

Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes

Chen Tang^{1,*}, Ben Abbatematteo^{1,*},
Jiaheng Hu^{1,*}, Rohan Chandra²,
Roberto Martín-Martín¹, Peter Stone^{1,3}

¹Department of Computer Science, The University of Texas at Austin, Austin, Texas 78712, United States; email: chen.tang@utexas.edu, abba@cs.utexas.edu, jiahengh@utexas.edu, robertomm@cs.utexas.edu, pstone@utexas.edu

²Department of Computer Science, The University of Virginia, Charlottesville, Virginia 22904, United States; email: rohanchandra@virginia.edu

³Sony AI

*Equal Contribution

Xxxx. Xxx. Xxx. Xxx. YYYY. AA:1–42

[https://doi.org/10.1146/\(\(please add article doi\)\)](https://doi.org/10.1146/((please add article doi)))

Copyright © YYYY by the author(s).
All rights reserved

Keywords

robotics, reinforcement learning, deep learning, learning for control, real-world applications

Abstract

Reinforcement learning (RL), particularly its combination with deep neural networks referred to as deep RL (DRL), has shown tremendous promise across a wide range of applications, suggesting its potential for enabling the development of sophisticated robotic behaviors. Robotics problems, however, pose fundamental difficulties for the application of RL, stemming from the complexity and cost of interacting with the physical world. This article provides a modern survey of DRL for robotics, with a particular focus on evaluating the real-world successes achieved with DRL in realizing several key robotic competencies. Our analysis aims to identify the key factors underlying those exciting successes, reveal underexplored areas, and provide an overall characterization of the status of DRL in robotics. We highlight several important avenues for future work, emphasizing the need for stable and sample-efficient real-world RL paradigms, holistic approaches for discovering and integrating various competencies to tackle complex long-horizon, open-world tasks, and principled development and evaluation procedures. This survey is designed to offer insights for both RL practitioners and roboticists toward harnessing RL's power to create generally capable real-world robotic systems.

1. Introduction

Reinforcement learning (RL) (1) refers to a class of decision-making problems in which an agent must learn through trial-and-error to act in such a way that maximizes its accumulated *return*, as encoded by a scalar reward function that maps the agent’s states and actions to immediate rewards. RL algorithms, particularly their combination with deep neural networks referred to as deep RL (DRL) (2), have shown remarkable capabilities in solving complex decision-making problems even with high-dimensional observations in domains such as board games (3), video games (4), healthcare (5), and recommendation systems (6).

These successes underscore the potential of DRL for controlling robotic systems with high-dimensional state or observation space and highly nonlinear dynamics to perform challenging tasks that conventional decision-making, planning, and control approaches (e.g., classical control, optimal control, sampling-based planning) cannot handle effectively. Yet, the most notable milestones of DRL so far have been achieved in simulation or game environments, where RL agents can learn from extensive experience. In contrast, robots need to complete tasks in the *physical world*, which presents additional challenges. It is often inefficient and/or unsafe for the RL agents to collect trial-and-error samples directly in the physical world, and it is usually impossible to create an exact replica of the complex real world in simulation. These challenges notwithstanding, recent advances have enabled DRL to succeed at some real-world robotic tasks. For instance, DRL has enabled champion-level drone racing (7) and versatile quadruped locomotion control integrated into production-level quadruped systems (e.g., ANYbotics¹, Swiss-Mile², and Boston Dynamics³). However, *the maturity of state-of-the-art DRL solutions varies significantly across different robotic applications*. In some domains, such as urban autonomous driving, DRL-based solutions remain limited to simulation or strictly confined field tests (8).

This survey aims to comprehensively evaluate the current progress of DRL in real-world robotic applications, identifying key factors behind the most exciting successes and open challenges that remain in less mature areas. Specifically, we assess the maturity of DRL for a variety of problem domains and contrast the DRL literature across domains to pinpoint broadly applicable techniques, under-explored areas, and common open challenges that need to be addressed to advance DRL’s applications in robotics. We aim for this survey to provide researchers and practitioners with a thorough understanding of the status of DRL in robotics, offering valuable insights to guide future research and facilitate broadly deployable DRL solutions for real-world robotic tasks.

2. Why Another Survey on RL for Robotics?

Although some previous articles have surveyed RL for robotics, we make three contributions that provide unique perspectives on the literature and fill gaps in knowledge. First, we focus on work that has demonstrated at least *some degree of real-world success*, aiming to assess the current state and open challenges of DRL for real-world robotic applications. Most existing surveys on RL for robotics do not explicitly address this topic, e.g., Dulac-Arnold et al. (9) discuss the general challenges of real-world RL not specific to robotics, and Ibarz et al. (10) list open challenges of DRL unique to real-world robotics settings but based on

¹<https://www.anybotics.com/news/superior-robot-mobility-where-ai-meets-the-real-world/>

²<https://www.swiss-mile.com/>

³<https://bostondynamics.com/blog/startng-on-the-right-foot-with-reinforcement-learning/>

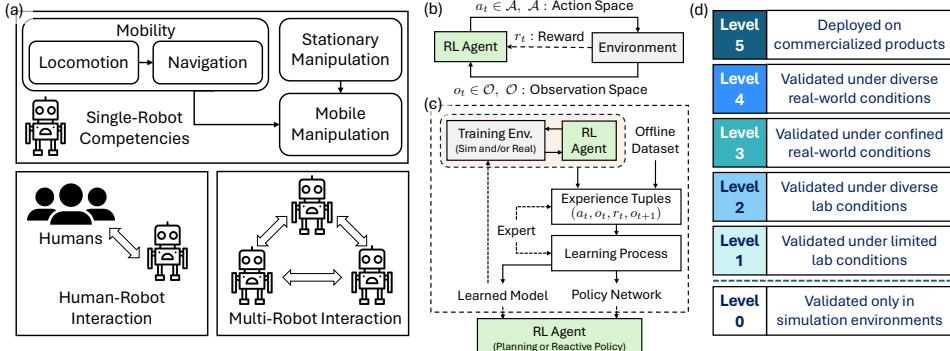


Figure 1: The four aspects of our taxonomy: (a) Robot competencies learned with DRL; (b) Problem formulation; (c) Solution approach; and (d) Levels of real-world success.

case studies drawn only from their own research. In contrast, our discussion is grounded in a comprehensive assessment of the real-world successes achieved by DRL in robotics, with one aspect of our evaluation being the level of real-world deployment (see Sec. 3.4).

Second, we present a *novel* and *comprehensive* taxonomy that categorizes DRL solutions along multiple axes: robot competencies learned with DRL, problem formulation, solution approach, and level of real-world success. Prior surveys on RL for robotics and broader robot learning have often focused on specific tasks (11, 12) or on particular techniques (13, 14). By contrast, our taxonomy allows us to survey the complete landscape of DRL solutions that are effective in robotics application domains, in addition to reviewing the literature of each application domain separately. Within this framework, we compare and contrast solutions and identify *common patterns*, *broadly applicable approaches*, *under-explored areas*, and *open challenges* for realizing successful robotic systems.

Third, while some past surveys have shared our motivation to provide a broad analysis of the field, the fast and impressive pace of DRL progress has created the need for a renewed analysis of the field, its successes, and limitations. The seminal survey by Kober et al. (15) was written before the deep learning era, and the general deep learning for robotics survey by Sunderhauf et al. (16) was written when DRL accomplishments were primarily in simulation. We provide a refreshed overview of the field by focusing on DRL, which is behind the most notable real-world successes of RL in robotics, paying particular attention to papers published in the last five years, during which most of the successes occurred.

3. Taxonomy

This section presents the novel taxonomy we introduce to categorize the literature on DRL. The unique focus of our survey on the real-world successes of DRL in robotics necessitates a new taxonomy to categorize and analyze the literature, which should enable us to assess the maturity of DRL solutions across various robotic applications and derive valuable lessons from both successes and failures. Specifically, we should identify the specific robotic problem addressed in each paper, understand how it has been abstracted as an RL problem, and summarize the DRL techniques applied to solve it. More importantly, we should evaluate the maturity of these DRL solutions, as demonstrated in their experiments. Consequently,

we introduce a taxonomy spanning four axes: **robot competencies learned with DRL**, **problem formulation**, **solution approach**, and **the level of real-world success**.

3.1. Robot Competencies Learned with DRL

Our primary axis focuses on the target robotic task studied in each paper. A robotic task, especially in open real-world scenarios, may require multiple competencies. One may apply DRL to synthesize an end-to-end system to realize all the competencies or learn sub-modules to enable a subset of them. Since our focus is DRL, we classify papers based on *the specific robot competencies learned and realized with DRL*. We first classify the competencies into *single-robot*—competencies required for a robot to complete tasks on its own—and *multi-agent*—competencies required to interact with other agents sharing the workspace with the robot and affecting its task completion.

When a single robot completes a task in a workspace, any competencies it requires can be considered as enabling specific ways to *interact with and affect the physical world*, which are further divided into **mobility**—moving in the environment—and **manipulation**—moving or rearranging (e.g., grasping, rotating) objects in the environment (17, 18, 19). In the robotics literature, mobility⁴ is typically split into two problems: **locomotion** and **navigation** (18, 20). Locomotion focuses on motor skills that enable robots of various morphologies (e.g., quadrupeds, humanoids, wheeled robots, drones) to traverse different environments, while navigation focuses on strategies that direct a robot to its destination efficiently without collision. Typical navigation policies generate *high-level* motion commands, such as desired states at the center of mass (CoM), while assuming effective locomotion control to execute them (18). Some works jointly address the locomotion and navigation problems, which is particularly useful for tasks in which the navigation strategies are heavily affected by the robot’s capability to traverse the environment, as determined by the robot dynamics and locomotion control (e.g., navigating through challenging terrains (20) or racing (21)). We review these papers alongside other navigation papers since their ultimate goal is navigation.

In the robotics literature, manipulation is often studied in table-top settings, e.g., robotic arms or hands mounted on a stationary base with fixed sensors observing the scene. Some other real-world tasks further require robots to interact with the environment while moving their base (e.g., household and warehouse robots), which necessitates a synergistic integration of manipulation and mobility capabilities. We review the former case under the **stationary manipulation** category and the latter under **mobile manipulation**.

When the task completion is affected by the other agents in the workspace, the robot needs to be further equipped with *abilities to interact with other agents*, which we place under the heading of *multi-agent* competencies. Note that some single-robot competencies may still be required while the robot interacts with others, such as crowd navigation or collaborative manipulation. In this category, we focus on papers where DRL occurs at the agent-interaction level, i.e., learning interaction strategies given certain single-robot competencies or learning policies that jointly optimize interaction and single-robot competencies. We further split these works into two subcategories based on the types of agents the robot interacts with: 1) *Human-robot interaction* concerns a robot’s ability to operate

⁴In the robotics literature, both **locomotion** and **navigation** have been used to refer to the ability to move in an environment. To avoid confusion, **mobility** is used in this survey to refer to the overarching category where DRL enables robot movement.

alongside humans. The presence of humans introduces additional challenges due to their sophisticated behavior and the stringent safety requirements for robots operating around humans. 2) *Multi-robot interaction* refers to a robot’s ability to interact with a group of robots. A class of RL algorithms, multi-agent RL (MARL), is typically applied to solve this problem. In MARL, each robot is a learning agent evolving its policy based on its interactions with the environment and other robots, which complicates the learning mechanism. Depending on whether the robots’ objectives align, their interactions could be cooperative, adversarial, or general-sum. In addition, practical scenarios often require decentralized decision-making under partial observability and limited communication bandwidth.

3.2. Problem Formulation

The second axis of our taxonomy is the formulation of the RL problem, which specifies the optimal control policy for the targeted robot competency. RL problems are typically modeled as Partially Observable Markov Decision Processes (POMDPs) for single-agent RL and Decentralized POMDPs (Dec-POMDP) for multi-agent RL. Specifically, we categorize the papers based on the following elements of the problem formulation: 1) *Action space*: whether the actions are *low-level* (i.e., joint or motor commands), *mid-level* (i.e., task-space commands), or *high-level* (i.e., temporally extended task-space commands or subroutines); 2) *Observation space*: whether the observations are *high-dimensional* sensor inputs (e.g., images and/or LiDAR scans) or estimated *low-dimensional* state vectors; 3) *Reward function*: whether the reward signals are *sparse* or *dense*. Due to space limitations, we provide detailed definitions of these terms in the supplementary materials.

3.3. Solution Approach

Another axis closely related to the previous one is the solution approach used to solve the RL problem, which is composed of the RL algorithm and associated techniques that enable a practical solution for the target robotic problem. Specifically, we classify the solution approach from the following perspectives: 1) *Simulator usage*: whether and how simulators are used, categorized into *zero-shot*, *few-shot sim-to-real transfer*, or directly learning offline or in the real world *without simulators*; 2) *Model learning*: whether (a part of) the transition dynamics model is learned from robot data; 3) *Expert usage*: whether expert (e.g., human or oracle policy) data are used to facilitate learning; 4) *Policy optimization*: the policy optimization algorithm adopted, including *planning* or *offline*, *off-policy*, or *on-policy RL*; 5) *Policy/Model Representation*: Classes of neural network architectures used to represent the policy or dynamics model, including *MLP*, *CNN*, *RNN*, and *Transformer*. Please refer to the supplementary materials for detailed term definitions.

3.4. Level of Real-World Success

To evaluate the practicality of DRL in real-world robotic tasks, we categorize the papers based on the maturity of their DRL methods. By comparing the effectiveness of DRL across different robotic tasks, we aim to identify domains where the gaps between research prototypes and real-world deployment are more or less significant. This requires a metric to quantify real-world success across tasks, which, to our knowledge, has not been attempted in the DRL for robotics literature. Inspired by the levels of autonomous driving (22) and Technology readiness level (TRL) for machine learning (23), we introduce the concept of

levels of real-world success. We classify the papers into six levels based on the scenarios where the proposed methods have been validated: 1) *Level 0*: validated only in simulation; 2) *Level 1*: validated in limited lab conditions; 3) *Level 2*: validated in diverse lab conditions; 4) *Level 3*: validated under confined real-world operational conditions; 5) *Level 4*: validated under diverse, representative real-world operational conditions; and 6) *Level 5*: deployed on commercialized products. We consider Levels 1-5 as achieving at least some degree of real-world success. The only information we can use to assess the level of real-world success is the experiments reported by the authors. However, many papers only described a single real-world trial. While we strive to provide accurate estimates, this assessment can be subjective due to limited information. Additionally, we use the level of real-world success to quantify the maturity of a solution for its target problem, irrespective of its complexity.

4. Competency-Specific Review

This section provides a detailed review of the DRL literature, with each subsection focusing on a specific robot competency. In each subsection, we further organize the review based on subcategories specific to each type of competency. After discussing the papers, we conclude each subsection by summarizing the trends and open challenges for learning the competency in question. To aid understanding, each subsection includes a table to overview the reviewed papers. Since our main objective is to assess the maturity of DRL solutions, we note the level of real-world success achieved by each paper in the table. For a comprehensive categorization of the papers, please refer to Tables 1–6 in the supplementary materials.

4.1. Locomotion

Locomotion research aims to develop motor skills for robots to traverse various real-world environments. Prior to the deep learning era, several pioneering works have explored RL for locomotion control and delivered promising hardware demos, e.g., quadruped walking (24) and helicopter control (25, 26). This subsection reviews DRL solutions for locomotion separately from navigation, where the controllers follow high-level navigation commands. Since locomotion mainly concerns motor skills, the problem complexity is primarily influenced by the system dynamics (27). We organize this subsection accordingly and review three representative locomotion problems: **quadruped and biped locomotion**, and **quadrotor flight control**. See Figure 2 for an overview of the papers reviewed.

4.1.1. Quadruped Locomotion. Quadruped locomotion is one of the robotic domains where DRL has provided mature real-world solutions. Multiple robotics companies, such as ANYbotics, Swiss-Mile, and Boston Dynamics, have reported that DRL was integrated into their quadruped control for applications including industrial inspection, last-mile delivery, and rescue operations. In the literature, DRL methods were first validated for *blind quadruped walking*, i.e., relying solely on proprioceptive sensors on flat indoor surfaces (28, 29). These policies were typically trained in simulation and deployed zero-shot in the real world. The main challenge lies in the sim-to-real gap in quadrupeds’ intrinsic dynamics. Several strategies have been explored to bridge the reality gap: 1) learning actuator models, either analytical (28) or neural network-based (29), from robot data to improve simulation fidelity; 2) randomizing dynamics parameters (28, 29) and, even further, randomizing morphology (30), which enables generalization to unseen quadrupeds; and 3) adopting a hierarchi-

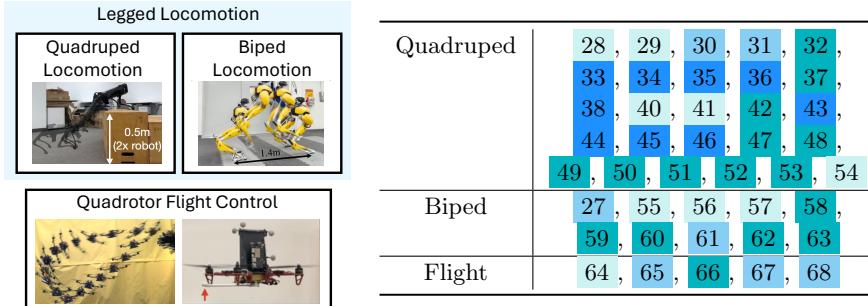


Figure 2: **Left:** An overview of the three locomotion problems reviewed in Sec. 4.1, including quadruped (49) and biped (63) locomotion, and quadrotor flight control (64, 67); **Right:** Locomotion papers reviewed in Sec. 4.1. The color map indicates the levels of real-world success: *Limited Lab*, *Diverse Lab*, *Limited Real*, and *Diverse Real*.

cal structure with a low-level, model-based controller to handle dynamics discrepancy and external disturbances while facilitating efficient learning. The interface between the DRL policy and the model-based controller could be defined at various levels, such as joint positions (31, 32, 33), leg poses (28), gait parameters (34, 35), or temporally-extended macro actions (36). As robots venture beyond controlled lab environments, they encounter more challenging terrains such as discontinuous, deformable, or slippery surfaces. Four main techniques have been used to address the additional challenges. First, the terrain and contact information are not directly observable. Privileged learning has been commonly adopted as a solution (34, 33), where a policy with privileged terrain information is trained first and then distilled into a student policy operating on realistic sensor inputs. Alternatively, end-to-end training can be achieved with the help of state estimation (37, 38) and asymmetric actor-critic (39, 38). In both cases, an extended history of observations is often set as input.

Second, policies should be exposed to diverse conditions during training for generalization in the wild. A learning curriculum that progressively increases task difficulty is often adopted to facilitate training (34, 33, 36, 38, 37). Advanced terrain models can also improve performance on terrains with complex contact dynamics, e.g., deformable surfaces (37).

Third, exteroceptive sensors are crucial for traversing risky terrains, as they allow the quadruped to adapt to terrains without stepping on them. For example, they have fostered more efficient and robust stair traversal (35, 43). Exteroceptive observations are typically in the form of terrain height maps (35, 36), depth images (44, 45), and RGB images (43). Privileged learning is widely used to facilitate policy learning from these high-dimensional observations (44, 35, 45). To reduce the sim-to-real gap in sensor inputs, techniques such as injecting simulated sensor noise (35), post-processing depth images (50), learning vision encoders from real-world samples (43) are shown effective. Additionally, policies benefit from improved representation via self-supervised learning (36, 35), cross-modal embedding matching (44, 43), or using models with higher capacity, such as transformers (69, 45).

Fourth, traversing certain complex terrains demands advanced locomotion skills beyond regular walking gaits. For example, end-to-end DRL policies typically struggle with terrains that have sparse contact regions. Jenelten et al. (46) showed that training an RL policy to track reference footholds provided by trajectory optimization results in more accurate and robust foot placement on sparse terrains. Jumping further extends the robots' ability to

cross gaps beyond their body length. For example, Yang et al. (47) trained a DRL policy to generate trajectories with a model-based tracking controller handling the complex jumping dynamics. Fall recovery is another essential skill, especially for automatic reset in real-world RL (48, 53). Several works have trained DRL policies for fall recovery (29, 31, 48, 32, 41). However, both jumping and fall recovery have only been validated on flat surfaces so far.

To effectively leverage agile locomotion skills for complex downstream tasks like parkour (49, 50), it is crucial to develop *multi-skill* policies. Learning multiple skills jointly has also been shown effective in fostering policy robustness (63). One approach is to create a set of RL policies (51, 32, 50), each tailored to a specific skill, and then train a high-level policy to select the optimal skill (32). Alternatively, a single policy can be distilled from specialized skill policies through BC (50). To avoid the cumbersome procedure of training multiple specialized policies, several works explored constructing a unified policy directly. For instance, MoB (52) encoded various locomotion strategies into a single policy conditioned on gait parameters. Cheng et al. (49) used a unified reward consisting of waypoint and velocity tracking terms to learn diverse parkour skills. Fu et al. (42) showed that energy minimization led to smooth gait transitions. Motion imitation reward is another widely used and unified approach for learning naturalistic and diverse locomotion skills (51, 40).

Remark on RL algorithms. We conclude the review on quadruped locomotion with a remark on the RL algorithms used in the literature. The most mature DRL solutions for quadruped locomotion followed the zero-shot sim-to-real transfer scheme, predominantly using on-policy model-free RL, e.g., PPO (70), due to its robustness to hyperparameters. Gangapurwala et al. (36) noted that on-policy RL could be less favorable when the action space is temporally extended or deterministic control actions are preferred. Meanwhile, researchers have explored few-shot adaptation and real-world RL, either model-free (48, 53) or model-based (54), to update policies using real-world rollouts to further generalize policies to novel situations without accurate simulation. However, most works along this line have only been validated in limited lab settings. The state-of-the-art performance for real-world fine-tuning (48) and learning from scratch (53) were achieved by using off-policy RL to learn walking and fall recovery. However, the tested conditions remain limited compared to mature zero-shot solutions.

4.1.2. Biped Locomotion. Compared to the quadruped case, the DRL literature on bipedal locomotion is sparser, and the real-world capabilities demonstrated are more limited. We confine the discussion to 3D bipedal robots, which can move freely in all spatial dimensions, unlike 2D bipeds that are attached to booms and confined to 2D planar motion (71), for their greater practical utility. The literature begins with walking on flat indoor surfaces (55, 57) and extends to walking on various indoor (56, 58, 27) and outdoor terrains (60, 62), and under external forces (27, 58). Other demonstrated skills include stair traversal (59), hopping (57), running (57, 63), jumping (63), and traversing obstacles and gaps (61). More advanced skills have been showcased by industrial companies⁵, but no technical reports are publicly available to reveal if RL was used in their demos. Notably, some of these works deployed their locomotion policies on humanoid robots (56, 60, 62) while others on bipedal robots without upper bodies (55, 58, 59, 27, 61, 63).

The DRL techniques for bipedal locomotion largely overlap with those for quadrupeds but show three distinct trends due to the complex and under-actuated dynamics of bipeds.

⁵For example, Unitree (<https://t.ly/s1FwW>) and Boston Dynamics (<https://t.ly/NaSa0>)

First, learning basic standing and walking skills is already challenging due to bipeds' non-statically stable dynamics (55). Thus, model-based approaches are frequently used to facilitate RL, either by generating reference gaits to guide RL (55, 58, 63) or handling low-level control for high-level RL policies (60). Alternatively, Siekmann et al. (57) offered an end-to-end solution with a reference-free periodic reward design based on periodic composition. Second, the role of state and action *memories* was particularly noted (55), especially a combination of both long- and short-term memories (63). Thus, most works adopted sequence models in their policy architecture (63, 61, 55, 57, 59, 62). Third, almost all these policies were zero-shot transferred from simulation. One exception is GAT (56), which collected real-world samples to refine a simulator iteratively, enabling an NAO to walk on uneven carpets. The limited real-world learning examples are likely due to bipeds' limited recovery capabilities, which hinder their resilience in trials, particularly their ability to auto-reset.

4.1.3. Quadrotor Flight Control. Flight control for unmanned aerial vehicles (UAVs), in particular quadrotors, is another problem where DRL has shown compelling performance. Hwangbo et al. (64) developed the first DRL quadrotor control policy that was successfully validated on hardware for waypoint tracking and recovery from harsh initialization. Later studies showed that carefully designed simulated dynamics, domain randomization (65), and carefully designed action space, specifically collective thrust and body rates (66), can facilitate policy robustness. Zhang et al. (67) applied RMA to train a robust near-hover position controller adaptable to unseen disturbances. Eschmann et al. (68) introduced the first off-policy RL paradigm for quadrotor control, capable of training a deployable control policy within 18 seconds for waypoint tracking. In summary, DRL has demonstrated better robustness than classical feedback controllers (e.g., PID) in hovering control (65, 67). However, DRL policies tend to have larger tracking errors than carefully designed optimization-based controllers for waypoint tracking (64, 66). Yet the fundamental advantage of RL over optimal control is it enables joint optimization for planning and control (21), making it an ideal candidate for agile navigation such as racing (see Sec. 4.2).

4.1.4. Trends and Open Challenges in Locomotion. In summary, DRL has shown effectiveness in synthesizing robust and adaptive locomotion controllers for challenging conditions. DRL Techniques used for quadruped, biped, and flight control heavily overlap. For instance, RMA (33), initially proposed for quadruped locomotion, has been adapted for both biped (27) and quadrotor flight control (67). However, the maturity of DRL solutions varies across domains. Quadrupeds can traverse various indoor and outdoor terrains via DRL, while real-world bipedal locomotion skills achieved by DRL are more limited. For quadrotors, most tests remain confined to controlled, obstacle-free indoor environments. Hardware accessibility is a contributing factor. The introduction of low-cost quadrupeds has spurred quadruped research and led to open-sourced and unified software packages. Conversely, the high cost of bipedal hardware limits extensive real-world testing, though recent advances in humanoid hardware are expected to boost biped research. More importantly, the quadruped dynamics are inherently more stable, whereas bipeds and quadrotors are more prone to catastrophic failures under control errors, imposing higher requirements on both robustness and precision of control (63). High-speed quadrotor control in outdoor scenarios with complex obstacles further requires the policy to ensure the *long-horizon* feasibility of the closed-loop trajectories (72). End-to-end RL integrating long-horizon planning and short-horizon control shows promise as a solution (7). In addition to ensuring long-horizon

feasibility, integrating locomotion with downstream tasks (e.g., loco-manipulation) is an exciting direction in general, but how to discover skills necessary for downstream tasks remains an open question.

Key Takeaways

- DRL has enabled mature quadruped locomotion control; yet, the maturity of DRL-based solutions for other locomotion problems is lower.
- Hardware accessibility is an important contributing factor. Low-cost and standard hardware platforms would facilitate DRL development.
- The inherently complex dynamics of certain locomotion problems present fundamental challenges to the reliable deployment of DRL locomotion controllers.
- Even in the mature quadruped locomotion domain, open questions remain, such as 1) effectively integrating locomotion with downstream tasks via RL, and 2) enabling efficient and safe real-world learning.

4.2. Navigation

Navigation focuses on the decision-making challenge in mobility: transporting an agent to a goal location while avoiding collisions, typically assuming effective locomotion. As a fundamental mobility capability, navigation has an extensive history in robotics research (18). “Classical” navigation approaches employ mapping, localization, and planning modules to determine and execute a path to a goal. Planning is typically decomposed into global planning, which produces a coarse path, and local planning, which tracks the global plan and handles collision avoidance. In this section, we delineate navigation works by embodiment: **wheeled, legged, and aerial navigation** and identify capabilities enabled by RL in each setting. Social navigation, where the robot navigates in the presence of humans, is deferred to Sec. 4.5. Multi-robot navigation is similarly deferred to Sec. 4.6.

4.2.1. Wheeled Navigation. Navigation for wheeled robots, in particular, has a long history in robotics (18). We discuss several common wheeled navigation settings, including geometric navigation, visual navigation, and offroad navigation.

Geometric Navigation. Early attempts aimed to verify RL’s capability in solving navigation problems typically solved with modular classical approaches (73). These RL policies directly map 2D laser scans to control actions, unlike classical methods that construct explicit maps from the laser scans. While showing promise, they often did not compare against classical approaches or failed to outperform them (12). Some recent studies have benchmarked such RL-based approaches and found them superior in challenging problems with dense obstacles and narrow passages (74). Instead of replacing the entire navigation stack with an RL policy, *modular* approaches replace specific components like the local planner (75) or the exploration algorithm (76) with RL, enabling better performance than classical baselines. However, these improvements were mainly observed in limited real settings. Most commercially deployed systems still primarily adopt classical stacks, owing to the lack of safety, interpretability, and generalization of RL-based methods (12, 74).

Visual Navigation. Visual navigation refers to problems where agents navigate to a goal based on visual observations. The additional input and task complexity pose challenges but enable agents to learn common strategies for navigating in similar environments (e.g.,

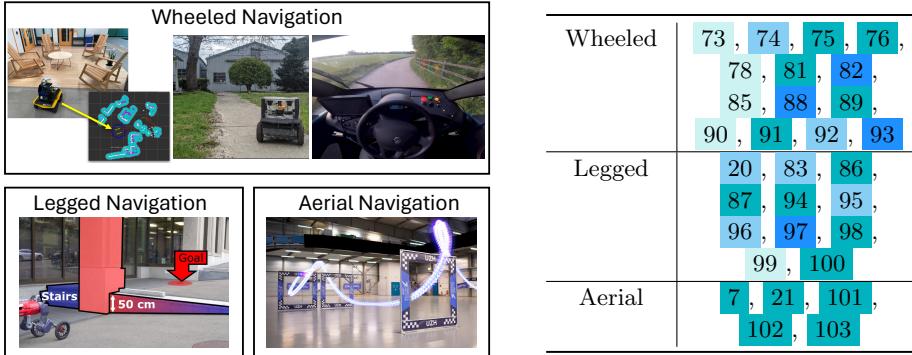


Figure 3: **Left:** An overview of the three navigation problems reviewed in Sec. 4.2, including wheeled navigation (74, 88, 92), legged navigation (97), and aerial navigation (21); **Right:** Navigation papers reviewed in Sec. 4.2. The color map indicates the levels of real-world success: *Limited Lab*, *Diverse Lab*, *Limited Real*, and *Diverse Real*.

homes), where structural patterns emerge in visual data. Goals are typically specified as a point relative to the agent (termed *pointgoal* navigation) or as an image of a particular object (*objectgoal* or *imagegoal*). RL is also commonly applied to vision-and-language navigation problems (77), though very little work has demonstrated these capabilities on a real robot. Many works (78, 79) map visual observations to actions directly without mapping or planning modules. These end-to-end methods have achieved near-perfect results on pointgoal tasks in visually realistic simulations (80). However, training such policies is challenging due to the need for scene understanding, intelligent exploration, and episodic memory. Their applicability for real-world navigation remains unclear, as they have mostly been validated in limited real or lab settings. Other works have investigated modular designs, e.g., using RL as a global exploration policy together with explicit mapping and local planning (81, 82). They have outperformed both classical and end-to-end learning baselines on pointgoal and imagegoal tasks. However, some challenges with such modular approaches exist, such as dynamic obstacles, where end-to-end methods have shown promise (83).

Despite the plethora of RL works on visual navigation, most are limited to simulation. While these simulators are typically constructed with real-world scans (77, 84), their transferability to the real world remains debatable. Some works reported poor transfer due to visual domain differences (82), while others found success through parameter tuning (85), abstraction of dynamics (86), or employing only depth images rather than RGB-D (83, 87).

Off-road navigation. Navigating off-road presents additional challenges due to the dynamics and traversability of different terrains. Some methods tackled these challenges with model-based RL to learn predictive models of events or disengagements (88), or utilizing demonstration data with offline RL (89). Success has also been achieved in high-speed, off-road driving with model-based RL (90) and, recently, vision-based model-free RL (91).

Autonomous Driving. Autonomous driving extends wheeled navigation to full-size passenger vehicles operating at higher speeds in more complex and safety-critical environments. RL has achieved limited real-world success for autonomous driving (8) with a few examples under specific conditions. Kendall et al. (92) trained a lane-following policy by learning to maximize its progress before the safety driver intervenes. More recently, Jang et al. (93) trained a cruise control policy, where the policy command is wrapped by manu-

ally specified thresholds to ensure safety. They deployed their policy onto 100 vehicles to smooth traffic flow in a field test. Their work suggested a pragmatic approach to embed RL into self-driving stacks and showed its potential benefits at the fleet level.

4.2.2. Legged Navigation. Legged navigation shares many challenges with wheeled navigation but also enables transversal of more complex terrains. Some have shown that robust visual-legged navigation policies can be learned with low-fidelity kinematic-only simulation for both indoors (83, 86) and outdoors (86, 94). The policies thus focus on kinematic-level control while assuming effective low-level locomotion control during deployment. Truong et al. (86) showed that this approach, in contrast to learning end-to-end policies with high-fidelity simulation, facilitates faster simulation and improves policy generalizability. With legged locomotion dynamics abstracted away, the approaches are similar to the wheeled case, with the main challenge being the visual domain gap. Unsupervised representation learning (83) and pre-trained vision models (94) have been used to facilitate robust visual policies. For outdoor scenes, Truong et al. (87) zero-shot transferred policies trained in well-established indoor simulators to outdoors, using goal vector normalization and camera pitch randomization to bridge the indoor-to-outdoor domain gap. Sorokin et al. (94) used a high-fidelity autonomous driving simulator and extracted visual features from a pre-trained semantic segmentation model for robust sim-to-real transfer to sidewalk navigation.

While abstracting away low-level locomotion has advantages, it limits the system from fully utilizing the agile locomotion skills endowed by advanced locomotion controllers. Recent research has explored DRL frameworks integrating locomotion with navigation, achieving high-speed obstacle avoidance (100) and agile navigation over challenging terrains (e.g., stairs, gaps, and boxes) (20, 96) and through confined 3D space (98, 99). Particularly, Lee et al. (97) demonstrated kilometer-scale navigation with a wheeled-legged robot in urban scenarios, overcoming challenging terrains and dynamic obstacles. The integrated policy network can be end-to-end, taking goal coordinates as input and outputting locomotion commands (20, 99). He et al. (100) further introduced a recovery policy coordinated using a learned reach-avoid value network. Alternatively, training efficiency can be improved with hierarchical architectures, where a high-level policy governs pre-trained low-level locomotion policies (95, 96, 97, 98). Despite the potential of integrating locomotion with navigation, policy training could be costly and unstable due to the complex low-level dynamics together with the long-horizon nature and sparse rewards of the navigation tasks (20, 96). Classical planning algorithms are often used for generating local waypoints to reduce the navigation horizon and synthesizing feasible paths to guide initial training (97).

4.2.3. Aerial Navigation. Compared to wheeled and legged robots, aerial vehicles such as quadrotors are more fragile, requiring higher robustness and safety in navigation policies. The weight and power constraints of quadrotors also limit the use of sophisticated sensors. Several works have explored DRL-based aerial navigation using low-cost monocular cameras (101, 102). Sadeghi et al. (101) leveraged visual domain randomization to achieve zero-shot sim-to-real transfer for indoor aerial navigation. Kang et al. (102) showed the values of 1) task-specific pre-training in simulation for learning generalizable visual representation and 2) the use of real-world data for learning accurate dynamics (79). Similar to quadruped navigation, DRL has been used to develop end-to-end navigation and locomotion policies for agile aerial navigation. Kaufmann et al. (7) achieved human champion-level performance in drone racing. A key recipe behind their success was augmenting simulation

with data-driven residual models of the drones perception and dynamics. Their subsequent study (21) showed that RLs advantage over model-based methods lies in its ability to directly optimize the long-horizon racing task objective. However, DRL-based policies are still less robust than human pilots, limiting their operational conditions. Integrating actor-critic RL with differential MPC has shown promise in enhancing robustness (103).

4.2.4. Trends and Challenges in Navigation. RL has shown potential for various submodules of navigation systems, such as local planning (75, 104) and global exploration (76, 81, 82), and for constructing end-to-end navigation solutions (74). However, RL-based solutions for navigation lack the generalization, explainability, and safety guarantees of classical systems and thus have not seen widespread real-world deployment (12, 74).

In visual navigation, model-free, end-to-end policies show promise for structured indoor environments like homes (105), while modular architectures boost performance without sacrificing guarantees and generalization (81, 82). Striking the right balance between learned and classical modules remains an open challenge. Hybrid approaches may be promising, for example, leveraging implicit map-like representations learned by end-to-end approaches (106), or using differentiable scene representations (107) to enable RL with algorithmic structure. RL-based vision-and-language navigation (77) is relatively under-explored in real-world settings but promising given the recent advances in vision-language models.

In legged navigation, abstracting away low-level dynamics has been shown to facilitate sim-to-real transfer for navigation (86). For agile legged and aerial navigation, where low-level complexity is unavoidable, jointly learning navigation and locomotion yields promising results (100, 20, 96, 7). Yet, involving locomotion complicates the training of long-horizon navigation policies, which requires future developments to stabilize learning.

Finally, learning navigation (collision avoidance, in particular) for safety-critical systems, like urban autonomous vehicles and drones, is challenging due to stringent robustness requirements in perception and control. These domains have seen fewer real-world successes as a result. Real-world data can help improve simulation fidelity for this purpose (7, 21, 103), though establishing guarantees on their performance remains difficult.

Key Takeaways

- While end-to-end RL excels at visual navigation in simulation, most real-world successes deploy modular designs and learn components of the navigation stack.
- Integrating RL into these modular architectures, e.g., for local planning or semantic exploration, is a promising avenue.
- Recent work reasoning jointly about navigation and locomotion enables agile legged and aerial navigation, yet how to learn long-horizon navigation stably and efficiently with low-level control in the loop remains an open challenge.
- Safety-critical applications like urban autonomous driving or outdoor drone flight have seen few real-world successes due to the higher requirements for robustness and the lack of explainability and generalization on the part of RL algorithms.



	Grasping	108 , 109 , 110 , 111 , 112
Pick-and-place	End-to-end Pick-and-place	54 , 113 , 114 , 115 , 116 , 117 , 118 , 119 , 120 , 121 , 122 , 123 , 124 , 125
Contact-rich	Assembly	126 , 127 , 128 , 129 , 130
	Articulated Objects	122 , 131 , 132 , 133
	Deformable Objects	134 , 135 , 136 , 137
In-hand	—	138 , 139 , 140 , 141 , 142 , 143 , 144
Non-prehensile	—	109 , 118 , 145 , 146 , 147

Figure 4: **Top:** An overview of the four manipulation problems reviewed in Sec. 4.3, including pick-and-place (108), contact-rich manipulation (130), in-hand manipulation (141), and non-prehensile manipulation (145); **Bottom:** Manipulation papers reviewed in Sec. 4.3. The color map indicates the levels of real-world success: [Limited Lab](#), [Diverse Lab](#), [Limited Real](#), and [Diverse Real](#).

4.3. Manipulation

Manipulation refers to an agent’s control of its environment through selective contact (19). To perform useful work in the world, robots require manipulation capabilities such as pick-and-place, mechanical assembly, in-hand manipulation, non-prehensile manipulation, and beyond. Manipulation poses several challenges for both analytical and learning-based methods (11), as the mechanics of contact are complex and difficult to model, and open-world manipulation requires strong generalization and fast online learning. RL is well-suited to these challenges, but manipulation poses fundamental difficulties for RL: large observation and action spaces make real-world exploration prohibitively time-consuming and unsafe; reward function design requires domain knowledge; tasks are often long-horizon; and instantaneous environment resets are usually unrealistic in real-world tasks. Despite these challenges, DRL has achieved notable successes in manipulation recently.

In this subsection, we review progress in several manipulation capabilities enabled by DRL, following the outline from Mason’s seminal review (19): **pick-and-place**, **contact-rich manipulation**, **in-hand manipulation**, and **non-prehensile manipulation**. See Figure 4 for an overview of the papers reviewed in this subsection. Note that this subsection focuses on stationary manipulators, and we defer mobile manipulation to Sec. 4.4.

4.3.1. Pick-and-place. Picking and placing objects is a longstanding challenge in manipulation, requiring the ability to perceive objects, grasp them, determine appropriate placements, and generate collision-free motion. *Structured* pick-and-place, in which the environment is engineered to reduce complexity and objects are known *a priori*, is well-understood and widely deployed in manufacturing contexts. Open-world, *unstructured* pick-and-place—rearranging arbitrary objects in the wild—remains a challenge. In recent years, more tra-

ditional robotic approaches have seen success in industrial applications like fulfillment, employing machine learning for object detection and grasping but deferring control to analytical methods (19). While pick-and-place tasks serve as a common testbed for new RL algorithms (115, 116, 117, 118, 119, 54), end-to-end RL methods still lack the ability to pick and place novel objects in the open world with generality. However, modular approaches, such as solving grasping with RL, have enabled some real-world successes. We will review RL-based solutions to the subproblem of grasping and then discuss end-to-end RL methods, omitting a discussion of motion generation for which RL is not commonly used.

4.3.1.1. Grasping. Grasping objects is a fundamental capability essential for pick-and-place and other downstream tasks, such as in-hand manipulation and assembly. Some of the first large-scale successes of DRL for manipulation were in grasping objects with unknown geometry and appearance (108). Where analytical methods had achieved grasping of known objects using taxonomies and databases, these works leveraged thousands or millions of grasp attempts to learn grasping behaviors through interaction. Many works frame grasping as a bandit or classification problem, where the action space consists of discrete grasp candidates and the picking motion is executed open-loop (108, 109). These methods commonly employ sparse rewards that indicate success when an object is lifted and collect data in a self-supervisory manner. Similar systems have been reportedly integrated into fulfillment applications⁶ with diverse objects. Closed-loop grasping—controlling the end-effector pose and/or fingers directly to achieve stable grasps—can be formulated as a sequential decision-making problem and solved with RL. While some successes have been seen (110, 111, 112), closed-loop grasping remains challenging due to the additional complexity of learning vision-based closed-loop control, and such systems have not seen the same level of real-world success as open-loop ones. In both closed- and open-loop grasping, while some works exclusively collect real-world data (109, 110, 112), the common recipe is to use simulation for data collection (108) or policy training (111), often employing domain adaptation to ensure visual similarity between the simulator and real world.

4.3.1.2. End-to-end Pick-and-place. Learning general-purpose pick-and-place in the open world remains daunting for end-to-end RL, owing to the sheer variety of objects and tasks and the limited generalization of current algorithms. This variety also precludes the common sim-to-real recipe successful in other domains like grasping and in-hand manipulation, where tasks and objects can be enumerated during training. Nonetheless, some major milestones in end-to-end pick-and-place have been observed: Levine et al. (113) demonstrated the potential of deep visuomotor policies; Riedmiller et al. (119) demonstrated pick-and-place manipulation with a hierarchical policy trained in the real world; and Lee et al. (116) achieved stacking of diverse objects through sim-to-real transfer. Augmenting the action space with primitives (123) can help in reducing the task horizon and is a natural means to incorporate human engineering. Recent work leveraging large vision-language models shows promise in handling open-ended diverse objects and task objectives specified by natural language (124). The potential of RL to solve this longstanding challenge is only now coming into focus with emerging large-scale robotic datasets and foundation models. Despite not yet achieving widespread success in real-world deployments, many important RL innovations have been demonstrated in pick-and-place problems, addressing challenges

⁶See examples from Ambi Robotics (https://t.ly/tSds_) and Covariant (<https://t.ly/S5pnz>).

such as multi-task learning (114, 115, 125), sample efficiency (54), defining and computing reward (120, 121), resetting the environment (117), and utilizing human demonstrations or offline data (122, 116, 124).

4.3.2. Contact-rich Manipulation. While pick-and-place tasks are often assumed to be strictly kinematic, contact-rich tasks like mechanical assembly (e.g., peg insertion), interacting with articulated objects (e.g., opening doors), and manipulating deformable objects, require reasoning about dynamics and relaxing the rigid-body assumption of the objects. We discuss several contact-rich tasks where RL has advanced the state of the art: assembly, articulated object manipulation, and deformable object manipulation.

4.3.2.1. Assembly. Assembly tasks are crucial in manufacturing, and automating them is a longstanding challenge in robotics. Existing industrial solutions tend to rely on extensive engineering of the environment and robot motions, resulting in behaviors sensitive to small perturbations and costly to design. Assembly is challenging for RL due to the difficulty in controlling contact-rich interactions and the stringent requirements for accuracy and precision, coupled with the need to handle diverse object parts. While RL has not seen widespread deployment in industrial contexts, some notable successes have been observed in recent years. Many approaches employ sim-to-real transfer to achieve assembly (130), though some train policies directly in the real world (126, 127, 128, 129), typically leveraging human demonstrations. Luo et al. (128) notably compare against solutions provided by integrators and find their RL-based policies more robust to perturbation. A common strategy among approaches to assembly is using residual RL (126), in which a residual policy is learned on top of a reference trajectory. Most works assume that the object is already grasped before assembly. By contrast, Tang et al. (130) present a sim-to-real RL framework for the entire assembly pipeline, including object detection, grasping, and insertion, achieving diverse assembly tasks by leveraging recent advances in contact simulation and developing algorithmic advances for sim-to-real transfer.

4.3.2.2. Articulated Objects. Some limited successes have been observed in constrained manipulation tasks like opening drawers. Most commonly, these tasks are used to demonstrate RL capabilities without dedicated efforts to realize practical deployment (122, 131). Other works target this class of skills in particular (132, 133) with limited success.

4.3.2.3. Deformable Objects. Deformable objects, such as cloth, present additional challenges owing to the difficulty in accurately modeling soft materials. Tasks like cloth folding (134, 135, 136) and assistive dressing (137) have thus received considerable attention in RL. These works often employ sim-to-real transfer (134, 136, 137), and often simplify the tasks using primitives such as pick-and-place (135) and flinging (136).

In summary, open-world contact-rich manipulation inherits the challenges of unstructured pick-and-place (namely, generalization to novel objects and tasks) and the additional challenge of controlling contact-rich interactions. Nonetheless, some successes have been demonstrated in contact-rich tasks, particularly assembly and deformable objects, where tasks are predefined, objects are enumerable, and rigid grasps are usually assumed.

4.3.3. In-hand Manipulation. Humans exhibit many in-hand manipulation behaviors, reorienting and re-positioning objects to facilitate downstream manipulation. Impressive

strides in the development of these capabilities have been made with DRL in recent years, allowing agents to learn such complex in-hand manipulation behaviors with impressive generalization. Several works focused on re-orienting single objects to target configurations (138, 139), employing pose estimation modules trained in simulation or using proprioception alone (143). Nagabandi et al. (140) similarly demonstrated rotating Baoding balls with model-based RL. While showing impressive dexterity, these works focus on manipulating known objects (e.g., a given cube) with low-dimensional observations. Recent methods leveraging vision and/or tactile data have demonstrated rotating arbitrary objects about arbitrary axes (141), even against gravity (142, 144). These approaches employ extensive domain randomization and typically leverage privileged information (e.g., object shape information, dynamic properties) and dense rewards when training in simulation. An open challenge is integrating these in-hand manipulation skills with other manipulation abilities (e.g., tool use), which require re-orientation to a target configuration suitable for a downstream task.

4.3.4. Non-prehensile Manipulation. Non-prehensile manipulation, namely moving objects without grasping, is crucial when objects are too large to be grasped, grasps are occluded, or in tool use. Object pushing abilities have long been demonstrated with RL (118), and studied in connection to grasping (109, 145). Recently, general non-prehensile re-orientation of diverse objects has been enabled through sim-to-real transfer of RL policies (146, 147). Similar to in-hand manipulation, learning with privileged information (i.e., object geometry) before distilling a student policy is a common approach. Further work is warranted to integrate these skills with prehensile and in-hand behaviors and to develop *extrinsic* dexterity, where the environment is used to facilitate manipulation. How to synthesize these capabilities for general-purpose, open-world manipulation remains an open question.

4.3.5. Trends and Open Challenges in Manipulation. RL is beginning to achieve real-world success in various manipulation problems. Generally, RL has been more successful in domains where the space of tasks is more constrained—grasping, in-hand manipulation, and assembly—rather than less, e.g., end-to-end pick-and-place. These more constrained tasks allow for a priori reward design and zero-shot sim-to-real transfer, whereas open-world pick-and-place and contact-rich manipulation require generalizing to diverse objects and tasks. The limitations of physical simulation may also preclude scaling sim-to-real for contact-rich tasks. Differentiable simulation has shown promise for this challenge (148). Open-world manipulation will require several advances, including scaling collections of simulated assets and tasks; few-shot sim-to-real (131); multi-task learning (114, 125); learning autonomously in the real world (120, 117, 54); learning reward functions from examples (120) or human videos (121); and utilizing human demonstrations (127), offline data (122) and foundation models (124). Incorporating priors, such as symmetry (112) and geometry (149), is promising for improving sample efficiency, generalization, and safety. Learning more complex behaviors, e.g. bimanual (150) or dynamic tasks like table tennis (151), is another important avenue for future work (11, 19).

Additionally, action spaces are typically chosen by domain experts to match each problem at hand. Open-loop grasping tends to employ an abstraction of motion generation for reaching and closing the fingers, whereas closed-loop grasping, assembly, and end-to-end pick-and-place methods typically control the end-effector Cartesian pose or velocity. Most in-hand manipulation approaches control the fingers in configuration space, keeping the

end-effector itself in a fixed position. Equipping one agent with these various manipulation abilities remains an important challenge for deploying capable manipulators in the real world. Moreover, many of these real-world successes are demonstrated on short-horizon tasks; further work is warranted to build agents that can reason over longer periods of time and compose learned abilities together to solve long-horizon tasks (11, 123, 132, 152, 153).

Key Takeaways

- RL solutions for manipulation are generally less mature than locomotion, with few deployments in the wild, yet there exist many impressive demonstrations in representative real-world conditions.
- Manipulation subproblems where tasks can be enumerated a priori—e.g., grasping, in-hand manipulation, assembly—allow for zero-shot sim-to-real transfer, facilitating many of the real-world successes.
- Integrating manipulation subfields and connecting with task planning to build a generally competent manipulator remains an open challenge.

4.4. Mobile Manipulation

Mobile manipulators are robotic agents combining mobility and manipulation competencies, unlocking applications in households, healthcare, and logistics. Mobile manipulation (MoMa) problems present unique challenges requiring more than a simple concatenation of locomotion and manipulation, including the need to control and synchronize many degrees-of-freedom governing multiple body components (e.g., head, arm(s), and base/legs), strong partial observability and tasks with a natural long horizon. DRL has been applied to tackle various types of MoMa tasks, including 1) learning precise, real-time **whole-body control**; 2) learning object perception and interaction in **short-horizon interactive tasks**; and 3) high-level decision-making in **long-horizon interactive tasks**. In this section, we review works addressing these three problems summarized in Figure 5.

4.4.1. Learning Whole-Body Control. The common goal in whole-body control (WBC) for mobile manipulators is to determine an action or sequence of actions for all degrees of freedom of the body to reach a desired configuration, possibly fulfilling additional constraints. Frequently, the desired configuration is specified as the desired position or pose of one or more of the links of the agent, e.g., the desired pose of the end-effector (154, 155, 156, 157). While there exist model-based analytical methods for whole-body control in advanced control theory literature (158), DRL has been explored as a powerful alternative in situations where either the system dynamics are hard to model (e.g., leg-ground contact, slippery wheels, unknown manipulator dynamics), or when the inference-time computation is constrained—a frequent problem in MoMa tasks due to the robot embodiment’s complexity. For example, Wang et al. (154) and Fu et al. (156) learned whole-body policies that enable a wheeled mobile manipulator and a quadruped with an arm to reach points in 3D space with their end-effector. Ma et al. (155) learned a locomotion policy robust to random wrench perturbation and used an MPC planner to control the arm for point reaching.

Typically, whole-body control problems focus on precise control of the end-effector without taking into account the agent’s surroundings: the policy takes proprioceptive sensing

as the observation and tries to minimize the difference to the desired configuration. Notably, recent works have explored how to integrate low-level whole-body control skills into hierarchical RL architectures (159, 160, 161), where the higher level perceives the surroundings and queries a low-level whole-body skill with the right desired pose as the goal. This extends the success of DRL in learning WBC to more complex interactive MoMa tasks.

4.4.2. Short-horizon Interactive Tasks. Short-horizon interactive tasks often focus on learning specific sensorimotor skills that require no memory or planning capabilities. Many works have explored applying DRL to these short-horizon tasks, including grasping (162, 163, 161), ball kicking (164, 160), collision-free target tracking (165, 166, 167), interactive navigation (168), and door opening (169, 170). Notably, Ji et al. (160) used hierarchical RL to learn soccer kicking skills, where a high-level policy generates the desired end-effector trajectory executed by a low-level policy. Hu et al. (165) improved the training efficiency by deriving a low-variance policy gradient update through action space decomposition. Cheng et al. (171) learned separate skills for locomotion and manipulation on a quadruped in simulation and chained different skills using a behavior tree. Ji et al. (164) learned a whole-body dribbling policy in simulation, transferring it zero-shot to the real world using extensive domain randomization in visual input and simulation parameters. Liu et al. (161) learned grasping policies via hierarchical RL and teacher-student distillation, where an image-based student policy is distilled from a state-based teacher policy. Interactive tasks require policies to make decisions based on sensor observations of their surroundings. Therefore, the policy usually takes in high-dimensional observations such as camera or lidar readings (Table 2). Meanwhile, these tasks often involve hard-to-model dynamics such as contact forces or articulated object motion, making model-free RL an appealing alternative both to classical methods and to model-based RL (Table 4).

4.4.3. Long-horizon Interactive Tasks. For a mobile manipulator to function in unstructured environments such as offices (172), homes, or kitchens (173), it needs to handle tasks with long horizons and strong partial observability. However, end-to-end RL struggles on long-horizon tasks due to the difficulty of exploring the state-action space to find successful strategies, requiring many samples to train. Partial observability is also challenging for DRL as it requires complex network architectures that can encode observation history (e.g., RNNs or LSTMs) or some other mechanism to aggregate observations and model the environment (e.g., mapping or 3D reconstruction). One possible way to mitigate this issue is to make use of expert demonstrations or simulation data to bootstrap the learning process. For instance, Herzog et al. (172) exploited simulation data and scripted policies to speed up the training process for off-policy RL in a waste sorting task. Another promising direction is to take a divide-and-conquer approach by sequentially chaining short-horizon interactive skills through planning (173) or hierarchical RL (159). Overall, solving long-horizon interactive tasks using DRL is an open challenge and under-explored area, but solving this type of task is necessary to create truly capable household and human-assistant robots.

4.4.4. Trends and Open Challenges in Mobile Manipulation. Thanks to the generalization of humanoids and other robot embodiments, and the advances in locomotion and stationary manipulation, DRL for MoMa is a growing field with increasing research attention. Based on our analysis, we infer some trends and open questions. First, compared to stationary manipulation, MoMa tasks have a significantly larger workspace, making safe real-world ex-

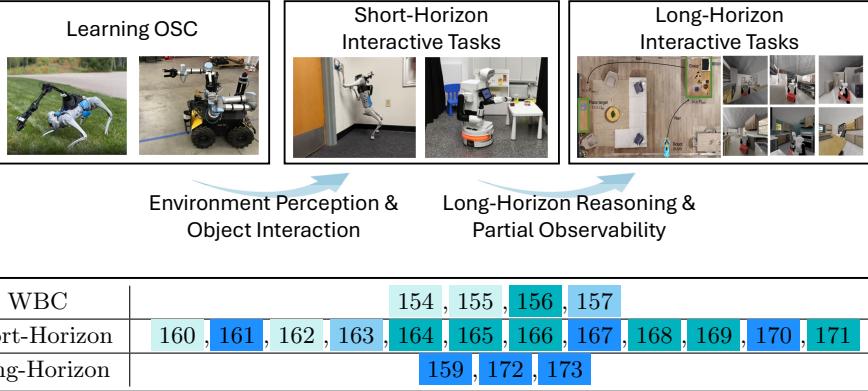


Figure 5: **Top:** An overview of the three MoMa challenges discussed in Sec. 4.4, including whole-body control (154, 156) (WBC) and short- (163, 171) and long-horizon (159, 173) interactive tasks; **Bottom:** MoMa papers reviewed in Sec. 4.4. Color map indicates levels of real-world success: *Limited Lab*, *Diverse Lab*, *Limited Real*, and *Diverse Real*.

ploration challenging. As such, existing works mainly perform training in simulations where safety is not a concern (Table 3). In the rare occurrences of real-world RL, strong domain knowledge, e.g., in the form of motion priors (170, 162) and/or demonstrations (170, 172), is used to enable safe and efficient exploration. Plus, MoMa’s large workspace demands a more sophisticated form of memory and scene representation. Representations that work well for navigation often fail to capture the dynamic characters in manipulation. While advances in sample efficiency, memory, and safe real-world RL promise new opportunities, scaling them to the open-worldness and vast workspaces inherent to MoMa remains challenging.

Second, mobile manipulators have very diverse morphologies compared to other types of robots, including wheeled robots with arms (172, 165, 169, 166, 154, 162, 163, 173, 170), quadrupeds with arms (155, 156, 161, 159), humanoids (157), and even quadrupeds using their legs for both locomotion and manipulation, i.e., loco-manipulation (160, 164, 171, 168). Each morphology brings unique challenges and opportunities. For example, wheeled mobile manipulators are easier to model and generally more kinematically stable, facilitating learning only for the manipulation component, while legged mobile manipulators can traverse uneven terrains but are harder to control, even for simple navigation phases. New research in both morphology-agnostic and morphology-specific RL methods is necessary for MoMa.

Third, perhaps due to the diverse morphologies, very diverse choices of action spaces are observed in the MoMa literature (Table 1), including direct joint control (165, 41, 169), task-space control with classical model-based (166, 163), task-space control with learned low-level controllers (171, 160, 159), and even factored actions that only controls a part of the embodiment (155, 166). Choosing the right action space is crucial for performance, as it affects the temporal abstraction levels and robot controllability. Yet, there is currently no principled way to select the appropriate action space for the diverse set of MoMa tasks.

Key Takeaways

- DRL has achieved initial success in mobile manipulation, in particular on short-horizon tasks, especially by leveraging training in simulation.

- Defining a suitable action space is critical for RL in MoMa, especially given the diversity in the morphologies of existing MoMa systems.
- The successes notwithstanding, existing methods are still insufficient for tackling multi-tasking, representing long-term memory, and performing safe exploration in the real world, providing opportunities for future improvements.

4.5. Human-Robot Interaction

In this subsection, we review works where DRL has been applied to human-robot interaction (HRI)—on robotic systems for use by or with humans. While HRI tasks can have varying objectives and involve robots with distinct morphology, the presence of humans introduces shared challenges, including safety, interpretability, and human modeling, that distinguish HRI from other robot problems not involving humans. Notice that this section focuses on robotic systems with HRI *competencies* (i.e., interact with humans *during task execution*), whereas works that only involve humans during *training* are out of the scope of this section. HRI tasks can be broadly classified into three main categories: **collaborative physical HRI (pHRI)**, where the robot and humans physically collaborate with a shared objective; **non-collaborative pHRI**, where the robot and humans share the same physical space but have distinct objectives; and **shared autonomy**, where humans act as teleoperators, and the robot autonomously interprets and executes the teleoperation command. In this section, we review works from these three categories. Figure 6 summarizes the papers reviewed.

4.5.1. Collaborative pHRI. The most intuitive type of HRI arises when a robot and a human physically collaborate toward accomplishing a shared goal—a common theme for service robots that assist humans in household activities. For example, Ghadirzadeh et al. (174) tackled the collective packaging task, where recurrent Q-learning is combined with a behavior tree to minimize the packaging time of a human worker. Christen et al. (175, 176) focused on object hand-over from a human to a robot, using RL to learn a simulated human hand-over policy and a robot policy to grasp the objects handed over by the human. Noticeably, existing works for collaborative pHRI share a similar procedure: learning a human model from pre-collected data to train a robot policy in simulation. This similarity is likely due to the high cost of collecting online interactions for collaborative tasks, which require continuous human attention and physical response to the robot’s behavior.

4.5.2. Non-collaborative pHRI. In non-collaborative pHRI tasks, a robot operates alongside humans in the same physical space but with different objectives. A representative example is social navigation where a robot navigates through crowded environments. Chen et al. (177) trained a robot for social navigation in simulation, where a hand-crafted reward is used to encourage socially compliant behavior, and zero-shot transferred the policy to a real-world corridor. Everett et al. (178) expanded on this work to incorporate human motion histories into decision-making by modeling the value network with an LSTM. Liang et al. (179) developed a high-fidelity simulator of human motions to train navigation policies taking lidar scans as inputs, and demonstrated reliable sim-to-real transfer capabilities. Hirose et al. (180) learned navigation policies alongside humans in the real world. A residual Q-function is learned on top of an offline pre-trained Q-function to generate adaptive behavior on the fly. Unlike collaborative tasks, humans do not actively participate

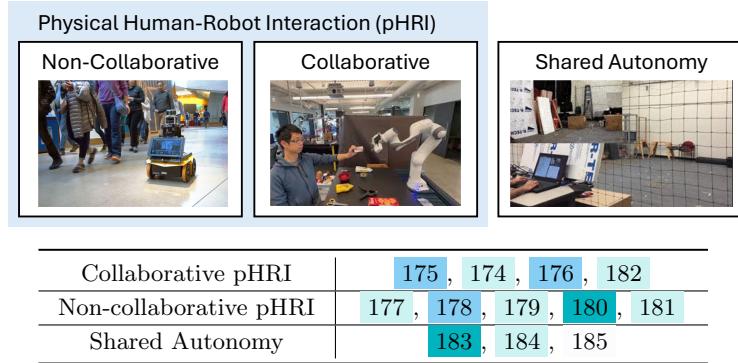


Figure 6: **Top:** An overview of the three types of HRI tasks discussed in Sec. 4.5, including collaborative (175) and non-collaborative (177) pHRI tasks, and shared autonomy (184); **Bottom:** Papers reviewed in Sec. 4.5. The color map indicates the levels of real-world success: *Sim Only* , *Limited Lab* , *Diverse Lab* , and *Limited Real* .

in the robot’s activities in non-collaborative tasks, making it easier to hard-code human behaviors (177, 178, 179) or train in the real world (180), resulting in successful real-world implementations. Aside from social navigation, Liu et al. (181) considered manipulation while avoiding collision with humans, where an action space transformation is conducted to ensure safe exploration in RL.

4.5.3. Shared Autonomy. Shared autonomy is an HRI paradigm that does not involve physical contact between humans and robots. Instead, the robot takes actions to complete tasks based on human instructions such as keyboard control or language commands. In this setting, RL can be used to learn a policy that conditions on human inputs and generates robot actions that optimize some external task rewards or constraints while aligning with the user instructions. For instance, Reddy et al. (184) tackled the quadrotor perching task, where a Q-function is learned based on task reward, and the robot chooses actions that are close to the user input and above a preset task value threshold. Schaff et al. (185) formulated shared autonomy for simulated quadrotor control as a constrained optimization problem, where a residual RL policy is learned to minimally change the human input policy while satisfying a set of task-invariant constraints. More recently, advances in NLP have opened up the possibility for shared autonomy through natural language instructions. For example, Nair et al. (183) learned a language-conditioned policy for table-top manipulation using model-based RL on a pre-collected dataset with hand-labeled language instructions.

4.5.4. Trends and Open Challenges in HRI. Despite the importance of HRI for household robot applications, RL has seen fewer successes in HRI compared to other robotics domains like locomotion and manipulation. A primary challenge for applying RL to HRI problems is properly incorporating human or human-like priors into the training process, which can often be non-markovian, have limited rationality, and are often costly to collect. Existing works have primarily tackled this challenge in three ways. First, a straightforward approach is to train the policies directly in real-world environments alongside humans. However, this approach presents significant challenges to the sample complexity of the algorithm since collecting real-world interaction data is costly, especially when humans are actively involved.

As such, works using this approach either focus on simple tasks (182) or rely on pretraining to derive a good initial policy and reduce sample complexity (180). Second, an alternative to avoid costly real-world learning is to learn a reasonable human model to simulate humans during training. This approach is particularly appealing in domains where human actions are fairly easy to model, such as shared autonomy, where a human policy can be learned by imitating a set of human actors (185, 184). In tasks where human actions are more complex, human models have been created using motion capture (174, 181), crowd-sourcing (183), and RL (175). Third, when human behaviors are simple, human models can be directly hardcoded using domain knowledge (177, 178, 179), and be incorporated either as parts of the simulation or as behavioral constraints. Although this approach is not scalable and inapplicable for many tasks, these simplified human models can serve as a useful source for pretraining to improve sample efficiency for real-world learning.

Overall, two promising future directions emerge: first, developing safe and sample-efficient RL algorithms to enable direct real-world RL, possibly by leveraging known human behavior models; second, building high-fidelity human behavior simulation to bridge sim-to-real gaps for zero-shot sim-to-real transfer. Future advances in these directions promise to broaden the application of RL to HRI problems significantly.

Key Takeaways

- Compared to other robotics domains, DRL has achieved limited success in HRI, especially on tasks that require the robot to collaborate with humans physically.
- A key challenge for applying RL to HRI lies in collecting realistic interactive experiences with humans, which can, in principle, be obtained by either directly training in the real world or by building high-fidelity human models for simulations.
- Existing works have explored both approaches in simple tasks. However, whether and how we can scale up these approaches to more difficult tasks remains unclear.

4.6. Multi-Robot Interaction

Multi-robot interaction is often solved as a MARL problem, which, in the most general case, is described using a partially observable stochastic game (POSG) with distinct reward functions and action and observation spaces, although most cooperative real-world problems model the problem as Decentralized POMDPs. We highlight three real-world domains where DRL has been successfully applied to learn multi-robot interaction: **collision avoidance and navigation**, **multi-agent loco-manipulation**, and **robot soccer**.

4.6.1. Multi-Agent Collision Avoidance. Chen et al. (186) and Everett et al. (187) model a Dec-MDP in which the policy takes the state vector consisting of positions, velocities, and radii of all the robots as input to predict the velocities for each robot. The policy is preconditioned via finetuning using ORCA (188). The reward function is sparse, consisting of a goal-reaching reward and collision penalties. These works successfully developed collision-avoidance policies in simulation and showcased hardware results on aerial and ground robots. The multicopters used onboard sensors and controllers to execute maneuvers suggested by the policy. The ground robot, equipped with affordable onboard sensors (under 1000 USD), was able to navigate through pedestrian traffic, effectively avoiding collisions despite imperfect perception and diverse pedestrian behaviors unseen during training.

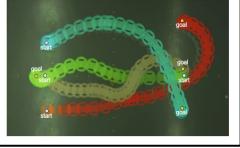
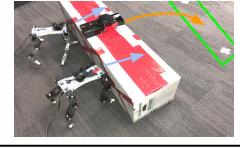
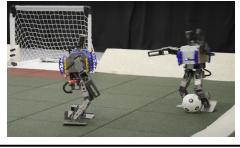
Multi-Robot Interaction Examples	
Collision Avoidance	
Multi-Robot Manipulation	
Robot Soccer	
Multi-Robot Collision Avoidance	186 , 187 , 189 , 190 , 191
Multi-Robot Loco-Manipulation	192
Robot Soccer	193

Figure 7: **Top:** An overview of the three representative multi-robot interaction domains reviewed in Sec. 4.6, including multi-robot collision avoidance (189), multi-robot manipulation via locomotion (192), and robot soccer (193); **Bottom:** Multi-robot interaction papers reviewed in Sec. 4.6. See the caption of Fig. 2 for color map description.

Other works (189, 190) have also modeled the problem as a Dec-MDP with the objective of time-to-goal minimization. These methods differ from the previous approaches in multiple respects. First, the policy takes raw lidar scans as input instead of the states of the other agents and thus does not depend on precise sensing and perception. Second, they do not precondition or finetune the policy using ORCA but instead employ curriculum learning and a dense reward function to facilitate training. Third, to deal with more complex multi-agent scenarios, it utilizes a hybrid controller to swap out the learned policy with a classical controller instead of restricting the other robots’ motion via constant linear velocity models.

Finally, Sartoretti et al. (191) used DRL to prevent agents from blocking each other in multi-agent pathfinding problems. A “blocking penalty” is applied when an agent reaches its goal but prevents another agent from doing the same. This strategy, combined with imitation learning and environment sampling, expedites convergence. The algorithm was tested on a small fleet of autonomous ground vehicles in a factory floor mock-up.

4.6.2. Multi-Agent Loco-Manipulation. We highlight a recent result (192) in multi-agent manipulation via locomotion (i.e., loco-manipulation). This involves multiple robots using movement to manipulate objects or interact with environments. Nachum et al. (192) focus on enabling multiple quadrupeds to perform complex tasks like manipulation and coordination using model-free RL. A significant challenge in applying RL to coordination or manipulation tasks with multiple legged robots is the complexity of interactions between agents or between agents and objects, which usually requires extensive real-world trial-and-error learning. To address this, this work employs a hierarchical sim2real approach demonstrating zero-shot sim-to-real transfer for object avoidance and targeted object pushing. Additionally, the work showcases a multi-agent scenario where two quadrupeds coordinate to move a heavy block to a specified location and orientation, illustrating the potential of using locomotion for coordinated multi-agent manipulation.

4.6.3. Robot Soccer. RL has also been successful in real physical soccer-playing robots in the RoboCup Standard Platform League. Many of these works focus on training a policy for a single robot, which is then transferred to multiple robots. See Sec. 4.1 and Sec. 4.4 for

discussions on these works focusing on single-robot competencies for robot soccer. A recent work (193) further applied RL to learn a variety of dynamic and complex movement skills like walking, turning, kicking, and rapid recovery from falls in *1v1 robot soccer play*. The agents learn to apply skills appropriately via self-play and showcase sophisticated multi-agent competencies such as opponent interception.

4.6.4. Trends and Open Challenges in Multi-Robot Interaction. One of the most significant challenges in multi-agent systems is managing the complexity and scalability of the systems as the number of agents increases. This challenge is evident in multi-agent manipulation via locomotion and robot soccer, where the increase in team size exponentially escalates the complexity of the interactions. The transition from controlled, simulated environments to unpredictable real-world conditions remains a formidable challenge. Although promising results have been shown in domains like collision avoidance, the variability in real-world dynamics, such as sensor inaccuracies, unexpected obstacles, and dynamic human interactions, often degrades system performance. Next, while RL has provided impressive results in learning complex behaviors autonomously, integrating these learned behaviors with classical control methods is an increasingly popular area of research. Finally, the ability of multi-robot systems to generalize across different tasks and environmental conditions presents a substantial opportunity for research.

Key Takeaways

- Current state-of-the-art in RL-based multi-robot interaction is limited to cooperative settings with identical reward functions, action spaces, and observation spaces.
- Predominantly, DRL in multi-robot settings is applied to collision avoidance among ground robots (as compared to manipulation via locomotion and robot soccer).
- Critical research areas moving forward include dealing with (*i*) communication and networking between agents, (*ii*) convergence and stability, (*iii*) scalability, (*iv*) general non-cooperative settings, (*v*) different robot morphologies and applications.

5. General Trends and Open Challenges

We conclude this survey by summarizing the patterns behind current real-world successes in robotics achieved with DRL and the characteristics of those less successful cases. Overall, more mature solutions (i.e., L3-4) have often followed the zero-shot sim-to-real transfer scheme (Table 3), which works particularly well for locomotion and navigation. The dynamics involved in these competencies, especially terrestrial locomotion and navigation, are relatively stable and easy to simulate. Dense and shaped rewards, which simplify exploration and improve sample efficiency, have also been effective (Table 2), leading to the predominant use of stable and robust model-free, on-policy algorithms in these domains (Table 5). The sim-to-real scheme has been successful for manipulation problems in which dense reward functions can be designed a priori (e.g., grasping, assembly, in-hand, non-prehensile manipulation), but less so in tasks with more diversity (e.g., pick-and-place). The community has been striving to explore alternative solutions that do not require simulation (Table 3) or reward shaping (Table 4) and adopt policy optimization algorithms with better sample efficiency (Table 5). Human demonstrations (Table 4) are effective for enabling real-world learning, particularly in manipulation tasks that are not prohibitively

complex to demonstrate. For competencies where both accurate simulation and real-world rollouts are prohibitive (e.g., HRI) or where stable, scalable RL algorithms are missing (e.g., multi-robot interaction), successful real-world examples are much sparser. In the remainder of this section, we identify several concrete open challenges that are opportunities for further extending DRL’s applications, in particular for those currently less successful domains.

Improving Stability and Sample-Efficiency in RL Algorithms. While on-policy RL methods are often preferred due to their robustness to hyperparameters, collecting large amounts of on-policy data can be prohibitive, especially for real-world RL. Even in the predominant zero-shot sim-to-real setting, the sample efficiency of on-policy RL is problematic for tasks such as long-horizon mobile manipulation (172, 173) and agile legged navigation (20, 96), where the long task horizons, large operational spaces, sparse rewards, and complex contact dynamics hinder efficient exploration and stable learning. Sample efficiency can also be a crucial issue in problems with temporally extended action spaces (32, 36). Fundamental algorithmic advances to develop RL algorithms that are at least as robust but more sample-efficient than on-policy methods are thus crucial for expanding RL’s applications in robotics. An appealing direction is leveraging off-policy or offline samples to complement or replace on-policy exploration. However, off-policy and offline RL are often less stable due to the distributional shift between behavioral and learning policy experiences. Promising efforts have been made to derive scalable and more stable off-policy (110) and offline RL algorithms (124) for manipulation and MoMa (172). Fine-tuning offline learned policies with online updates can further enhance performance in an efficient manner (48, 122). However, stable online fine-tuning is non-trivial, especially for value-based RL (194, 195). Combining model-free and model-based approaches is another promising direction to derive sample-efficient RL algorithms (196). Lastly, these advances have primarily focused on single-robot problems. Multi-robot problems present greater challenges as the complexity of multi-robot interaction escalates exponentially with the number of robots. The scalability and stability of MARL remain open questions that hinder RL’s application for multi-robot interaction.

Real-World Learning. In our analysis of RL for robot competencies (Sec. 4), real-world learning was often mentioned as one of the open challenges. A learning process carried out in the real world is crucial for robotic problems where the zero-shot sim-to-real transfer procedure is impractical due to the lack of high-fidelity simulation, such as open-world and contact-rich manipulation, lightweight quadrotor navigation, and physical HRI. Although some progress has been made, particularly for manipulation (Table 3), successful real-world learning examples are much rarer than zero-shot sim-to-real transfer, presenting exciting opportunities for future research. Two main issues need to be addressed for real-world RL learning. The first issue is *how to collect many useful experiences in a safe manner?* In domains where oracle policies, like humans (124) and scripts (172), are available, demonstrations can be collected for offline learning. However, offline RL faces challenges such as distributional shifts, and the demonstration data can be suboptimal and costly to collect for human experts. Real-world rollouts require automatic resets (120, 117, 53) and safe exploration mechanisms (170) to minimize human effort and ensure safety. Such mechanisms are still missing in most problem domains and present an opportunity for future development, especially for safety-critical applications (92, 102). To date, human-in-the-loop learning (for resets and safety) is currently the only alternative (92), leaving automated real-world learning a desirable future capability. In addition to procedural and algorithmic improve-

ments, safe real-world exploration may also be facilitated through hardware advances, such as adaptive and less fragile hardware and mechanisms that ensure safety passively (197). The second issue is *how do we accelerate training to require fewer experiences?* A promising avenue is to explore what modules can be updated with real-world samples and how. Instead of updating the entire policy with model-free RL, some solutions explore adapting vision encoders (43) or learning (residual) dynamics models (7, 102, 56) from real-world samples. These alternatives improve efficiency; we predict future successful real-world training procedures exploring alternative combinations of frozen-trainable modules.

Learning for Long-Horizon Robotic Tasks. Long-horizon tasks pose a fundamental challenge to RL algorithms, requiring directed exploration and temporal credit assignment over long stretches of time. Many such real-world tasks require integrating diverse abilities. By contrast, the vast majority of the RL successes we have reviewed are in short-horizon problems, e.g., controlling a quadruped to walk at a given velocity or controlling a manipulator to rotate an object in hand. A promising avenue for solving long-horizon tasks is learning *skills* and composing them, enabling compositional generalization. This approach has seen success in navigation (96, 97), manipulation (11, 115, 132, 152), and MoMa (159, 173). A critical question for future work is: *what skills should the robot learn?*. While some successes have been achieved with manually specified skills and reward functions (32, 50, 96, 123, 152, 114), these approaches heavily rely on domain knowledge. Some efforts have been made to explore unified reward designs for learning multi-skill locomotion policies (42, 51, 49). Formulating skill learning as goal-conditioned (125) or unsupervised RL (198, 199) is promising for more general problems. A second critical question is: *how should these skills be combined to solve long-horizon tasks?* Various designs have been explored, including hierarchical RL (32, 159), end-to-end training (123, 49), and planning (132, 152, 173). This question will also be central to integrating various competencies toward general-purpose robots; recent advances along this line have opened up exciting possibilities, including wheel-legged navigation (97) and loco-manipulation (51, 160, 164, 171, 168).

Designing Principled Approaches for RL Systems. For each robotic task, an RL practitioner must choose among the many alternatives that will define its RL system, both in the problem formulation and solution space (see Table 1–6). Many of these choices are made based on expert knowledge and heuristics, which are not necessarily optimal and can even harm performance (200, 201). Principled approaches for RL system design, relying less on heuristics and manual efforts, will be essential in the future for scalable development and deployment, especially for open-world tasks. Here, we note some particularly important examples. First, many real-world successes have been achieved with dense and shaped rewards designed with heavy engineering efforts, particularly in locomotion and navigation (Table 2). Efforts are being made to explore principled reward designs for specific competencies (42, 51, 49) and more general problems using goal-conditioned (125) or unsupervised RL (198, 199). Second, various action spaces are used, particularly for manipulation and MoMa (Table 1). The action space choices affect the temporal abstraction levels and robustness of the RL policies. Some studies have attempted to benchmark different action spaces (66, 86, 201), but such principled studies and guidelines are still lacking for many problems. Another related design choice is the integration of RL with classical planning and control modules. The different levels of integration result in different action spaces for the RL policies (i.e., low-, mid-, and high-level). The effectiveness of end-to-end versus

hybrid modular solutions varies by problem (82, 86, 100, 21, 202). Neither approach is universally superior. There are many other dimensions that require such principled investigations, which are crucial for advancing DRL’s real-world success, in addition to exploring new frontiers in algorithms and applications.

Benchmarking Real-World Success. In this survey, we classify papers into six levels of real-world success to assess the maturity of DRL-based solutions. However, precisely determining these levels can be challenging since the only source of information is the experimental results reported by the authors, but the varying testing conditions and evaluation metrics make direct comparison difficult. This highlights the need for *standard evaluation protocols and benchmarks for real-world performance*. While widely adopted, low-cost hardware, as seen with quadrupeds, is helpful by enabling standardized experimental platforms, it is not sufficient alone. Test environments and tasks must also resemble real-world conditions and, more importantly, be *reproducible*. Multiple real-world benchmarks have been established, including those for manipulation (203, 204) and domestic service robots (205). However, when it comes to complex open-world problems, the evaluation procedure must also scale up to be realistic and informative (206). Overall, developing scalable evaluation protocols and benchmarks remains an exciting open research direction for many problems.

Leveraging Foundation Models. Lastly, recent advances in large-scale robot dataset (207, 208) and robot foundation models (209, 210) present exciting open opportunities for RL successes in the real world. Foundation models have demonstrated impressive generalization capabilities across domains for reasoning and decision-making tasks (211), showing promise for addressing several of the aforementioned challenges of DRL for robotics. For instance, the recently introduced DrEureka (212) algorithm leverages large language models (LLMs) to automate reward design and domain randomization configuration for sim-to-real transfer without manual tuning. In addition, LLMs and vision-language models (VLMs) open up new opportunities to create language-conditioned RL policies for novel applications. We refer readers to existing surveys for detailed discussions on the opportunities foundation models offer in general (209, 210), but we anticipate an increased integration of foundation models into RL solutions for real-world robotic tasks.

6. Conclusion

Deep reinforcement learning has recently played an important role in the development of many robotic capabilities, leading to many real-world successes. Here, we have reviewed and categorized these successes, delineating them based on the specific robotic competency, problem formulation, and solution approach. Our analysis across these axes has revealed general trends and important avenues for future work, including algorithmic and procedural improvements, ingredients for real-world learning, and holistic approaches toward synthesizing all the competencies discussed herein. Harnessing RL’s power to produce capable real-world robotic systems will require solving fundamental challenges and innovations in its application; nonetheless, we expect that RL will continue to play a central role in the development of generally intelligent robots.

Acknowledgements

We thank Pieter Abbeel, Yuchen Cui, Shivin Dass, George Konidaris, Jan Peters, Eric Rosen, Koushil Sreenath, Eugene Vinitsky, and Zhaoming Xie for their feedback on the manuscript. We also thank Google DeepMind for permission to use representative images from their work on robot soccer. A portion of this work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory at the University of Texas at Austin. LARG research is supported in part by the National Science Foundation (FAIN-2019844, NRT-2125858), the Office of Naval Research (N00014-18-2243), Army Research Office (E2061621), Bosch, Lockheed Martin, and Good Systems, a research grand challenge at the University of Texas at Austin. The views and conclusions contained in this document are those of the authors alone. Peter Stone serves as the Chief Scientist of Sony AI and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

LITERATURE CITED

1. Sutton RS, Barto AG. 2018. *Reinforcement learning: An introduction*. MIT press, 2nd ed.
2. François-Lavet V, Henderson P, Islam R, Bellemare MG, Pineau J, et al. 2018. An introduction to deep reinforcement learning. *Found. Trends in Mach. Learn.* 11(3-4):219–354
3. Schrittwieser J, Antonoglou I, Hubert T, Simonyan K, Sifre L, et al. 2020. Mastering atari, go, chess and shogi by planning with a learned model. *Nature* 588(7839):604–609
4. Wurman PR, Barrett S, Kawamoto K, MacGlashan J, Subramanian K, et al. 2022. Outracing champion gran turismo drivers with deep reinforcement learning. *Nature* 602(7896):223–228
5. Yu C, Liu J, Nemati S, Yin G. 2021. Reinforcement learning in healthcare: A survey. *ACM Comput. Surv.* 55(1):1–36
6. Afsar MM, Crump T, Far B. 2022. Reinforcement learning based recommender systems: A survey. *ACM Comput. Surv.* 55(7):1–38
7. Kaufmann E, Bauersfeld L, Loquercio A, Müller M, Koltun V, Scaramuzza D. 2023. Champion-level drone racing using deep reinforcement learning. *Nature* 620(7976):982–987
8. Kiran BR, Sobh I, Talpaert V, Mannion P, Al Sallab AA, et al. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Trans. Intell. Transp. Syst.* 23(6):4909–4926
9. Dulac-Arnold G, Levine N, Mankowitz DJ, Li J, Paduraru C, et al. 2021. Challenges of real-world reinforcement learning: definitions, benchmarks, and analysis. *Mach. Learn.* 110(9):2419–2468
10. Ibarz J, Tan J, Finn C, Kalakrishnan M, Pastor P, Levine S. 2021. How to train your robot with deep reinforcement learning: lessons we have learned. *Int. J. Robot. Res.* 40(4-5):698–721
11. Kroemer O, Niekum S, Konidaris G. 2021. A review of robot learning for manipulation: Challenges, representations, and algorithms. *J. Mach. Learn. Res.* 22(30):1–82
12. Xiao X, Liu B, Warnell G, Stone P. 2022. Motion planning and control for mobile robot navigation using machine learning: a survey. *Auton. Robots* 46(5):569–597
13. Deisenroth MP. 2011. A survey on policy search for robotics. *Found. Trends Robot.* 2(1-2):1–142
14. Brunke L, Greeff M, Hall AW, Yuan Z, Zhou S, et al. 2022. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning. *Annu. Rev. Control Robot. Auton. Syst.* 5(1):411–444.eprint: <https://doi.org/10.1146/annurev-control-042920-020211>
15. Kober J, Bagnell JA, Peters J. 2013. Reinforcement learning in robotics: A survey. *Int. J. Robot. Res.* 32(11):1238–1274

16. Sünderhauf N, Brock O, Scheirer W, Hadsell R, Fox D, et al. 2018. The limits and potentials of deep learning for robotics. *Int. J. Robot. Res.* 37(4-5):405–420
17. Mason MT. 2001. *Mechanics of robotic manipulation*. MIT press
18. Siciliano B, Khatib O, Kröger T. 2008. *Springer handbook of robotics*, vol. 200. Springer
19. Mason MT. 2018. Toward robotic manipulation. *Annu. Rev. Control Robot. Auton. Syst.* 1:1–28
20. Rudin N, Hoeller D, Bjelonic M, Hutter M. 2022. *Advanced skills by learning locomotion and local navigation end-to-end*. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 2497–2503. IEEE
21. Song Y, Romero A, Müller M, Koltun V, Scaramuzza D. 2023. Reaching the limit in autonomous racing: Optimal control versus reinforcement learning. *Sci. Robot.* 8(82):eadg1462
22. On-Road Automated Driving (ORAD) committee. 2018. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. Tech. rep., SAE International
23. Lavin A, Gilligan-Lee CM, Visnjic A, Ganju S, Newman D, et al. 2022. Technology readiness levels for machine learning systems. *Nat. Commun.* 13(1):6039
24. Kohl N, Stone P. 2004. *Policy gradient reinforcement learning for fast quadrupedal locomotion*. In *IEEE Int. Conf. Robot. Autom.*, vol. 3, pp. 2619–2624. IEEE
25. Bagnell JA, Schneider JG. 2001. *Autonomous helicopter control using reinforcement learning policy search methods*. In *IEEE Int. Conf. Robot. Autom.*, vol. 2, pp. 1615–1620. IEEE
26. Abbeel P, Coates A, Quigley M, Ng A. 2006. *An application of reinforcement learning to aerobatic helicopter flight*. In *Adv. Neural Inf. Process. Syst.*, vol. 19
27. Kumar A, Li Z, Zeng J, Pathak D, Sreenath K, Malik J. 2022. *Adapting rapid motor adaptation for bipedal robots*. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 1161–1168. IEEE
28. Tan J, Zhang T, Coumans E, Iscen A, Bai Y, et al. 2018. *Sim-to-real: Learning agile locomotion for quadruped robots*. In *Robot. Sci. Syst.*
29. Hwangbo J, Lee J, Dosovitskiy A, Bellicoso D, Tsounis V, et al. 2019. Learning agile and dynamic motor skills for legged robots. *Sci. Robot.* 4(26):eaau5872
30. Feng G, Zhang H, Li Z, Peng XB, Basireddy B, et al. 2023. *Genloco: Generalized locomotion controllers for quadrupedal robots*. In *Conference on Robot Learning*, pp. 1893–1903. PMLR
31. Lee J, Hwangbo J, Hutter M. 2019. Robust recovery controller for a quadrupedal robot using deep reinforcement learning. *arXiv preprint arXiv:1901.07517*
32. Yang C, Yuan K, Zhu Q, Yu W, Li Z. 2020. Multi-expert learning of adaptive legged locomotion. *Sci. Robot.* 5(49):eabb2174
33. Kumar A, Fu Z, Pathak D, Malik J. 2021. *RMA: Rapid motor adaptation for legged robots*. In *Robot. Sci. Syst.*
34. Lee J, Hwangbo J, Wellhausen L, Koltun V, Hutter M. 2020. Learning quadrupedal locomotion over challenging terrain. *Sci. Robot.* 5(47):eabc5986
35. Miki T, Lee J, Hwangbo J, Wellhausen L, Koltun V, Hutter M. 2022. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Sci. Robot.* 7(62):eabk2822
36. Gangapurwala S, Geisert M, Orsolino R, Fallon M, Havoutis I. 2022. RLOC: Terrain-aware legged locomotion using reinforcement learning and optimal control. *IEEE Trans. Robot.* 38(5):2908–2927
37. Choi S, Ji G, Park J, Kim H, Mun J, et al. 2023. Learning quadrupedal locomotion on deformable terrain. *Sci. Robot.* 8(74):eade2256
38. Nahrendra IMA, Yu B, Myung H. 2023. *DreamWaQ: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning*. In *IEEE Int. Conf. Robot. Autom.*, pp. 5078–5084. IEEE
39. Pinto L, Andrychowicz M, Welinder P, Zaremba W, Abbeel P. 2018. *Asymmetric actor critic for image-based robot learning*. In *Robot. Sci. Syst.*
40. Escontrela A, Peng XB, Yu W, Zhang T, Iscen A, et al. 2022. *Adversarial motion priors make good substitutes for complex reward functions*. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 25–32. IEEE

41. Ma Y, Farshidian F, Hutter M. 2023. *Learning arm-assisted fall damage reduction and recovery for legged mobile manipulators*. In *IEEE Int. Conf. Robot. Autom.*, pp. 12149–12155. IEEE
42. Fu Z, Kumar A, Malik J, Pathak D. 2022. *Minimizing energy consumption leads to the emergence of gaits in legged robots*. In *Conf. Robot Learn.*, pp. 928–937. PMLR
43. Loquercio A, Kumar A, Malik J. 2023. *Learning visual locomotion with cross-modal supervision*. In *IEEE Int. Conf. Robot. Autom.*, pp. 7295–7302. IEEE
44. Agarwal A, Kumar A, Malik J, Pathak D. 2023. *Legged locomotion in challenging terrains using egocentric vision*. In *Conf. Robot Learn.*, pp. 403–415. PMLR
45. Yang R, Yang G, Wang X. 2023. *Neural volumetric memory for visual locomotion control*. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 1430–1440. PMLR
46. Jenelten F, He J, Farshidian F, Hutter M. 2024. DTC: Deep tracking control. *Sci. Robot.* 9(86):eadh5401
47. Yang Y, Shi G, Meng X, Yu W, Zhang T, et al. 2023. *CAJun: Continuous adaptive jumping using a learned centroidal controller*. In *Conf. Robot. Learn.* PMLR
48. Smith L, Kew JC, Peng XB, Ha S, Tan J, Levine S. 2022. *Legged robots that keep on learning: Fine-tuning locomotion policies in the real world*. In *IEEE Int. Conf. Robot. Autom.*, pp. 1593–1599. IEEE
49. Cheng X, Shi K, Agarwal A, Pathak D. 2024. *Extreme parkour with legged robots*. In *IEEE Int. Conf. Robot. Autom.* IEEE
50. Zhuang Z, Fu Z, Wang J, Atkeson CG, Schwertfeger S, et al. 2023. *Robot parkour learning*. In *Conf. Robot. Learn.* PMLR
51. Vollenweider E, Bjelonic M, Klemm V, Rudin N, Lee J, Hutter M. 2023. *Advanced skills through multiple adversarial motion priors in reinforcement learning*. In *IEEE Int. Conf. Robot. Autom.*, pp. 5120–5126. IEEE
52. Margolis GB, Agrawal P. 2023. *Walk these ways: Tuning robot control for generalization with multiplicity of behavior*. In *Conf. Robot. Learn.*, pp. 22–31. PMLR
53. Smith L, Kostrikov I, Levine S. 2023. Demonstrating a walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning. *Robot. Sci. Syst.* 2(3):4
54. Wu P, Escontrela A, Hafner D, Abbeel P, Goldberg K. 2023. *Daydreamer: World models for physical robot learning*. In *Conf. Robot Learn.*, pp. 2226–2240. PMLR
55. Siekmann J, Valluri S, Dao J, Bermillo L, Duan H, et al. 2020. *Learning memory-based control for human-scale bipedal locomotion*. In *Robot. Sci. Syst.*
56. Hanna JP, Desai S, Karnan H, Warnell G, Stone P. 2021. Grounded action transformation for sim-to-real reinforcement learning. *Mach. Learn.* 110(9):2469–2499
57. Siekmann J, Godse Y, Fern A, Hurst J. 2021. *Sim-to-real learning of all common bipedal gaits via periodic reward composition*. In *IEEE Int. Conf. Robot. Autom.*, pp. 7309–7315. IEEE
58. Li Z, Cheng X, Peng XB, Abbeel P, Levine S, et al. 2021. *Reinforcement learning for robust parameterized locomotion control of bipedal robots*. In *IEEE Int. Conf. Robot. Autom.*, pp. 2811–2817. IEEE
59. Siekmann J, Green K, Warila J, Fern A, Hurst J. 2021. *Blind Bipedal Stair Traversal via Sim-to-Real Reinforcement Learning*. In *Robot. Sci. Syst.*
60. Castillo GA, Weng B, Zhang W, Hereid A. 2022. Reinforcement learning-based cascade motion policy design for robust 3d bipedal locomotion. *IEEE Access* 10:20135–20148
61. Duan H, Pandit B, Gadde MS, van Marum BJ, Dao J, et al. 2024. *Learning vision-based bipedal locomotion for challenging terrain*. In *IEEE Int. Conf. Robot. Autom.*
62. Radosavovic I, Xiao T, Zhang B, Darrell T, Malik J, Sreenath K. 2024. Real-world humanoid locomotion with reinforcement learning. *Sci. Robot.* 9(89):eadi9579
63. Li Z, Peng XB, Abbeel P, Levine S, Berseth G, Sreenath K. 2024. Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control. *arXiv preprint arXiv:2401.16889*
64. Hwangbo J, Sa I, Siegwart R, Hutter M. 2017. Control of a quadrotor with reinforcement learning. *IEEE Robot. Autom. Lett.* 2(4):2096–2103

65. Molchanov A, Chen T, Hönig W, Preiss JA, Ayanian N, Sukhatme GS. 2019. *Sim-to-(multi)-real: Transfer of low-level robust control policies to multiple quadrotors*. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 59–66. IEEE
66. Kaufmann E, Bauersfeld L, Scaramuzza D. 2022. *A benchmark comparison of learned control policies for agile quadrotor flight*. In *Int. Conf. Robot. Autom.*, pp. 10504–10510. IEEE
67. Zhang D, Loquercio A, Wu X, Kumar A, Malik J, Mueller MW. 2023. *Learning a single near-hover position controller for vastly different quadcopters*. In *IEEE Int. Conf. Robot. Autom.*, pp. 1263–1269. IEEE
68. Eschmann J, Albani D, Loianno G. 2024. Learning to fly in seconds. *IEEE Robot. Autom. Lett.* 9(7):6336–6343
69. Yang R, Zhang M, Hansen N, Xu H, Wang X. 2021. *Learning vision-Guided quadrupedal locomotion end-to-end with cross-modal transformers*. In *Int. Conf. Learn. Represent.*
70. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*
71. Grizzle JW, Hurst J, Morris B, Park HW, Sreenath K. 2009. *MABEL, a new robotic bipedal walker and runner*. In *Am. Control Conf.*, pp. 2030–2036. IEEE
72. Loquercio A, Kaufmann E, Ranftl R, Müller M, Koltun V, Scaramuzza D. 2021. Learning high-speed flight in the wild. *Sci. Robot.* 6(59):eabg5810
73. Tai L, Paolo G, Liu M. 2017. *Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation*. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 31–36
74. Xu Z, Liu B, Xiao X, Nair A, Stone P. 2023. *Benchmarking Reinforcement Learning Techniques for Autonomous Navigation*. In *IEEE Int. Conf. Robot. Autom.*, pp. 9224–9230
75. Chiang HTL, Faust A, Fiser M, Francis A. 2019. Learning navigation behaviors end-to-end with autorl. *IEEE Robot. Autom. Lett.* 4(2):2007–2014
76. Stein GJ, Bradley C, Roy N. 2018. *Learning over subgoals for efficient navigation of structured, unknown environments*. In *Conf. Robot Learn.*, pp. 213–222. PMLR
77. Anderson P, Wu Q, Teney D, Bruce J, Johnson M, et al. 2018. *Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments*. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 3674–3683
78. Zhu Y, Mottaghi R, Kolve E, Lim JJ, Gupta A, et al. 2017. *Target-driven visual navigation in indoor scenes using deep reinforcement learning*. In *IEEE Int. Conf. Robot. Autom.*, pp. 3357–3364
79. Kahn G, Villaflor A, Ding B, Abbeel P, Levine S. 2018. *Self-Supervised Deep Reinforcement Learning with Generalized Computation Graphs for Robot Navigation*. In *IEEE Int. Conf. Robot. Autom.*, pp. 5129–5136. IEEE
80. Wijmans E, Kadian A, Morcos A, Lee S, Essa I, et al. 2020. DD-PPO: Learning Near-Perfect PointGoal Navigators from 2.5 Billion Frames. *ArXiv:1911.00357 [cs]*
81. Chaplot DS, Gandhi DP, Gupta A, Salakhutdinov RR. 2020. Object goal navigation using goal-oriented semantic exploration. *Adv. Neural Inf. Process. Syst.* 33:4247–4258
82. Gervet T, Chintala S, Batra D, Malik J, Chaplot DS. 2023. Navigating to objects in the real world. *Sci. Robot.* 8(79)
83. Hoeller D, Wellhausen L, Farshidian F, Hutter M. 2021. Learning a State Representation and Navigation in Cluttered and Dynamic Environments. *IEEE Robot. Autom. Lett.* 6(3):5081–88
84. Savva M, Kadian A, Maksymets O, Zhao Y, Wijmans E, et al. 2019. *Habitat: A platform for embodied ai research*. In *IEEE/CVF Int. Conf. Comput. Vis.*, pp. 9339–9347
85. Kadian A, Truong J, Gokaslan A, Clegg A, Wijmans E, et al. 2020. Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robot. Autom. Lett.* 5(4):6670–6677
86. Truong J, Rudolph M, Yokoyama NH, Chernova S, Batra D, Rai A. 2023. *Rethinking sim2real: Lower fidelity simulation leads to higher sim2real transfer in navigation*. In *Conf. Robot.*

- Learn.*, pp. 859–70. PMLR
87. Truong J, Zitkovich A, Chernova S, Batra D, Zhang T, et al. 2024. Indoorsim-to-outdoorreal: learning to navigate outdoors without any outdoor experience. *IEEE Robot. Autom. Lett.*
 88. Kahn G, Abbeel P, Levine S. 2021. BADGR: An Autonomous Self-Supervised Learning-Based Navigation System. *IEEE Robot. Autom. Lett.* 6(2):1312–1319
 89. Shah D, Bhorkar A, Leen H, Kostrikov I, Rhinehart N, Levine S. 2023. *Offline Reinforcement Learning for Visual Navigation*. In *Conf. Robot Learn.*, pp. 44–54. PMLR
 90. Williams G, Wagener N, Goldfain B, Drews P, Rehg JM, et al. 2017. *Information theoretic mpc for model-based reinforcement learning*. In *IEEE Int. Conf. Robot. Autom.*, pp. 1714–1721
 91. Stachowicz K, Shah D, Bhorkar A, Kostrikov I, Levine S. 2023. *FastRLAP: A System for Learning High-Speed Driving via Deep RL and Autonomous Practicing*. In *Conf. Robot Learn.*, pp. 3100–3111. PMLR
 92. Kendall A, Hawke J, Janz D, Mazur P, Reda D, et al. 2019. *Learning to drive in a day*. In *Int. Conf. Robot. Autom.*, pp. 8248–8254. IEEE
 93. Jang K, Lichtlé N, Vinitsky E, Shah A, Bunting M, et al. 2024. Reinforcement learning based oscillation dampening: Scaling up single-agent rl algorithms to a 100 av highway field operational test. *arXiv preprint arXiv:2402.17050*
 94. Sorokin M, Tan J, Liu CK, Ha S. 2022. Learning to navigate sidewalks in outdoor environments. *IEEE Robot. Autom. Lett.* 7(2):3906–13
 95. Zhang C, Jin J, Frey J, Rudin N, Mattamala Aravena ME, et al. 2024. *Resilient legged local navigation: Learning to traverse with compromised perception end-to-end*. In *IEEE Int. Conf. Robot. Autom.*
 96. Hoeller D, Rudin N, Sako D, Hutter M. 2024. Anymal parkour: Learning agile navigation for quadrupedal robots. *Sci. Robot.* 9(88):eadi7566
 97. Lee J, Bjelonic M, Reske A, Wellhausen L, Miki T, Hutter M. 2024. Learning robust autonomous navigation and locomotion for wheeled-legged robots. *Sci. Robot.* 9(89):eadi9641
 98. Miki T, Lee J, Wellhausen L, Hutter M. 2024. *Learning to walk in confined spaces using 3d representation*. In *Int. Conf. Robot. Autom.*
 99. Xu Z, Raj AH, Xiao X, Stone P. 2024. *Dexterous Legged Locomotion in Confined 3D Spaces with Reinforcement Learning*. In *Int. Conf. Robot. Autom.*
 100. He T, Zhang C, Xiao W, He G, Liu C, Shi G. 2024. Agile but safe: Learning collision-free high-speed legged locomotion. *Robot.: Sci. Syst.*
 101. Sadeghi F, Levine S. 2017. Cad2rl: Real single-image flight without a single real image. *Robot.: Sci. Syst.*
 102. Kang K, Belkhale S, Kahn G, Abbeel P, Levine S. 2019. *Generalization through simulation: Integrating simulated and real data into deep reinforcement learning for vision-based autonomous flight*. In *Int. Conf. Robot. Autom.*, pp. 6008–6014. IEEE
 103. Romero A, Song Y, Scaramuzza D. 2024. *Actor-critic model predictive control*. In *Int. Conf. Robot. Autom.*
 104. Xie L, Wang S, Markham A, Trigoni N. 2017. Towards monocular vision based obstacle avoidance through deep reinforcement learning. *arXiv preprint arXiv:1706.09829*
 105. Wijmans E, Kadian A, Morcos A, Lee S, Essa I, et al. 2020. *DD-PPO: Learning Near-Perfect PointGoal Navigators from 2.5 Billion Frames*. In *Int. Conf. Learn. Represent.*
 106. Wijmans E, Savva M, Essa I, Lee S, Morcos AS, Batra D. 2023. *Emergence of Maps in the Memories of Blind Navigation Agents*. In *11th Int. Conf. Learn. Represent.*
 107. Rosinol A, Leonard JJ, Carlone L. 2023. *Nerf-slam: Real-time dense monocular slam with neural radiance fields*. In *2023 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 3437–3444
 108. Mahler J, Matl M, Satish V, Danielczuk M, DeRose B, et al. 2019. Learning ambidextrous robot grasping policies. *Sci. Robot.* 4(26):eaau4984
 109. Zeng A, Song S, Welker S, Lee J, Rodriguez A, Funkhouser T. 2018. *Learning synergies between pushing and grasping with self-supervised deep reinforcement learning*. In *IEEE/RSJ*

- Int. Conf. Intell. Robots Syst.*, pp. 4238–4245. IEEE
110. Kalashnikov D, Irpan A, Pastor P, Ibarz J, Herzog A, et al. 2018. *Scalable deep reinforcement learning for vision-based robotic manipulation*. In *Conf. Robot. Learn.*, pp. 651–73
 111. James S, Wohlhart P, Kalakrishnan M, Kalashnikov D, Irpan A, et al. 2019. *Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks*. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 12627–12637
 112. Wang D, Jia M, Zhu X, Walters R, Platt R. 2023. *On-Robot Learning With Equivariant Models*. In *Conf. on Robot Learn.*, pp. 1345–1354. PMLR
 113. Levine S, Finn C, Darrell T, Abbeel P. 2016. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.* 17(1):1334–1373
 114. Kalashnikov D, Varley J, Chebotar Y, Swanson B, Jonschkowski R, et al. 2022. *Scaling up multi-task robotic reinforcement learning*. In *Conf. Robot Learn.*, pp. 557–575. PMLR
 115. Chebotar Y, Hausman K, Lu Y, Xiao T, Kalashnikov D, et al. 2021. *Actionable Models: Unsupervised Offline Reinforcement Learning of Robotic Skills*. In *Int. Conf. Mach. Learn.*, pp. 1518–1528. PMLR
 116. Lee AX, Devin CM, Zhou Y, Lampe T, Bousmalis K, et al. 2021. *Beyond pick-and-place: Tackling robotic stacking of diverse shapes*. In *Conf. on Robot Learn.*
 117. Walke HR, Yang JH, Yu A, Kumar A, Orbik J, et al. 2023. *Dont start from scratch: Leveraging prior data to automate robotic reinforcement learning*. In *Conf. Robot Learn.*, pp. 1652–1662
 118. Ebert F, Finn C, Dasari S, Xie A, Lee A, Levine S. 2018. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*
 119. Riedmiller M, Hafner R, Lampe T, Neunert M, Degrave J, et al. 2018. *Learning by Playing Solving Sparse Reward Tasks from Scratch*. In *Int. Conf. Mach. Learn.*, pp. 4344–4353. PMLR
 120. Zhu H, Yu J, Gupta A, Shah D, Hartikainen K, et al. 2020. *The Ingredients of Real World Robotic Reinforcement Learning*. In *Int. Conf. Learn. Represent.*
 121. Ma YJ, Sodhani S, Jayaraman D, Bastani O, Kumar V, Zhang A. 2022. *VIP: Towards Universal Visual Reward and Representation via Value-Implicit Pre-Training*. In *Int. Conf. Learn. Represent.*
 122. Nair A, Gupta A, Dalal M, Levine S. 2020. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*
 123. Nasiriany S, Liu H, Zhu Y. 2022. *Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks*. In *Int. Conf. Robot. Autom.*, pp. 7477–7484. IEEE
 124. Chebotar Y, Vuong Q, Hausman K, Xia F, Lu Y, et al. 2023. *Q-Transformer: Scalable offline reinforcement learning via autoregressive Q-functions*. In *Conf. on Robot Learn.*, pp. 3909–3928. PMLR
 125. Nair AV, Pong V, Dalal M, Bahl S, Lin S, Levine S. 2018. *Visual reinforcement learning with imagined goals*. In *Adv. Neural Inf. Process. Syst.*, vol. 31
 126. Johannink T, Bahl S, Nair A, Luo J, Kumar A, et al. 2019. *Residual reinforcement learning for robot control*. In *Int. Conf. Robot. Autom.*, pp. 6023–6029
 127. Vecerik M, Hester T, Scholz J, Wang F, Pietquin O, et al. 2017. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*
 128. Luo J, Sushkov O, Pevceviciute R, Lian W, Su C, et al. 2021. *Robust multi-modal policies for industrial assembly via reinforcement learning and demonstrations: A large-scale study*. In *Robot.: Sci. Syst.*
 129. Zhao TZ, Luo J, Sushkov O, Pevceviciute R, Heess N, et al. 2022. *Offline meta-reinforcement learning for industrial insertion*. In *IEEE Int. Conf. Robot. Autom.*, pp. 6386–6393
 130. Tang B, Lin MA, Akinola I, Handa A, Sukhatme GS, et al. 2023. *IndustReal: Transferring contact-rich assembly tasks from simulation to reality*. In *Robot.: Sci. and Sys.*
 131. Chebotar Y, Handa A, Makoviychuk V, Macklin M, Issac J, et al. 2019. *Closing the sim-to-real loop: Adapting simulation randomization with real world experience*. In *IEEE Int. Conf.*

- Robot. Autom.*, pp. 8973–8979. IEEE
132. Abbatematteo B, Rosen E, Thompson S, Akbulut T, Rammohan S, Konidaris G. 2024. *Composable interaction primitives: A structured policy class for efficiently learning sustained-contact manipulation skills*. In *IEEE Int. Conf. Robot. Autom.* IEEE
 133. Wu R, Zhao Y, Mo K, Guo Z, Wang Y, et al. 2022. *VAT-Mart: Learning visual action trajectory proposals for manipulating 3D articulated objects*. In *Int. Conf. Learn. Represent.*
 134. Matas J, James S, Davison AJ. 2018. *Sim-to-real reinforcement learning for deformable object manipulation*. In *Conf. on Robot Learn.*, pp. 734–743
 135. Wu Y, Yan W, Kurutach T, Pinto L, Abbeel P. 2020. *Learning to manipulate deformable objects without demonstrations*. In *Robot.: Sci. Syst.*
 136. Avigal Y, Berscheid L, Asfour T, Kröger T, Goldberg K. 2022. *Speedfolding: Learning efficient bimanual folding of garments*. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 1–8. IEEE
 137. Wang Y, Sun Z, Erickson Z, Held D. 2023. *One policy to dress them all: Learning to dress people with diverse poses and garments*. In *Robot.: Sci. and Syst.*
 138. Andrychowicz OM, Baker B, Chociej M, Jozefowicz R, McGrew B, et al. 2020. Learning dexterous in-hand manipulation. *Int. J. Robot. Res.* 39(1):3–20
 139. Handa A, Allshire A, Makoviychuk V, Petrenko A, Singh R, et al. 2023. *Dextreme: Transfer of agile in-hand manipulation from simulation to reality*. In *IEEE Int. Conf. Robot. and Autom.*, pp. 5977–5984
 140. Nagabandi A, Konolige K, Levine S, Kumar V. 2020. *Deep Dynamics Models for Learning Dexterous Manipulation*. In *Proc. Conf. Robot. Learn.*
 141. Qi H, Yi B, Suresh S, Lambeta M, Ma Y, et al. 2023. *General in-hand object rotation with vision and touch*. In *Conf. on Robot Learn.*, pp. 2549–2564. PMLR
 142. Chen T, Tippur M, Wu S, Kumar V, Adelson E, Agrawal P. 2023. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Sci. Robot.* 8(84):eadc9244
 143. Sievers L, Pitz J, Bum B. 2022. *Learning Purely Tactile In-Hand Manipulation with a Torque-Controlled Hand*. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2745–2751
 144. Pitz J, Röstel L, Sievers L, Burschka D, Bäuml B. 2024. *Learning a Shape-Conditioned Agent for Purely Tactile In-Hand Manipulation of Various Objects*. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*
 145. Zhou W, Held D. 2023. *Learning to grasp the ungraspable with emergent extrinsic dexterity*. In *Conf. Robot Learn.*, pp. 150–160. PMLR
 146. Zhou W, Jiang B, Yang F, Paxton C, Held D. 2023. *HACMan: Learning hybrid actor-critic maps for 6D non-prehensile manipulation*. In *Conf. Robot Learn.* PMLR
 147. Cho Y, Han J, Cho Y, Kim B. 2024. *CORN: Contact-based object representation for nonprehensile manipulation of general unseen objects*. In *Int. Conf. on Learn. Represent.*
 148. Lv J, Feng Y, Zhang C, Zhao S, Shao L, Lu C. 2023. *SAM-RL: Sensing-Aware Model-Based Reinforcement Learning via Differentiable Physics-Based Simulation and Rendering*. In *Robot.: Sci. Sys.*
 149. Van Wyk K, Handa A, Makoviychuk V, Guo Y, Allshire A, Ratliff ND. 2024. Geometric fabrics: a safe guiding medium for policy learning. *arXiv preprint arXiv:2405.02250*
 150. Chitnis R, Tulsiani S, Gupta S, Gupta A. 2020. *Efficient bimanual manipulation using learned task schemas*. In *IEEE Int. Conf. Robot. Autom.*, pp. 1149–1155
 151. Büchler D, Guist S, Calandra R, Berenz V, Schölkopf B, Peters J. 2022. Learning to play table tennis from scratch using muscular robots. *IEEE Trans. Robot.* 38(6):3850–3860
 152. Cheng S, Xu D. 2023. League: Guided skill learning and abstraction for long-horizon manipulation. *IEEE Robot. Autom. Lett.*
 153. Funk N, Chalvatzaki G, Belousov B, Peters J. 2022. *Learn2assemble with structured representations and search for robotic architectural construction*. In *Conf. Robot. Learn.*, pp. 1401–1411

154. Wang C, Zhang Q, Tian Q, Li S, Wang X, et al. 2020. Learning mobile manipulation through deep reinforcement learning. *Sensors* 20(3):939
155. Ma Y, Farshidian F, Miki T, Lee J, Hutter M. 2022. Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators. *IEEE Robot. Autom. Lett.* 7(2):2377–2384
156. Fu Z, Cheng X, Pathak D. 2023. *Deep whole-body control: learning a unified policy for manipulation and locomotion*. In *Conf. on Robot Learn.*, pp. 138–149. PMLR
157. Fu Z, Zhao Q, Wu Q, Wetzstein G, Finn C. 2024. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*
158. Sentis L, Khatib O. 2006. *A whole-body control framework for humanoids operating in human environments*. In *IEEE Int. Conf. Robot. Autom.*, pp. 2641–2648. IEEE
159. Yokoyama N, Clegg AW, Undersander E, Ha S, Batra D, Rai A. 2023. Adaptive skill coordination for robotic mobile manipulation. *IEEE Robot. Autom. Lett.*
160. Ji Y, Li Z, Sun Y, Peng XB, Levine S, et al. 2022. *Hierarchical reinforcement learning for precise soccer shooting skills using a quadrupedal robot*. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 1479–1486. IEEE
161. Liu M, Chen Z, Cheng X, Ji Y, Yang R, Wang X. 2024. Visual whole-body control for legged loco-manipulation. *arXiv preprint arXiv:2403.16967*
162. Sun C, Orbik J, Devin CM, Yang BH, Gupta A, et al. 2022. *Fully autonomous real-world reinforcement learning with applications to mobile manipulation*. In *Conf. on Robot Learn.*, pp. 308–319. PMLR
163. Jauhri S, Peters J, Chalvatzaki G. 2022. Robot learning of mobile manipulation with reachability behavior priors. *IEEE Robot. Autom. Lett.* 7(3):8399–8406
164. Ji Y, Margolis GB, Agrawal P. 2023. *Dribblebot: Dynamic legged manipulation in the wild*. In *IEEE Int. Conf. Robot. Autom.*, pp. 5155–5162. IEEE
165. Hu J, Stone P, Martín-Martín R. 2023. *Causal Policy Gradient for Whole-Body Mobile Manipulation*. In *Robot.: Sci. and Syst.*
166. Honerkamp D, Welschehold T, Valada A. 2023. N^2M^2 : Learning navigation for arbitrary mobile manipulation motions in unseen and dynamic environments. *IEEE Trans. Robot.*
167. Uppal S, Agarwal A, Xiong H, Shaw K, Pathak D. 2024. *SPIN: Simultaneous perception interaction and navigation*. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 18133–18142
168. Kumar KN, Essa I, Ha S. 2023. Cascaded compositional residual learning for complex interactive behaviors. *IEEE Robot. Autom. Lett.* 8(8):4601–4608
169. Yang R, Kim Y, Kembhavi A, Wang X, Ehsani K. 2023. Harmonic mobile manipulation. *arXiv preprint arXiv:2312.06639*
170. Xiong H, Mendonca R, Shaw K, Pathak D. 2024. Adaptive mobile manipulation for articulated objects in the open world. *arXiv preprint arXiv:2401.14403*
171. Cheng X, Kumar A, Pathak D. 2023. *Legs as Manipulator: Pushing Quadrupedal Agility Beyond Locomotion*. In *IEEE Int. Conf. Robot. and Autom.*
172. Herzog A, Rao K, Hausman K, Lu Y, Wohlhart P, et al. 2023. *Deep rl at scale: Sorting waste in office buildings with a fleet of mobile manipulators*. In *Robot.: Sci. and Syst.*
173. Wu B, Martin-Martin R, Fei-Fei L. 2023. *M-EMBER: Tackling Long-Horizon Mobile Manipulation via Factorized Domain Transfer*. In *IEEE Int. Conf. Robot. Autom.*
174. Ghadirzadeh A, Chen X, Yin W, Yi Z, Björkman M, Kräigc D. 2020. Human-centered collaborative robots with deep reinforcement learning. *IEEE Robot. Autom. Lett.* 6(2):566–571
175. Christen S, Feng L, Yang W, Chao YW, Hilliges O, Song J. 2024. Synh2r: Synthesizing hand-object motions for learning human-to-robot handovers. *Int. Conf. Robot. Autom.*
176. Christen S, Yang W, Pérez-D'Arpino C, Hilliges O, Fox D, Chao YW. 2023. *Learning human-to-robot handovers from point clouds*. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 9654–9664

177. Chen YF, Everett M, Liu M, How JP. 2017. *Socially aware motion planning with deep reinforcement learning*. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 1343–1350. IEEE
178. Everett M, Chen YF, How JP. 2021. Collision avoidance in pedestrian-rich environments with deep reinforcement learning. *IEEE Access* 9:10357–10377
179. Liang J, Patel U, Sathyamoorthy AJ, Manocha D. 2021. *Crowd-steer: Realtime smooth and collision-free robot navigation in densely crowded scenarios trained using high-fidelity simulation*. In *Int. Conf. Int. Joint Conf. Artif. Intell.*, pp. 4221–4228
180. Hirose N, Shah D, Stachowicz K, Sridhar A, Levine S. 2024. Selfi: Autonomous self-improvement with reinforcement learning for social navigation. *arXiv preprint arXiv:2403.00991*
181. Liu P, Zhang K, Tateo D, Jauhri S, Hu Z, et al. 2023. *Safe reinforcement learning of dynamic high-dimensional robotic tasks: navigation, manipulation, interaction*. In *IEEE Int. Conf. Robot. Autom.*, pp. 9449–9456. IEEE
182. Dimeas F, Aspragathos N. 2015. *Reinforcement learning of variable admittance control for human-robot co-manipulation*. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 1011–1016
183. Nair S, Mitchell E, Chen K, Savarese S, Finn C, et al. 2022. *Learning language-conditioned robot behavior from offline data and crowd-sourced annotation*. In *Conf. Robot Learn.*, pp. 1303–1315. PMLR
184. Reddy S, Dragan AD, Levine S. 2018. *Shared autonomy via deep reinforcement learning*. In *Robot.: Sci. and Sys.*
185. Schaff C, Walter MR. 2020. *Residual policy learning for shared autonomy*. In *Robot. Sci. Syst.*
186. Chen YF, Liu M, Everett M, How JP. 2017. *Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning*. In *IEEE Int. Conf. Robot. Autom.*, pp. 285–292. IEEE
187. Everett M, Chen YF, How JP. 2018. *Motion planning among dynamic, decision-making agents with deep reinforcement learning*. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 3052–3059
188. Van Den Berg J, Guy SJ, Lin M, Manocha D. 2011. *Reciprocal n-body collision avoidance*. In *Int. Symp. Robot. Res.*, pp. 3–19. Springer
189. Fan T, Long P, Liu W, Pan J. 2020. Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. *Int. J. Robot. Res.* 39(7):856–892
190. Han R, Chen S, Wang S, Zhang Z, Gao R, et al. 2022. Reinforcement learned distributed multi-robot navigation with reciprocal velocity obstacle shaped rewards. *IEEE Robot. Autom. Lett.* 7(3):5896–5903
191. Sartoretti G, Kerr J, Shi Y, Wagner G, Kumar TS, et al. 2019. Primal: Pathfinding via reinforcement and imitation multi-agent learning. *IEEE Robot. and Autom. Lett.* 4(3):2378–2385
192. Nachum O, Ahn M, Ponte H, Gu S, Kumar V. 2020. *Multi-agent manipulation via locomotion using hierarchical sim2real*. In *Conf. Robot Learn.*, vol. 100, pp. 110–121. PMLR
193. Haarnoja T, Moran B, Lever G, Huang SH, Tirumala D, et al. 2024. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *Sci. Robot.* 9(89):eadi8022
194. Uchendu I, Xiao T, Lu Y, Zhu B, Yan M, et al. 2023. *Jump-start reinforcement learning*. In *Int. Conf. Mach. Learn.*, pp. 34556–34583. PMLR
195. Li C, Tang C, Nishimura H, Mercat J, Tomizuka M, Zhan W. 2023. *Residual Q-learning: offline and online policy customization without value*. In *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 61857–61869
196. Hansen N, Su H, Wang X. 2023. *TD-MPC2: Scalable, robust world models for continuous control*. In *Conf. Learn. Represent.*
197. Jeong GC, Bahety A, Pedraza G, Deshpande AD, Martín-Martín R. 2023. Bariflex: A robotic gripper with versatility and collision robustness for robot learning. *arXiv preprint arXiv:2312.05323*
198. Eysenbach B, Gupta A, Ibarz J, Levine S. 2019. *Diversity is all you need: Learning skills*

- without a reward function.* In *Int. Conf. Learn. Represent.*
199. Schwarke C, Klemm V, Van der Boon M, Bjelonic M, Hutter M. 2023. *Curiosity-driven learning of joint locomotion and manipulation tasks*. In *Conf. Robot Learn.*, vol. 229, pp. 2594–2610
 200. Xie Z, Da X, Van de Panne M, Babich B, Garg A. 2021. *Dynamics randomization revisited: A case study for quadrupedal locomotion*. In *IEEE Int. Conf. Robot. Autom.*, pp. 4955–4961
 201. Martín-Martín R, Lee MA, Gardner R, Savarese S, Bohg J, Garg A. 2019. *Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks*. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 1010–1017. IEEE
 202. Xia F, Li C, Martín-Martín R, Litany O, Toshev A, Savarese S. 2021. *ReLMoGen: Integrating motion generation in reinforcement learning for mobile manipulation*. In *IEEE Conf. Robot. Autom.*, pp. 4583–4590
 203. Luo J, Xu C, Liu F, Tan L, Lin Z, et al. 2024. Fmb: A functional manipulation benchmark for generalizable robotic learning. *Int. J. Robot. Res.*
 204. Heo M, Lee Y, Lee D, Lim JJ. 2023. *FurnitureBench: Reproducible Real-World Benchmark for Long-Horizon Complex Manipulation*. In *Robot. Sci. Syst.*
 205. van der Zant T, Iocchi L. 2011. *Robocup@ home: Adaptive benchmarking of robot bodies and minds*. In *Int. Conf. Soc. Robot.*, pp. 214–225. Springer
 206. Li X, Hsu K, Gu J, Pertsch K, Mees O, et al. 2024. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*
 207. Padalkar A, Pooley A, Jain A, Bewley A, Herzog A, et al. 2023. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*
 208. Khazatsky A, Pertsch K, Nair S, Balakrishna A, Dasari S, et al. 2024. *Droid: A large-scale in-the-wild robot manipulation dataset*. In *Robot.: Sci. and Sys.*
 209. Firooz R, Tucker J, Tian S, Majumdar A, Sun J, et al. 2023. Foundation models in robotics: Applications, challenges, and the future. *arXiv preprint arXiv:2312.07843*
 210. Hu Y, Xie Q, Jain V, Francis J, Patrikar J, et al. 2023. Toward general-purpose robots via foundation models: A survey and meta-analysis. *arXiv preprint arXiv:2312.08782*
 211. Yang S, Nachum O, Du Y, Wei J, Abbeel P, Schuurmans D. 2023. Foundation models for decision making: Problems, methods, and opportunities. *arXiv preprint arXiv:2303.04129*
 212. Ma YJ, Liang W, Wang HJ, Wang S, Zhu Y, et al. 2024. *DrEureka: Language Model Guided Sim-To-Real Transfer*. In *Robot.: Sci. Sys.*

A. Term Definition

As presented in Sec. 3 of the main article, we classify the literature based on a taxonomy consisting of four axes: **robot competencies learned with DRL**, **problem formulation**, **solution approach**, and **the level of real-world success**. In this section, we provide a detailed definition and discussion of the elements along the **problem formulation** and **solution approach** axes.

A.1. Problem Formulation

As discussed in Sec. 3, we categorize the papers based on the following elements of the problem formulation: 1) *Action space*: whether the actions are *low-level* (i.e., joint or motor commands), *mid-level* (i.e., task-space commands), or *high-level* (i.e., temporally extended task-space commands or subroutines); 2) *Observation space*: whether the observations are *high-dimensional* sensor inputs (e.g., images and/or LiDAR scans) or estimated *low-dimensional* state vectors; 3) *Reward function*: whether the reward signals are *sparse* or *dense*. This subsection provides detailed definitions and a discussion of these terms.

A.1.1. Action Space.

Low-level Actions: We define low-level actions as those that directly operate in the robot’s joint space, such as controlling torques of individual joints in a robot arm or velocities of individual wheels in a mobile robot. A low-level action space requires minimal domain knowledge and allows the policy to have fine-grained control over the robot’s behavior. However, performing learning in low-level action spaces presents several challenges: 1) exploration with low-level actions is difficult, as random joint actions often result in trivial behaviors; 2) the action space scales linearly with the robot’s degrees of freedom, often resulting in high-dimensional action spaces; and 3) joints are often controlled at a high frequency, resulting in extended task horizons and inference-time constraints.

Mid-level Actions: Mid-level actions control the robot in its workspace, such as adjusting the end-effector pose of a robot arm or controlling the velocity of the center of mass of a mobile robot. Once the policies generate these mid-level actions, they are often executed by an external controller, such as an inverse kinematics (IK) controller, to produce the joint-level torques. As such, operating in a mid-level action space requires domain knowledge to define an appropriate operational space and to design and implement the external controller effectively. When chosen correctly, mid-level action spaces strike a balance between incorporating domain knowledge and maintaining generality for various tasks. This approach is a popular choice in many RL applications for robotics, as it leverages specific expertise while allowing flexibility across different robotic functions.

High-level Actions: High-level actions control the robot through temporally extended “skills” that can realize certain short-horizon behaviors, such as “grasping certain objects” or “moving to certain rooms.” A well-designed high-level action space can greatly enhance the efficiency of the RL agent’s exploration by drastically shortening the task horizon and ensuring that the robot performs task-relevant actions most of the time. However, designing an appropriate set of skills for the high-level action space is a complex problem, often requiring each skill to be formulated as an RL problem in itself. Additionally, these skills may not always be transferable across tasks, posing challenges to their scalability.

A.1.2. Observation Space.

Low-dimensional Observations: The robot’s observations are represented as a compact, low-dimensional vector, which can include proprioceptive information, object locations, and task information.

High-dimensional Observations: The robot’s observations include high-dimensional sensor data for exteroceptive information, which can be in the form of lidar readings, camera images, and/or point clouds.

A.1.3. Reward Function.

Sparse Reward: A sparse reward signal means the agent receives trivial reward signals for most of the potential transitions in a (PO)MDP and only receives non-trivial reward signals sparsely. One natural way of defining a sparse reward for a task is to have +1 for all transitions into a success termination state, -1 for all transitions into a failure termination state, and 0 for any other transitions.

Dense Reward: A dense reward means the reward signal is abundant, providing rich feedback to the agent. In certain tasks, such as locomotion, the reward is naturally dense (e.g., the error between the robot’s current forward velocity and the instructed velocity). In other scenarios where the task reward is inherently sparse, such as navigation tasks, a dense reward can be defined by adding shaping components to the sparse reward (e.g., the distance between the robot and the navigation target). Such kind of shaped and dense rewards are often used to facilitate learning efficiency, especially for long-horizon tasks.

A.2. Solution Approach

As introduced in Sec. 3, we classify the solution approach from the following perspectives:

- 1) *Simulator usage*: whether and how simulators are used, categorized into *zero-shot*, *few-shot sim-to-real transfer*, or directly learning offline or in the real world *without simulators*;
 - 2) *Model learning*: whether (a part of) the transition dynamics model is learned from robot data;
 - 3) *Expert usage*: whether expert (e.g., human or oracle policy) data are used to facilitate learning;
 - 4) *Policy optimization*: the policy optimization algorithm adopted, including *planning* or *offline*, *off-policy*, or *on-policy RL*;
 - 5) *Policy/Model Representation*: Classes of neural network architectures used to represent the policy or dynamics model, including *MLP*, *CNN*, *RNN*, and *Transformer*.
- This subsection provides detailed definitions of these terms.

A.2.1. Simulator Usage.

Zero-shot sim2real: The training is performed entirely in a simulator, where the trained policy is deployed directly in the real world without additional learning.

Few-shot sim2real: The robot is pre-trained in the simulator and fine-tuned in the real world with limited additional real-world interactions.

No Simulator: The training is conducted in the real world without using a simulator.

A.2.2. Model Learning. RL algorithms can be broadly classified into two categories: model-free RL and model-based RL, based on whether they learn a dynamics model. In model-free RL algorithms, such as PPO and SAC, the robot directly learns a policy or value function without explicitly modeling the environment’s dynamics. Model-free RL is often easier to implement and superior when learning a good policy is simpler than learning a good model. In contrast, model-based RL algorithms, such as TD-MPC and Dreamer, involve the robot learning a world model that can predict the consequences of its actions. This world model can be used either for model-based planning and control or for generating experiences for a model-free RL agent, potentially increasing the agent’s sample efficiency. Instead of learning the full dynamics, some methods learn a residual or a part of the dynamics model (e.g., actuator dynamics model) to complement the simulation for model-free RL, which we also mark as involving model learning in our categorization.

A.2.3. Expert Usage. *Tabula rasa* RL begins with random initialization, training entirely through trial and error. However, in robotics, it is sometimes possible to utilize an external expert to expedite the learning process. These experts may include human demonstrations, trajectory planners, oracle actions, and so on. In this survey, we classify all works that utilize an external expert, either offline or online, to facilitate learning as works “with experts”, which gives them an advantage over methods that do not assume access to experts.

A.2.4. Policy Optimization.

Planning: The robots policy is derived by solving an optimal control problem online using a learned world model. Representative algorithms include A* and MPPI (Model Predictive Path Integral).

Offline: The robot does not interact with the environment during learning. Instead, it learns a policy and, optionally, a value function directly from offline data. Representative algorithms include CQL (Conservative Q-Learning) and DT (Decision Transformer).

On-policy: The robot interacts with the environment during learning and only updates the policy with transitions collected by the current policy. Representative algorithms include PPO (Proximal Policy Optimization) and TRPO (Trust Region Policy Optimization).

Off-policy: The robot interacts with the environment during learning and updates the policy with transitions collected by both the current policy and other/previous policies. Representative algorithms include SAC (Soft Actor-Critic) and DQN (Deep Q-Network).

A.2.5. Policy/Model Representation.

MLP Only: Multi-layer Perceptron (MLP) models take 1D vector inputs and consist solely of fully connected layers. They are widely used for processing low-dimensional observations.

CNN: Convolutional Neural Networks (CNNs) are a specialized type of MLP that preserves local spatial coherence, initially designed for image processing. Later works have extended CNNs to process 1D data like lidar readings and observation memory, as well as 3D data such as point clouds.

RNN: Recurrent Neural Networks (RNNs), including LSTM and GRU, are bidirectional

neural networks with internal memory. They are suitable for processing time-series data, such as trajectories over time.

Transformer: Transformers take a sequence of vectors (tokens) as input and use multi-head self-attention to generate outputs. Recently, transformers have been widely used to process time-series data, natural language instructions, and visual information. They have also proven powerful in fusing tokenized multi-modal information.

B. Additional Tables

This section contains tables that present the complete categorization of the reviewed papers along all four axes of our taxonomy. Tables 1-2 categorize the papers based on problem formulation for each robot competency, while Tables 3-6 categorize the papers based on solution approach for each robot competency. As in the tables in the main article, the color map indicates the levels of real-world success: *Sim Only*, *Limited Lab*, *Diverse Lab*, *Limited Real*, and *Diverse Real*.

In these tables, we add a superscript * to papers that appear in multiple columns, which means they adopt two different elements jointly (e.g., a hierarchical policy that outputs both low-level and mid-level actions, a policy network consists of both CNN and RNN).

Application	Action Space								
	Low-Level			Mid-Level			High-Level		
Locomotion	27 , 28 , 29 , 30 , 31 * , 32 * , 33 , 36 * , 37 , 38 , 40 , 41 , 42 , 43 , 44 , 45 , 46 , 48 , 49 , 50 , 51 , 52 , 53 , 54 , 55 , 56 , 57 , 58 , 59 , 61 , 62 , 63 , 64 , 65 , 68	31 * , 32 * , 34 , 35 , 47 , 66 , 67					36 * , 60		
Navigation	20 , 90 , 96 * , 97 * , 99 , 100 ,	7 , 21 , 73 , 74 , 75 , 78 , 83 , 85 , 88 , 89 , 91 , 92 , 93 , 94 , 95 * , 98 , 101 , 102 , 103					76 , 81 , 82 , 86 , 87 , 95 * , 96 * , 97 *		
Manipulation	113 , 122 , 127 , 131 , 138 , 139 , 140 , 141 , 142 , 143 , 144	54 , 110 , 111 , 112 , 114 , 115 , 116 , 117 , 118 , 119 , 120 , 121 , 124 , 125 , 126 , 128 , 129 , 130 , 132 , 133 , 134 , 137 , 145 , 146 , 147					108 , 109 , 123 , 135 , 136		
MoMa	155 , 156 , 157 , 165 , 169 , 171 , 160 , 164 , 161 , 168	154 , 166 , 162 , 163 , 173 , 159 , 172 , 167					170		
HRI	177 , 178 , 179 , 180 , 181 , 182	175 , 176 , 183 , 184 , 185					174		
Multi-Robot Interaction	186 , 187 , 189 , 190 , 192 , 193	191							

Table 1: Categorizing Literature based on Problem Formulation

Application	Observation Space						Reward Function			
	High-dim			Low-dim			Sparse	Dense		
Locomotion	35 , 36 , 43 , 44 , 45 , 49 , 50 , 54 , 61 ,	27 , 28 , 29 , 30 , 31 , 32 , 33 , 34 , 37 , 38 , 40 , 41 , 42 , 46 , 47 , 48 , 51 , 52 , 53 , 55 , 56 , 57 , 58 , 59 , 60 , 62 , 63 , 64 , 65 , 66 ,	56	27 , 28 , 29 , 30 , 31 , 32 , 33 , 34 , 35 , 36 , 37 , 38 , 40 , 41 , 42 , 43 , 44 , 45 , 46 , 47 , 48 , 49 , 50 , 51 , 52 , 53 , 54 , 55 , 57 , 58 , 59 , 60 , 61 , 62 , 63 , 64 , 65 , 66 , 67 , 68	67 , 68					
Navigation	73 , 74 , 75 , 76 , 78 , 81 , 82 , 83 , 85 , 86 , 87 , 88 , 89 , 91 , 92 , 94 , 95 , 96 , 97 , 98 , 99 , 101 , 102	7 , 20 , 21 , 90 , 93 , 100 , 103	78 , 96 *	7 , 20 , 21 , 73 , 74 , 75 , 76 , 81 , 82 , 83 , 85 , 86 , 87 , 88 , 89 , 90 , 91 , 92 , 93 , 94 , 95 , 96 * , 97 , 98 , 99 , 100 , 101 , 102 , 103						
Manipulation	54 , 108 , 109 , 110 , 111 , 112 , 113 , 114 , 115 , 116 , 117 , 118 , 119 , 120 , 121 , 124 , 125 , 128 , 133 , 134 , 135 , 136 , 137 , 141 , 142 , 145 , 146 , 147	122 , 123 , 126 , 127 , 129 , 130 , 131 , 132 , 138 , 139 , 140 , 143 , 144 147	54 , 108 , 110 , 111 , 112 , 114 , 115 , 117 , 118 , 122 , 124 , 128 , 129 , 134 , 131 , 132 , 131 , 136 , 135 , 138 , 138 , 139 , 141 , 142 , 144 , 145 , 147	109 , 113 , 116 , 118 , 119 , 120 , 121 , 123 , 125 , 126 , 127 , 130 , 131 , 132 , 133 , 135 , 136 , 137 , 138 , 139 , 140 , 141 , 142 , 143 , 144 , 145 , 146 , 147						
MoMa	165 , 169 , 166 , 162 , 170 , 167 , 161 , 173 , 159 , 172	155 , 156 , 154 , 157 , 171 , 160 , 164 , 163 , 168	170 , 173 , 172	155 , 156 , 154 , 157 , 165 , 169 , 171 , 160 , 164 , 166 , 162 , 163 , 167 , 161 , 168 , 159						
HRI	175 , 176 , 179 , 180 , 183	174 , 177 , 178 , 181 , 182 , 184 , 185	174	175 , 176 , 177 , 178 , 179 , 180 , 181 , 182 , 183 , 184 , 185						
Multi-Robot Interaction		186 , 187 , 189 , 190 , 191 , 192 , 193	186 , 187	189 , 190 , 191 , 192 , 193						

Table 2: Categorizing Literature based on Problem Formulation (Cont.)

Application	Simulator Usage					
	Zero-shot Sim-to-Real		Few-shot Sim-to-Real		No Simulator	
Locomotion	27 , 28 , 29 , 30 ,		43 , 48 , 56		53 , 54	
	31 , 32 , 33 , 34 ,					
	35 , 36 , 37 , 38 ,					
	40 , 41 , 42 , 44 ,					
	45 , 46 , 47 , 49 ,					
	50 , 51 , 52 , 55 ,					
	57 , 58 , 59 , 60 ,					
	61 , 62 , 63 , 64 ,					
	65 , 66 , 67 , 68					
Navigation	20 , 21 , 73 , 74 ,		7 , 102		88 , 90 , 91 , 92	
	75 , 76 , 78 , 81 ,					
	82 , 83 , 85 , 86 ,					
	87 , 93 , 94 , 95 ,					
	96 , 97 , 98 , 99 ,					
	100 , 101 , 103					
Manipulation	108 , 111 , 123 ,		116 , 131		54 , 109 , 110 ,	
	130 , 133 , 134 ,				112 , 113 , 114 ,	
	135 , 137 , 138 ,				115 , 117 , 118 ,	
	139 , 141 , 142 ,				119 , 120 , 121 ,	
	143 , 144 , 145 ,				122 , 124 , 125 ,	
	146 , 147				126 , 127 , 128 ,	
					129 , 132 , 136 ,	
					140	
MoMa	155 , 156 , 154 ,		160 , 172		162 , 170	
	157 , 165 , 169 ,					
	171 , 164 , 166 ,					
	163 , 167 , 161 ,					
	168 , 173 , 159					
HRI	174 , 175 , 176 ,		184		180 , 183 , 182	
	177 , 178 , 179 ,					
	181					
Multi-Robot Interaction	186 , 187 , 189 ,					
	190 , 191 , 192 ,					
	193					

Table 3: Categorizing Literature based on Solution Approach

Application	Model Learning				Expert Usage			
	with Model Learning		No Model Learning		No Expert		with Expert	
Locomotion	29 , 31 , 34 , 35 , 36 , 41 , 51 , 52 , 54 , 56	27 , 28 , 30 , 33 , 37 , 38 , 42 , 43 , 44 , 46 , 47 , 48 , 50 , 53 , 55 , 58 , 59 , 60 , 62 , 63 , 64 , 66 , 67 , 68	32 , 40 , 45 , 48 , 49 , 57 , 61 , 65 , 68	28 , 29 , 31 , 33 , 34 , 35 , 38 , 41 , 42 , 44 , 45 , 47 , 49 , 50 , 52 , 53 , 56 , 57 , 59 , 61 , 62 , 65 , 67 , 68	32 , 37 , 43 , 42 , 47 , 46 , 53 , 55 , 59 , 60 , 65 , 66 , 67 , 68	27 , 30 , 36 , 40 , 46 , 48 , 51 , 55 , 58 , 63 , 64		
Navigation	7 , 20 , 74 * , 88 , 90 , 95 , 96 , 97 , 98 , 102	21 , 73 , 74 * , 75 , 76 , 78 , 82 , 83 , 85 , 87 , 89 , 91 , 93 , 94 , 99 , 100 , 101 , 103	74 , 75 , 78 , 82 , 83 , 85 , 87 , 88 , 93 , 95 , 96 , 98 , 100 , 101 , 102 , 103	21 , 73 , 78 , 81 , 86 , 87 , 92 , 91 , 94 , 99 , 101 , 102 , 103	76 , 89 , 90 , 91 , 92 , 97			
Manipulation	54 , 113 , 118 , 140 , 144	108 , 109 , 110 , 111 , 112 , 114 , 115 , 116 , 117 , 119 , 120 , 121 , 122 , 123 , 124 , 125 , 126 , 127 , 128 , 129 , 130 , 131 , 132 , 133 , 134 , 135 , 136 , 137 , 138 , 139 , 141 , 142 , 143 , 145 , 146 , 147	54 , 108 , 109 , 110 , 111 , 114 , 115 , 116 , 118 , 119 , 123 , 125 , 130 , 131 , 133 , 135 , 137 , 138 , 139 , 140 , 141 , 142 , 143 , 144 , 145 , 146 , 147	112 , 113 , 117 , 120 , 121 , 122 , 124 , 126 , 127 , 128 , 129 , 132 , 134 , 136				
MoMa		155 , 156 , 154 , 157 , 165 , 169 , 171 , 160 , 164 , 166 , 162 , 163 , 170 , 167 , 161 , 168 , 173 , 159 , 172	155 , 156 , 154 , 165 , 169 , 171 , 160 , 164 , 166 , 162 , 163 , 167 , 161 , 168 , 173	157 , 170 , 159 , 172				
HRI	175 , 176 , 183	174 , 177 , 178 , 179 , 180 , 181 , 182 , 184 , 185	175 , 176 , 179 , 181 , 182 , 184 , 185	174 , 177 , 178 , 180 , 183				
Multi-Robot Interaction		186 , 187 , 189 , 190 , 191 , 192 , 193	189 , 190 , 192	186 , 187 , 191 , 193				

Table 4: Categorizing Literature based on Solution Approach (Cont.)

Application	Planning	Policy Optimization									
		Offline		Off-Policy				On-Policy			
		32	36*	48	27	28	29	30	31	32	33
Locomotion		53	54	68	31	32	33	34	35	36*	37
		38	40	41	42	43	44	45	46	47	49
		50	51	52	55	56	57	58	59	60	61
		62	63	64	65	66	67				
Navigation	74*, 76*, 88, 90, 102, 103*	76*, 89, 91*, 94, 101	73, 74*, 75, 78, 91*, 92, 94, 101	7, 20, 21, 82, 83, 85, 87, 93, 95, 96, 97, 98, 99, 100, 103*	81, 86,						
Manipulation	118, 140	108, 115, 121, 122, 124, 129	54, 109, 110, 111, 112, 114, 116, 117, 119, 120, 123, 125, 126, 127, 128, 132, 133, 134, 135, 136, 137, 143, 145, 146	113, 130, 131, 138, 139, 141, 142, 144, 147							
MoMa			160*, 166, 162, 163, 173, 172	155, 156, 154, 157, 165, 169, 171, 160*, 164, 170, 167, 161, 168, 159							
HRI	183	180*	174, 175, 176, 180*, 181, 182, 184	177, 178, 179, 185							
Multi-Robot Interaction			193	186, 187, 189, 190, 191, 192							

Table 5: Categorizing Literature based on Solution Approach (Cont.)

Application	Policy/Model Representation				
	MLP Only	CNN	RNN	Transformer	
Locomotion	28 , 29 , 30 ,	27 , 33 , 34 ,	35 , 37 , 44 * ,		62
	31 , 32 , 38 ,	36 , 42 , 43 ,	49 , 50 , 54 * ,		
	40 , 41 , 46 ,	44 * , 45 , 49 * ,	55 , 57 , 59 ,		
	47 , 48 , 51 ,	54 * , 63 , 67		61	
	52 , 53 , 56 ,				
	58 , 60 , 64 ,				
	65 , 66 , 68				
Navigation	7 , 20 , 21 ,	78 , 81 , 82 ,	74 * , 83 * ,		74 *
	73 , 74 * , 75 ,	83 * , 85 , 86 * ,	85 , 86 * ,		
	76 , 90 , 93 ,	87 * , 89 , 91 ,	87 * , 95 ,		
	99 , 100 , 103 §	92 , 94 , 96 ,	98 * , 102 * ,		
		97 , 98 * , 101 ,			
		102 *			
Manipulation	123 , 129 ,	54 * , 108 ,	109 ,	54 * , 118 * ,	116 , 124 ,
	131 , 132 ,	110 , 111 ,	112 ,	130 , 144 * ,	141 , 147
	138 , 139 ,	113 , 114 ,	115 ,		
	140 , 143 ,	117 , 118 * ,	119 ,		
	144 * , 145	120 , 121 ,	122 ,		
		125 , 126 ,	127 ,		
		128 , 133 ,	134 ,		
MoMa	135 , 136 ,	136 ,	137 ,		
		142 , 146			
	155 , 156 ,	165 , 169 * ,	166 ,	169 * , 167 * ,	
	154 , 157 ,	162 , 170 ,	167 * ,		172 *
	171 , 160 ,	161 , 173 ,	159 ,		
HRI	164 , 163 , 168		172 *		
	177 , 181 ,	175 , 176 ,	179 ,	174 , 178	183 *
	182 , 184 , 185	180 , 183 *			
Multi-Robot Interaction	186 , 189 ,			187 , 190 ,	
	192 , 193			191	

Table 6: Categorizing Literature based on Solution Approach (Cont.)