial Manipulator Based on Reinforcement Learning and Artificial Potential Field," *International Journal of Intelligent Robotics and Applications* 5, no. 2 (2021): 186–202, https://doi.org/10.1007/s41315-021-00172-5.

[100] Z. Fang and X. Liang, "Intelligent Obstacle Avoidance Path Planning Method for Picking Manipulator Combined with Artificial Potential Field Method," *Industrial Robot: The International Journal of Robotics Research and Application* 49, no. 5 (2022): 835–850, https://doi.org/10.1108/IR-09-2021-0194.

[101] L. Zheng, Y. H. Wang, R. Yang, S. Wu, R. Guo, and E. Dong, "An Efficiently Convergent Deep Reinforcement Learning-Based Trajectory Planning Method for Manipulators in Dynamic Environments," *Journal of Intelligent and Robotic Systems* 107, no. 4 (2023): 50, https://doi.org/10.1007/s10846-023-01822-5.

[102] C. Bai, J. Zhang, J. Guo, and C. P. Yue, "Adaptive Hybrid Optimization Learning-Based Accurate Motion Planning of Multi-Joint Arm," *IEEE Transactions on Neural Networks and Learning Systems* 34, no. 9 (2023): 5440–5451, https://doi.org/10.1109/TNNLS.2023.3262109.

[103] S. M. LaValle and J. J. Kuffner, "Rapidly-Exploring Random Trees: Progress and Prospects," *Algorithmic and Computational Robotics* (2001): 303–307.

[104] W. Jing, C. F. Goh, M Rajaraman, et al., "A Computational Framework for Automatic Online Path Generation of Robotic Inspection Tasks via Coverage Planning and Reinforcement Learning," *IEEE Access* 6 (2018): 54854–54864, https://doi.org/10.1109/ACCESS.2018.2872693.

[105] T. Kawabe and T. Nishi, "A Flexible Collision-free Trajectory Planning for Multiple Robot Arms by Combining Q-Learning and RRT," *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)* 2022 (2022): 2363–2368, https://doi.org/10.1109/CASE49997.2022.9926603.

[106] X. Li, H. Liu, and M. Dong, "A General Framework of Motion Planning for Redundant Robot Manipulator Based on Deep Reinforcement Learning," *IEEE Transactions on Industrial Informatics* 18, no. 8 (2022): 5253–5263, https://doi.org/10.1109/TII.2021.3125447.

[107] D. Zhou, R. Jia, H. Yao, and M. Xie, "Robotic Arm Motion Planning Based on Residual Reinforcement Learning," in *2021 13th International Conference on Computer and Automation Engineering (ICCAE)* (2021), 89–94, https://doi.org/10.1109/ICCAE51876.2021.9426160.

[108] J. Zhang, J. Guo, and C. Bai, "Heuristic Reward Function for Reinforcement Learning Based Manipulator Motion Planning," *2022 IEEE International Conference on Unmanned Systems (ICUS)* (2022): 1545–1550, https://doi.org/10.1109/ICUS55513.2022.9986816.

[109] J. Yamada, Y. Lee, G. Salhotra, et al., "Motion Planner Augmented Reinforcement Learning for Robot Manipulation in Obstructed Environments," (2020), 1–15, http://arxiv.org/abs/2010.11940.

[110] K. Ota, D. K. Jha, T. Oiki, et al., "Trajectory Optimization for Unknown Constrained Systems Using Reinforcement Learning," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2019), 3487–3494, https://doi.org/10.1109/IROS40897.2019.8968010.

[111] N. Sacchi, B. Sangiovanni, G. P. Incremona, and A. Ferrara, "Scenario-Based Collision Avoidance Control with Deep Q-Networks for Industrial Robot Manipulators," *2021 60th IEEE Conference on Decision and Control (CDC)* 2021 (2021): 4388–4393, https://doi.org/10.1109/CDC45484.2021.9683056.

[112] Í. Elguea-Aguinaco, A. Serrano-Muñoz, D. Chrysostomou, I. Inziarte-Hidalgo, S. Bøgh, and N. Arana-Arexolaleiba, "A Review on Reinforcement Learning for Contact-Rich Robotic Manipulation Tasks," *Robotics and Computer-Integrated Manufacturing* 81 (2023): 102517, https://doi.org/10.1016/j.rcim.2022.102517.

[113] S. Nasiriany, S. Lin, and S. Levine, *Planning with Goal-Conditioned Policies* (NeurIPS, 2019).

[114] S. Pateria, B. Subagdja, Ah Tan, and C. Quek, "Hierarchical Reinforcement Learning: A Comprehensive Survey," *ACM Computing Surveys* 54, no. 5 (2021): 1–35, https://doi.org/10.1145/3453160.

[115] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," *35th International Conference on Machine Learning* 5 (2018): 2976–2989.

[116] T. P. Lillicrap, J. J. Hunt, A. Pritzel, et al., "Continuous Control with Deep Reinforcement Learning," (2016), https://arxiv.org/abs/1509.02971.

[117] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," (2017), 1–12, http://arxiv.org/abs/1707.06347.

[118] A. Ray, J. Achiam, and D. Amodei, "Benchmarking Safe Exploration in Deep Reinforcement Learning" (2019).

[119] L. Brunke, M. Greeff, A. W. Hall, et al., "Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning," *Annu. Rev. Control. Robot. Auton. Syst.* 5, no. 1 (2022): 411–444, https://doi.org/10.1146/annurev-control-042920-020211.

[120] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A Lyapunov-Based Approach to Safe Reinforcement Learning," *Advances in Neural Information Processing Systems* 2018 (2018): 8092–8101.

[121] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa, "Safe Exploration in Continuous Action Spaces" (2018).

[122] H. Odriozola-Olalde, N. Arana-Arexolaleiba, M. Zamalloa, J. Perez-Cerrolaza, and J. Arozamena-Rodríguez, "Fear Field: Adaptive Constraints for Safe Environment Transitions in Shielded Reinforcement Learning," *CEUR Workshop Proceedings* 3505 (2023): 1–9.

[123] J. Thumm and M. Althoff, "Provably Safe Deep Reinforcement Learning for Robotic Manipulation in Human Environments," *2022 International Conference on Robotics and Automation (ICRA)* (2022): 6344–6350, https://doi.org/10.1109/ICRA46639.2022.9811698.

[124] N. K. Sreenivas and S. Rao, "Safe Deployment of a Reinforcement Learning Robot Using Self Stabilization," *Intelligent Systems with Applications* 16 (2022): 200105, https://doi.org/10.1016/j.iswa.2022.200105.

[125] E. W. Dijkstra, "Self-stabilizing Systems in Spite of Distributed Control," *Edsger Wybe Dijkstra* (2022): 333–338, https://doi.org/10.1145/3544585.3544606.

[126] International Organization of Standardization, "ISO 10218-1. Robots and Robotic Devices - Safety Requirements for Industrial Robots," *Part 1: Robot* (2014): 56.

[127] International Organization of Standardization, "ISO 10218-2. Robots and Robotic Devices - Safety Requirements for Industrial Robots," *Part 2: Robot systems and integration* (2016): 86.

[128] International Organization of Standardization, "ISO/TS 15066," *Robots and Robotic Devices - Collaborative Robots* (2016): 40.

[129] N. Akalin, A. Kristoffersson, and A. Loutfi, "Do You Feel Safe with Your Robot? Factors Influencing Perceived Safety in Human-Robot Interaction Based on Subjective and Objective Measures," *International Journal of Human-Computer Studies* 158 (2022): 102744, https://doi.org/10.1016/j.ijhcs.2021.102744.

[130] H. B. Suay and S. Chernova, "Effect of Human Guidance and State Space Size on Interactive Reinforcement Learning," *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication: Foreword* (2011): 1–6, https://doi.org/10.1109/ROMAN.2011.6005223.

[131] J. Lin, Z. Ma, R. Gomez, K. Nakamura, B. He, and G. Li, "A Review on Interactive Reinforcement Learning from Human Social Feedback," *IEEE Access* 8 (2020): 120757–120765, https://doi.org/10.1109/ACCESS.2020.3006254.

[132] J. Collins, S. Chand, A. Vanderkop, and D. Howard, "A Review of Physics Simulators for Robotic Applications," *IEEE Access* 9 (2021): 51416–51431, https://doi.org/10.1109/ACCESS.2021.3068769.