A Review of Mobile Robot Motion Planning Methods: from Classical Motion Planning Workflows to Reinforcement Learning-based Architectures

Lu Dong, *Member, IEEE,* Zichen He, Chunwei Song, and Changyin Sun, *Senior Member, IEEE*

***Abstract*—Motion planning is critical to realize the autonomous operation of mobile robots. As the complexity and randomness of robot application scenarios increase, the planning capability of the classical hierarchical motion planners is challenged. With the development of machine learning, deep reinforcement learning (DRL)-based motion planner has gradually become a research hotspot due to its several advantageous feature. DRL-based motion planner is model-free and does not rely on the prior structured map. Most importantly, DRL-based motion planner achieves the unification of the global planner and the local planner. In this paper, we provide a systematic review of various motion planning methods. First, we summarize the representative and state-of-the-art works for each submodule of the classical motion planning architecture and analyze their performance features. Subsequently, we concentrate on summarizing RL-based motion planning approaches, including motion planners combined with RL improvements, map-free RL-based motion planners, and multi-robot cooperative planning methods. Last but not least, we analyze the urgent challenges faced by these mainstream RL-based motion planners in detail, review some state-of-the-art works for these issues, and propose suggestions for future research.**

***Index Terms*—Mobile robot, Reinforcement learning, Motion planning, Multi-robot cooperative planning.**

I. Introduction

WITH the rapid development of AI, autonomous intelligent mobile robots (MRs) are always at the forefront of scientific research due to their compact size, flexible mobility, diverse functions, and modularity. MR can replace human beings to perform complicated and dangerous missions on various occasions by carrying different sensing modules. Therefore, it plays a vital role in ocean exploration, urban rescue, security patrol, and epidemic control [1]–[3], etc.

The motion planning (MP) technology is one of the most critical modules that endows the MR with autonomous capabilities. The specific function of the motion planner is to integrate local or global state information of the robot system and produce optimal or near-optimal planning decisions in the face of various environments. The standard MP module consists of the global motion planner and the local motion planner. The global motion planner is responsible for generating optimal or near-optimal, kinodynamic feasible, safe, and executable trajectories on the basis of the structured prior map information. The local motion planner is responsible for helping the MR make real-time motion decisions in local dynamic environments (e.g., pedestrian participation environment, outdoor environments, etc.).

The standard map-based MP framework is hierarchical and multi-level cascading and has a certain degree of customization. In the framework, the global planner and the local planner are independent of each other, and both need to be configured separately for different scenarios. These features make it difficult for classical motion planners to adapt to unstructured, complex, and dynamic environments. Therefore, studying a map-less motion planner with a certain generalization, robustness, and adaptability is of great significance. With the development of reinforcement learning (RL), RL-based motion planners can be independent of the map prior data and obtain a stronger generalization by learning interactively with various scenarios during the training stage. Therefore, it gradually becomes a research hotspot.

The research of RL has experienced a long history. The dynamic programming algorithm proposed by Bellman in 1956 has laid the foundation for the subsequent development of this field [4]. The primary research issue of the RL is the tradeoff between exploration and exploitation at each time step. The agent explores to discover different policies that can bring more incredible benefits or exploits current optimal policies. Along with the substantial improvement in the computing power and the storage capacity of hardware systems, deep learning (DL) technology has been widely used in RL. Researchers utilize DRL to integrate the learning module and the decision module, and realize the nonlinear mapping procedure from the raw state inputs to the final action decisions. With its superior properties, DRL has been successfully applied to gaming AI, autopilot, transportation schedules, power system optimization [3], [5], [6], etc. In the field of mobile robot MP, RL-based motion planners achieve end-to-end planning, get rid of the tedious hierarchical multi-level coupling planning framework, and unify the global motion planner with the local motion planner. By constructing specific reward forms and training paradigms according to

L.Dong is with the School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China (e-mail:ldong90@seu.edu.cn).

Z.He and C.Song are with the College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China (e-mail:1910646@tongji.edu.cn; 2030739@tongji.edu.cn).

C.Sun is with the School of Automation, Southeast University, Nanjing 210096, China, and also with the College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China (e-mail: cysun@seu.edu.cn).

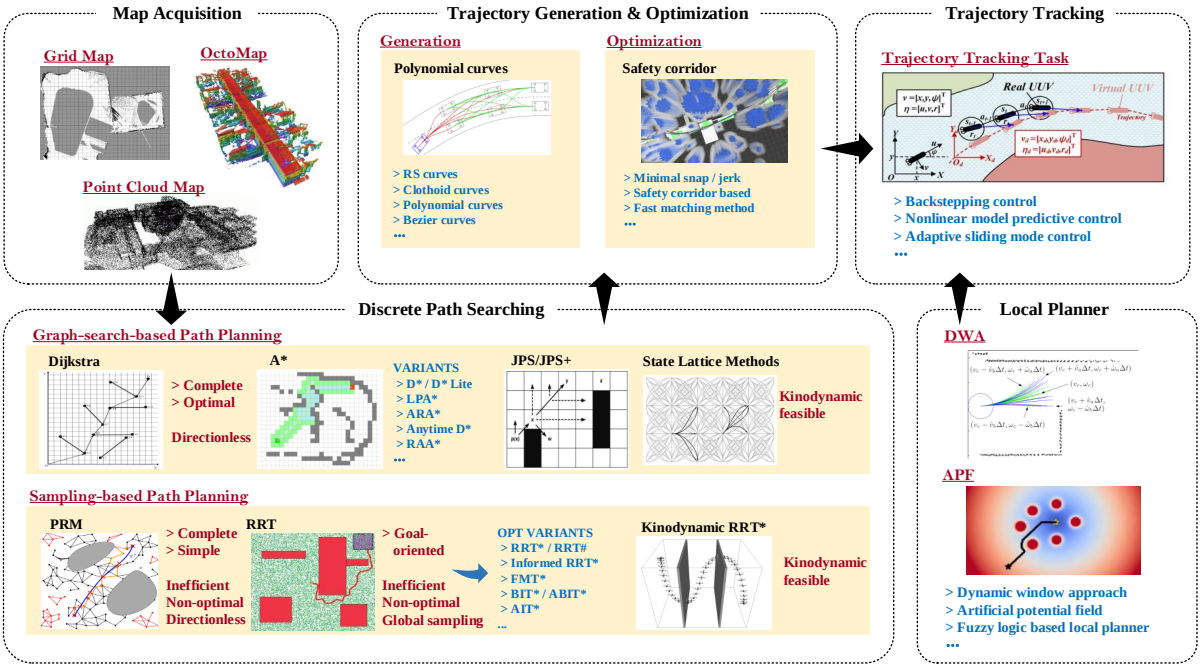

Fig. 1. An overall architecture diagram of the classical motion planning workflow.

different task objectives, the state updating policy can be improved iteratively based on the feedback signals from the environment during the process of interaction. These features mentioned above offer the possibility of deploying RL-based motion planners to robots operating in some unstructured and dynamic environments where the real-time mapping process is challenging to perform.

This paper is a systematic review of current mainstream and state-of-the-art mobile robot motion planning methods. Its content mainly covers robot types, including wheeled mobile robots (WMRs), autonomous underwater vehicles (AUVs), unmanned aerial vehicles (UAVs), etc. The overall structure of this paper consists of three parts. The first part is the summary and comparison of different representative algorithms of each sub-module in the pipeline of the classical motion planner. The second part is an overview of RL-based MP approaches. It consists of three sections. The first section is a summary of classical motion planning methods incorporating the RL optimization module. This type of motion planner still relies on the map prior. The role of the RL module is to make local planning decisions (e.g., obstacle avoidance) or select the optimal functional hyper-parameters in classical motion planners. The second section is an overview of sensor-level end-to-end RL-based motion planners. This type of motion planner is map-free. We mainly review two mainstream sensor-based works: lidar-based methods and vision sensor-based methods. The third section is an overview of RL-based multi-robot collaborative motion planning methods. In this section, we focus on reviewing some works of multi-robot collaborative planning based on centralized training with decentralized execution (CTDE) RL paradigm. The last part of this survey is a discussion. In this part, we systematically summarize several challenges faced by current RL-based motion planners, which are reality gap, social etiquette, catastrophic forgetting problem, reward sparsity issue, lidar data pre-processing problem, low sample efficiency problem, and generalization problem. These issues hinder the application and deployment of the pretrained RL agents in realistic physical environment. Also, we review several representative works aimed at addressing these issues.

To sum up, the rest of this paper is organized as follows. Section II is a review of classical hierarchical MP approaches. Section III focuses on summarizing several map-based classical motion planners combined with the RL optimization algorithms. Section IV discusses the map-less and end-to-end RL-based MP methods at the sensor level. Section V provides a survey of RL-based multi-robot motion planning methods. Section VI concludes several current challenges in RL-based motion planners and gives future directions. The conclusion of this paper is drawn in section VII.

II. Classical Motion Planning of MRs

Before we start designing the workflow of the classical motion planner, we need to obtain the map representation of the environment. Commonly used structured maps include occupancy grid map, point cloud map, Voronoi diagram map, Euclidean signed distance fields, etc [7], [8]. The quality and accuracy of these prior maps directly determine the final planning performance. We represent the classical hierarchical architecture of the classical motion planner as shown in Fig. 1. This framework can be divided into four submodules: discrete path searching, trajectory generation and optimization, trajectory tracking, and local planner [1], [9], [10]. It can be found that the classical motion planning approach is map-based and highly customized for different mission scenarios. Each internal submodule is interdependent. In this section,

we will describe the main features and principles of each submodule, list representative algorithms and their limitations, and provide an overview of recent progress.

A. Discrete Path Searching

The goal of discrete path searching (DPS) process is to find a feasible path that consists of a series of discrete waypoints from the initial point to the target point. DPS works in the configuration space (C-space). Each configuration of the robot can be represented as a point. This approximation process reduces the complexity of the computation and improves search efficiency. Notably, in C-space, specific expansion operations are required for robots of different sizes and shapes [11].

Traditional global DPS algorithms can be divided into two categories: the graph-searching-based algorithm (GSBA) and the sampling-based algorithm (SBA) [12].

1) Graph-search-based algorithms: Depth-first search (DFS) and breadth-first search (BFS) are two fundamental graph search algorithms. On the basis of BFS, Dijkstra is proposed. This algorithm is greedy, complete, and optimal [13]. However, Dijkstra lacks directionality in the process of path search. In [14], A* is proposed. The researchers introduce the heuristic function to measure the distance between the real-time search position and the target position. This function makes the search more oriented and improves the search speed compared to Dijkstra. In [15], Anthony Stentz presents Dynamic A* (D*). They replace the heuristic rule in A* with an incremental reverse rule. Based on A*, SvenKoenig et al. develop Lifelong planning A* (LPA*) in [16]. They combine incremental search with A*. LPA* avoids the problem of re-calculating the whole graph due to changes in the environment. In [17], Dorota Belanová et al. propose D* Lite. D* Lite is a path planning algorithm with the variable start point and the fixed target point. This method incorporate the reverse searchint trick with the heuristic mechanism. The difference between D* Lite and LPA* is the search direction. Jump point search (JPS) [18] is another type of GSBA with different search principles. The proposers seek to improve the search efficiency by optimizing subsequent nodes searching process on the basis of A*. Notably, JPS only adds the jump points that are searched according to the specific rules into the open list. This operation excludes a large number of meaningless nodes. Therefore, JPS occupies less memory and can search faster than A*. However, JPS is only applicable to the uniform grid map [9], [19]. In [20], Changjiang Jiang et al. propose the JPS+ based path planning method. They further improve the search efficiency of JPS by adding the pre-processing section but is less suitable for dynamic environments.

Some researchers focus on improving the real-time performance of A*, such as [21], [22]. However, none of the above methods consider the kino-dynamics of the robot. For some WMRs with non-holonomic constraints, sometimes the discrete path planned by the above methods cannot be executed well. State lattice methods [23]–[25] have been proposed to handle this problem. These approaches perform spatial discretization, and use a hyper-dimensional grid of states to represent the planning area. The sampling process allows the planner to generate a series of dynamically feasible motion connections. Finally, the algorithm would search for the optimal path fragment among these connections.

2) Sampling-based algorithms: GSBAs are mainly applied to path planning problems on low-dimensional spaces. The completeness of these methods depends on the entire modeling process of the environment. In a higher-dimensional space, such methods would be suffering from the curse of dimensionality. Sampling-based algorithm (SBA) are more suitable for high-dimensional path searching scenarios. These algorithms have probabilistic completeness and further improve the search efficiency of feasible waypoints [26].

Probabilistic road map (PRM) and rapid-exploring random tree (RRT) are two fundamental types of SBAs [9]. PRM builds a graph during the learning stage and utilizes it to search for valid discrete paths during the query stage [19]. It is simple with few parameters but lacks optimality. RRT is more goal-oriented than PRM. It generates an extended tree by selecting leaf nodes with random sampling. When the leaf node expands to the target region, the discrete path from the root node to the goal is obtained. RRT is not sufficient because of the whole-space-sampling process. And, RRT is not an optimal or asymptotically optimal algorithm. Limited by these restrictions, RRT cannot plan a feasible path quickly in the narrow passage environment. Until now, there are still many researchers dedicated to optimizing these problems in RRT [26]. RRT* [27] introduces prune optimization and random geometric graphs (RGG) during the node extension phase of RRT. It is an asymptotically optimal algorithm. In [28], [29], RRT*-smart and RRT# are proposed respectively. The main idea of these two methods is to improve the convergence speed of RRT*. In [30], kino-dynamic RRT* is presented. This method additionally deals with the kino-dynamics of robots. Also, kino-dynamic RRT* samples in full state space and can handle the problem of robot motion planning with non-holonomic constraints. Informed RRT* in [31] directly limits the sampling interval to improve the overall convergence efficiency, which can effectively solve the discrete path search problem for narrow passages. Batch informed trees (BIT*) is presented in [32]. This method unifies the advantages of GSBAs and SBAs and introduces a heuristic method to search for a sequence of increasingly dense implicit RGGs iteratively. Compared with RRT*, informed-RRT*, and fast match trees (FMT*) [33], BIT* enhances planning performance significantly in the experiments. In [34], Marlin P. Strub et al. further extend BIT* utilize advanced truncated anytime graph-based search techniques to enhance real-time planning performance. In [35], they introduce asymmetric bidirectional search on the basis of BIT*. This trick can help the planning process converge towards the optimal result as fast as RRT-connect. Real-time RRT* (RT-RRT*) [36] and information-driven RRT* (ID-RRT*) [37] focus on enhancing the real-time performance of the RRT*, improving the planning capability in unknown and dynamic environments.

B. Trajectory Generation and Optimization

Most of the DPS algorithms only consider the geometric constraints of the workspace. In some cases, the final optimal

or near-optimal segmented trajectories are not executable for the actual MR. The trajectory generation and optimization (TGO) process considers multiple constraints (e.g., safety constraints, kino-dynamic constraints, etc.) of the robot and incorporates the time-allocation mechanism in the planning process to empower the motion planner with more executable capabilities. By cascading with TGO, the planner can finally generate a kino-dynamic feasible, collision-free, executable, and trackable trajectory that satisfies several optimization objectives (time optimal, energy optimal, etc.).

The interpolation-curve-based method is one of the most commonly used approaches for trajectory generation. This method can generate trajectories with good continuity and differentiability. The typical interpolating curves include Reeds and Sheep (RS) curves [38], clothoid curves [39], polynomial curves [26], Bezier curves [40], etc. The minimum snap presented in [41] is an effective optimization paradigm for trajectory optimization and has inspired many scholars. The method proposer Mellinger utilizes the differential flatness of the drone to reduce the dimension of the state space and the action space. By solving the quadratic programming (QP) problem with several constraints, the minimal snap algorithm can minimize the thrust change rate to achieve the objective of optimal energy consumption. Later, Richter et al. solved the minimum snap problem in closed-form to avoid the numerical instability [42]. In [43], Chen Jing et al. introduce safety corridor constraints in the minimum snap algorithm to enforce the safety of the robot during planning. However, the process of iterative detection of the boundary extremum safety is time-consuming. Fei Gao et al. use Bezier curves with the features of convex hull and hodograph to substitute traditional polynomial curves [40]. This approach has fewer constraints and avoids the tedious iterative checking process. In [44], Fei Gao et al. present an online TGO method based on the Euclidean distance field (EDF). The cost function of this method is composed of the elastic band smoothness term, the safety term, and the kino-dynamic term. Then, the nonlinear optimization method is used to solve the final trajectory. EDF-based TGO method is real-time, and has great local replanning ability. It overcomes the problem of low clearance between the MR and the obstacle in previous approaches.

C. Trajectory Tracking

Generally, the purpose of the trajectory tracking (TT) phase is to enable the MR to track the trajectory planned by the TGO process. The early TT task belongs to the control level. It mainly focuses on designing virtual controllers based on dynamics equations so that the MR can track a given reference trajectory asymptotically. Common trajectory tracking control methods are input-output linearization [45], backstepping control [46], sliding mode control (SMC) [47], robust control [48], etc. However, these approaches suffer from several limitations. For example, the mathematical models of some robots (e.g., AUVs) contain complex nonlinear or uncertain terms. These terms increase the difficulty of modeling. Besides, the realistic operating environments of MR are often changeable, such as crowded environments, multi-robot interaction environments. These challenges require that the TT algorithms should have certain anti-disturbance and local replanning capabilities.

In [49], an adaptive sliding mode controller (ASMC) is designed to handle the TT problem of the WMRs. This algorithm takes nonlinear model and disturbances into account and utilizes the discontinuous projection mapping to adjust the performance parameters of the controller. Model predictive control (MPC) is also a mainstream approach used to deal with TT problems. MPC belongs to the category of optimal online control method and can handle various state and control constraints [50]. In [51], the MPC-based iterative trajectory tracking scheme is presented and applied to UAV navigation. In [52], Avraiem Iskander et al. select to adopt the nonlinear MPC (NMPC) to cooperate with RRT* and the minimum snap algorithm to build a closed-loop of UAV motion planning in three-dimensional (3D) space. Björn Lindqvist et al. focus on coping with dynamic obstacle avoidance problems. They couple the dynamic obstacle avoidance strategies in the TT controller. The proposed architecture of the novel NMPC is based on the PANOC non-convex solver and the trajectory classification scheme [53]. Caicha Cui et al. combine the MPC with the robust sliding mode dynamic control (RSMDC). The MPC module is responsible for replanning trajectory to avoid local obstacles. The RSMDC plays the role of controlling the tracking speed to reduce the impact of UUV model uncertainty and external disturbance on the final planning effect [54].

D. Local Planning

In the real world, MRs are always deployed in environments full of uncertain factors. For example, serving robots in airports or stations must reasonably predict and react to pedestrians. Therefore, the MR needs to have the local replanning ability to deal with unexpected situations while tracking the global trajectory. Local motion planners serve this purpose.

The commonly used local planners include artificial potential field [55], reactive replanning method [56], [57], fuzzy algorithm based method [58], etc. APF is relatively simple and has good real-time planning performance. However, the traditional APF method is prone to fall into local optimum and has accessibility problem when the target point is surrounded by obstacles. Therefore, many researchers pay their attention to solving the shortcomings of the traditional APF. In [55], Di Wang et al. solve the above limitations by introducing the left tuning potential field and the virtual target point.

Reactive replanning methods can also avoid unknown dynamic obstacles in the environment. Common methods include directional approach [56], dynamic window approach (DWA) [57], etc. DWA is an active velocity selection algorithm and considers the kino-dynamics of the robot. DWA samples multiple velocities in the velocity space and generates a series of intrinsic motion trajectories in a certain time. By comparing the scores of different trajectories, the algorithm would select the optimal trajectory for the robot. DWA is usually coupled with the global trajectory tracking process and can endow the MR with a high degree of flexibility.

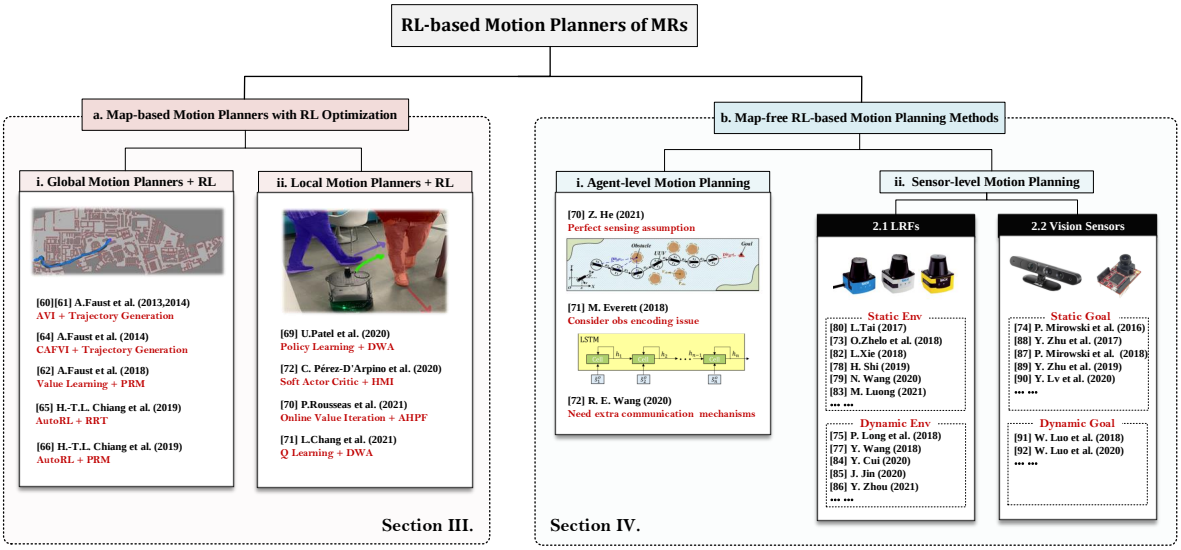

Fig. 2. A summary diagram of RL-based motion planning algorithms for the single mobile robot operation. We mainly review two types of research methods: map-based motion planning methods with RL optimization (the left branch, part a.), and the map-free end-to-end RL-based motion planning methods (the right branch, part b.).

III. Map-based Classical Motion Planning Algorithms with RL Optimization

The research of classical motion planning is relatively mature and systematic. However, there are several limitations of these works. For example, the global planner and the local planner of the planning architecture are independent of each other. Researchers need to develop them separately. In addition, to obtain the final executable trajectory, researchers have to solve the optimization problem with several constraints. This process increases the complexity of the planning method. Besides, the classical motion planner contains various submodules, each of which requires a tedious parameter adjusting process. As shown in Fig. 2 (the left branch), there are considerable works dedicated to combining the advantages of the RL and the classical motion planner to achieve better planning results. This section provides an overview of these classical motion planning methods combined with RL improvements.

A. Global Planners Combined with RL Improvements

Some researches are devoted to combining RL algorithms with classical global motion planning methods to address some bottlenecks such as long-distance navigation, dynamic obstacle interaction, etc.

In [59], Aleksandra Faust et al. design an approximate value iteration (AVI) RL-based motion planning architecture to help the drone with a suspended load to find a safety trajectory with minimal residual oscillations in the static environment. The results of the simulation and the actual experiment show that their planning method has stronger robustness and better scenario generalization ability. Later, they extend this approach in [60]. They apply the global planner PRM to generate a collision-free path in the global static environment and utilize the RL agent as a local planner to generate the swing-free trajectory for the rotorcraft. This improved task decomposition scheme reduces the task difficulty, narrows the action space of the RL agent, and effectively improves the training speed.

In [61], Aleksandra Faust et al. further propose the PRM-RL algorithm to optimize the indoor long-range navigation problem over complex maps. They utilize PRM to partition an entire planar map into multiple local maps, and deploy the local navigation algorithm on each partial map. Different from previous research routes, the PRM-RL can work in the continuous action space with better control accuracy. The pipeline of PRM-RL consists of three phases. In the first phase, the DDPG agent [62] or the continuous action fitted value iteration (CAFVI) [63] agent is trained to learn the policy of point-to-point navigation. These pre-trained RL agents play the role of local planners. They can choose the optimal motion commands and transfer these motion strategies to the next phase. In the second phase, the RL-improved PRM can generate the roadmap of the deployment environment. In this global roadmap, the connection between any two configurations is a dynamic feasible trajectory output by the RL agent instead of a straight line. In the last phase, a graph search algorithm is deployed to search the optimal path from the roadmap. In [64], Aleksandra Faust et al. continue to follow their research route and present RL-RRT. Similar to the previous work, RL-RRT uses a DDPG agent to learn a local collision avoidance policy. The DDPG agent acts as a local planner to generate dynamically feasible trajectories to replace the steer function in the RRT framework. RL-RRT avoids the process of solving the optimal boundary value problem (OBVP) in kino-dynamic RRT algorithms [30]. Later, Aleksandra Faust et al. apply AutoRL [65] technology to the previous PRM-RL [66]. Based on the covariance matrix adaptation evolution strategy, AutoRL can automatically adjust the parameters of different reward terms and the architecture of the policy networks. AutoRL-based PRM-RL avoids the tedious process of hyper-parameters tuning, improves the algorithm accuracy,

and makes the motion planner more robust to the sensor noise and the environmental uncertainties.

B. Local Planners Combined with RL Improvements

Some researchers integrate RL modules in local planners. Their purpose is to give the robot a stronger ability to operating in unstructed, dynamic, and uncertain environments.

Utsav Patel et al. propose the hybrid DWA-RL motion planning approach [67]. They utilize the RL algorithm as the upper-level policy optimizer and adopt DWA as the low-level observation space generator. DWA-RL introduces the benefit of the DWA to perform kino-dynamic feasible planning and uses RL to select the optimal velocity commands to maximize the global returns for complex environments. Panagiotis Rousseas et al. integrate the artificial harmonic potential fields (AHPF) with the RL algorithm [68]. They retain the efficient and real-time features of AHPG, while utilizing the value iteration to improve the planning policy iteratively. AHPG-RL endows the local planner with optimality. Lu Chang et al. propose the Q-learning-based DWA [69]. The evaluation function of the classical DWA approach is crucial for the final planning result. In different scenarios, the proportion of each weight of the evaluation function should be different. Q-learning-based DWA uses a Q-learning RL module to auto-tune the weights in the DWA evaluation function at each timestep to improve the optimality of the planner.

In some works, researchers choose to leverage RL algorithms to help robots better predict behavioral patterns of uncertain factors (like pedestrians) in the surrounding environment. In [70], Pérez-D'Arpino et al. present a soft actor-critic (SAC)-based local planner for constrained indoor navigation problems with pedestrian participation. This hybrid planner is composed of the planning module and the RL module. The planning module is responsible for generating feasible waypoints on the basis of the prior map. These waypoints are utilized as guidance and concatenated with the Lidar data and the goal state as input to the policy network of the SAC agent. The output of the policy network is the speed controller command. The planning module helps the robot reach the target. The RL module focuses on learning different interaction patterns (slowing down, detouring, going backward, etc.) between robots and pedestrians.

IV. RL-based Mapless Motion Planning Methods

In some works reviewed in the previous section, RL is only used as a supplement and improvement module to the classical motion planner. In this section, we shift the focus of the review to the end-to-end RL-based motion planning methods. These motion planners are map-free and realize the unification of the global planner and the local planner. Researchers do not need to construct and maintain a prior geometric map of the operating environment that directly affects the final effect of the motion planning. In addition, compared with some supervised learning-based mapless motion planning methods [71], [72], RL-based motion planning methods can learn and evolve directly from the interaction data between robots and external environments. This mode avoids the construction process of the complex labeled expert dataset.

As shown in Fig. 2 (the right branch), RL-based mapless motion planning approaches can be further divided into two types: agent-level methods and sensor-level methods. Agent-level methods are based on the preset state estimation process and can directly acquire the upper-level state information of the environment. The agent-level methods are easier to train, which allows the robot to obtain optimal planning strategies faster. The observation of the agent-level method has a lower dimensionality, and contains more useful information. The state of the agent-level methods can be equivalent to the output after feature extraction of the raw sensor data. More importantly, the simulator of agent-level methods are much less difficult to develop. However, such methods rely on the assumption of perfect perception [73], or require to consider the observation encoding issue [74], or need extra communication mechanisms to share the state information [75]. These restrictions limit the scalability and application of agent-level methods.

Sensor-level methods are end-to-end. These methods directly establish the nonlinear mapping from the raw sensor data to the planning decisions. Although the offline training process is more time-consuming than agent-level methods, sensor-level methods do not rely on the perfect sensing assumption. Since the observation input dimension of sensor data is fixed at each time step, there is no necessity to consider the encoding and representation problems in the dynamic environment. Thus, sensor-level methods have better scalability, scenario generalization, and sim-to-real capabilities than agent-level methods. This section is an overview of these sensor-level and end-to-end RL-based motion planning methods. According to the mainstream research trends, as well as the commonly used robot perception methods, we further divide sensor-level RL motion planning methods into two categories: laser range finder (LRF) based methods and visual-based methods.

1) Laser Range Finder based: LRF equipment is widely used in map modeling, mobile robot navigation, autonomous driving, etc. In this section, we summarize some state-of-the-art sensor-level RL-based motion planning approaches with LRF.

In [76], Oleksii Zhelo et al. consider motion planning problems of several specific scenarios, including long corridors, dead corners, etc. that are not suitable for RL-based planners to learn the optimal or near-optimal policies. Different from some works that pre-define the reward form of navigation [77], [78], researchers introduce the intrinsic curiosity module [79] to help RL agent obtain the intrinsic reward. This exploration trick helps the planner acquire better generalization ability in the 2D virtual environment. Likewise, in [80], Haobin Shi et al. also introduce the intrinsic curiosity module and present a more general end-to-end motion planner based on the A3C framework with the input of the sparse Lidar data. They successfully deploy their planner from the physical engine to the realistic mixed scene. Yuanda Wang et al. consider the end-to-end motion planning task in the static virtual environment with dynamic obstacles [81]. They decompose

the whole motion planning task into an obstacle avoidance subtask and a navigation subtask. The collision-free planning module takes the raw sensor data of LRF as input and outputs a 5-dimensional force vector. It is important to note that the Q network in their planner has two streams: the spatial stream and the temporal stream. The spatial stream deals with the raw sensor data, while the temporal stream processes the difference between two consecutive frames of ranging data. On the contrary, the navigation module is training by a conventional Q network architecture. Experiment results show that this approach could obtain a high-performance motion planner in the 2D dynamic simulation environment. In [82], Ning Wang et al. find those extant methods generally require retraining the RL agent in different motion planning scenarios to reduce the generalization error caused by environmental changes. To overcome this catastrophic forgetting problem, they propose the elastic weight consolidation DDPG (EWC-DDPG)-based motion planning algorithm. EWC-DDPG enables the RL agent to acquire continuous learning capabilities without forgetting previous knowledge. This feature allows the planner to generalize across different scenarios.

Most of the above motion planners can only be deployed in the numerical simulation space. As we all know, the sim-to-real problem has been hindering the real-world application of those learning-based motion planners. In [83], asynchronous DDPG (ADDPG)-based motion planning algorithm is applied in mapless navigation of the differential WMR. The action space consists of the velocity and the angular velocity of the robot at each time step. The state space in the training stage includes three parts: (1) the 10-dimensional sparse findings of LRF. (2) the 2-dimensional previous action of the DMR. (3) the 2-dimensional relative position of the goal. A goal-reaching reward is set when the WMR approaches the target point, and a certain penalty is given when the DMR collides with the obstacle. Otherwise, $r_t(s_t, a_t) = C(d_{t-1} - d_t)$. C is a hyper-parameter, $d_{t-1} - d_t$ is the distance difference between the robot and the target point in adjacent timesteps.The whole motion planner is training in VREP [84] virtual engine and inference in the realistic scenarios. The final results show that the proposed ADDPG-based motion planner is more robust during pedestrian interaction than *Move Base*. Linhai Xie et al. present assisted-DDPG (AsDDPG) for training agents to learn local planning policy in the realistic and static obstacle environment without the prior map [85]. This DRL framework integrates DDPG with a classical controller (like a PID controller) to replace the random exploration strategy (e.g., ϵ-greedy). The classical controller can output control policy based on the position error between the current position and the goal. It should be noted that the triggering of this controller is determined by the DQN branch in the whole architecture. Sim-to-real experiments suggest that this trick is able to accelerate and stabilize the training phase effectively. In [86], Manh Luong et al. present an incremental learning paradigm to address the inefficient training issue in sensor-level RL-based motion planners. The incremental learning phase in the training stage is beneficial to optimize the current policy before loop termination. The researchers utilize sim-to-real technology to verify the performance of their motion planner on the *Gazebo* simulator and the real *Pioneer P3-DX* robot platform. Furthermore, in [87]–[89], researchers expect to endow the MR with social safety awareness while performing end-to-end motion planning tasks. Training robots to safely and carefully interact with pedestrians in the environment instead of simply treating them as static or dynamic obstacles.

In [87], Yuxiang Cui et al. develop a model-based RL motion planning method with social safety awareness. They first obtain priori data from the interaction process between the robot with the realistic scenarios and utilize this dataset to train a world transition model with pedestrian participation in the self-supervised learning paradigm. Finally, they concatenate the real data with the virtual data generated by the world environment model as the observation and input it into the RL architecture to train the planning policy. In [88] and [89], researchers introduce rectangular social-safety zones for robots and pedestrians, and design the corresponding safety interaction reward term based on these zones. Moreover, they adopt end-to-end RL framework and train the motion planning policy in the mixed and complex environment.

2) Vision sensors: In addition to Lidar sensors, vision sensors are also widely used as sensing modules for mobile robots. Many scholars have found that the end-to-end planning and decision-making for robots can also be achieved based on the raw visual data input. DRL framework combined with convolutional neural networks (CNNs) is naturally suitable for this task. In [77], researchers in DeepMind propose the NavA3C algorithm to teach the agent to find the goal and navigate in different 3D mazes. This work encourages robots to learn motion planning policies while learning auxiliary tasks such as environmental loop closure detection and image depth prediction. The whole architecture of NavA3C is shown in Fig. 3. The inputs contains RGB visual input $\mathbf{x}_t$, past reward r_{t-1}, previous action $\mathbf{a}_{t-1}$, and the agent-relative velocity $\mathbf{v}_t$. The outputs including motion policy π, value function V, depth predictions $g_d(\mathbf{f}_t), g'_d(\mathbf{h}_t)$, and closure detection $g_l(\mathbf{h}_t)$. It should be noted that the closed-loop detection and the depth prediction in this paper are based on the supervised learning method. Optimizers need to aggregate all gradients of various loss functions from different tasks in the parameters updating process. Multi-group experiments prove that this trick of adding auxiliary tasks can avoid reward sparsity issues, improve the richness of the training samples, and improve the training efficiency. Later, researchers in DeepMind extend this work, and apply NavA3C to outdoor navigation scenarios on the basis of the realistic street panoramas dataset of Google [90]. Inspired by NavA3C of DeepMind, Jonáš Kulhánek et al. design auxiliary tasks in the planning architecture to facilitate the domain randomization of the model in simulation environments [91]. They directly utilize unlabeled images of segmentation masks that are not readily available in the environment. This operation can significantly improve the planning effect when the planner is deployed in the actual physical environment.

In [92], Li Fei-Fei et al. propose a DRL-based target-driven visual navigation approach for indoor scenarios. Unlike previous works about visual navigation, their method has a stronger generalization ability to various environments and can

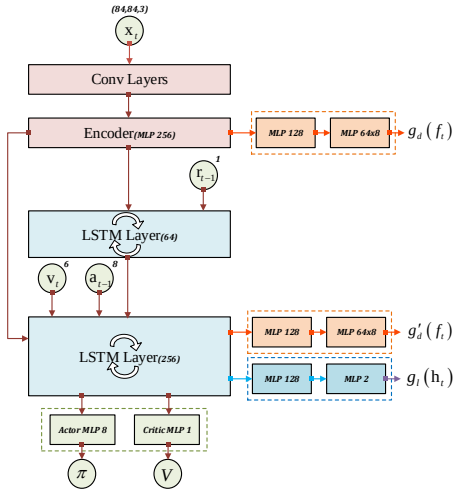

Fig. 3. The architectures of NavA3C visual navigation method with several auxiliary tasks including depth prediction and closure detection [77].

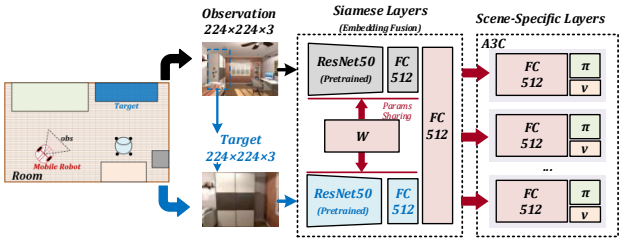

Fig. 4. The overall architecture of the deep siamese actor-critic based target-driven indoor navigation algorithm [92]. This algorithm takes the current observation and the goal image as the input. Two parameter-sharing ResNet-50 are responsible for encoding these two batches of observations, and generating embedding vectors. After fusion operation, the vectors are converted into one embedding vector and input to the A3C as the state input. The A3C scene-specific layers output the final action policy of the mobile robot.

be easily deployed to the realistic world just by fine-tuning. The framework of their method is shown in Fig. 4. The inputs include the current observation image and the target image. Weights-sharing siamese networks transfer the input features to the same embedding space. Scene-specific networks based on the A3C output the motion policy and the action value V. Li Fei-Fei and her group have pioneered target-driven visual navigation. They also develop and open-source a high-quality 3D indoor simulation platform: The House Of inteRactions (AI2-THOR) which has subsequently facilitated other scholars to conduct their visual navigation research (e.g., [93], [94])

Above visual-based motion planning methods are based on the premise that the static target is known. Researchers in Peking University and Tencent AI Lab have been jointly working on the visual-based end-to-end motion planning of MRs in active dynamic target tracking scenarios. Their methods utilize the Conv-LSTM network to establish the mapping from the raw sensor image to the control command of the actuator. Visual navigation is generally challenging to achieve the sim-to-real process due to the gap between the simulator and the realistic scene. Aiming at this issue, the researchers perform environment complexity augmentation and virtual model detail augmentation to enhance the robustness and generalization of the algorithm [95], [96].

V. RL-based Multi-robot Motion Planning

There exist several performance limitations in single-robot motion planning operation, such as limited sensing range, low mission reliability, and time inefficiency in some specific sceanrios (e.g., underwater searching, hospital disinfection, etc.). Collaborative motion planning of multiple robots is more flexible, robust, and efficient. Therefore, it is widely applied to marine exploration, smart agriculture, disaster rescue, etc [2]. Unlike the Markov properties of the single-agent RL-based motion planning methods, the RL-based multi-robot motion planning (MRMP) procedure requires consideration of the influence of the local observability and the uncertainty of the environment. Therefore, most of the mainstream works extend the interaction process of robots with the environment from the Markov decision process (MDP) to the partially observable Markov decision process (POMDP) or decentralized POMDP (Dec-POMDP). The major RL-based MRMP research works can be further divided into two main categories: centralized RL-based MRMP (CeRL-MRMP) methods and decentralized RL-based MRMP (DecRL-MRMP) methods. CeRL-MRMP is simple and intuitive. It uses a centralized Q network to build a map from joint trajectories of all agents to the global action-state value and aims at learning the joint planning policy that maximizes the total rewards [97]. CeRL-MRMP methods have some bottlenecks,such as the dimensionality problem of the joint space representation, the exploration problem of the high-dimensional joint policy, and the scalability problem [2]. DecRL-MRMP can be further subdivided into the independent MRMP and the centralized training and decentralized execution (CTDE)-based MRMP. Independent MRMP has better scalability [98]. Each robot only gets partial information of the environment and cannot directly obtain the action policies of other active agents. In independent MRMP, every agent is self-interested and only considers how to maximize own return. Therefore, this type of methods exists credit assignment problem. And for each agent, the entire environment changes dynamically in each timestep, which further leads to the convergence difficulty of the algorithm.

CTDE-based methods incorporate the advantages of the above paradigms. During the training procedure, each agent can extract global environmental information through a variety of centralized methods, such as total action-state value [99], historical trajectories from the global experience buffer [100], and other sensor-based explicit approaches like [101]. During the planning execution process, each robot only requires its own local observations to make online inference decisions. In the section, we give an overview of some of these representative works.

In [2], [100], [102], researchers focus on improving CTDE-based multi-agent RL (MARL) algorithms. They usually adopt the cooperative navigation scenario of Multi-agent Particle Environment (MPE) [100] as an experimental benchmark task to test the performance of their proposed MARL algorithms. Specific optimizing objectives of these works include the efficiency of information sharing, the cooperative ability of

agents, overall convergence speed, credit assignment strategy, etc. However, agents in MPE navigation environment are too idealized and do not match the characteristics of real MRs. Also, the planner ignores the kinodynamic constraints of the robot in the process of state updating. These limitations hinder the deployment of these algorithms in realistic scenarios.

Researchers in the ACL laboratory of MIT have made great achievements in the field of DecRL-MRMP. Michael Everett et al. are dedicated to studying the problem of MRMP in complex and dynamic scenarios without communication [74], [103]–[105]. They describe this type of problem as a sequential decision-making problem. In a n-agent scenario, the state vector of agent i is $\mathbf{s}_\mathrm{i}$, and the observation vector of other $n-1$ agents (MRs or pedestrians) is $\widetilde{\mathbf{S}}_i^o$. $\mathbf{s}_\mathrm{i} = [\mathbf{s}_\mathrm{i}^\mathrm{o}, \mathbf{s}_\mathrm{i}^\mathrm{h}]$ where $\mathbf{s}_i^o = [p_x, p_y, v_x, v_y, r]$ represents observable states including the position, velocity and radius of the agent i and $\mathbf{s}_i^h = [p_{gx}, p_{gy}, v_{pref}, \psi]$ represents the unobservable states including the position of goal, the preferred velocity and the orientation of agent i. The continuous action space for the agent i at time step t is $\mathbf{u}_\mathrm{t} = [v_t, \psi_t]$, where v is the speed and ψ is the heading angular. π_i is the policy. The objective of each agent i is to develop a policy π^* to minimize the time consumption t_g from the start position to the goal with several constraints. The details are as follows. Eqs.(2)-(4) respectively represent the safety constraint, the target point constraint and the kinodynamic constraint [105].

$$\underset{\pi_i}{\operatorname{argmax}} \quad \mathbb{E}[t_g|\mathbf{s}_\mathrm{i}, \widetilde{\mathbf{S}}_i^o, \pi_i] \tag{1}$$

$$s.t. \ \left\|\mathbf{p}_{\mathrm{i,t}} - \widetilde{\mathbf{p}}_{\mathrm{j,t}}\right\|_2 \geq r_i + r_j \quad \forall i \neq j, \ \forall t \tag{2}$$

$$\mathbf{p}_{\mathrm{i,t_g}} = \mathbf{p}_{\mathrm{i,goal}} \quad \forall i \tag{3}$$

$$\mathbf{p}_{\mathrm{i,t}} = \mathbf{p}_{\mathrm{i,t-1}} + \Delta t \cdot \pi_i \quad \forall i \tag{4}$$

In [103], Michael Everett et al. propose the collision avoidance with deep RL (CADRL) algorithm and apply it to solve (1)-(4). They adopt a non-communicating and offline CTDE MARL paradigm to efficiently avoid the time-consuming online computing process in some classical motion planning methods. In the training phase, they utilize a value network V to evaluate the performance of the current policy and iteratively retrieve the optimal time-efficient motion policy from this value function through $\pi^* = \underset{\mathbf{u}_\mathrm{t} \in \mathrm{U}}{\operatorname{argmax}} \ R([\mathbf{s}_i^o, \mathbf{s}_i^h], \mathbf{u}_\mathrm{t}) + \gamma V(\hat{s}_{t+1}, \widetilde{\hat{S}}_{t+1}^o)$. CADRL is real-time and has great superiority compared to other mainstream methods. However, the cooperative behaviors of each agent in CADRL-based MRMP method cannot be controlled. In [104], they further extend CADRL and propose the socially aware CADRL (SA-CADRL) algorithm. SA-CADRL introduces social behaviors to the multi-agent motion planning task (The social behaviors here mainly refer to various interacting patterns between pedestrians and MRs). Unlike the existing model-based or learning-based approaches, SA-CADRL integrates the behavior rule of humans (time-efficient rule) and the social norms (passing on the right and overtaking on the left) into the reward function of the RL architecture. Moreover, they deploy the SA-CADRL-based cooperative planner on real MR hardware platform to realize automatic navigation

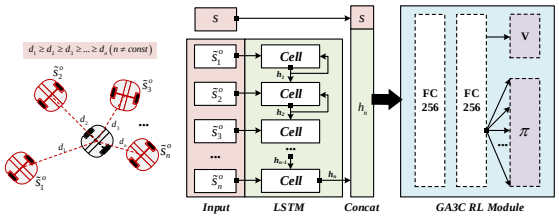

Fig. 5. The overall architecture of GA3C-CADRL in [74].

at human-walking speed in pedestrian-rich environments. In [74], Michael Everett et al. further consider the stochastic behaviour model and the uncertain number of other agents in the environment on the basis of CADRL and present the GA3C-CADRL MRMP algorithm. The most critical contribution of GA3C-CADRL is that it can tackle the agent-level state representation issue where the number of obstacles or agents in the environment varies randomly. Due to the fixed input dimension constraint of neural network, some researchers choose to define a maximum number of agents and pad the excess space with zero. The utilization of these tricks would introduce several issues like additional parameter calculations and the observation sparsity. Also, compared to those sensor-level MRMP approaches like [83], Michael Everett et al. hope to extract an agent-level representation that implies the motion plans of other agents. In response to these challenges, they propose an LSTM-based encoding method for environment information and integrate it into the actor-critic RL framework. The details are shown in Fig. 5. $\mathbf{s} = [\left\|\mathbf{p}_{goal} - \mathbf{p}\right\|_2, v_{pref}, \psi, r]$ represents the observation of the agent itself and $\widetilde{\mathbf{s}}^o = [\widetilde{p}_x, \widetilde{p}_y, \widetilde{v}_x, \widetilde{v}_y, \widetilde{r}, \widetilde{r} + r, \|\widetilde{\mathbf{p}} - \mathbf{p}\|_2]$ represents the observation of other agents in the vicinity. Whatever the dimension of $\widetilde{\mathbf{s}}^o$ is, the final output h_n can encodes the entire the observation of environment in a fixed-length vector. It is worth noting that Michael Everett et al. open-source their code of simulation environment, which provides a studying platform for other researchers. However, GA3C-CADRL still has certain limitations. First, GA3C-CADRL integrates the supervised learning module. This module relies on the prior dataset and increases the difficulty of training process to some extent. Also, GA3C-CADRL does not improve the reward function in CADRL and still has the reward sparsity issue. To overcome these problems, Michael Everett et al. further improve GA3C-CADRL and propose GA3C-CADRL-NSL [106]. They replace the original navigation reward form with a special goal-distance-based proxy reward function and eliminate the supervised learning stage. The detailed form of this reward function is given in (5).

$$\begin{aligned} R(s^{jn}) &= R_c + R_g \\ R_c &= \begin{cases} -1 & \text{if } d_{min} < 0 \\ 10 d_{min} - 1 & \text{if } 0 < d_{min} < 0.1 \\ 0 & \text{otherwise} \end{cases} \\ R_g &= \begin{cases} 1 & \text{if } p = p_g \\ \propto (\mathrm{goal}_{\mathrm{dist}}^{t-1} - \mathrm{goal}_{\mathrm{dist}}^{t}) & \text{otherwise} \end{cases} \end{aligned} \tag{5}$$

where R_c is responsible for monitoring d_{min} which represents the distance between the current agent i and its closest

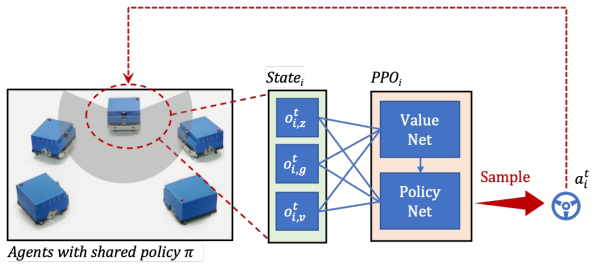

Fig. 6. The overall framework of the distributed PPO-based multi-scale mobile robots motion planning algorithm in [78].

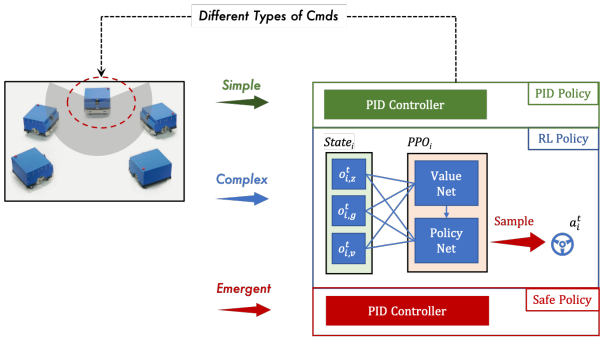

Fig. 7. The hybrid motion planning framework in [101]. The robot will choose an appropriate action command according to the type of the current external environment.

agent, and punishing dangerous actions. R_g is responsible for rewarding goal-reaching discrete action pairs. This reward shaping trick can generate continuous reward signals. It should also be noted that GA3C-CADRL-NSL introduces a hybrid motion planning architecture that combines DRL and force-based motion planning(FMP) [107] method. Once the mobile robot falls into high-risk situations, the FMP algorithm will take effect and help the robot get out of trouble.

Jia Pan et al. have been working on DecRL-MRMP for large-scale multi-robots in dense environments [78]. Different from previous works [74], they pay more attention to the sensor-level motion planning methods. They argue that the assumptions underlying the agent-level approaches are too strong and not general. In addition, they think that agent-level methods have to cooperate with environmental encoding methods to obtain the fixed-dimensional observation input in a dynamic environment [103]. In their CTDE-based MRMP approach, each robot is independent of others, which makes their method has strong scalability [108]. During the training phase, rewards, policy networks, and value networks are shared among each robot. Also, the shared transition samples are utilized to guide the development of implicit collaboration mechanisms. The main idea of [78] is shown in Fig. 6. Later, they combine this method with the PID controller and propose a hybrid architecture that can be deployed in real-world scenarios [101]. The detail is shown in Fig. 7.

VI. Discussion

At present, RL-based motion planners still face plenty of challenges. These challenges stem from the characteristics of the algorithm itself, the limitations of sensors, and the external environment. Therefore, many factors still need to be considered before realizing the large-scale deployment of the RL motion planner to real navigation scenarios. In this section, we mainly analyze and summarize these challenges. The specific contents are summarized in Fig. 8. Then, in conjunction with these analyses, we provide suggestions for future research directions.

A. Challenges

1) Reality Gap: There is a real-world gap in applying DRL to realistic robotic tasks. For instance, unexpected actions may cause potential safety problems (a real robot could cause real damage) of robots in real-world scenes, low sample efficiency in the real world may lead to convergence difficulty of the training process. Besides, sensors and actuators of real robots cannot be as ideal as virtual environments, which brings plenty of uncertainties. At present, many scholars in Robotics have committed to the research of innovative Sim-to-Real methods [109]. Mainstream Sim-to-Real approaches include domain adaption methods [110], disturbances learning-based robust methods [111], domain randomization methods [91], [112], knowledge distillation methods [113], etc.

As for motion planning tasks, training planning policy in the simulation platforms with the physical engine (Some popular platforms include CARLA, Pybullet, CoppeliaSim, Gazebo, Unity 3D, etc.) and transferring to the real-world navigation scenario is a commonly used research pipeline to alleviate the influence of the reality gap problem [114], [115]. For example, Thomas Chaffre et al. present a depth-map-based Sim-to-Real robot navigation method. They first set up several scenarios with increasing complexity in the Gazebo platform for incremental training. In the real-world scenario training stage, the learning phase is deployed on the fixed ground truth Octomap and utilizes a PRM* path planner to ensure safety in the resetting stage of every episode. The velocity and the angular velocity commands are output to the low-level controller of the MR in the testing phase [112]. Jingwei Zhang et al. handle the Sim-to-Real motion planning problem by adapting the real camera streams to the synthetic modality during the actual deployment stages. This operation is lightweight and flexible, and could transfer the style of realistic image to the simulated style, which can be used in the stage of training RL agent [110]. Similarly, Jing Liang et al. propose a brand new learning-based local navigation approach named CrowdSteer to solve the motion planning problem of MRs in dense environments [116].

2) Sparse reward problem: The motion planning process of mobile robots is often target-driven. The positive reward of the environmental feedback is generally at the final goal point. For long-distance navigation in obstacle environments, it is hard for robots to obtain final positive reward signals. Besides, long-term training of robots under the negative reward signals might develop abnormal behavior patterns , such as timidity. Moreover, the sparse reward problem can lead to slow learning problems and convergence difficulties.

Curiosity-driven is a way to solve the problem of sparse rewards using existing trajectories [79]. The main idea of this

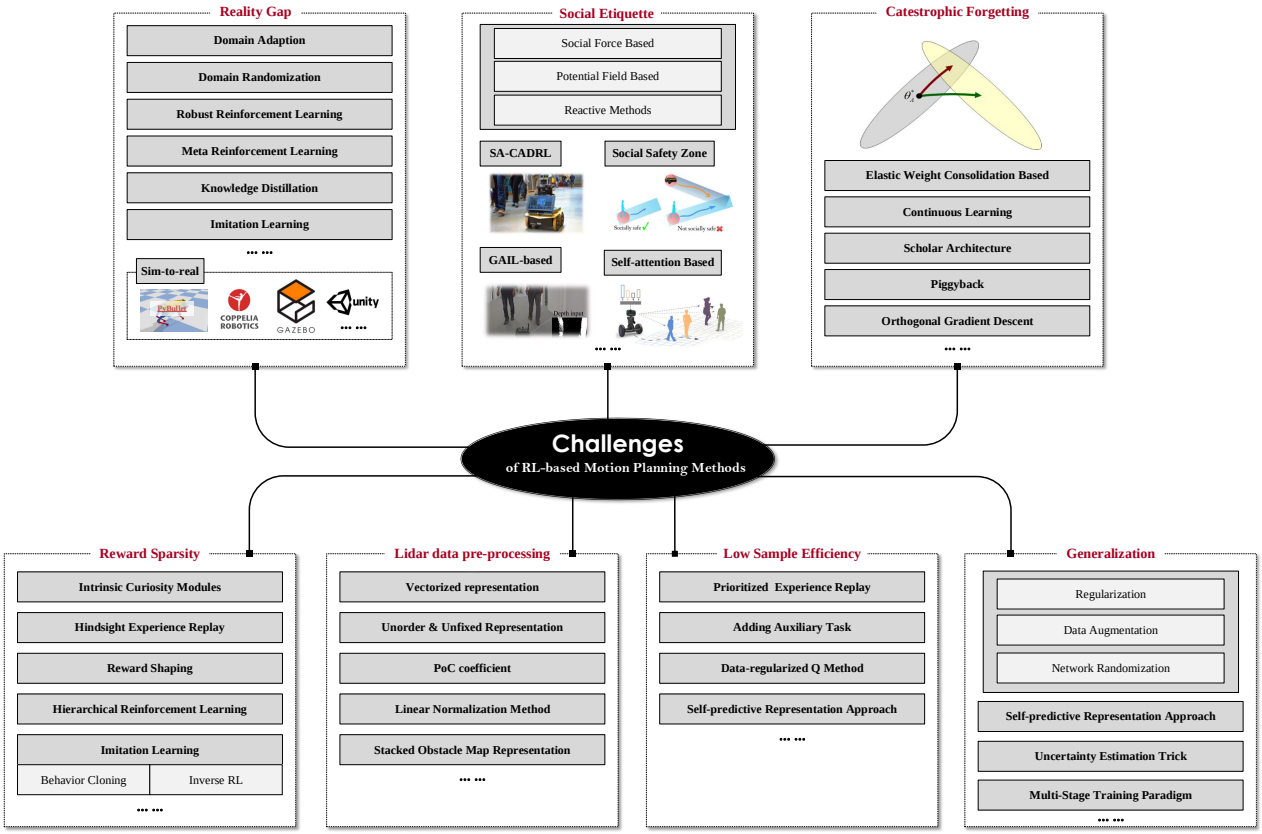

Fig. 8. A summary diagram of several issues existing in RL-based motion planning methods. Current RL-based motion planning approaches have many performance limitations, such as reality gap, reward sparsity, generalization, low sample efficiency, social etiquette, lidar data pre-processing issue, catastrophic forgetting problem.

method is to build intrinsic curiosity modules (ICM) to extract additional intrinsic reward signals from the environment to encourage more effective exploration of the agent. In [76], [80], researchers choose to utilize this trick to develop map-free and end-to-end motion planning frameworks of MRs. Hindsight experience replay (HER) [117] is another approach used to solve the reward sparsity problem with existing data. HER is based on the multi-objective RL algorithms. The main idea of this algorithm is to encourage learning from unrewarded states. By mapping the unrewarded state as the new target and replacing the previous target, the agent is encouraged to explore and obtain additional reward signals during the training process. Different from above methods, reward shaping is more intuitive. Researchers using this trick must be patient and manually adjust and modify the refined reward signal values of the robot under different states [118]. So, reward shaping skill is highly dependent on expert experience. An improper reward can lead to the change of the optimal policy and cause the anomalous behaviors of the agent [119]. Another type of method to solve reward sparsity is the hierarchical RL (HRL). HRL tends to decompose the original task hierarchically into multiple discrete or continuous and easy-to-solve subtasks, and then divide and conquer to provide the agent with dense reward signals. Besides, there are already some researchers studying HRL-based motion planners, such as [120], [121]. For some more specific and complex planning tasks, the reward functions are hard to configure. Expert demonstrations can be utilized to help mobile robots learn better, i.e., imitation learning. Common imitation learning paradigms include behavior cloning and inverse RL (IRL). Behavior cloning relies on a supervised learning process and suffers from the mismatch problem between the actor and the expert policy. In [122], Zijing Chi et al. utilize this method to help the robot develop the collision avoidance ability. In contrast, IRL is not just a simple imitation of expert behaviors. IRL makes good use of the expert trajectories to learn the reward function in the RL architecture in reverse and performs the policy optimization after obtaining the reward function. IRL methods are commonly applied to the autonomous driving field [123], [124].

3) Generalization: The generalization ability of the RL-based motion planners determines whether the mobile robot can perform safe and reasonable motion primitives in unseen scenarios different from the training stage. Most of the RL-based motion planners rely heavily on the inference performance of neural networks. However, for the unpredictable far-from-training test cases or the out-of-distribution test data, neural networks cannot guarantee the security and effectiveness of the planning process [111]. For example, cleaning robots have to learn to interact with pedestrians in a school campus. If the robot cannot correctly predict the movement intention of passers-by, there will be some potential

safety hazards. At the algorithm level, many scholars design improving methods to enhance the generalization ability of RL agents, such as the regularization method [125], the data augmentation trick [126], etc. In [127], Kimin Lee et al. propose a method to improve the generalization ability of the RL agent across different tasks by using random neural networks to generate random observations. In addition, some generalization enhancing techniques have been applied to the RL-based motion planning methods. In [101], Tingxiang Fan et al. adopt the multi-stage stochastic training policy to improve the generalization of the MR in different environments. In practical applications, the pre-trained RL policy is combined with a PID policy and a safe policy to ensure the stability and safety of the robot in unseen scenarios. In [111], a safe RL architecture is proposed to handle dynamic collision avoidance problems in novel scenarios. Unlike the traditional RL framework, this method integrates the collision prediction networks based on the LSTM ensemble, the uncertainty estimation based on Monte-Carlo dropout, bootstrapping process, and the safest action selection method based on the model predictive control (MPC). The final results demonstrate that this type of uncertainty-aware pipeline endows the motion planner with stronger robustness and generalization.

4) Low sample efficiency: Unlike the manner of human learning, the RL agent learns from scratch for different tasks. In the early stage of training, too much useless transitions make it difficult for the agent to learn and update effectively. Therefore, how to better explore valuable strategies and enhance the sampling efficiency of valuable experience are the hot directions in the RL-related research, and they are also the key to improving the performance of RL-based motion planners.

Schaul et al. propose prioritized experience replay (PER) method [128]. In order to improve the learning efficiency, Schaul et al. compute the probability of a sample according to the importance (i.e., temporal-difference error). PER is indeed an effective trick to improve the efficiency of off-policy RL methods. There is already some work applying PER to RL-based motion planning methods [129], [130]. Lasjub et al. utilize constractive unsupervised representations as the auxiliary task to speed up sample efficiency. [131]. They extract valid features from the raw sensor data through the comparative learning process and then feed the features into the RL module. Kostrikov et al. proposed Data-regularized Q (DrQ) method [132]. They do augmentation on the observation input before starting the sampling training process and calculating the target-Q and the current Q simultaneously. Also, through combining with the regularization trick, the sample efficiency of the raw input data is significantly improved. Schwarzer et al. propose the self-predictive representation (SPR) approach [133]. They improve the sample efficiency by training agents to predict multi-step representations of their potential future states. This operation allows agents to learn temporally predictive and consistent representations under different environmental observations.

5) Social etiquette: At present, more and more robots are deployed in crowded places such as airports and stations. In these specific scenarios, mobile robots using classical motion planners may cause the freezing problem (Robots cannot find any feasible action) because the probabilistic evolution of pedestrians could expand to the entire workspace [134]. Therefore, it is challenging but essential to deploy learning-based algorithms to train MRs to learn social etiquette and interact with humans in a safe, effective, and socially compliant manner.

Pioneer research works including social force-based methods [135], potential field-based methods [136], reactive methods (RVO, ORCA) [137], [138], etc. These methods are overly dependent on the hand-crafted process and lack a certain generalization ability for complex scenarios. Some works simplify the motion pattern of pedestrians. They treat pedestrians as static obstacles or dynamic obstacles with simple kinematics over short timescales [103], [139]. The effectiveness of these approaches is based on several strong assumptions. When the planner is deployed to the real dynamic environment, robots may produce unsafe decisions due to their inaccurate prediction of human behaviors.

Typical RL-based methods including SA-CADRL [104], GAIL-based planning methods [140], etc. These methods effectively constrain the specific interaction norm between the robot and the pedestrian, but rely on effective explicit pedestrian detection approaches. In [88] and [89], researchers construct rectangular social-safety zones for the MR and pedestrians respectively, and design corresponding safety reward terms. In [141], Changan Chen et al. propose a self-attention and deep V learning-based agent-level crowd-aware robot motion planning approach. They consider more practical crowd-robot interaction patterns rather than the first-order human-robot interaction pattern problem. The state value estimation network of this deep V learning framework consists of three modules: an interaction module, a pooling module, and a planning module. The interaction module has a multi-layer perception (MLP) to extract the pairwise interaction feature between the robot and the nearby pedestrians. The pooling module outputs embedding tensors of above pairwise interactions by self-attention model. The final planning module estimates the state value based on previous compact embeddings.

6) Lidar data pre-processing issues: Lidar data represents the distance information between the MR and the surrounding environments. Compared with the visual sensors, Lidar data naturally contains depth information and is much easier to achieve the sim-to-real process. Therefore, Lidar is widely utilized in end-to-end motion planning tasks. However, improper Lidar data pre-processing may cause the degradation of the planning capability of the robot in unknown scenarios.

Many works have directly utilized the distance vector read from Lidar as part of the observation in the RL framework (e.g., [83], [142]). This operation may cause some issues. For example, if the Lidar observation at a certain timestep in the testing scenario is similar to the training scenario but with different passability, the agent may not make different action decisions. Also, if the Lidar data occupies most of the dimensional space in the observation input, the agent may not have a good goal-reaching ability for planning in obstacle-free scenarios. Francisco Leiva et al. propose an unordered

Lidar data representation method with the non-fixed dimension [143]. This method integrates the relative distance and the orientation information of obstacles, making the whole motion planning algorithm more robust. Wei Zhang et al. present a Lidar data preprocessing approach with the parameter self-learning mechanism [144]. They introduce the PoC (proportion of distance values considered "close") ratio coefficient to differentiate similar scenarios and help the MR judge the complexity of the surrounding environment. Yuxiang Cui et al. in [87] find that the stacked obstacle map generated based on the 2D laser scan data has a lower reconstruction error and can represent the difference between the static and dynamic obstacles in the environment more accurately than the angle range representation method.

7) Catastrophic forgetting problem: The catastrophic forgetting problem of RL-based motion planners refers to the forgetting of previously learned knowledge by agents when performing task-to-task continuous learning processes. Since the motion planning process of robots generally involves multiple optimization objectives, the weights that are important for previous tasks might be changed to adapt to a new task. If changes contain highly relevant parameters to historical information, the new knowledge will overwrite the old knowledge, resulting in catastrophic forgetting issues. Kirkpatrick et al. propose the elastic weight consolidation algorithm [145]. They solve the forgetting problem by calculating the Fisher information matrix to quantify the importance of the network parameter weights to the previous task. Next, they add this term as a regularization to constrain the update direction of the neural network while learning a new task. In [82], Ning Wang et al. combine EWC with DDPG algorithm and apply it to the multiple target motion planning task of the mobile robot. In [146], Shin et al. propose a scholar architecture with a generator and a solver. The old generator generates replay data and mixes it with the current task data as the training sets for the new task. This operation ensures that the new scholar does not forget the previous knowledge while learning a new task. Mallya et al. present Piggyback [147]. Researchers fix a backbone network and train a binary mask network for each task. Different binary masks are combined with the backbone network to perform different policies to simplify the computation process and improve reusability. Farajtabar et al. propose the orthogonal gradient descent method [148]. This approach reduces the forgetting problem of existing knowledge by orthogonally projecting the updated gradients of the new task on the gradient parameter space of the previous task.

B. Future Directions

1) Task-free RL-based general motion planner : A complete motion planning task generally consists of several subtask goals. The main idea of the commonly used continuous learning approach is to learn task by task and to overcome the problem of forgetting during task transferring. These operations cause the training phase to be multi-stage and cumbersome. Combining the multi-task motion planning process with the state-of-the-art task-free continuous learning paradigm can directly determine the state of the model based on the fluctuation information of the loss function. It facilitates breaking the rigid boundary between individual subtasks and trains the general motion planner that accomplishes multiple task goals.

2) Meta RL-based motion planning methods: Meta learning helps the model to learn how to learn by acquiring sufficient prior knowledge in a large number of tasks. In the process of training the RL-based motion planner, the meta learning mechanism can be introduced to inspire the robot to learn to inference in unknown environments. Meta RL can make the robot adapt to the planning task in the new environment quickly by using the prior knowledge gained from the previous planning experience, and improve the generalization ability of the motion planners.

3) Multi-modal fusion based RL motion planning methods: Multi-modality consists of two levels: the sensor level and the data representation level. At the sensor level, the utilization of single-type perception sensors has performance limitations. By combining mapless end-to-end motion planning methods with multi-sensor fusion techniques, the advantageous features of each perception module can be fully leveraged and complemented. Moreover, this fusion trick improves the environment cognition and understanding ability of robots and enhances the fault tolerance and robustness of motion planners. At the data representation level, multi-modal RL-based motion planning refers to integrating data features from multiple modalities (e.g., images, languages, etc.) during the training process. For instance, it is possible to improve the planning performance through integrating human language instructions with the visual information as the observation to improve the motion planning performance of the robot for real deployment applications.

4) Multi-task objectives based RL motion planning methods: A practical motion planner is often developed with multiple task objectives. For example, researchers usually expect the planner to guarantee the shortest path length and the safety of the MR while having as little time and energy consumption as possible during the whole process. Classical motion planner separates the planning process and the optimizing process. Therefore, it is worthwhile to research how to introduce multi-task objectives learning in map-less end-to-end planning architecture. The breakthrough point is to improve the generalization of the planning methods by treating the domain information contained in the training signals of related tasks as induction bias. This operation helps the algorithm learn general skills that could be shared and utilized across various related tasks, and maintain a competitive balance between multi-task objectives. Finally, The representation of the planning policy can be obtained by merging operation multiple task-specific policies into a unified single optimal policy.

5) Human-Machine interaction mode based motion planning methods: Most of the current planning application scenarios for RL-based motion planners are point-to-point. The robot learns independently throughout the planning loop. In more complex and changing unstructured environments, human intelligence can be coupled with machine intelligence. A prevalent human-in-the-loop approach is to endow the human

with the role of supervisor. The robot autonomously performs the human-assigned task for a period of time, then stops and waits for the next cycle of planning commands. This approach makes the MR unable to respond effectively to sudden external changes. Therefore, it is necessary to study how to integrate human and robot intelligence to improve the human-machine interaction motion planning performance in-depth, such as teaching by demonstration, learning based on human judgment and experience, etc.

6) RL-based motion planning of multiple heterogeneous MRs: Most of the multi-robot RL-based motion planning approaches in this survey work in 2D space, and each robot of the system shares the same action space and the same observation space. The multiple heterogeneous MRs system has a more extensive application domain (e.g., air-ground cooperation planning and air-sea cooperation planning, etc.). It can leverage the unique advantages of each single-structured robot in the system. On this basis, the advantages of the centralized critic and distributed actors architecture of the MARL could be utilized to realize the dynamic task allocation of each heterogeneous robot and achieve optimal joint planning decisions in 3D space without the prior map.

7) Multi-MR flexible formation planning methods: Multiple MR swarming and formation planning is widely used in military, logistics, transportation, intelligent agriculture, and resource exploration, etc. To realize the combination of the mainstream RL-based motion planner with the flexible formation, researchers can utilize the hierarchical learning paradigm and continuous learning paradigm or design multimodal reward function to implement the overall planning of the robots group and the formation-keeping within the robots group. In addition, another breakthrough point is to design a hybrid planning framework that incorporates RL-based motion planners and the mainstream formation algorithm to enhance the formation and planning performance.

VII. Conclusion

In this paper, we systematically review the state-of-the-art motion planning methods of mobile robots and give an overview of RL-based motion planners. There are three mainstream research directions: motion planners combined with RL improvements, map-free RL-based motion planning methods, and RL-based multi-robot cooperative planning methods. Although there are many representative research works, RL-based motion planners still have several performance bottlenecks that hinder its pratical application, such as reality gap, reward sparsity problem, low sample efficiency, generalization problem, catestrophic forgetting problem, social etiquette, Lidar data pre-processing issue, etc. At last, we analyze these challenges and predict the future directions of RL-based motion planning methods.

References

[1] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion control for mobile robot navigation using machine learning: a survey," *arXiv preprint arXiv:2011.13112*, 2020.

[2] C. Sun, W. Liu, and L. Dong, "Reinforcement learning with task decomposition for cooperative multiagent systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 5, pp. 2054–2065, 2021.

[3] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[4] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[5] L. Dong, X. Zhong, C. Sun, and H. He, "Event-triggered adaptive dynamic programming for continuous-time systems with control constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 8, pp. 1941–1952, 2017.

[6] L. Dong, X. Yuan, and C. Sun, "Event-triggered receding horizon control via actor-critic design," *Science China Information Sciences*, vol. 63, no. 5, p. 150210, 2020.

[7] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1366–1373.

[8] L. Han, F. Gao, B. Zhou, and S. Shen, "Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4423–4430.

[9] L. Quan, L. Han, B. Zhou, S. Shen, and F. Gao, "Survey of uav motion planning," *IET Cyber-systems and Robotics*, vol. 2, no. 1, pp. 14–21, 2020.

[10] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2019.

[11] H. M. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, S. Thrun, and R. C. Arkin, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[12] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.

[13] H. Wang, Y. Yu, and Q. Yuan, "Application of dijkstra algorithm in robot path-planning," in *2011 Second International Conference on Mechanic Automation and Control Engineering*, 2011, pp. 1067–1069.

[14] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[15] A. Stentz, "Optimal and efficient path planning for partially known environments," in *Intelligent unmanned ground vehicles*. Springer, 1997, pp. 203–220.

[16] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning A*," *Artificial Intelligence*, vol. 155, no. 1-2, pp. 93–146, 2004.

[17] D. Belanová, M. Mach, P. Sinčák, and K. Yoshida, "Path planning on robot based on D* lite algorithm," in *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*. IEEE, 2018, pp. 125–130.

[18] D. Harabor and A. Grastien, "Online graph pruning for pathfinding on grid maps," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, no. 1, 2011.

[19] Z. He, L. Dong, C. Sun, and J. Wang, "Asynchronous multithreading reinforcement-learning-based path planning and tracking for unmanned underwater vehicle," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–13, 2021.

[20] C. Jiang, S. Sun, J. Liu, and Z. Fang, "Global path planning of mobile robot based on improved JPS+ algorithm," in *2020 Chinese Automation Congress (CAC)*, 2020, pp. 2387–2392.

[21] V. Bulitko and G. Lee, "Learning in real-time search: A unifying framework," *Journal of Artificial Intelligence Research*, vol. 25, pp. 119–157, 2006.

[22] S. Koenig and M. Likhachev, "Real-time adaptive A*," in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, 2006, pp. 281–288.

[23] M. Pivtoraiko and A. Kelly, "Generating state lattice motion primitives for differentially constrained motion planning," in *Proceedings of the International Conference on Intelligent Robots and Systems*, 2012, pp. 101–108.

[24] M. Pivtoraiko and A. Kelly, "Kinodynamic motion planning with state lattice motion primitives," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 2172–2179.

[25] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019.

[26] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2015.

[27] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[28] J. Nasir, F. Islam, U. Malik, Y. Ayaz, O. Hasan, M. Khan, and M. S. Muhammad, "RRT*-SMART: a rapid convergence implementation of RRT," *International Journal of Advanced Robotic Systems*, vol. 10, no. 7, p. 299, 2013.

[29] O. Arslan and P. Tsiotras, "Use of relaxation methods in sampling-based algorithms for optimal motion planning," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 2421–2428.

[30] D. J. Webb and J. van den Berg, "Kinodynamic rrt*: Asymptotically optimal motion planning for robots with linear dynamics," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 5054–5061.

[31] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2997–3004.

[32] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3067–3074.

[33] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 883–921, 2015.

[34] M. P. Strub and J. D. Gammell, "Advanced BIT*(ABIT*): Sampling-based planning with advanced graph-search techniques," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 130–136.

[35] ——, "Adaptively Informed Trees (AIT*): Fast asymptotically optimal path planning through adaptive heuristics," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3191–3198.

[36] K. Naderi, J. Rajamäki, and P. Hämäläinen, "RT-RRT*: a real-time path planning algorithm based on rrt," in *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, 2015, pp. 113–118.

[37] J. M. Pimentel, M. S. Alvim, M. F. Campos, and D. G. Macharet, "Information-driven rapidly-exploring random tree for efficient environment exploration," *Journal of Intelligent & Robotic Systems*, vol. 91, no. 2, pp. 313–331, 2018.

[38] T. Fraichard and A. Scheuer, "From reeds and shepp's to continuous-curvature paths," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 1025–1035, 2004.

[39] M. Brezak and I. Petrović, "Real-time approximation of clothoids with bounded error for path planning applications," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 507–515, 2013.

[40] F. Gao, W. Wu, Y. Lin, and S. Shen, "Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 344–351.

[41] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 2520–2525.

[42] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research*. Springer, 2016, pp. 649–666.

[43] J. Chen, T. Liu, and S. Shen, "Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1476–1483.

[44] F. Gao, Y. Lin, and S. Shen, "Gradient-based online safe trajectory generation for quadrotor flight in complex environments," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 3681–3688.

[45] K. Majd, M. Razeghi-Jahromi, and A. Homaifar, "A stable analytical solution method for car-like robot trajectory tracking and optimization," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 39–47, 2020.

[46] B. Sun, D. Zhu, and S. X. Yang, "A bioinspired filtered backstepping tracking control of 7000-m manned submarine vehicle," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3682–3693, 2013.

[47] A. E.-S. B. Ibrahim, "Wheeled mobile robot trajectory tracking using sliding mode control." *J. Comput. Sci.*, vol. 12, no. 1, pp. 48–55, 2016.

[48] J. Osusky and J. Ciganek, "Trajectory tracking robust control for two wheels robot," in *2018 Cybernetics & Informatics (K&I)*. IEEE, 2018, pp. 1–4.

[49] B. B. Mevo, M. R. Saad, and R. Fareh, "Adaptive sliding mode control of wheeled mobile robot with nonlinear model and uncertainties," in *2018 IEEE Canadian Conference on Electrical Computer Engineering (CCECE)*, 2018, pp. 1–5.

[50] T. P. Nascimento, C. E. Dórea, and L. M. G. Gonçalves, "Nonholonomic mobile robots' trajectory tracking model predictive control: a survey," *Robotica*, vol. 36, no. 5, p. 676, 2018.

[51] F. Gavilan, R. Vazquez, and E. F. Camacho, "An iterative model predictive control algorithm for uav guidance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 2406–2419, 2015.

[52] A. Iskander, O. Elkassed, and A. El-Badawy, "Minimum snap trajectory tracking for a quadrotor uav using nonlinear model predictive control," in *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, 2020, pp. 344–349.

[53] B. Lindqvist, S. S. Mansouri, A. Agha-mohammadi, and G. Nikolakopoulos, "Nonlinear mpc for collision avoidance and control of uavs with dynamic obstacles," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6001–6008, 2020.

[54] C. Cui, D. Zhu, and B. Sun, "Trajectory re-planning and tracking control of unmanned underwater vehicles on dynamic model," in *2018 Chinese Control And Decision Conference (CCDC)*, 2018, pp. 1971–1976.

[55] W. Di, L. Caihong, G. Na, S. Yong, G. Tengteng, and L. Guoming, "Local path planning of mobile robot based on artificial potential field," in *2020 39th Chinese Control Conference (CCC)*. IEEE, 2020, pp. 3677–3682.

[56] J. Minguez and L. Montano, "Nearness diagram navigation (nd): A new real time collision avoidance approach," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)*, vol. 3. IEEE, 2000, pp. 2094–2100.

[57] M. Seder and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 1986–1991.

[58] W.-d. Chen and Q.-g. Zhu, "Mobile robot path planning based on fuzzy algorithms," *ACTA ELECTONICA SINICA*, vol. 39, no. 4, p. 971, 2011.

[59] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia, "Learning swing-free trajectories for uavs with a suspended load," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 4902–4909.

[60] ——, "Aerial suspended cargo delivery through reinforcement learning," *Department of Computer Science, University of New Mexico, Tech. Rep*, vol. 151, 2013.

[61] A. Faust, K. Oslund, O. Ramirez, A. Francis, L. Tapia, M. Fiser, and J. Davidson, "PRM-RL: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5113–5120.

[62] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International Conference on Machine Learning*. PMLR, 2014, pp. 387–395.

[63] A. Faust, P. Ruymgaart, M. Salman, R. Fierro, and L. Tapia, "Continuous action reinforcement learning for control-affine systems with unknown dynamics," *IEEE/CAA Journal of Automatica Sinica*, vol. 1, no. 3, pp. 323–336, 2014.

[64] H.-T. L. Chiang, J. Hsu, M. Fiser, L. Tapia, and A. Faust, "RL-RRT: Kinodynamic motion planning via learning reachability estimators from RL policies," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4298–4305, 2019.

[65] H.-T. L. Chiang, A. Faust, M. Fiser, and A. Francis, "Learning navigation behaviors end-to-end with AutoRL," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2007–2014, 2019.

[66] A. Francis, A. Faust, H.-T. L. Chiang, J. Hsu, J. C. Kew, M. Fiser, and T.-W. E. Lee, "Long-range indoor navigation with PRM-RL," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1115–1134, 2020.

[67] U. Patel, N. Kumar, A. J. Sathyamoorthy, and D. Manocha, "Dynamically feasible deep reinforcement learning policy for robot navigation in dense mobile crowds," *arXiv preprint arXiv:2010.14838*, 2020.

[68] P. Rousseas, C. Bechlioulis, and K. J. Kyriakopoulos, "Harmonic-based optimal motion planning in constrained workspaces using reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2005–2011, 2021.

[69] L. Chang, L. Shan, C. Jiang, and Y. Dai, "Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment," *Autonomous Robots*, vol. 45, no. 1, pp. 51–76, 2021.

[70] C. Pérez-D'Arpino, C. Liu, P. Goebel, R. Martín-Martín, and S. Savarese, "Robot navigation in constrained pedestrian environments using reinforcement learning," *arXiv preprint arXiv:2010.08600*, 2020.

[71] B. Ichter and M. Pavone, "Robot motion planning in learned latent spaces," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2407–2414, 2019.

[72] A. H. Qureshi, A. Simeonov, M. J. Bency, and M. C. Yip, "Motion planning networks," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2118–2124.

[73] Z. He, L. Dong, C. Sun, and J. Wang, "Asynchronous multithreading reinforcement-learning-based path planning and tracking for unmanned underwater vehicle," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[74] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3052–3059.

[75] R. E. Wang, M. Everett, and J. P. How, "R-maddpg for partially observable environments and limited communication," *arXiv preprint arXiv:2002.06684*, 2020.

[76] O. Zhelo, J. Zhang, L. Tai, M. Liu, and W. Burgard, "Curiosity-driven exploration for mapless navigation with deep reinforcement learning," *arXiv preprint arXiv:1804.00456*, 2018.

[77] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu *et al.*, "Learning to navigate in complex environments," *arXiv preprint arXiv:1611.03673*, 2016.

[78] P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, and J. Pan, "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6252–6259.

[79] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2778–2787.

[80] H. Shi, L. Shi, M. Xu, and K.-S. Hwang, "End-to-end navigation strategy with deep reinforcement learning for mobile robots," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2393–2402, 2019.

[81] Y. Wang, H. He, and C. Sun, "Learning to navigate through complex dynamic environment with modular deep reinforcement learning," *IEEE Transactions on Games*, vol. 10, no. 4, pp. 400–412, 2018.

[82] N. Wang, D. Zhang, and Y. Wang, "Learning to navigate for mobile robot with continual reinforcement learning," in *2020 39th Chinese Control Conference (CCC)*. IEEE, 2020, pp. 3701–3706.

[83] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 31–36.

[84] E. Rohmer, S. P. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1321–1326.

[85] L. Xie, S. Wang, S. Rosa, A. Markham, and N. Trigoni, "Learning with training wheels: speeding up training with a simple controller for deep reinforcement learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6276–6283.

[86] M. Luong and C. Pham, "Incremental learning for autonomous navigation of mobile robots based on deep reinforcement learning," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 1, pp. 1–11, 2021.

[87] Y. Cui, H. Zhang, Y. Wang, and R. Xiong, "Learning world transition model for socially aware robot navigation," *arXiv preprint arXiv:2011.03922*, 2020.

[88] J. Jin, N. M. Nguyen, N. Sakib, D. Graves, H. Yao, and M. Jagersand, "Mapless navigation among dynamics with social-safety-awareness: a reinforcement learning approach from 2d laser scans," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6979–6985.

[89] Y. Zhou, S. Li, and J. Garcke, "R-SARL: Crowd-aware navigation based deep reinforcement learning for nonholonomic robot in complex environments," *arXiv preprint arXiv:2105.13409*, 2021.

[90] P. Mirowski, M. K. Grimes, M. Malinowski, K. M. Hermann, K. Anderson, D. Teplyashin, K. Simonyan, K. Kavukcuoglu, A. Zisserman, and R. Hadsell, "Learning to navigate in cities without a map," *arXiv preprint arXiv:1804.00168*, 2018.

[91] J. Kulhánek, E. Derner, and R. Babuška, "Visual navigation in real-world indoor environments using end-to-end deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4345–4352, 2021.

[92] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3357–3364.

[93] Y. Wu, Z. Rao, W. Zhang, S. Lu, W. Lu, and Z.-J. Zha, "Exploring the task cooperation in multi-goal visual navigation." in *IJCAI*, 2019, pp. 609–615.

[94] Y. Lv, N. Xie, Y. Shi, Z. Wang, and H. T. Shen, "Improving target-driven visual navigation with attention on 3d spatial relationships," *arXiv preprint arXiv:2005.02153*, 2020.

[95] W. Luo, P. Sun, F. Zhong, W. Liu, T. Zhang, and Y. Wang, "End-to-end active object tracking via reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3286–3295.

[96] ——, "End-to-end active object tracking and its real-world deployment via reinforcement learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 6, pp. 1317–1332, 2020.

[97] A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru, and R. Vicente, "Multiagent cooperation and competition with deep reinforcement learning," *PLOS ONE*, vol. 12, no. 4, p. e0172395, 2017.

[98] K. Sivanathan, B. Vinayagam, T. Samak, and C. Samak, "Decentralized motion planning for multi-robot navigation using deep reinforcement learning," in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*. IEEE, 2020, pp. 709–716.

[99] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4295–4304.

[100] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *NIPS*, 2017.

[101] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 856–892, 2020.

[102] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of MAPPO in cooperative, multi-agent games," *arXiv preprint arXiv:2103.01955*, 2021.

[103] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 285–292.

[104] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1343–1350.

[105] M. Everett, Y. F. Chen, and J. P. How, "Collision avoidance in pedestrian-rich environments with deep reinforcement learning," *IEEE Access*, vol. 9, pp. 10 357–10 377, 2021.

[106] S. H. Semnani, H. Liu, M. Everett, A. de Ruiter, and J. P. How, "Multi-agent motion planning for dense and dynamic environments via deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3221–3226, 2020.

[107] S. H. Semnani, A. H. de Ruiter, and H. H. Liu, "Force-based algorithm for motion planning of large agent," *IEEE Transactions on Cybernetics*, 2020.

[108] S. Tang, J. Thomas, and V. Kumar, "Hold or take optimal plan (hoop): A quadratic programming approach to multi-robot trajectory generation," *The International Journal of Robotics Research*, vol. 37, no. 9, pp. 1062–1084, 2018.

[109] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: a survey," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 737–744.

[110] J. Zhang, L. Tai, P. Yun, Y. Xiong, M. Liu, J. Boedecker, and W. Burgard, "Vr-goggles for robots: Real-to-sim domain adaptation for

visual control," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1148–1155, 2019.

[111] B. Lütjens, M. Everett, and J. P. How, "Safe reinforcement learning with model uncertainty estimates," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8662–8668.

[112] T. Chaffre, J. Moras, A. Chan-Hon-Tong, and J. Marzat, "Sim-to-real transfer with incremental environment complexity for reinforcement learning of depth-based robot navigation," *arXiv preprint arXiv:2004.14684*, 2020.

[113] R. T. Kalifou, H. Caselles-Dupré, T. Lesort, T. Sun, N. Diaz-Rodriguez, and D. Filliat, "Continual reinforcement learning deployed in real-life using policy distillation and sim2real transfer," in *ICML Workshop on Multi-Task and Lifelong Learning*, 2019.

[114] A. A. Rusu, M. Večerík, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," in *Conference on Robot Learning*. PMLR, 2017, pp. 262–270.

[115] Y. Zhu, D. Schwab, and M. Veloso, "Learning primitive skills for mobile robots," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7597–7603.

[116] J. Liang, U. Patel, A. J. Sathyamoorthy, and D. Manocha, "Realtime collision avoidance for mobile robots in dense crowds using implicit multi-sensor fusion and deep reinforcement learning," *arXiv e-prints*, pp. arXiv–2004, 2020.

[117] M. Andrychowicz, D. Crow, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, "Hindsight experience replay," in *NIPS*, 2017.

[118] Y. Sun, J. Cheng, G. Zhang, and H. Xu, "Mapless motion planning system for an autonomous underwater vehicle using policy gradient-based deep reinforcement learning," *Journal of Intelligent & Robotic Systems*, vol. 96, no. 3-4, pp. 591–601, 2019.

[119] A. Y. Ng, D. Harada, and S. J. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *ICML*, 1999, pp. 278–287.

[120] Z. Qiao, J. Schneider, and J. M. Dolan, "Behavior planning at urban intersections through hierarchical reinforcement learning," *arXiv preprint arXiv:2011.04697*, 2020.

[121] S. Christen, L. Jendele, E. Aksan, and O. Hilliges, "Learning functionally decomposed hierarchies for continuous control tasks with path planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3623–3630, 2021.

[122] Z. Chi, L. Zhu, F. Zhou, and C. Zhuang, "A collision-free path planning method using direct behavior cloning," in *International Conference on Intelligent Robotics and Applications*. Springer, 2019, pp. 529–540.

[123] C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning," *Robotics and Autonomous Systems*, vol. 114, pp. 1–18, 2019.

[124] S. Rosbach, V. James, S. Großjohann, S. Homoceanu, and S. Roth, "Driving with style: Inverse reinforcement learning in general-purpose planning for automated driving," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2658–2665.

[125] K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman, "Quantifying generalization in reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1282–1289.

[126] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas, "Reinforcement learning with augmented data," *arXiv preprint arXiv:2004.14990*, 2020.

[127] K. Lee, K. Lee, J. Shin, and H. Lee, "Network randomization: A simple technique for generalization in deep reinforcement learning," in *International Conference on Learning Representations*, 2019.

[128] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.

[129] H. Zijian, G. Xiaoguang, W. Kaifang, Z. Yiwei, and W. Qianglong, "Relevant experience learning: A deep reinforcement learning method for uav autonomous motion planning in complex unknown environments," *Chinese Journal of Aeronautics*, vol. 34, no. 12, pp. 187–204, 2021.

[130] Z. He, L. Dong, C. Sun, and J. Wang, "Reinforcement learning based multi-robot formation control under separation bearing orientation scheme," in *2020 Chinese Automation Congress (CAC)*, 2020, pp. 3792–3797.

[131] M. Laskin, A. Srinivas, and P. Abbeel, "CURL: Contrastive unsupervised representations for reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5639–5650.

[132] I. Kostrikov, D. Yarats, and R. Fergus, "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels," *arXiv preprint arXiv:2004.13649*, 2020.

[133] M. Schwarzer, A. Anand, R. Goel, R. D. Hjelm, A. Courville, and P. Bachman, "Data-efficient reinforcement learning with self-predictive representations," *arXiv preprint arXiv:2007.05929*, 2020.

[134] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 797–803.

[135] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1688–1694.

[136] G. Ferrer and A. Sanfeliu, "Behavior estimation for a complete framework for human motion prediction in crowded environments," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 5940–5945.

[137] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1928–1935.

[138] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*. Springer, 2011, pp. 3–19.

[139] M. Phillips and M. Likhachev, "Sipp: Safe interval path planning for dynamic environments," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 5628–5635.

[140] L. Tai, J. Zhang, M. Liu, and W. Burgard, "Socially compliant navigation through raw depth inputs with generative adversarial imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1111–1117.

[141] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6015–6022.

[142] L. Xie, Y. Miao, S. Wang, P. Blunsom, Z. Wang, C. Chen, A. Markham, and N. Trigoni, "Learning with stochastic guidance for robot navigation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 166–176, 2020.

[143] F. Leiva and J. Ruiz-del Solar, "Robust RL-based map-less local planning: Using 2D point clouds as observations," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5787–5794, 2020.

[144] W. Zhang, Y. Zhang, and N. Liu, "Enhancing the generalization performance and speed up training for DRL-based mapless navigation," *arXiv preprint arXiv:2103.11686*, 2021.

[145] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[146] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *arXiv preprint arXiv:1705.08690*, 2017.

[147] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 67–82.

[148] M. Farajtabar, N. Azizan, A. Mott, and A. Li, "Orthogonal gradient descent for continual learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 3762–3773.