

A review of mobile robot motion planning methods: from classical motion planning workflows to reinforcement learning-based architectures

Zichen He, Jiawei Wang, and Chunwei Song,

Abstract—Motion planning is critical to realize the autonomous operation of mobile robots. As the complexity and stochasticity of robot application scenarios increase, the planning capability of the classical hierarchical motion planners is challenged. In recent years, with the development of intelligent computation technology, the deep reinforcement learning (DRL) based motion planning algorithm has gradually become a research hotspot due to its advantageous features such as not relying on the map prior, model-free, and unified global and local planning paradigms. In this paper, we provide a systematic review of various motion planning methods. First, we summarize the representative and cutting-edge algorithms for each submodule of the classical motion planning architecture and analyze their performance limitations. Subsequently, we concentrate on reviewing RL-based motion planning approaches, including RL optimization motion planners, map-free end-to-end methods that integrate sensing and decision-making, 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

NOWADAYS, with the rapid development of the artificial intelligence technology, autonomous intelligent mobile robots (MRs) are always at the forefront of scientific research owing 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 fields of ocean exploration, urban rescue, security patrol, and epidemic prevention and control [1]–[3], etc.

Generally, motion planning (MP) is one of the most fundamental and essential components among the key technologies to empower MRs with a high degree of autonomy. MP is responsible for coordinating the entire robot system so that the robot or the robot group can obtain planning and decision-making capabilities in a messy environment. The standard MP module of the MR consists of the global motion planner and the local motion planner. Global motion planner in charge of planning and generating a series of feasible waypoints based on the priori map information, which contains the passable area and the obstacle state of the environment and selecting

the optimal and kino-dynamic feasible trajectory according to different optimization objectives. The local motion planner is responsible for making specific action strategies in the local environment on the basis of external information collected by the robot sensing module, such as dynamic obstacle avoidance and pedestrian interaction. The standard MP framework is hierarchical and multi-level cascading, with a certain degree of customization. The global planner and the local planner are independent of each other and need to be developed separately. This feature makes it difficult for classical motion planners to adapt to complex, dynamic and changeable application scenarios. Therefore, it is of great significance to study motion planners with the ability of self-learning. In recent years, the rise of reinforcement learning (RL) has enabled the learning-based MP methods to be independent of the map prior data and thus have gained the favor of many scholars.

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 whether other behaviors might bring more incredible benefits or exploits the optimal behaviours to obtain the maximum return. Along with the substantial improvement in the computing power and the storage capacity of hardware systems, deep learning (DL) has been widely used in AI. In this context, deep reinforcement learning (DRL), a combination of DL and RL, was born. DRL help the researchers integrate the learning module and the decision module, and realize the nonlinear mapping procedure from the raw input to the action space. With its superior properties, DRL has been successfully applied to gaming AI, autopilot, transportation scheduling, power system optimization [3], [5], [6], etc. In the field of the mobile robot motion planning, RL-based motion planners achieve end-to-end planning, avoid tedious hierarchical multi-level coupling planning framework, and unify global motion planners and local motion planners. Furthermore, RL-based MP technology absorbs the perception module, which makes it not rely on the knowledge of the prior map. By constructing specific reward forms and training patterns, the state update policy can be improved iteratively based on the feedback from the environment during the process of interaction. These characteristics mentioned above offer the possibility of deploying RL-based MP methods in some unstructured environments where the mapping operation is difficult to perform.

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

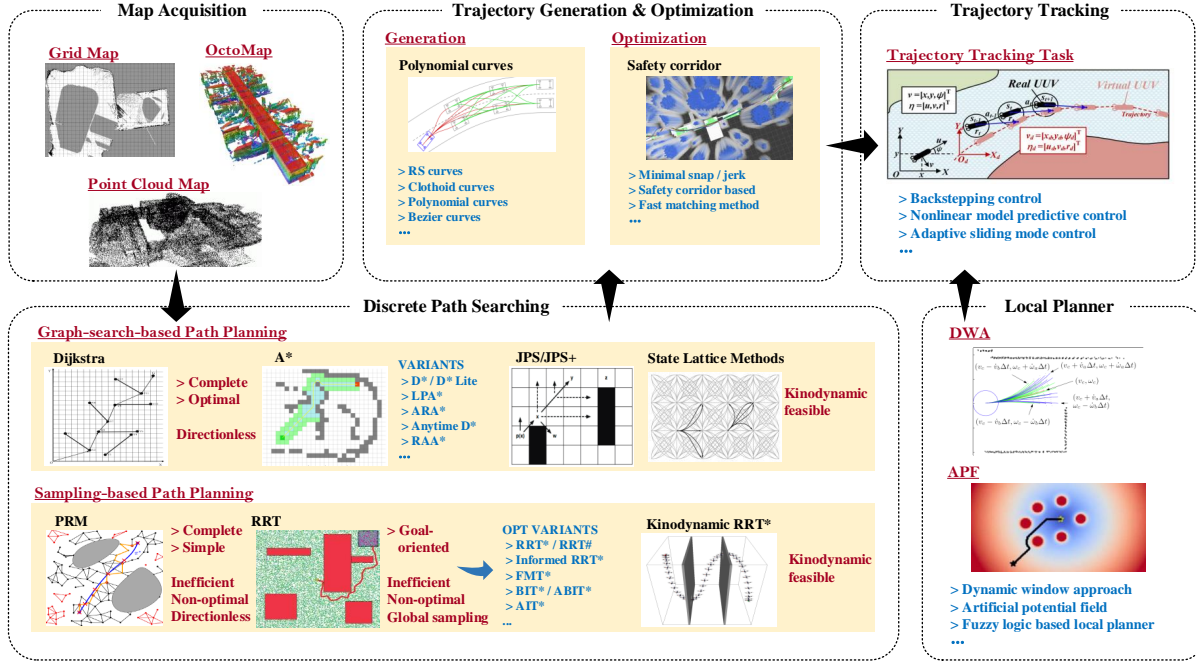


Fig. 1. The architecture of the classical motion planning approach. Each subpart contains some optimized variants of the corresponding algorithm. This figure is a general abstraction of section II.

This paper is a survey of current mainstream and state-of-the-art mobile robot motion planning algorithms. Its content mainly covers robot types, including wheeled mobile robots, 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 for each submodules of the classical motion planners. The second part is an overview of RL-based MP approaches. It consists of three sections. The first section is a summary of MP methods that incorporate the advantageous features of both classical motion planners and RL algorithms. This type of algorithm still relies on the map prior. The role of the RL within the algorithm 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 end-to-end RL-based motion planners with different sensors. This type of algorithm is map-free. Its main research hotspots including laser range finder based methods and vision sensor based target driven methods. The third section is an overview of RL-based multi-robot MP methods. In this section, we focus on reviewing some works of multi-robot collaborative planning on the basis of centralized training and decentralized execution (CTDE) RL architecture. The last part of this survey is a discussion. In this part, we systematically summarize the current challenges faced by RL-based motion planners for practical deployment in real life, including reality gap, social etiquette, catastrophic forgetting problem, reward sparsity issue, lidar data pre-processing issue, low sample efficiency problem, and generalization problem. Also, we review several representative works aimed at addressing these issues. Finally, we provide an outlook on the future research directions of RL-based MP

algorithms.

To sum up, the rest of this paper is organized as follows. Section II is a review of classical hierarchical MP approaches. Section III is a review of map-based classical MP algorithms with RL optimization. Section IV is a review of map-less RL-based motion planning methods. Section V is a review of RL-based multi-robot motion planning methods. Section VI is the discussion. It mainly focuses on summarizing the current challenges in current RL-based motion planners and giving suggestions on future directions. The conclusion is draw in section VII.

II. CLASSICAL MOTION PLANNING OF MRS

The classical hierarchical architecture of the motion planning methods of mobile robots (MRs) is shown in Fig. 1. It can be divided into five submodules: discrete path searching, trajectory generation, trajectory optimization, trajectory tracking, and local planning [1], [7], [8]. 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 some recent relevant works.

A. Map Acquisition

Before we start the classical motion planning workflow of MRs, we need to obtain the map representation of the environment. The quality of the constructed priori map directly determines the final planning performance. Common maps include occupancy grid map, octo-map, voxel map, point cloud map, Voronoi diagram map, etc.

B. Discrete Path Searching

The goal of discrete path searching (DPS) is to find a feasible and discrete path from the start point to the target point. DPS works in configuration space (C-space). Each configuration of the robot can be represented as a point. This setting largely 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 [9].

Traditional global DPS algorithms can be divided into two categories: the graph-searching-based algorithm (GSBA) and the sampling-based algorithm (SBA) [10].

1) *Graph-search-based algorithms*: Depth first search (DFS) and breadth first search (BFS) are two fundamental graph search algorithms. Build on the basis of BFS, Dijkstra is proposed. This algorithm is greedy and has completeness, and optimality [11]. However, it lacks directionality in the process of path search. In [12], A* is proposed. This algorithm introduces 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 [13], Anthony Stentz presents Dynamic A* (D*), he replaces the heuristic rule in A* with an incremental reverse rule. In [14], SvenKoenig et al. develop Lifelong planning A* (LPA*). They combine incremental search with A*. LPA* avoids the problem of recalculating the whole graph due to changes in the environment. In [15], Dorota Belanová et al. propose D* lite. D* Lite is a path planning algorithm with the variable start point and the fixed target point. Dorota Belanová et al. utilize the reverse search with the heuristic mechanism. The difference between D* Lite and LPA* is the search direction. Jump point search (JPS) [16] is another type of GSBA with different search principles. It improves the search efficiency for subsequent nodes on the basis of A* and can explore more intelligently. Notably, JPS only adds the jump points searched according to specific rules into the open list. This operation excludes a large number of meaningless nodes. Therefore, it occupies less memory and can search faster than A*. However, JPS is only applicable to the uniform grid map [7], [17]. In [18], Changjiang Jiang et al. propose the JPS+ based path planning method. They further improves the search efficiency of JPS by adding the pre-processing section but is less suitable for dynamic environments. Several other A*-based anytime heuristic GSBA include Anytime Repairing A* [19] and Anytime D* [20], etc. Several real-time A*-based GSBA include Learning Real-time A* [21] and Real-time Adaptive A* [22], etc.

None of the above methods considers the kino-dynamics of the MR. For some particular MRs with non-holonomic constraints, sometimes the discrete path planned by the above methods cannot be executed well. State lattice methods [23]–[25] have been widely used to handle this problem. These approaches first perform spatial discretization, use a hyper-dimensional grid of states to represent the planning area, generate a graph with feasible motion connections based on the sampling process, and finally search for an optimal discrete

path.

2) *Sampling-based algorithms*: GSBA are mainly applied to path planning problems on low-dimensional spaces. The completeness of these algorithms is based on the full modelling process of the environment. In high-dimensional spaces, 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 path points [26].

Probabilistic road map (PRM) and rapid-exploring random tree (RRT) are two fundamental SBAs [7]. PRM builds a graph during the learning stage and utilizes it to search for valid discrete paths during the query stage [17]. 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 it is not optimal or asymptotically optimal. 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. Both RRT*-smart [28] and RRT# [29] are optimized for the slow convergence speed of RRT*. Kino-dynamic RRT* [30] additionally deals with the kino-dynamic constraints of MRs. This algorithm samples in full state space, and has applications in MRs with non-holonomic properties. Informed RRT* [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*) [32] unifies the advantages of GSBA and SBAs. It introduces a heuristic method to search for a sequence of increasingly dense implicit RGGs iteratively. Compared to RRT*, Informed-RRT*, and fast match trees (FMT*) [33], BIT* enhances planning performance significantly in the experiments. Advanced-BIT* (ABIT*) [34] further extends BIT*. It applies advanced truncated anytime graph-based search techniques to enhance real-time planning performance. Adaptively informed trees (AIT*) [35] adds the trick of asymmetric bidirectional search on the basis of BIT*. It can quickly converge towards the optimal result and plan 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, real-time, and dynamic environment.

C. Trajectory Generation and Optimization

Most of the DPS algorithms only consider the geometric constraints of the workspace. Therefore, the trajectory generation and optimization (TGO) process that adds a time scale and handles various constraints faced by the MR (e.g., safety constraints, kino-dynamic constraints, maximum velocity constraints, etc.) is introduced to further improve the

executability of the planned paths. The purpose of TGO is to generate a collision-free, high-quality, time-efficient, and energy-efficient trajectory that meets safety and kino-dynamics feasibility based on the discrete path waypoints.

The interpolation-curve-based method is one of the most commonly used approaches for trajectory generation. It can generate trajectories with favourable continuity and differentiability. In the field of motion planning of MRs, the typical interpolating curves include Reeds and Sheep (RS) curves [38], clothoid curves [39], polynomial curves [26], Bezier curves [40], etc. For the trajectory optimization, the minimum snap [41] algorithm utilizes differential flatness of MRs to reduce the dimension of the state space and the action space. Then, by solving the constrained quadratic programming (QP) problem to minimize the rate of thrust change, the energy consumption is optimized. Later, Richter et al. solved the minimum snap problem in closed-form to avoid the numerical instability [42]. The algorithm in [43] introduces safety corridor constraints on the minimum snap method to enforce the security of the MR during motion, but the process of iteratively checking 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. An online TGO approach based on the Euclidean distance field is applied in [44]. The cost function of this method is composed of the elastic band smoothness term, the safety term, and the dynamical feasibility term. Then, the nonlinear optimization method is used to solve the final trajectory. This 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.

D. Trajectory Tracking

Generally, the purpose of the trajectory tracking (TT) phase is to enable the MR to track a time-dependent, safe, smooth, and dynamically feasible 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, such as input-output linearization [45], backstepping control [46], sliding mode control (SMC) [47], robust control [48], etc. However, these approaches suffer from several limitations. First, some MRs, like AUVs, contain complex nonlinear or uncertain terms in the kino-dynamic equations. This increases the complexity of modelling process. Besides, in practice, the operating environments of MRs are changeable. For instance, there may exist pedestrians or other cooperative robots in different scenarios. 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 wheeled mobile robots. This algorithm takes nonlinear model and disturbances into account and utilizes the discontinuous projection mapping to adjust the parameters of the WMR. Model predictive control

(MPC) is also a mainstream technique used to deal with TT problems. It belongs to the category of online optimal control and can handle various state and control constraints [50]. In [51], the MPC-based iterative trajectory tracking scheme is presented to be deployed into the navigation of UAV. Avraïem 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 [52]. Björn Lindqvist et al. focus on coping with dynamic obstacle avoidance problems. They couple the dynamic obstacle avoidance behaviour with the TT control. The proposed architecture of 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 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].

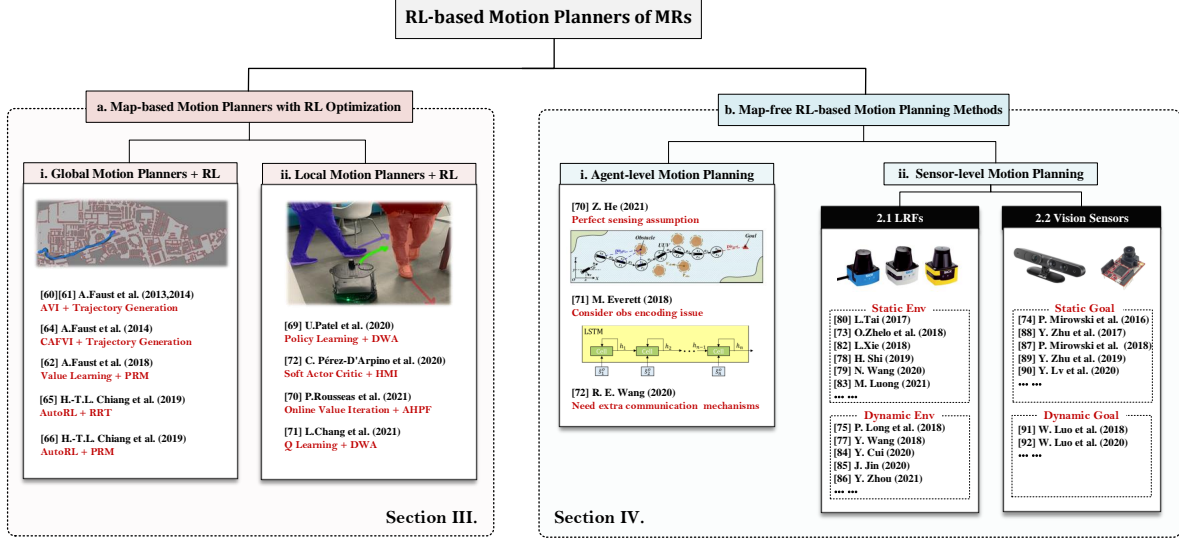
E. Local planning

In most application scenarios where MRs are deployed, uncertainties exist, such as pedestrians. Therefore, the MR should also need to be capable of handling uncertainties when tracking the trajectory generated by the global motion planner. 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. At present, many types of research focus on APF-based optimization algorithms. For example, [55] solves the above limitations by introducing the left tuning potential field and virtual target point. Reactive replanning methods can avoid unknown dynamic obstacles in the environment. Commonly used methods include directional approach [56], dynamic window approach (DWA) [57], etc. DWA is an active velocity selection algorithm that incorporates the kino-dynamics of the MR itself. It samples multiple velocities in the velocity space and simulates the intrinsic trajectory sets in a certain time, and finally selects the optimal trajectory to drive the MR. DWA is usually coupled with the global trajectory tracking process. It can endow the MR with a high degree of flexibility.

III. MAP-BASED CLASSICAL MOTION PLANNING ALGORITHMS WITH RL OPTIMIZATION

The research of the classical motion planning is relatively mature and systematic, but still exists some performance limitations, such as the separation of the global planner and the local planner, the necessity of solving optimization problems with constraints, the difficulty of dealing with changes of the environment and the sensor noise, and the reliance on handcrafting performance parameters, etc. As shown in Fig. 2. There are considerable works dedicated to combining the



method retains the fast and real-time advantages of AHPF while utilizing the value iteration to iteratively improve the planning policy. It endows the local planner with optimality. Lu Chang et al. propose the global dynamic window approach (GDWA) based on Q-learning optimization [70]. The evaluation function of GDWA is crucial for its effective planning. In different scenarios, the proportion of each weight of the evaluation function should be different. For instance, when the MR approaches an obstacle, the weight of the obstacle distance evaluation term should be increased. Inspired by this idea, Lu Chang et al. choose to compose the action space of the RL agent by these performance weights. The RL agent learns the optimal weight combination policy and realize the adaptive adjustment of the performance weights of the DWA.

In some scenarios with the pedestrian involvement, optimization at the map-based motion planning level can also be performed by combining RL algorithms. For example, recently, to address the problem that classical global motion planner have difficulty handling robot navigation in constrained indoor spaces with pedestrian participation. Pérez-D'Arpino et al. propose an indoor navigation framework that combines the soft actor critic (SAC) algorithm with global motion planning method [71]. The global motion planner is responsible for generating planned waypoints on the basis of the prior map. The waypoints are utilized as guidance and concatenate 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. This approach makes the MR more flexible. The global motion planner help the MR reach the target and the RL agent focus on learning different interaction patterns (slowing down, detouring, going backwards, etc.) with pedestrians in different motion states.

IV. RL-BASED MAPLESS MOTION PLANNING METHODS

Different from classical motion planning methods based on the improvement of RL algorithms. Most of RL-based motion planning approaches is map-free and realizes the unification of the global planner and the local planner. Researchers do not need to construct and maintain a geometric prior map of the environment, whose accuracy will directly influence the final effect of the motion planning. In addition, compared with the supervised learning based mapless motion planning methods [72], [73], RL based methods can directly learn from numerous trials and reward signals instead of the labeled data. So, it avoids the procedure of expert data collection as the ground truth.

As shown in part b of Fig. 2. RL-based mapless motion planning approaches can be further divided into two categories: agent-level methods and sensor-level methods. Agent-level methods can directly acquire the upper-level information of the environment. That is, the observation space of the agent-level method contains the state information (shapes, positions, velocities, etc.) of other obstacles or pedestrians. The agent-level methods are easier to train, which allow the MR to obtain optimal planning strategies faster. Also, the corresponding simulators are much simpler to develop. However, such methods either rely on the assumption of

perfect perception [74], or require to consider observation encoding issue [75], or need extra communication mechanisms to share the state information [76]. These undoubtedly limit the scalability and application of agent-level methods. Sensor-level methods are end-to-end. They directly establish the mapping between the raw sensor data to the planning actions of the MR. Although the offline training process is more time-consumption compared to agent-level methods, these 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 of the environment. Thus, they has more scalability and sim-to-real ability compared to agent-level methods. This section is mainly an overview of this more widely used end-to-end RL-based mapless motion planning methods. According to the mainstream research trends and different sensors carried by MRs, we further divided such algorithms into two categories: laser range finder (LRF) based methods and visual-based methods.

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

In [77], Oleksii Zhelo et al. consider some scenarios including long corridors or dead corners that are not suitable for RL agent to learn the optimal mapless motion planning strategies. Different from some RL-based end-to-end motion planning methods [78], [79], this approach utilizes the intrinsic curiosity module [80] to help RL agent obtain the intrinsic reward. This exploration trick help the A3C agent acquire better generalization ability in the 2D virtual environment. Yuanda Wang et al. consider the end-to-end motion planning task in static virtual environment with dynamic obstacles [81]. They decompose the whole motion planning task into an obstacle avoidance subtask and a navigation subtask. Obstacle avoidance module takes the raw sensor data of LRF as input, and output a 5-dimensional force vector. The Q network of the obstacle avoidance module has two streams: the spatial stream and the temporal stream. Spatial stream directly processes the raw sensor data, while the temporal stream processes the difference between the two frames of ranging data. Navigation module is training by a conventional Q network architecture. Experiments show that this approach could obtain a high-performance motion planning strategy in the 2D dynamic simulation environment. Likewise, [82] introduces the intrinsic curiosity module and presents a more general end-to-end motion planning method based on the A3C framework with the sparse Lidar data and successfully deploys from the physical engine to the realistic mixed scene. In [83], Ning Wang et al. find that extant methods generally require retraining the RL agent in different motion planning scenarios to reduce the generalization error caused by environmental changes, which can lead to the catastrophic forgetting problem. They propose the elastic weight consolidation DDPG (EWC-DDPG) based navigation algorithm and enable the RL agent to acquire continuous learning capabilities without forgetting previous knowledge.

Above methods do not consider issues such as sensor noise and safety issues in real scenes. In [84], asynchronous DDPG (ADDPG) based motion planning algorithm is applied in map-less navigation of the differential mobile robot (DMR). The action space consists of the angular and the linear velocities of DMR at each time step. The state space in training stage includes three parts: (1) 10-dimensional sparse findings of LRF. (2) 2-dimensional previous action of the DMR. (3) 2 dimensional relative position of the goal. A reward is set when the DMR arrive near the target, and a certain penalty is given when the DMR collides with the obstacle. Otherwise, $r_t(s_t, a_t) = C(d_{t-1} - d_t)$ where C is a hyper-parameter, $d_{t-1} - d_t$ is the distance difference between the DMR and the goal in adjacent time steps. The whole motion planner is training in VREP [85] virtual engine and evaluating in real environment. Final results show that the proposed ADDPG-based end-to-end motion planning approach is more robust than *Move Base* method in the real static environment with human interruption. Linhai Xie et al. present the Assisted DDPG (AsDDPG) for training agents to learn local planning strategy in static obstacle environment without the prior map [86]. This DRL framework integrates DDPG with classical controller (like a PID controller). This naive controller would replace the random exploration strategy (e.g. ϵ -greedy) can output control policy on the basis of the errors 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. The experiments from Gazebo to real world verify that this trick is able to accelerate and effectively stabilize the training phase. In [87], Manh Luong et al. present an incremental learning paradigm to address the training inefficiencies in end-to-end RL-based motion planning. In the training stage, the learning phase is incremental to enhance the current policy until the loop terminates. The performance of this algorithm is verifying on the *Gazebo* simulation platform and the real *Pioneer P3-DX* mobile robot. Furthermore, [88]–[90] etc. expect to endow the MR with the social safety awareness while performing end-to-end motion planning task. That is, training robots to safely and carefully interact with pedestrians in the environment instead of simply treating them as static or dynamic obstacles. In [88], Yuxiang Cui et al. developed a model-based RL motion planning method with social safety awareness. They first obtain data by interacting the MR with the realistic scene, and then utilize them to train a world transition model with pedestrian participation in the self-supervised learning paradigm. Finally, they combine the real data with the virtual data generated by the environment model as the observation to the RL architecture to train the motion planning policy. [89] and [90] construct rectangular social-safety zones for the MR and pedestrians, respectively, and design the corresponding safety reward term based on them. And the end-to-end RL framework is utilized to train the motion planning policy in the mixed and complex environment.

2) *Vision sensors*: The general end-to-end visual-based motion planning or navigation means that the robot can find a real-time and collision-free trajectory with the less time or energy consumption from the initial position to the target in a

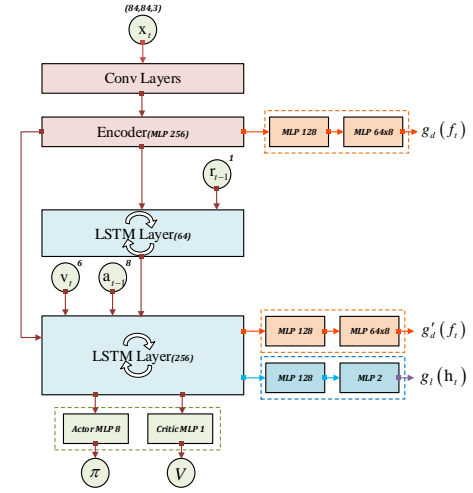


Fig. 3. The architectures of Nav A3C visual navigation algorithm with additional tasks including depth prediction and closures detection [92].

complex space with the perception information of cameras. To accomplish this, the core algorithm is required to be able to establish a mapping relationship from the raw image data to the motion commands of robots. DRL framework combined with convolutional neural networks (CNNs) is naturally suitable for this task. DeepMind proposes NavA3C algorithm to teach the agent to find the goal and navigate in 3D mazes [78]. The main idea of this work is to enable the agent to learn to accomplish the main task of finding the target location, while learning the auxiliary tasks including the loop closure and the depth prediction. The whole architecture of this algorithm 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 are based on the supervised learning. Parameters updating process need to aggregating all gradients of various loss functions from different tasks. Multi-group experiments prove that this trick of adding auxiliary tasks can avoid sparse reward signals, improve the richness of the training data, and optimize the training efficiency. Later, DeepMind extend this work, and apply it on outdoor navigation scenarios on the basis of the realistic street panoramas dataset of Google [91]. Inspired by this work, designs auxiliary tasks in the planning architecture to facilitate the domain randomization of the model in simulation environments. Even better, this method directly utilizes unlabeled images image of segmentation masks that are not readily available in the environment. This operation improves the planning performance of the algorithm in the realistic environment.

In [93], Li Fei-Fei group proposes a DRL-based target-driven visual navigation approach in indoor scenes. Different from previous research about visual navigation, their method has strong generalization ability to various scenarios and different targets and can be easily deployed to the realistic world just by fine-tuning. The framework of their method is shown in Fig. 4. The inputs including current observation

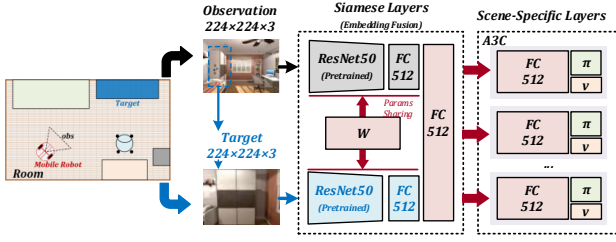


Fig. 4. The overall architecture of the deep siamese actor-critic based target-driven indoor navigation algorithm [93]. This algorithm takes the current observation and the goal image as the input. Two ResNet50 sharing parameters are responsible for encoding these two batches of observations, and generating two vectors. After fusion operation, These two vectors are converted into one embedding vectors and input to the A3C as the state input. The A3C scene-specific layers output the final action policy of the mobile robot.

image and the target image. Weights-sharing siamese networks transfer the inputs 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 group has pioneered the target-driven visual navigation. They also developed a high quality 3D indoor simulation platform named The House Of InterActions (AI2-THOR) which has subsequently facilitated other scholars to conduct their visual navigation research (e.g. [94], [95])

Visual-based motion planning methods mentioned above is on the basis of environments with priori static goals. 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 signal. Visual navigation is generally difficult to achieve sim-to-real process due to the gap between the simulator and the realistic scene. Aim at this issue, they perform environment complexity augmentation and virtual model detail augmentation to enhance the robustness and generalization of the algorithm [96], [97].

V. RL-BASED MULTI-ROBOT MOTION PLANNING

In specific application scenarios, relying solely on a single MR operation suffers from performance bottlenecks such as limited sensing capabilities, low task reliability, and inefficient execution. Collaborative motion planning of multiple MRs is more flexible, robust, and time-efficient. Therefore, it is widely utilized in marine exploration, smart agriculture, disaster rescue, etc [2].

Unlike the Markov properties of the single agent RL-based motion planning methods. 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 approaches will extend the interaction process of robots with their environment to the partial observable Markov decision process (POMDP) or decentralized POMDP (Dec-POMDP). The major RL-based MRMP research work can be further divided into two main categories: the centralized RL-based MRMP (CeRL-MRMP) and decentralized RL-based MRMP (DecRL-MRMP). CeRL-MRMP is simple and intuitive. It uses

a centralized network to build a map from joint trajectories of all agents to total action-state value and aims to learn the joint planning policy that maximizes the total rewards [98]. The limitations of this approach are 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 [99]. Each robot only gets partial observation of the environment and does not consider the state and action of others during the training phase. Every individual is self-interested and only considers how to maximize its own return. So, independent MRMP exists credit assignment problem. And, since each robot performs state update at each timestep, resulting in the entire environment is dynamic changing. This is not beneficial for the reward convergence. CTDE-based methods merge the advantages of the above paradigms. During the training procedure, each agent can obtain state information about others through the combined total Q value [100] or historical trajectories from the global experience buffer [101] or other sensor-based explicit approaches [102]. During the planning execution process, each robot requires only its own local observations to make online inference decisions. Some of these representative works will be reviewed below.

Some works focus on studying multi-agent reinforcement learning (MARL) algorithm based on CTDE architecture [2], [101], [103]. They usually adopt cooperative navigation scenario of Multi-agent Particle Environment (MPE) [101] as an experimental benchmark task to test the performance of their proposed algorithms. These works concentrate on optimizing the performance of the MARL itself, such as the information sharing efficiency, collaboration ability of agents, overall convergence speed, etc. However, the agent in MPE environment differs significantly from the real MR and ignores the kino-dynamic constraint, making it difficult to deploy in real operation scenarios. The ACL laboratory of MIT has 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 [75], [104]–[106]. They describe this type of problem as a sequential decision making problem. In an n -agent scenario, the state vector of agent i is \mathbf{s}_i , and the observable states vector of other $n - 1$ agents (MRs or pedestrians) is $\tilde{\mathbf{S}}_i^o$. $\mathbf{s}_i = [\mathbf{s}_i^o, \mathbf{s}_i^h]$ 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 $\mathbf{u}_t = [v_t, \psi_t]$, 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. (2),(3),(4) respectively represent the safety constraint, the target point constraint and the kinodynamic constraint [106].

$$\arg\max_{\pi_i} \mathbb{E}[t_g | \mathbf{s}_i, \tilde{\mathbf{S}}_i^o, \pi_i] \quad (1)$$

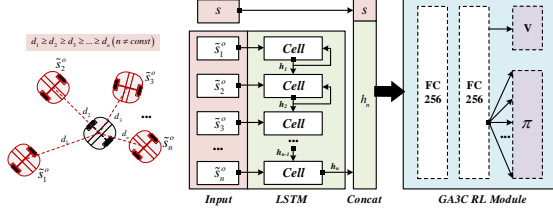


Fig. 5. The overall architecture of GA3C-CADRL in [75].

$$s.t. \quad \|\mathbf{p}_{i,t} - \tilde{\mathbf{p}}_{j,t}\|_2 \geq r_i + r_j \quad \forall i \neq j, \forall t \quad (2)$$

$$\mathbf{p}_{1,t_g} = \mathbf{p}_{1,\text{goal}} \quad \forall i \quad (3)$$

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

In [104], they propose the Collision Avoidance with Deep RL (CADRL) algorithm and applies it to solve (1)-(4). CADRL adopts a non-communicating and off-line CTDE framework to efficiently avoid the time-consuming online computation process in some classical motion planning methods. In the training phase, it utilizes 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 $\pi^* = \underset{\mathbf{u}_t \in \mathcal{U}}{\operatorname{argmax}} R([\mathbf{s}_i^o, \mathbf{s}_i^h], \mathbf{u}_t) + \gamma V(\hat{\mathbf{s}}_{t+1}, \hat{\mathbf{s}}_{t+1}^o)$. CADRL is real-time and has great superiority compared to other mainstream methods. However, The cooperative behaviors of CADRL cannot be controlled.

In [105], they further extend CADRL and propose 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 the interaction between pedestrians and mobile robots). Different from 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 algorithm on real MR hardware platform to realize automatic navigation at human-walking speed in pedestrian-rich environment.

In [75], Michael Everett et al. further consider the stochastic behavior model and the uncertain number of other agents in the environment on the basis of CADRL and present the GA3C-CADRL algorithm. The most critical contribution of GA3C-CADRL is that it can tackle agent-level state representation issue where the number of obstacles or agents in the environment vary 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 this trick would introduce additional parameter calculations and the extra issue of observation sparsity, which is detrimental to the final convergence of the algorithm. Also, compare to those sensor-level MRMP approaches [84], Michael Everett et al. think these methods cannot extract an agent-level representation that imply the motion plans of other agents. In response to these challenges, they propose a LSTM-based encoding method

for environment information and integrate this method into the Actor-Critic framework. The details is shown in Fig. 5. $\mathbf{s} = [\|\mathbf{p}_{\text{goal}} - \mathbf{p}\|_2, v_{\text{pref}}, \psi, r]$ represents the observation of the agent itself and $\tilde{\mathbf{s}}^o = [\tilde{p}_x, \tilde{p}_y, \tilde{v}_x, \tilde{v}_y, \tilde{r}, \tilde{r} + r, \|\tilde{\mathbf{p}} - \mathbf{p}\|_2]$ represents the observation of other agents in the vicinity. Whatever the dimension of $\tilde{\mathbf{s}}^o$ is, the final output hidden state h_n encodes the entire the observation of environment in a fixed-length vector. It is worth noting that their team open source their code of simulation environment, which provide a studying platform for other researchers. GA3C-CADRL combines Supervised Learning with multi-process RL framework and introduces curriculum learning paradigm, which increase the training difficulty to some extent. Also, GA3C-CADRL does not improve the form of reward function in CADRL. Therefore, there exists the risk of appearing sparse reward problem.

Later, this group continues to optimize the GA3C-CADRL, and proposes GA3C-CADRL-No Supervised Learning (GA3C-CADRL-NSL) algorithm [107]. This work design a special goal-distance based proxy reward function to eliminate the supervised learning stage. The detailed form of this reward function is given in (5).

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

where R_c is responsible for monitoring d_{\min} which represents the distance between current agent i and its closest agent, and punishing dangerous actions. R_g is responsible for rewarding those actions that enable the agent approaching the goal. This trick of reward shaping 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) [108] method. Once the mobile robot fall into the high-risk situations, FMP algorithm will take effect and help the robot get out of the trouble.

Tingxiang Fan, Jia Pan et al., from the department of computer science at university of Hong Kong, have been working on DecRL-MRMP for large-scale multi-robots in dense environments [79]. Different from the related work of ACL Lab [75], they pay more attention to the research of sensor-level motion planning methods which can directly map the raw sensor data to desired steering commands rather than agent-level motion planning policies on the basis of the full observable or perfect sensing assumption [104]. In this decentralized approach, each robot is independent of others, so it has strong scalability and would not suffer from the curse of observation space dim disasters of the observation and the action space dimensionality faced by some centralized methods [109]. During the training phase, rewards, policy networks, and value function networks are shared among individual robots. Also, the shared experience samples are utilized to guide the development of implicit collaboration

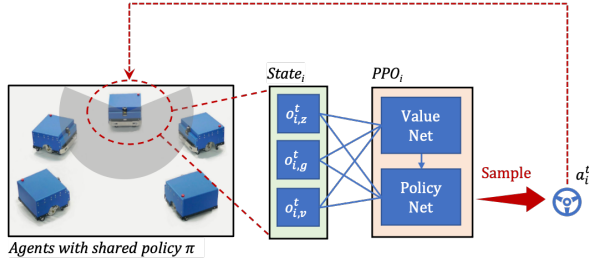


Fig. 6. The overall framework of the distributed PPO-based multi-scale mobile robots motion planning algorithm in [79].

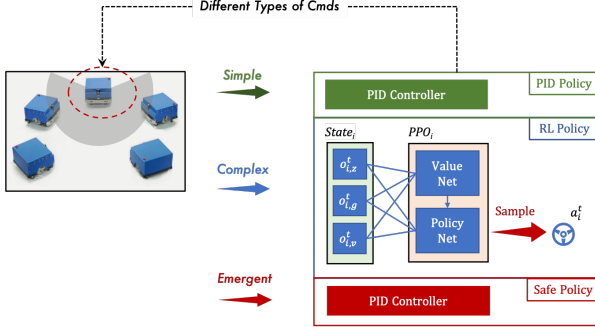


Fig. 7. Hybrid control policy framework in [102]. The robot will choose an appropriate action command according to the type of the current scenario.

mechanisms. The main idea of [79] 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 MRMP scenarios [102]. The detail is shown in Fig. 7.

VI. DISCUSSION

In this section, we mainly analyze and summarize several challenges of the current RL-based motion planning methods based on the current performance bottlenecks in the development of current RL algorithms, as well as some details that still need to be considered when the actual deployment is implemented. The specific contents are shown 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 problem (a real robot could cause real damage) of robots in real-life scenes, low sample efficiency in 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 and result in plenty uncertainties. At present, many scholars in Robotics have devoted to the research of innovative Sim-to-Real methods [111]. Mainstream Sim-to-Real approaches include domain adaption methods [112], disturbances learning based robust RL methods [113], domain randomization methods [110], [114], knowledge distillation [115], etc. As for motion planning of mobile robots, 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 [116], [117]. 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 learning stage, they adopt the architecture shown in Fig. 9. 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 linear and angular speed commands are output to the low-level controller of the MR in the testing phase [110]. 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 is used in the stage of training RL agent [112]. Jing Liang et al. propose a brand new learning-based local navigation approach named CrowdSteer to solve the motion planning problem of MRs in the real-world scenarios of dense crowded environments [118].

2) *Sparse reward problem*: The motion planning process of mobile robots are often target-driven. The reward of environmental feedback is generally at the final goal point. Coupled with the presence of various obstacles that would bring negative penalties in the task scenario. It is hard for robots to obtain positive reward signals and might develop abnormal behavior patterns such as timidity. The sparse reward problem can lead to slow learning of the agent and difficulty in algorithm convergence.

Curiosity driven is a way to solve the problem of sparse rewards using existing trajectories [80]. The main idea of this method is to build intrinsic curiosity modules (ICM) to extract additional intrinsic reward signals from environment to encourage more effective exploration of the agent. In [77], [82], the researchers choose to utilize this trick in developing map-free and end-to-end motion planning frameworks of MRs. Hindsight experience replay (HER) [119] is another approach used to solve the sparse reward problem with existing data 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 the above methods, reward shaping is more subjective. Researchers using this trick need to be patient and manually adjust and modify the additional design reward signal values under different states [120]. So, reward shaping skill is highly dependent on expert experience. Improper reward can lead to changes of optimal policy, causing the agent to exhibit anomalous behaviors [121]. Another type of method to solve the issue of reward sparsity is called hierarchical RL (HRL) algorithm. HRL tends to decompose the original task uniformly or hierarchically into discrete or continuous and easy-to-solve subtasks, and then divide and conquer to provide the agent with dense reward signals.

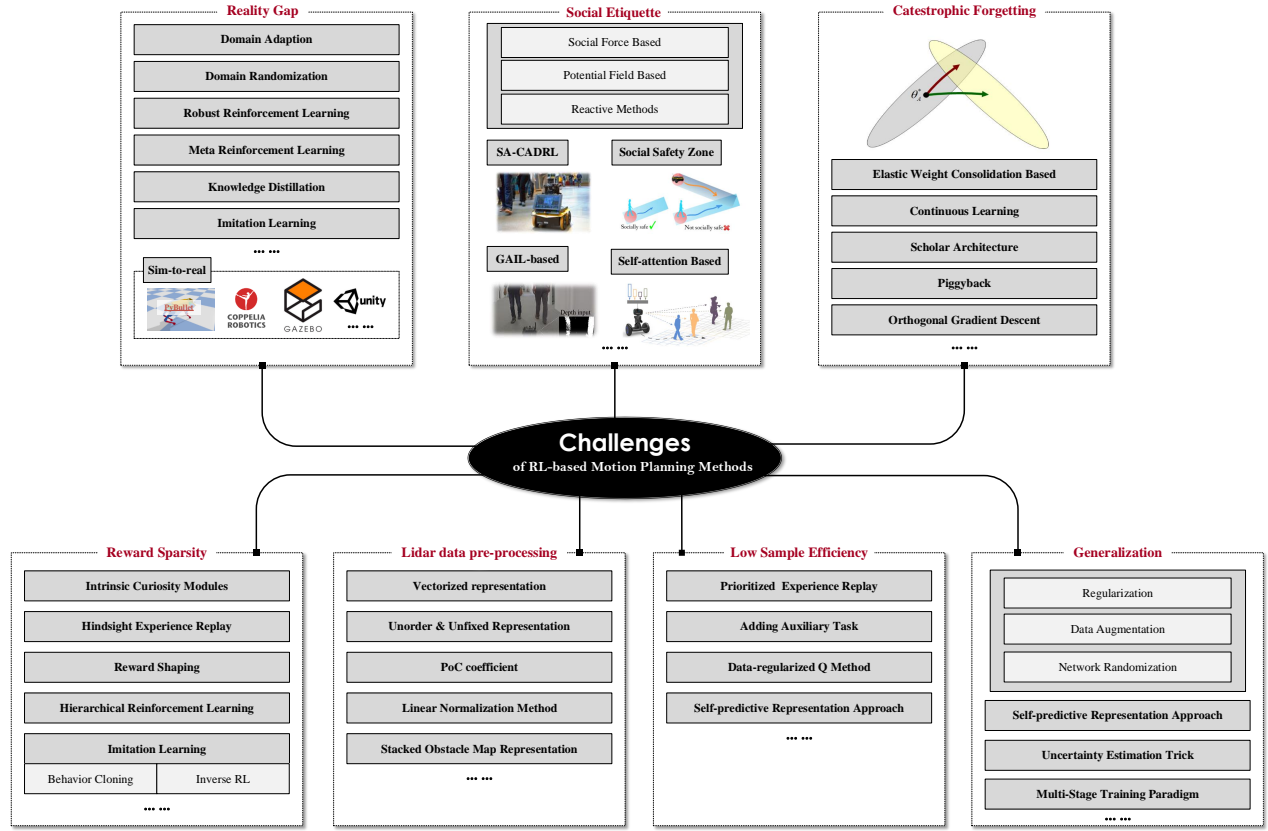


Fig. 8. Current RL-based motion planning approaches have many performance limitations, common challenges include reality gap problem, sparse reward problem, generalization, low sample efficiency, social etiquette, lidar data pre-processing issue, catastrophic forgetting problem, etc.

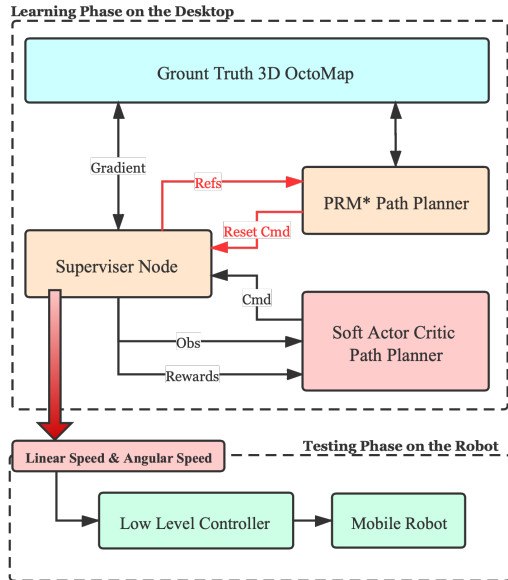


Fig. 9. The learning and testing phase architecture in the real-world scenarios of the depth-map-based Sim-to-Real robot navigation method in [110].

Besides, there are already some researchers studying HRL-based motion planning algorithms for mobile robots, such as [122], [123]. For more complex motion planning tasks that are

difficult to set up reward functions, 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 supervised learning process and suffers from the problem of mismatch problem between the actor and the expert policy. In [124], Zijng 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 the expert behavior. It utilizes 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 is commonly applied in the field of autopilot [125], [126].

3) *Generalization:* The generalization ability of the RL-based motion planning algorithm determines whether the mobile robot can perform safe and reasonable motion primitives in a novel scene different from the training stage. Most of the RL-based approaches for MR motion planning 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 [113]. In some scenes such as the interaction between pedestrians and mobile robots in campus, there exist potential safety hazards.

In the research field of underlying RL algorithm design, there are many researchers intend to enhance the generalization ability of RL agents. For example, the regularization

method [127], the data augmentation trick [128], etc. In [129], 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 techniques have been applied to the RL-based motion planning of MRs. In [102], 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 the safety of operation in new scenarios. In [113], a safe RL architecture is proposed to handle dynamic collision avoidance problem in novel scenarios. Different from traditional RL framework, it integrates the collision prediction networks based on the LSTM ensemble, the uncertainty estimation based on Monte-Carlo dropout and Bootstrapping process, and the safest action selection based on the model predictive control (MPC). The final results demonstrate that this type of uncertainty-aware pipeline helps the motion planning algorithm to cope with more complex or even brand new physical scenes.

4) *Low sample efficiency*: RL-based motion planning methods generate training data by interacting with the environment and updating the network parameters of actors and critics to obtain a satisfactory policy. Due to this mechanism, the sample efficiency of these methods are quite inefficient. This result will lead to excessive training time consumption, difficult convergence of the algorithm, and sometimes even training failure, which must be avoided in realistic robotics.

Schaul et al. propose prioritized experience replay (PER) method [130]. It computes the probability of a transition being sampled according to the importance (i.e., temporal-difference error) to improve the learning speed. PER is indeed an effective trick to improve the efficiency of off-policy RL methods. There is already some work applying it to RL-based motion planning methods [131], [132]. Lasjub et al. present constractive unsupervised representations as auxiliary task to speed up sample efficiency when input information is raw data such as images [133]. They extract valid information from the raw data through comparative learning process, and then feed the extracted information into the RL module for the whole training process. Kostrikov et al. proposed Data-regularized Q (DrQ) method [134]. This algorithm does data augmentation on the input observation before start sampling training process while calculating the target-Q and the current Q. Also, through combining with regularization trick, the sample efficiency of raw input data are significantly improved. Schwarzer et al. propose the self-predictive representation (SPR) approach [135]. It improves the sample efficiency by training agents to predict multi-step representations of their own potential future states. This operation allows agents to learn representations that are temporally predictive and consistent under different environmental observations.

5) *Social etiquette*: Currently, common MRs such as cleaning robots, disinfection robots, and security robots are mostly operating in pedestrian social scenes. In addition, in some specific densely crowded scenarios, such as train stations, hospitals, etc. Mobile robots using classical motion planning methods may cause freezing problem (Robots cannot find

any feasible action) because the probabilistic evolution of pedestrian could expand to the entire workspace [136]. Therefore, It is challenging but essential to deploy learning-based algorithms to train MRs to learn the social etiquette, anticipate human behaviors, and interact with humans in a safe, effective and socially-compliant manner.

Pioneer research works including social force based methods [137], potential field based methods [138], reactive methods (RVO, ORCA) [139], [140], etc. These methods are overly dependent on the hand-crafted function and lack a certain generalization ability for complex scenarios. Some works simplify the movement paradigm of pedestrians. They treats pedestrians as static obstacles or dynamic obstacles with simple kinematics over short timescales [104], [141]. These approaches are overly ideal. In practice, robots may produce unsafe action decision-makings due to their inaccurate prediction of human behaviors.

Typical RL-based methods including SA-CADRL [105], GAIL-based navigation methods [142], etc. These methods are effective, and constrain the specific interaction norm between the robot and the pedestrian, but rely on effective explicit pedestrian detection approaches. [89] and [90] construct rectangular social-safety zones for the MR and pedestrians, respectively, and design the corresponding safety reward term based on them. In [143], Changan Chen et al. propose a self-attention and deep v learning based agent-level crowd-aware robot motion planning approach. The main contribution of their work is to consider the more practical crowd-robot interaction problem rather than the first-order human-robot interaction problem. The state value estimation network of this deep v learning RL framework consists of three modules, an interaction module, a pooling module, and a planning module. The interaction module utilize a multi-layer perception (MLP) to extract the pairwise interaction feature between the robot and the nearby pedestrians. The pooling module output a weighted sum of above pairwise interactions by using the self-attention mechanism. The final planning module estimates the state value based on the compact representation of pairwise.

6) *Lidar data pre-processing issues*: Lidar data represents the distance information between the MR and the surrounding obstacles. Compared with the visual sensors, it is easier to realize the sim-to-real process. Therefore, Lidar is widely utilized in end-to-end motion planning tasks. However, improper Lidar data processing may lead to planning capability degradation of the MR in unknown scenarios. Many works have directly utilized the distance vector read from Lidar as part of the observation space in the RL framework (e.g. [84], [144]). This operation may cause some issues. For example, if the Lidar observation at a certain time-step in the inference scenario is similar to the training scenario, but with different passability. The agent may not be able to make different action decisions. Also, if the dimensionality of the sparse Lidar data is relatively high in the observation space, the agent may not have 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 [145]. 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 [146]. They introduce the PoC (proportion of distance values considered "close") ratio coefficient to achieve the differentiation of similar scenarios and help the MR to judge the complexity of the surrounding environment. Experiments show that the performance of their method is better than mainstream linear normalization method in [147]. Yuxiang Cui et al. in [88] 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 accurate compared to the angle range representation method.

7) *Catastrophic forgetting problem*: Catastrophic forgetting problem in RL-based motion planning of MRs refers to the forgetting of previously learned knowledge by agents when performing task-to-task continuous learning process. Since the motion planning process of robots generally involves multiple optimization objectives, the weights that are of importance for previous tasks might be changed to adapt to a new task. If changes contain parameters that are highly relevant 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 [148]. It solves the forgetting problem by calculating the Fisher information matrix to quantify the importance of the network parameter weights to the previous task, and adds it as a regularization term to constrain the update direction of the neural network while learning a new task. [83] combines EWC with DDPG algorithm and applies it to the multiple target motion planning problem of the mobile robot. Shin et al. propose a scholar architecture with a generator and a solver [149]. The old generator generates replay data and mixes it with the current task data as the training sets of the new task. It ensures that the new scholar does not forget the previous knowledge while learning new task. Mallya et al. present Piggyback [150]. It fixes a backbone network and learns a binary mask network for each task. Different binary masks are combined with the backbone network to perform different tasks to simplify computation and improve reuseability. Farajtabar et al. propose the orthogonal gradient descent method [151]. This approach reduces the forgetting of existing knowledge by orthogonally projecting the updated gradients of the new task onto 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. This causes the training process 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 on the basis of the fluctuation information of the loss function. It facilitates breaking the hard

boundary between individual subtasks and train the general motion planner that accomplishes multiple task goals.

2) *Meta RL based motion planning methods*: Meta learning help 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 MR to learn to explore in unknown environments. This not only prevents the planner from falling into the local optimum, but improves the learning speed and the generalization in different scenarios.

3) *Multi-modal fusion based RL motion planning methods*: At the sensor level, the utilization of individual type perception sensors exists performance limitations. By combining mapless end-to-end motion planning methods with multi-sensor fusion techniques to leverage the advantageous features of each perception modules. Moreover, it improves the environment cognition and understanding ability of MRs and enhances the fault tolerance and robustness of planners. Broadly speaking, multi-modal RL-based motion planning refers to the integration of 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 vision information as the observation in to improve the motion planning performance of the MR.

4) *Multitask objectives based RL motion planning methods*: Practical motion planner is often deployed with multiple task objectives. Such as ensuring the planning safety with the shortest distance while achieving the least global time or energy consumption. Classical motion planner separates the planning process and the optimization process. Therefore, it is worthwhile to research how to introduce multitask objectives learning in map-less end-to-end planning architecture. The breaking point is to improve the generalization by treating the domain information contained in the training signals of related tasks as induction bias, to learn general skills that could be shared and utilized across various related tasks, and to maintain a competitive balance between multitask objectives. Finally, merging multiple task-specific policies into a unified single optimal policy to enhance the representation of the planning policy.

5) *Human-Machine interaction mode based motion planning methods*: The currently application scenario for most RL-based motion planners is point-to-point navigation. The robot learns independently throughout the planning loop. In more complex and changing unstructured environments, human intelligence can be coupled with the 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 manual commands. This approach makes the MR unable to respond effectively to sudden external changes. Therefore, it is necessary to research on more in-depth ways integrate human and the robot intelligence to improve the human-machine interaction motion planning performance, 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. For multiple heterogeneous MRs system, it has a more extensive application domain (e.g air-ground cooperation planning and air-sea cooperation planning, etc.) and 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 priori map.

7) *Multi-MR flexible formation planning methods*: Multiple MR swarming and formation planning is widely used in military, logistics and transportation, intelligent agriculture, and resource exploration, etc. To realize the combination of the mainstream RL-based motion planner with the flexible formation, hierarchical learning paradigm, continuous learning paradigm can be developed or multimodal reward function can be designed to realize the passability performance of the external planner for different application scenarios and maintain the relative distance and angle relationships of the internal formation keeper. In addition, another focus of the research can be placed on the design of hybrid planning framework that incorporating RL-based motion planner and the mainstream formation algorithm to enhance the formation transformation and the formation recovery ability of the system.

VII. CONCLUSION

In this paper, we review the researches of mobile robot motion planning algorithms in recent years, with the focus on RL-based motion planning methods. We find that there exist three mainstream research directions: RL optimization motion planning methods, map-free RL-based motion planning methods, and RL-based multi-robot cooperative planning methods. Although, there are many research cases from simulations to real life applications, RL-based motion planners still have many performance bottlenecks if it is to be commercialized at scale, such as the generalization problem of different scenarios, the human-machine interaction problem of dense environments, and the catastrophic forgetting problem of multi-task training, etc. At last, combining these issues and the state-of-the-art techniques in the DRL research field, we present some suggestions for future research directions for RL-based motion planning methods, which provide a reference for the development of mobile robots with general artificial intelligence.

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, 2020.
- [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, 2016.
- [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] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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*. IEEE, 2011, pp. 1067–1069.
- [12] 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.
- [13] A. Stentz, "Optimal and efficient path planning for partially known environments," in *Intelligent unmanned ground vehicles*. Springer, 1997, pp. 203–220.
- [14] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning a*," *Artificial Intelligence*, vol. 155, no. 1-2, pp. 93–146, 2004.
- [15] D. Belanová, M. Mach, P. Šinčák, 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] M. Likhachev, G. J. Gordon, and S. Thrun, "Ara*: Anytime a* with provable bounds on sub-optimality," *Advances in neural information processing systems*, vol. 16, pp. 767–774, 2003.
- [20] M. Likhachev, D. I. Ferguson, G. J. Gordon, A. Stentz, and S. Thrun, "Anytime dynamic a*: An anytime, replanning algorithm," in *ICAPS*, vol. 5, 2005, pp. 262–271.
- [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 *2011 IEEE international conference on robotics and automation*. 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. Tengting, 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 ELECTRONICA 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*. 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 autolr," *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] N. Hansen, A. Ostermeier, and A. Gawelczyk, "On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation," in *ICGA*. Citeseer, 1995, pp. 57–64.
- [68] 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.

- [69] P. Rousseeas, 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.
- [70] 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.
- [71] 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.
- [72] 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.
- [73] 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.
- [74] 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.
- [75] 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.
- [76] R. E. Wang, M. Everett, and J. P. How, "R-maddpg for partially observable environments and limited communication," *arXiv preprint arXiv:2002.06684*, 2020.
- [77] 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.
- [78] 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.
- [79] 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.
- [80] 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.
- [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] 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.
- [83] 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.
- [84] 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.
- [85] 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.
- [86] 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.
- [87] 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.
- [88] Y. Cui, H. Zhang, Y. Wang, and R. Xiong, "Learning world transition model for socially aware robot navigation," *arXiv preprint arXiv:2011.03922*, 2020.
- [89] 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.
- [90] 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.
- [91] 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.
- [92] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, D. Kumaran, and R. Hadsell, "Learning to navigate in complex environments," 2017.
- [93] 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.
- [94] 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.
- [95] 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.
- [96] 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.
- [97] —, "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.
- [98] 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.
- [99] 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.
- [100] 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.
- [101] 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.
- [102] 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.
- [103] C. Yu, A. Velu, E. Vinitzky, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of mappo in cooperative, multi-agent games," *arXiv preprint arXiv:2103.01955*, 2021.
- [104] 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.
- [105] 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.
- [106] 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.
- [107] 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.
- [108] 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.
- [109] 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.
- [110] 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.
- [111] 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.
- [112] 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.
- [113] 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.
- [114] 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.
- [115] 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.
- [116] 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.
- [117] 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.
- [118] J. Liang, U. Patel, A. J. Sathyaamoorthy, 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.
- [119] 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.
- [120] 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.
- [121] A. Y. Ng, D. Harada, and S. Russell, “Policy invariance under reward transformations: Theory and application to reward shaping,” in *Icml*, vol. 99, 1999, pp. 278–287.
- [122] Z. Qiao, J. Schneider, and J. M. Dolan, “Behavior planning at urban intersections through hierarchical reinforcement learning,” *arXiv preprint arXiv:2011.04697*, 2020.
- [123] 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.
- [124] 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.
- [125] C. You, J. Lu, D. Filev, and P. Tsotras, “Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning,” *Robotics and Autonomous Systems*, vol. 114, pp. 1–18, 2019.
- [126] 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.
- [127] 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.
- [128] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas, “Reinforcement learning with augmented data,” *arXiv preprint arXiv:2004.14990*, 2020.
- [129] 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.
- [130] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, “Prioritized experience replay,” *arXiv preprint arXiv:1511.05952*, 2015.
- [131] 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*, 2021.
- [132] 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.
- [133] 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.
- [134] 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.
- [135] M. Schwarzer, A. Anand, R. Goel, R. D. Hjelm, A. Courville, and P. Bachman, “Data-efficient reinforcement learning with self-predictive representations,” 2021.
- [136] 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.
- [137] 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.
- [138] 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.
- [139] 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.
- [140] 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.
- [141] 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.
- [142] 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.
- [143] 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.
- [144] 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.
- [145] 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.
- [146] 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.
- [147] M. Pfeiffer, S. Shukla, M. Turchetta, C. Cadena, A. Krause, R. Siegwart, and J. Nieto, “Reinforced imitation: Sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4423–4430, 2018.
- [148] 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.
- [149] H. Shin, J. K. Lee, J. Kim, and J. Kim, “Continual learning with deep generative replay,” *arXiv preprint arXiv:1705.08690*, 2017.
- [150] 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.
- [151] 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.