# A Survey on Reinforcement Learning Applications in SLAM

Mohammad Dehghani Tezerjani [1], Mohammad Khoshnazar [2], Mohammadhamed Tangestanizadeh [3], Arman Kiani[4], Qing Yang[5]
[1, 5] Computer Science and Engineering, University of North Texas, Denton, USA
[2] Institute of Artificial Intelligence, University of Bremen, Bremen, Germany
[3] Computer Science and Engineering, University of California Santa Cruz, Santa Cruz, USA
[4, 5] Electrical and Computer Engineering, University of Maine, Maine, USA
Email: [1] mike.degany@unt.edu, [2] Khoshnam@uni-bremen.de, [3] mtangest@ucsc.edu,
[4] arman.kiani@maine.edu, [5] qing.yang@unt.edu

***Abstract*—The emergence of mobile robotics, particularly in the automotive industry, introduces a promising era of enriched user experiences and adept handling of complex navigation challenges. The realization of these advancements necessitates a focused technological effort and the successful execution of numerous intricate tasks, particularly in the critical domain of Simultaneous Localization and Mapping (SLAM). Various Artificial Intelligence (AI) methodologies, such as deep learning and Reinforcement Learning (RL), present viable solutions to address the challenges in SLAM. This study specifically explores the application of RL in the context of SLAM. By enabling the agent (the robot) to iteratively interact with and receive feedback from its environment, RL facilitates the acquisition of navigation and mapping skills, thereby enhancing the robot's decision-making capabilities. This approach offers several advantages, including improved navigation proficiency, increased resilience, reduced dependence on sensor precision, and refinement of the decision-making process. The findings of this study, which provides an overview of RL's utilization in SLAM, reveal significant advancements in the field. The investigation also highlights the evolution and innovative integration of these techniques.**

***Keywords—Simultaneous Localization and Mapping, Reinforcement Learning, Path Planning, Loop Closure Detection, Active SLAM.***

## I. INTRODUCTION

Mobile robotics involves designing, constructing, operating, and utilizing robots to perform tasks in dynamic, non-fixed environments. These robots are usually designed to be mobile and autonomous, capable of operating without direct human control [1]. Autonomous driving vehicles are one of the specific applications of mobile robotics, focusing on developing vehicles that can navigate and operate on their own in real-world environments, such as roads and highways. These vehicles use a combination of sensors, cameras, radar, Light Detection and Ranging (LiDAR), as well as advanced algorithms to perceive their surroundings and make decisions about how to navigate safely to their destination [2].

Simultaneous Localization and Mapping (SLAM) is a key technology in Mobile Robotics and Autonomous Driving (MRAD) [3]. SLAM enables a robot to navigate and create a map in an unknown environment by continuously observing map features to determine its own position and orientation [4, 5]. Localization enables the robot to determine its position within an environment, while mapping involves constructing a representation of the environment (the map) as the robot explores it [6]. The ability to accurately localize a robot using a map of its surroundings is essential for tasks ranging from spatial exploration to autonomous driving, as it provides the necessary information for predicting obstacle movements and determining optimal maneuvers [7, 8].

Accurate localization in robotics is a complex task due to the inherent noise in sensor measurements. Addressing outliers, occlusions, and sensor failures, as well as resolving scale differences between the map and robot motion, is essential for successful localization [4]. Additionally, detecting revisited locations, known as loop closures, presents a challenge due to perceptual aliasing and sensor limitations. It is crucial to maintain a consistent estimate of the robot's pose over time, especially in dynamic environments [6, 9]. Real-time performance is also a key consideration in ensuring effective robot navigation and localization [10].

Mapping presents several challenges that require attention to ensure accuracy and reliability. One such challenge is aligning sensor measurements with map features, which demands careful consideration and precise alignment [3, 11]. Furthermore, extracting meaningful features from sensor data, such as point clouds and images, is essential for constructing an accurate map [12]. It is also crucial to accurately estimate vehicle motion during mapping and to identify revisited locations to close loops in the map [13]. Choosing an appropriate map representation, whether it's 2D grids, 3D point clouds, or another format tailored to specific mapping needs, is another key consideration. Finally, maintaining map consistency as new data is incorporated is important for preserving its accuracy and reliability over time [6].

These challenges drive research and innovation in SLAM algorithms, aiming to improve accuracy, robustness, and efficiency. Artificial Intelligence (AI) algorithms are integral to SLAM, as they enable robots to navigate and map their

Received: 05- 09- 2024
Revised: 30-12- 2024
Published: 31- 12- 2024

environment in real-time. Specifically, Reinforcement Learning (RL) offers promising opportunities for improving exploration, localization, and map building in SLAM applications [14]. The integration of RL in the context of MRAD aims to offer a systematic approach for robots or vehicles to acquire the skills needed for navigating environments and making informed decisions using sensor data [15, 16]. Therefore, this study explores the practical uses of RL within the context of SLAM. The key contributions are detailed as follows:

- The concepts of SLAM were meticulously categorized into two distinct parts: passive and `active. This nuanced classification enhances the analysis of SLAM approaches in studies, offering a clearer framework for evaluating and comparing different methodologies.
- The sources of data utilized for input into the SLAM algorithm have been examined.
- RL in SLAM is divided into four main categories: path planning, loop closure detection, environment exploration, obstacle detection, and Active SLAM. This classification enables researchers to explore the various applications of RL in SLAM and stimulates the development of new ideas for improvement.

The rest of the paper is organized as follows: In Section II, types of SLAM approaches are introduced. Section III describes the modalities used for capturing information about the surrounding environment. Section IV explains RL and its operational principles. Section V focuses on reviewing and categorizing studies that have utilized RL in the context of SLAM. Section VI outlines the challenges in applying RL to SLAM. Finally, Section VII and VIII conclude the paper and depict future directions.

## II. Simultaneous Localization And Mapping

The SLAM technology was first introduced at a conference in San Francisco in 1986. It combines map recognition and initialization to achieve simultaneous positioning and map creation . SLAM is a collection of approaches utilized by robots to autonomously determine their location and map the surrounding environment as they traverse through it. The concept of SLAM can be further categorized into two main components: (1) localization, which involves estimating the robot's position in relation to the map, and (2) mapping, which involves reconstructing the environment using visual, visual–inertial, and laser sensors mounted on the robot . In modern SLAM techniques, a graphical approach is commonly adopted, specifically a bipartite graph where nodes represent either the robot or landmark poses, and edges represent measurements between poses or poses and landmarks. Imagine a robot characterized by a state vector $x \in R^2$ that defines its position and orientation (pose). The primary aim of the SLAM issue is to determine the optimal state vector $x^*$, minimizing the measurement error $e_i(x)$ weighted by the covariance matrix $\Omega_i \in R^{lxl}$, which accounts for the uncertainty in pose measurements, with l representing the state vector's dimension, as illustrated in (1).

$$x^* = arg\ \min_x \sum_t e_t^T(x)\Omega_t e_t(x) \quad (1)$$

There are two primary approaches to SLAM: Active SLAM and passive SLAM [20], which are discussed in the following.

### A. Passive SLAM approach

Passive SLAM systems do not involve navigating a robot to explore unfamiliar environments. Instead, they rely on predetermined routes or manual guidance and do not actively adapt to changes in the environment. Passive SLAM is particularly suitable for scenarios where precise robot motion is not critical, focusing instead on effective mapping and localization [21]. This approach facilitates more predictable and controlled movement, which can be advantageous in specific applications. However, it also implies that the robot may lack the ability to autonomously adjust to unforeseen changes in its environment without additional intervention [17]. Passive SLAM separates the estimation of robot motion from map estimation. Manual control or adherence to predetermined waypoints is characteristic of the robot's operation [4]. In passive SLAM, Particle Filters (PF) are commonly used to estimate robot poses and build maps. PF is a probabilistic method that represents the posterior distribution using a set of particles (samples), where each particle represents a potential robot pose and map hypothesis [22].

### B. Active SLAM approach

Active SLAM involves surveying the environment using sensors that are in motion, while simultaneously estimating the status of these sensors and constructing a map. Active SLAM setups use sensor readings as input and generate real-time decisions or actions to influence future measurements [23, 24]. Typically, it involves a three-part process [25]:

1. The recognition of all potential locations for exploration (ideally infinite),
2. The calculation of the efficacy or benefit derived from the actions that would transition the robot from its present coordinates to each of those locations,
3. The choice and implementation of the most advantageous course of action.

Active SLAM includes modules for planning waypoints and generating trajectories. It uses methods from information theory, optimal control theory, and RL to actively steer the robot towards its destination [4]. We discussed this approach in Section V and explored the RL application in Active SLAM.

## III. Data Source

SLAM in autonomous driving typically involves integrating data from multiple sensors to create a comprehensive perception of the surroundings [26, 27]. These sensors include LiDAR, camera, Global Navigation Satellite System / Inertial Navigation System (GNSS/INS), Inertial Measurement Unit (IMU), wheel odometry, and

radar. The sensor measurements are combined to provide a detailed and accurate understanding of the environment [22].

Furthermore, control commands from the vehicle's actuators such as steering, throttle, and brakes affect the robot's movement. RL techniques can optimize these control maneuvers to improve SLAM performance [4, 22]. SLAM algorithms usually start with an initial estimate of the robot's pose, derived from sources like GNSS or wheel odometry [28]. RL approaches can refine this pose estimation throughout the SLAM process, enhancing accuracy and reliability [29].

*LiDAR:* LiDAR, a remote sensing technology, uses laser light to measure distances and create detailed 3D maps of environments [30]. By emitting laser beams and calculating the time it takes for them to reflect off objects, LiDAR generates accurate point cloud data. This data is instrumental in creating comprehensive maps, identifying obstacles, and ensuring collision-free paths [31]. Additionally, LiDAR enhances precise localization by matching current scans with previously mapped features. LiDAR scans are thus pivotal for both mapping and obstacle detection, providing critical support for autonomous navigation systems [32].

*Cameras:* Cameras capture images of the surroundings, extracting visual features such as key points and edges to aid in robot localization and map construction [33]. These images are crucial for visual odometry and feature extraction, enabling the estimation of motion by tracking features across consecutive frames. Additionally, cameras help identify previously visited locations and contribute to the creation of detailed 3D maps [34].

*GNSS/INS:* GNSS offers global positioning information via satellite signals, providing data on latitude, longitude, and altitude. In contrast, INS utilizes accelerometers and gyroscopes to estimate position and orientation by measuring accelerations and angular rates [22]. GNSS delivers an initial position estimate, while INS ensures continuous tracking and compensates for GNSS signal outages, maintaining accurate navigation even in challenging conditions [35].

*IMU:* IMUs measure both accelerations and angular rates by combining accelerometers and gyroscopes to capture linear acceleration and angular velocity. IMUs provide short-term motion estimates, and their data is integrated with other sensors, such as LiDAR and cameras, to enhance the robustness of SLAM systems [36].

*Radar:* Radar sensors identify objects using radio waves, making them particularly valuable for obstacle detection in challenging weather conditions. In SLAM systems, radar sensors are enhancing accuracy, especially under poor lighting or occlusions [37]. The number of radar sensors used in a SLAM system varies depending on the application, requirements, and desired accuracy [38]. Some systems utilize a single radar sensor for both motion estimation and environmental mapping. This approach is particularly beneficial in low-light conditions or adverse weather conditions, where other sensors such as cameras or LiDAR may have limitations. Advanced SLAM systems may use multiple radar sensors to improve robustness against sensor failures and cover a full 360-degree field of view [39]. Combining data from multiple radars enhances accuracy and reduces blind spots [40]. Radar sensors are often integrated with other sensors to provide complementary information [41]:

- LiDAR: Offers long-range detection and high-resolution mapping.
- Cameras: Radar-camera fusion helps handle challenging lighting conditions.
- IMU: Improves motion estimation.

Ongoing research is focused on finding optimal configurations for radar-based SLAM, balancing factors such as cost, power consumption, and sensor placement [42].

*Wheel odometry:* Wheel odometry is a fundamental localization technique that uses wheel encoders to measure the rotation of a robot's wheels, allowing it to estimate incremental movement in terms of distance and direction [43]. This method is computationally inexpensive and widely available, providing continuous pose updates as the robot moves [44]. Encoders track the distance traveled, offering incremental pose updates, but they can accumulate errors over time due to factors like wheel slippage and uneven terrain. Encoder readings are also subject to noise, which can impact accuracy [45]. In SLAM systems, wheel odometry is commonly used as one of the sensor inputs [43]. RL algorithms can learn to integrate wheel odometry information for pose estimation. In RL-based SLAM, agents are trained to fuse wheel odometry data with other sensor modalities, such as LiDAR, cameras, and IMUs, to enhance overall system robustness and accuracy [26].

## IV. Reinforcement Learning

To effectively analyze data and develop intelligent automated applications, a solid understanding of RL is crucial [46]. In RL, an agent learns through interactions with its environment and receiving rewards [47]. The agent explores various actions to determine which yield the highest rewards over time. Given that actions can have lasting impacts, the return value *R*, defined by the reward function, is computed by summing discounted future rewards across episodes, as shown in (2) where γ denotes the discount factor, and $r(s_t,a_t)$ represents the reward for action $a_t$ in state $s_t$.

$$R = \sum_{t=0}^{T} \gamma^t r(s_t,a_t) \tag{2}$$

Each state is assigned a state value $V(s)$, which represents the expected return an agent can obtain by selecting actions. Equation (3) illustrates how $V(s)$ is computed.

$$\begin{aligned} V(s) &= E[R \vee s_0 = s] \; or \\ V(s) &= E[r(s) + \gamma V(s_{next})] \end{aligned} \tag{3}$$

The expectation symbol $\mathbb{E}$ represents the value of a state as the expected return while following a specific policy. The value of a state, denoted as $r(s)$, is the total expected sum of rewards achievable in that state [48]. Additionally, each action within a state has an action value $Q(s,a)$ determined by the Bellman equation (4) [49].

$$Q(s,a) = E[R \vee s_0 = s, a_0 = a] \text{ or}$$
$$Q(s,a) = E[r(s,a) + \gamma Q(s_{\text{next}}, a_{\text{next}})] \quad (4)$$

The state-action value, or Q-value, is the expected return of taking a specific action $a_0$ in a state $s_0$ and then following the policy throughout the episode. The relationship between state value and Q-value can be described by (5).

$$V(s) = \sum_{a \in A} \pi(a \vee s) Q(s,a) \quad (5)$$

Given a policy $\pi$ that selects action $a$ in state $s$, a Q-table can be constructed for a Markov Decision Problem (MDP) where all states and actions are known and limited. In this process, the agent takes actions and updates the Q-value in the Q-table based on the return received from the environment. The Q-value is updated using the (6), Where the updated $Q$ value, denoted as $Q'$, is calculated using a learning rate represented by $\alpha$ [48].

$$Q'(s_t, a_t) = Q(s_t, a_t) + \alpha([r_t + \gamma Q max(s_{t+1}, a_{t+1})] - [Q(s_t, a_t)]) \quad (6)$$

Fig. 1 illustrates an agent that learns from its environment using a RL algorithm. The algorithm updates the policy based on the actions taken, with the environment providing feedback in the form of rewards and the next state. This information is then used by the RL algorithm to improve the selection of actions.

Value-based and policy-based methods are two primary approaches used in RL to train agents to solve decision-making problems. In practice, choosing between value-based and policy-based methods depends on the specifics of the problem, such as the nature of the state and action spaces, stability concerns, and desired convergence properties [50].

### A. *Value-Based Methods*

Value-based methods focus on estimating the value function, which represents the expected return or reward that an agent can achieve from a given state (or state-action pair) while following a particular policy. The primary goal is to find an optimal value function that can be used to derive an optimal policy [51]. Common value-based methods include:

- Q-Learning: This algorithm aims to learn the action-value function (Q-function), which estimates the expected return of taking an action in a specific state and following the best policy thereafter. Once the Q-function is learned, the agent can act by selecting actions that maximize this value [52].
- Deep Q-Networks (DQN): An extension of Q-learning that uses deep neural networks to approximate the Q-function, making it feasible to scale Q-learning to environments with high-dimensional state spaces [53].
- SARSA: State-Action-Reward-State-Action (SARSA) updates the Q-values based on the action taken by the agent, ensuring that the learned policy reflects the actions taken during training. SARSA represents the sequence of events used to update the Q-values. In each step, after the agent takes an action and receives a reward, SARSA updates its Q-value estimates by considering the next state and the next action that the policy would take. [54].

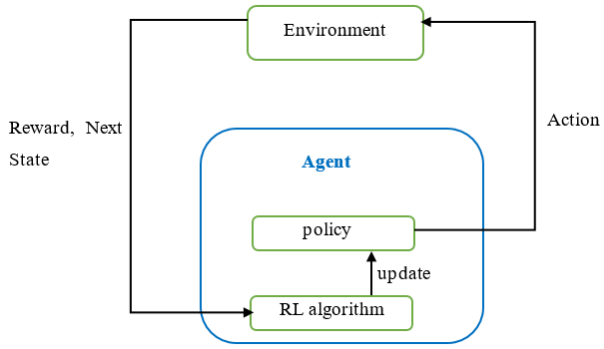


Fig. 1. Reinforcement Learning

### B. *Policy-Based Methods*

Policy-based methods focus on learning a policy directly. The policy can be either deterministic or stochastic, and it maps states (or observations) directly to actions, without necessarily using a value function to do so. In policy-based methods, the objective is typically to optimize the policy itself to maximize some measure of cumulative reward [50]. Key policy-based methods include:

*Policy Gradient Methods:* These methods specifically use gradient ascent to optimize the policy. The idea is to compute the gradient of the expected return with respect to the policy parameters and use this gradient to update the policy in a direction that improves performance [55].

- REINFORCE: This is the foundational policy gradient algorithm proposed by Williams (1992). It uses a Monte Carlo estimation of the policy gradient and is simple but often suffers from high variance, making convergence slower in some scenarios. REINFORCE directly optimizes the expected return of the policy by adjusting the weights of the neural network using sampled trajectories [56].
- Trust Region Policy Optimization (TRPO): TRPO algorithm enhances the foundational policy gradient method by incorporating a trust region constraint. This constraint is crucial as it ensures that the newly updated policy does not significantly diverge from the existing policy, a divergence quantified by the Kullback-Leibler (KL) divergence metric. By maintaining this constraint, TRPO stabilizes the training process. However, this stability comes at the expense of increased computational complexity [57]. TRPO guarantees incremental policy updates, thereby enhancing both stability and performance. It has been effectively applied in SLAM for making robust decisions in dynamic environments [58].
- Proximal Policy Optimization (PPO): PPO is a refinement of TRPO, developed to address its computational inefficiencies. It replaces the

complex trust region constraint with a simpler clipping mechanism in the surrogate loss function, which limits large policy updates. PPO is computationally efficient, easier to implement, and performs well across a wide range of tasks, making it one of the most popular policy gradient algorithms [59]. It has been applied to SLAM for efficient exploration and mapping [55].

- Actor-Critic Models: Actor-Critic Models refine the learning framework by integrating the advantages of both value-based and policy-based methods. The Actor component suggests actions based on current policy, while the Critic evaluates these actions by estimating the value function, providing a feedback loop that continually refines both policy and action value estimations [60]. This dual approach enables efficient learning and convergence, facilitating quick adaptation to changes in the environment, which is critical for real-time autonomous navigation and mapping [61].

### *C. Deep Reinforcement Learning*

Deep Reinforcement Learning (DRL) is a ML approach that combines RL with deep neural networks [62]. It differs from traditional methods by learning feature representations directly from raw data, such as images and sensor readings, rather than relying on handcrafted features [63]. This approach has shown great promise in solving complex decision-making problems and has been applied in various fields such as robotics, gaming, and autonomous systems [48, 64].

Value-based DRL methods, such as DQN and their extensions, have proven effective in addressing decision-making problems in discrete and low-dimensional action spaces. However, these methods face significant limitations when applied to tasks that require navigating continuous, high-dimensional action spaces. Active SLAM, which involves intricate path planning and precise control, exemplifies such challenges [55]. To overcome these limitations, researchers have increasingly turned to Policy Gradient Methods and Actor-Critic Models. These alternative approaches offer distinct advantages, enabling robots to autonomously navigate and map their environments while effectively handling the complexities of high-dimensional action spaces. This paradigm shift underscores the growing role of advanced RL techniques in addressing the multifaceted demands of active SLAM tasks [48].

### *D. RL-based SLAM applications*

RL has significantly impacted SLAM in practical robotics applications, particularly in autonomous vehicles, drones, and robotics. Traditionally, SLAM faced challenges in dealing with complex and dynamic environments where sensor noise, data association, and computational constraints could hinder performance.

*Robotics:* RL-based SLAM enables robots to autonomously navigate and map complex environments by learning optimal navigation strategies through trial and error. This method enhances robots' abilities to make real-time decisions and adapt to dynamic conditions in unfamiliar indoor settings while avoiding obstacles [48]. Traditional systems lack independent learning capabilities, but Lee et al. [65] introduced an innovative end-to-end approach using DRL for autonomous navigation in uncharted environments. They developed two deep Q-learning agents, the DQN and the double DQN, which allow robots to independently learn skills for collision avoidance and navigation. The process begins with target object detection using a deep neural network model, followed by navigation guidance using deep Q-learning algorithms. Moreover, recent research focused on improving SLAM for mobile robots in complex environments using advanced methodologies. Wong et al. [66] developed a multi-sensor fusion approach using DRL and Multi-Model Adaptive Estimation (MMAE), enhancing the localization precision and stability of robots in challenging conditions. Su et al. [67] presented Adaptive SLAM Fusion Degradation (ASLAM-FD), a framework combining adaptive multi-sensor fusion with degradation detection and DRL to maintain SLAM accuracy in dynamic environments. Liu et al. [68] created snake-like robots using SLAM and DRL for autonomous navigation and obstacle avoidance in hard-to-access areas. Their innovations, featuring lightweight structures with 2D LiDAR and IMU, significantly improved path planning and reduced collision rates compared to traditional methods.

*Autonomous Vehicles:* RL methods have become essential in addressing real-time SLAM tasks for autonomous cars, particularly self-driving vehicles operating in dynamic environments such as urban streets. These vehicles utilize RL to handle decision-making processes, including lane-keeping [69], obstacle avoidance [70], and route planning [71], while concurrently constructing and updating maps of their surroundings.

RL excels in such applications due to its ability to handle high-dimensional sensory data (e.g., LiDAR, cameras) and learn optimal control policies through interaction with the environment. For example, RL enables autonomous cars to adapt to dynamic traffic conditions, manage sensor noise, and make split-second decisions to ensure safety and efficiency [72]. Advanced RL techniques, such as Actor-Critic models and PPO, enhance these capabilities by improving stability and convergence during training. These methods allow the vehicles to navigate effectively in complex scenarios, such as unstructured roads, crowded intersections, and multi-agent environments, where traditional SLAM approaches might struggle [73].

*Drones:* RL is particularly impactful for drones due to their reliance on SLAM in GPS-denied environments. For example, RL allows drones to perform high-speed racing, where they must process rapid sensor inputs and dynamically adjust trajectories. Experiments demonstrate that RL-based controllers achieve better performance under real-world

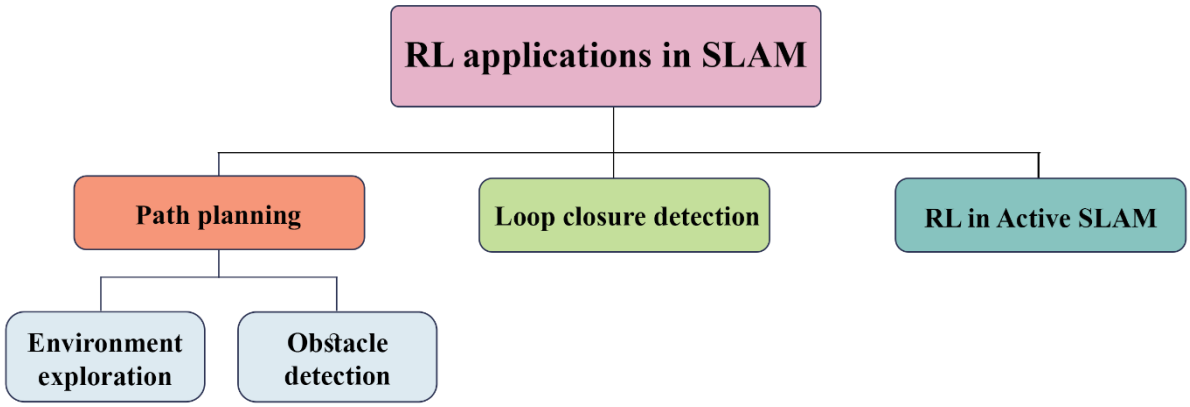


Fig. 2. Classification of RL applications in SLAM.

conditions compared to classical optimal control methods, especially when unmodeled dynamics or disturbances are present [72, 73].

## V. Classification of Existing Methods

RL has many applications within SLAM, an intersection rich with research endeavors. This study reviews the diverse applications of RL within SLAM, drawing upon an array of articles for insights and analysis. These applications span various domains, including path planning, loop closure detection, and Active SLAM, as delineated in Fig. 2, underscoring the multifaceted utility of RL techniques in advancing SLAM methodologies.

### A. Path Planning

Robot path planning technology, pivotal in robotics research, optimizes criteria such as minimizing work costs and finding the shortest route, ensuring efficient navigation while avoiding obstacles [74]. Path planning methods can generally be categorized into traditional and AI approaches. When considering environmental information, these methods can be further classified into global planning with known environmental data and local planning with unknown environmental information [75].

Wang [76] developed a visual SLAM system utilizing the ORB-SLAM3 framework. The system's primary function involves the generation of a dense point cloud map. Subsequently, this dense point cloud map from the visual SLAM system is converted into an octomap, followed by a projection transformation to the grid map. The next stage involves the development of a path planning algorithm rooted in RL. Experimental comparisons were conducted among the Q-learning algorithm, the DQN algorithm, and the SARSA algorithm. The outcomes showed that the DQN algorithm exhibits the swiftest convergence rate and superior performance, particularly in intricate environments characterized by high dimensions.

Nam et al. [29] introduced a novel framework for the navigation of mobile robots, integrating two established approaches (SLAM and DRL), to improve operational efficiency. The framework leverages SLAM to construct maps and pinpoint the robot's coordinates, while employing an Ant Colony Optimization (ACO) algorithm to formulate a predetermined route. In scenarios characterized by varying obstacles within the environment, the framework adopts DRL-based techniques for localized path planning. Furthermore, the suggested framework conducts a comparative analysis and assessment of the efficacy of three distinct DRL-based navigation algorithms: Deep Generative Network (DGN), Twin Delayed Deep Deterministic Policy Gradient (TD3), and Proximal Policy Optimization (PPO).

*a) Environment Exploration:*

The robot exploration model integrates various exploration methods and technologies, empowering robots to autonomously navigate, map, and explore unfamiliar environments efficiently. It leverages advancements in robotics, AI, and sensor technology to seamlessly fulfill these objectives [77].

Chen et al. [75] introduced a DRL-based robot exploration model designed for navigating unknown environments without any collisions. This innovative approach integrates SLAM technology and a DRL dual-mode structure to address local-minimum issues. After 30 training rounds, the model successfully achieved zero collisions and minimized repeated exploration. It surpasses existing methods for exploring unknown environments by a margin of less than 5%.

Li et al. [78] examined the concept of automatic exploration within unfamiliar environments through the application of DRL alongside a graph-based SLAM technique known as Karto SLAM. The proposed framework incorporates decision-making, planning, and mapping components that make use of a deep neural network to acquire knowledge pertaining to exploration strategies.

*b. Obstacle Detection:*

The obstacle detection model involves the development of algorithms and technologies to enable vehicles to detect obstacles in their surroundings accurately and in real time to ensure safe navigation and collision avoidance [79].

Wen et al. [80] used a fully convolutional residual networks method to identify road obstacles. The dueling DQN algorithm is also used in designing the robot's path. A two-dimensional map of the route is created by FastSLAM.

According to Nam et al. study in [81], SLAM algorithms are effective for mapping in the environment and DRL

algorithms can find dynamic obstacles well, but SLAM algorithms and DRL algorithms alone do not perform perfectly. In this method, SLAM algorithms combine data from several sensors, including LiDAR, to make a map of the environment. The ACO algorithm is used for planning to find the shortest optimal global path. DRL algorithms help the robot in planning the local path. In this way, the robot can make decisions based on its current situation and goal.

Fayjie et al. [82] used DRL for autonomous navigation and obstacle avoidance in self-driving cars. This study uses camera and laser sensor data and a trained neural network for driving. The DQN approach has also been implemented for autonomous driving simulation tests.

### *B. Loop Closure Detection*

Loop closure, a crucial process in robotics and autonomous vehicles, addresses inaccuracies in sensor measurements [83]. It also tackles various issues affecting reliability, such as drifting, acceleration changes, and weather conditions. By detecting when a vehicle revisits a previously visited location, loop closure helps to correct any accumulated errors in the system's map or position estimate [84]. This process is essential for ensuring the accuracy and credibility of the vehicle's navigation system and overall performance [85].

Iqbal et al. [86] investigated loop closure detection in simulated environments using a DRL approach. In this study, training was improved with entropy maximization for batch size selection. Furthermore, Bag-of-Words (BOW) method is used for loop closure and localization in maps, which represent an image using locally created features. DRL trains the probabilistic policy for loop closure detection. Furthermore, Convolutional Neural Networks (CNN) and region-based features are used for landmark proposal and matching.

In another study, Iqbal [87] presented two approaches to solve the problem of loop closure detection. The first approach uses statistical and clustering methods. In the second approach using DRL, loop closure detection is considered as a reward-driven optimization process. The proposed structure is implemented in a simulated grid environment. After generating the data, the learning process is done for the agent and the agent learns to detect the loop closure in variant environments.

### *C. RL in Active SLAM*

As mentioned in Section 2, Active SLAM is a method used by robots and autonomous systems to actively explore their surroundings. This approach allows the system to continuously enhance its understanding of the environment while also updating its position in real-time [88]. By making informed decisions on where to move next, the robot can effectively gather the most valuable data for mapping and localization purposes. This dynamic technique enables the system to adapt to changing environments and efficiently navigate through unknown areas [89, 90].

Fang et al. [91] harnessesed the power of MuZero to improve agents' planning abilities for joint Active SLAM and navigation tasks. These tasks involve navigating through unfamiliar environments while creating a map and determining the agent's location simultaneously. The paper introduces the SLAMuZero framework, which combines SLAM with the tree-search-based MuZero. SLAMuZero employs an explicit encoder-decoder architecture for mapping, along with a prediction function to assess policy and value using the generated map. The integration of SLAMuZero leads to a substantial decrease in training time.

Placed et al. [88] utilized deep Q-learning architecture with laser measurements for navigation and focused on reducing uncertainty in robot localization and map representation. Trained agents reduce uncertainty, transfer knowledge to new maps and learn to navigate and explore in simulations.

Pei et al. [92] introduced Active relative localization for multi-agent SLAMs. The task allocation algorithm is based on DRL and utilizes a Multi-Agent System DQN (MAS-DQN) to enhance collaboration efficiency in SLAM.

Alcalde et al. [93] used two agents namely completeness-based and uncertainty-based agents. According to the results, these agents completed maps without collisions. The uncertainty-based agent generated longer paths but better maps, and the Active SLAM DRL solution improved performance in complex environments.

Table 1 offers a comparative examination of the scrutinized research endeavors utilizing RL in the context of SLAM applications. It delves into the simulation environment, deep learning techniques, SLAM methodologies, and RL algorithms employed in these studies.

TABLE 1: COMPARISON OF REVIEWED STUDIES.

| | Ref. | Year | Simulation environment | Deep learning method | SLAM method | RL algorithm | Advantage | Disadvantage |
|---|---|---|---|---|---|---|---|---|
| ***Path planning*** | [76] | 2024 | Simple maze | Deep neural network | ORBSLAM3 | - Q-learning<br>- DQN<br>- SARSA | • Effective map conversion<br>• Contribution to autonomous navigation | • Dependence on sensor quality<br>• Computational demands |
| | [29] | 2023 | - Gazebo<br>- ROS<br>- TurtleBot | Deep neural network | SLAM-MCL | - Q-learning<br>- SARSA<br>- actor-critic | • Comprehensive framework<br>• Extensive experimental validation using various | • Limited generalization<br>• Potential overfitting |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | simulated environments | |
| ***Environment exploration*** | [75] | 2024 | - Gazebo<br>- ROS | CNN | - | DQN | • Effective Training Strategy<br>• Addressing Real-World Challenges | • Slower Exploration Speed<br>• No Real-World Implementation |
| | [78] | 2019 | ROS | CNN | Karto SLAM | DQN | • Modular Framework<br>• Efficient Exploration Strategy<br>• Generalization Performance | • Complexity of Implementation<br>• High Computational Burden |
| ***Obstacle detection*** | [81] | 2023 | - ROS2<br>- DDS Communication<br>- Gazebo | CNN | SLAM-MCL | - DQN<br>- PPO<br>- TD3 | • Adaptability to Dynamic Environments<br>• Exploration of Multiple DRL Algorithms | • Complexity in Implementation<br>• Restricted Problem Scope |
| | [80] | 2020 | Gazebo | Fully convolutional residual network | FastSLAM | Dueling DQN | • Scalable Action Space<br>• Improved Learning Efficiency | • High Computational Complexity<br>• Sensor Dependency |
| | [82] | 2018 | Unity Game Engine | CNN | - | DQN | • Sensor Fusion<br>• Efficient Simulation Environment<br>• Improved Training Stability | • Simulation-Limited Validation<br>• Simplistic Action Set<br>• Limited Scalability |
| ***Loop closure detection*** | [86] | 2022 | Turtlebot | CNN | VSLAM | Markov Decision Process | • Real-World Application<br>• Improved Feature Utilization | • Dependence on Prior Knowledge<br>• Computational Demands<br>• Hardware-Dependent Constraints |
| | [87] | 2019 | - Zoox<br>- Autonomous driving platform | CNN | VSLAM | Markov decision process | • Robust Data Association Method<br>• Scalable to Unknown Environments | • Dependency on Accurate Depth Estimation<br>• Computational Overheads |
| ***Active SLAM*** | [91] | 2024 | Habitat | Encoder-Decoder | - | - | • Efficiency in Training<br>• Improved Performance | • High Initial Complexity<br>• Unclear Scalability<br>• Potential for Overfitting |
| | [88] | 2020 | Gazebo | Deep Neural Network | - | - DQN<br>- DDQN<br>- D3QN | • Strong generalization capabilities<br>• Complex Simulation Validation | • Limited Real-world Testing<br>• Performance Variability |
| | [92] | 2020 | - ROS<br>- Telobot | Deep neural network | ORBSLAM | MAS-DQN | • Scalable Design<br>• Realistic Simulation | • Lack of Distributed Alternative<br>• Dependence on Centralized Coordination |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | [93] | 2022 | - Gazebo<br>- ROS<br>- ROBOTIS TurtleBot3-Burger | Deep Neural Network | Lightweight Passive SLAM | Partially Observable Markov Decision Process (POMDP) | • Focus on Map Quality and Robustness<br>• Efficiency through Dimensionality Reduction<br>• Flexible and Lightweight Approach | • Limited Physical World Validation<br>• Dependency on Reward Function Design |

## VI. Challenges in Applying RL to SLAM

In the rapidly evolving field of AI, RL has garnered significant attention for its potential to empower autonomous systems with the ability to learn and adapt through interaction with their environment. Despite its promising outlook, deploying RL in real-world applications faces considerable challenges [94]. These challenges posing unique hurdles that must be navigated to harness the full potential of RL technologies. Some of these limitations including:

*High Computational Demands:* RL models, especially those that extensively utilize deep learning frameworks, frequently require significant computational resources to perform optimally. The necessity for high processing power and substantial memory can pose a critical limitation in contexts where rapid decision-making is essential [95]. Consequently, attaining real-time performance with these models on edge devices, such as robots or drones, is particularly challenging. Such devices often possess limited processing capabilities, which can impede the implementation of sophisticated RL algorithms, even though these algorithms could potentially enhance the devices' autonomy and operational effectiveness [96].

*Safety and Reliability:* In the context of RL-based SLAM, guaranteeing safe exploration and robust decision-making is of paramount importance, especially for autonomous vehicles navigating through dynamic and complex urban environments. This necessity arises from the intricate nature of urban streets, which present a variety of unpredictable challenges such as varying traffic patterns, pedestrian movement, and environmental changes [97]. These factors can impact a vehicle's ability to accurately map and understand its surroundings for optimal path planning and hazard avoidance [95].

*Generalization Issues:* RL models often face challenges in adapting to diverse environmental conditions due to significant discrepancies between the controlled settings in which they are typically trained and the complex, unstructured nature of real-world scenarios. This limitation arises because the assumptions and constraints inherent in simulation environments fail to capture the full spectrum of variability and unpredictability present in natural settings. As a result, the application of RL models outside their training domains frequently leads to suboptimal performance, highlighting a critical gap in their generalization capabilities [96].

*High-dimensional state and action spaces:* The state space encapsulates all possible states in the environment. This encompasses the intricate array of sensor data vital for SLAM operations. Meanwhile, the action space delineates the spectrum of feasible actions available to the agent [98]. Within SLAM, these actions pertain to the movements, encompassing maneuvers like turning and accelerating. SLAM systems operate amidst a milieu of high-dimensional sensor data, ranging from intricate camera images to intricate LiDAR point clouds, essential for navigating complex environments [99]. However, the efficacy of RL agents in handling such expansive input spaces is challenged by the escalating computational complexity inherent in high-dimensional realms [30].

*Sample efficiency:* The ability of a RL algorithm to learn from a small number of interactions (samples) with the environment is known as sample efficiency [100]. Sample efficiency is important since real-world data collecting for autonomous vehicles in SLAM can be costly and time-consuming (e.g., using laser sensors) [101].

*Sensor/actuator delays:* Sensor/actuator delays epitomize the temporal gap between perceiving an event and enacting a response. This latency, inherent to the system, poses a critical challenge. In these domains, the journey from sensing to decision-making to action execution encompasses finite intervals, demanding precise synchronization [102]. RL algorithms must grapple with these delays to orchestrate timely and precise responses, ensuring seamless navigation and operation [103]. Within the intricate landscape of SLAM, this temporal precision assumes paramount importance, since the essence of success lies in the precision of real-time processing. This crucial element not only upholds but enhances the quality of localization and mapping, but also ensures a seamless fusion of navigational prowess [30].

## VII. Future Directions

In future research on the application of RL in SLAM, several key areas hold promise for advancing the state of the art:

- Adaptive sensor fusion: Combining data from various sensors, such as cameras, LiDAR, and Inertial Measurement Units (IMUs), is crucial for achieving robust SLAM. Future work could focus on developing RL agents capable of learning how to adaptively fuse information from these different modalities. By doing so, the overall performance and reliability of SLAM systems could be significantly enhanced, particularly in diverse and dynamic environments.
- Self-Supervised learning and data augmentation: The integration of Self-Supervised Learning (SSL) and data augmentation techniques offers substantial potential for improving RL-based SLAM, especially

in MRAD applications. Leveraging large amounts of unlabeled data and generating diverse training samples can enhance the robustness and generalization capabilities of SLAM systems. Future research should explore innovative SSL strategies and data augmentation methods to maximize the efficacy of RL in SLAM.

- Knowledge transfer across environments: For RL agents to be truly effective in real-world applications, they must be able to transfer knowledge across different maps or environments. Future studies should investigate techniques such as domain adaptation and meta-learning to facilitate better generalization of RL-based SLAM systems. These approaches can enable RL agents to apply learned knowledge from one environment to another, thereby improving their adaptability and performance in previously unseen settings.

By addressing these areas, future research can contribute to the development of more robust, efficient, and versatile RL-based SLAM systems, paving the way for advancements in the navigation of MRAD.

## VIII. Conclusion

SLAM is a technique used in robotics and autonomous vehicles to create a map of an unknown environment while simultaneously keeping track of an agent's location within that environment. It is a key technology for enabling MRAD to navigate and operate in real-world settings. SLAM involves the use of various sensors such as cameras, LiDAR, and odometry to gather information about the surrounding environment and then process this data to construct a map and estimate the agent's pose. In this survey, applications that have used RL in SLAM were investigated. According to the searches, the most use of RL in SLAM was in path planning, loop closure detection, environment exploration, obstacle detection, and Active SLAM. In these problems, RL helps the agent to design an intelligent map and facilitate navigation. SLAM methods can be effectively applied in MRAD, but the sensors, and environmental factors may need to be tailored to the respective application domain.

## References


[1] N. Sharma, J. K. Pandey, and S. Mondal, "A review of mobile robots: Applications and future prospect," International Journal of Precision Engineering and Manufacturing, vol. 24, no. 9, pp. 1695–1706, Sep. 2023, doi: 10.1007/s12541-023-00876-7.

[2] K. Othman, "Public acceptance and perception of autonomous vehicles: A comprehensive review," *AI and Ethics*, vol. 1, no. 3, pp. 355–387, Aug. 2021, doi: 10.1007/s43681-021-00041-8.

[3] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[4] M. F. Ahmed, K. Masood, V. Fremont, and I. Fantoni, "Active SLAM: A review on last decade," *Sensors*, vol. 23, no. 19, p. 8097, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/19/8097.

[5] J. Ren and D. Xia, "SLAM in autonomous driving," in *Autonomous Driving Algorithms and Its IC Design*. Singapore: Springer Nature Singapore, 2023, pp. 127–152.

[6] J. A. Placed *et al.*, "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *IEEE Transactions on Robotics*, 2023.

[7] X. Wu, G. Wang, and N. Shen, "Research on obstacle avoidance optimization and path planning of autonomous vehicles based on attention mechanism combined with multimodal information decision-making thoughts of robots," *Frontiers in Neurorobotics*, vol. 17, p. 1269447, 2023, doi: 10.3389/fnbot.2023.1269447.

[8] H. Taheri and Z. C. Xia, "SLAM; definition and evolution," *Engineering Applications of Artificial Intelligence*, vol. 97, p. 104032, Jan. 2021, doi: https://doi.org/10.1016/j.engappai.2020.104032.

[9] R. Zeng, Y. Wen, W. Zhao, and Y.-J. Liu, "View planning in robot active vision: A survey of systems, algorithms, and applications," *Computational Visual Media*, vol. 6, pp. 225–245, 2020.

[10] M. Diginsa, N. Shafie, and N. Yusuf, "Review: Issues and challenges of simultaneous localization and mapping (SLAM) technology in autonomous robot," *International Journal of Innovative Computing*, vol. 13, pp. 59–63, Nov. 2023, doi: 10.11113/ijic.v13n2.408.

[11] R. Eyvazpour, M. Shoaran, and G. Karimian, "Hardware implementation of SLAM algorithms: A survey on implementation approaches and platforms," *Artificial Intelligence Review*, vol. 56, no. 7, pp. 6187–6239, Jul. 2023, doi: 10.1007/s10462-022-10310-5.

[12] S. Li, Z. Li, X. Liu, C. Shan, Y. Zhao, and H. Cheng, "Research on map-SLAM fusion localization algorithm for unmanned vehicle," *Applied Sciences*, vol. 12, no. 17, p. 8670, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/17/8670.

[13] T. Liu, C. Xu, Y. Qiao, C. Jiang, and J. Yu, "Particle filter SLAM for vehicle localization," *arXiv preprint*, arXiv:2402.07429, 2024.

[14] C. Fan, Z. Li, W. Ding, H. Zhou, and K. Qian, *Integrating Artificial Intelligence with SLAM Technology for Robotic Navigation and Localization in Unknown Environments*, 2024.

[15] N. Botteghi, B. Kallfelz Sirmacek, R. Schulte, M. Poel, and C. Brune, *Reinforcement Learning Helps SLAM: Learning to Build Maps*, 2020, pp. 329–335.

[16] A. Hamza, "Deep reinforcement learning for mapless mobile robot navigation," 2022.

[17] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006, vol. 2, pp. 2161–2168.

[18] S. Saeedi, L. Paull, M. Trentini, and H. Li, "Multiple robot simultaneous localization and mapping," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 853–858.

[19] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[20] S. Frintrop, P. Jensfelt, and H. Christensen, *Simultaneous Robot Localization and Mapping Based on a Visual Attention System*, 2007, pp. 417–430.

[21] S. Thrun, "Simultaneous localization and mapping," in *Robotics and Cognitive Approaches to Spatial Mapping*. Springer, 2008, pp. 13–41.

[22] S. Zheng, J. Wang, C. Rizos, W. Ding, and A. El-Mowafy, "Simultaneous localization and mapping (SLAM) for autonomous driving: Concept and analysis," *Remote Sensing*, vol. 15, no. 4, p. 1156, 2023. [Online]. Available: https://www.mdpi.com/2072-4292/15/4/1156.

[23] H. Carrillo, I. Reid, and J. A. Castellanos, "On the comparison of uncertainty criteria for active SLAM," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2080–2087.

[24] W. Burgard, D. Fox, and S. Thrun, "Active mobile robot localization," in *IJCAI*. Citeseer, 1997, pp. 1346–1352.

[25] A. A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte, "An experiment in integrated exploration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1. IEEE, 2002, pp. 534–539.

[26] J. Lin, J. Peng, Z. Hu, X. Xie, and R. Peng, "ORB-SLAM, IMU and wheel odometry fusion for indoor mobile robot localization and navigation," *Academic Journal of Computer and Information Science*, vol. 3, 2020.

[27] A. Kumar, Maneesha, and P. K. Pandey, "Advances in simultaneous localization and mapping (SLAM) for autonomous mobile robot navigation," in *Proceedings of International Joint Conference on Advances in Computational Intelligence*. Springer Nature Singapore, Singapore, 2024, pp. 481–493.

[28] E. Reitbauer, C. Schmied, F. Theurl, and M. Wieser, "LIWO-SLAM: A LiDAR, IMU, and wheel odometry simultaneous localization and mapping system for GNSS-denied environments based on factor graph optimization," in *Proceedings of the 36th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2023)*, 2023, pp. 1669–1683.

[29] S. Nam, C. Woo, S. Kang, T. A. Nguyen, and D. Min, "SLAM-DRLnav: A SLAM-enhanced deep reinforcement learning navigation framework for indoor self-driving," in *2023 International Conference on Mechatronics, Control and Robotics (ICMCR)*, Feb. 2023, pp. 44–48, doi: 10.1109/ICMCR56776.2023.10181042.

[30] G. Dulac-Arnold *et al.*, "Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis," *Machine Learning*, vol. 110, no. 9, pp. 2419–2468, Sep. 2021, doi: 10.1007/s10994-021-05961-4.

[31] C. Debeunne and D. Vivet, "A review of visual-LiDAR fusion based simultaneous localization and mapping," *Sensors*, vol. 20, no. 7, p. 2068, 2020.

[32] J. Tang *et al.*, "LiDAR scan matching aided inertial navigation system in GNSS-denied environments," *Sensors*, vol. 15, no. 7, pp. 16710–16728, 2015.

[33] Y. Yang, D. Tang, D. Wang, W. Song, J. Wang, and M. Fu, "Multi-camera visual SLAM for off-road navigation," *Robotics and Autonomous Systems*, vol. 128, p. 103505, 2020.

[34] X. Lang *et al.*, "Gaussian-LIC: Photo-realistic LiDAR-Inertial-Camera SLAM with 3D Gaussian splatting," *arXiv preprint*, arXiv:2404.06926, 2024.

[35] L. Chang, X. Niu, and T. Liu, "GNSS/IMU/ODO/LiDAR-SLAM integrated navigation system using IMU/ODO pre-integration," *Sensors*, vol. 20, p. 4702, Aug. 2020, doi: 10.3390/s20174702.

[36] N. Abdelaziz and A. El-Rabbany, "INS/LIDAR/Stereo SLAM integration for precision navigation in GNSS-denied environments," *Sensors*, vol. 23, no. 17, p. 7424, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/17/7424.

[37] A. Woo, B. Fidan, and W. W. Melek, "Localization for autonomous driving," in *Handbook of Position Location: Theory, Practice, and Advances*, 2nd ed., 2018, pp. 1051–1087.

[38] M. S. A. Khan, D. Hussain, K. Naveed, U. S. Khan, I. Q. Mundial, and A. B. Aqeel, "Investigation of widely used SLAM sensors using analytical hierarchy process," *Journal of Sensors*, vol. 2022, pp. 1–15, 2022.

[39] Z. Hong, Y. Petillot, A. Wallace, and S. Wang, "Radar SLAM: A robust SLAM system for all weather conditions," *arXiv preprint*, arXiv:2104.05347, 2021.

[40] R. Huang, K. Zhu, S. Chen, T. Xiao, M. Yang, and N. Zheng, "A high-precision and robust odometry based on sparse MMW radar data and a large-range and long-distance radar positioning data set," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 98–105.

[41] J. Zhu, H. Li, and T. Zhang, "Camera, LiDAR, and IMU based multi-sensor fusion SLAM: A survey," *Tsinghua Science and Technology*, vol. 29, no. 2, pp. 415–429, 2023.

[42] Y.-E. Lu, S. Tsai, M.-L. Tsai, and K.-W. Chiang, "A low-cost visual radar-based odometry framework with mmWave radar and monocular camera," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 235–240, 2022.

[43] M. Quan, S. Piao, M. Tan, and S.-S. Huang, "Tightly-coupled monocular visual-odometric SLAM using wheels and a MEMS gyroscope," *IEEE Access*, vol. 7, pp. 97374–97389, 2019.

[44] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An overview to visual odometry and visual SLAM: Applications to mobile robotics," *Intelligent Industrial Systems*, vol. 1, no. 4, pp. 289–311, Dec. 2015, doi: 10.1007/s40903-015-0032-7.

[45] J. Mahmoud and A. Penkovskiy, *Dynamic Environments and Robust SLAM: Optimizing Sensor Fusion and Semantics for Wheeled Robots*, 2023, pp. 185–191.

[46] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Computer Science*, vol. 2, no. 3, p. 160, 2021.

[47] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[48] R. Singh, J. Ren, and X. Lin, "A review of deep reinforcement learning algorithms for mobile robot path planning," *Vehicles*, vol. 5, no. 4, pp. 1423–1451, 2023. [Online]. Available: https://www.mdpi.com/2624-8921/5/4/78.

[49] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A brief survey of deep reinforcement learning," *arXiv preprint*, arXiv:1708.05866, 2017.

[50] M. Sewak, S. K. Sahay, and H. Rathore, "Policy-approximation based deep reinforcement learning techniques: An overview," in *Information and Communication Technology for Competitive Strategies (ICTCS 2020): ICT Applications and Social Interfaces*, 2022, pp. 493–507.

[51] H. Byeon, "Advances in value-based, policy-based, and deep learning-based reinforcement learning," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 8, 2023.

[52] J. Clifton and E. Laber, "Q-learning: Theory and applications," *Annual Review of Statistics and Its Application*, vol. 7, no. 1, pp. 279–301, 2020.

[53] Y. Huang, "Deep Q-networks," in *Deep Reinforcement Learning: Fundamentals, Research and Applications*, 2020, pp. 135–160.

[54] A. Momenikorbekandi and M. Abbod, "Intelligent scheduling based on reinforcement learning approaches: Applying advanced Q-learning and state–action–reward–state–action reinforcement learning models for the optimisation of job shop scheduling problems," *Electronics*, vol. 12, no. 23, p. 4752, 2023.

[55] S. Zhao and S.-H. Hwang, "Exploration-and exploitation-driven deep deterministic policy gradient for active SLAM in unknown indoor environments," *Electronics*, vol. 13, no. 5, p. 999, 2024.

[56] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, pp. 229–256, 1992.

[57] J. Schulman, "Trust region policy optimization," *arXiv preprint*, arXiv:1502.05477, 2015.

[58] V. T. Potter, *Exploring Deep Reinforcement Learning Techniques for Autonomous Navigation*, 2022.

[59] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint*, arXiv:1707.06347, 2017.

[60] D. Joel, Y. Niv, and E. Ruppin, "Actor–critic models of the basal ganglia: New anatomical and computational perspectives," *Neural Networks*, vol. 15, no. 4–6, pp. 535–547, 2002.

[61] I. Grondman, M. Vaandrager, L. Busoniu, R. Babuska, and E. Schuitema, "Efficient model learning methods for actor–critic control," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 3, pp. 591–602, 2011.

[62] X. Wang *et al.*, "Deep reinforcement learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[63] A. Zai and B. Brown, *Deep Reinforcement Learning in Action*. Manning Publications, 2020.

[64] X. Lei, Z. Zhang, and P. Dong, "Dynamic path planning of unknown environment based on deep reinforcement learning," *Journal of Robotics*, vol. 2018, 2018.

[65] M.-F. R. Lee and S. H. Yusuf, "Mobile robot navigation using deep reinforcement learning," *Processes*, vol. 10, no. 12, p. 2748, 2022. [Online]. Available: https://www.mdpi.com/2227-9717/10/12/2748.

[66] C.-C. Wong, H.-M. Feng, and K.-L. Kuo, "Multi-sensor fusion simultaneous localization mapping based on deep reinforcement learning and multi-model adaptive estimation," *Sensors*, vol. 24, no. 1, p. 48, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/1/48.

[67] W. Su, T. Huang, F. Liu, and H. Wang, "ASLAM-FD: A multi-sensor adaptive collaborative fusion SLAM framework based on degradation detection and deep reinforcement learning," *Measurement Science and Technology*, vol. 35, no. 12, p. 126312, Sep. 2024, doi: 10.1088/1361-6501/ad7bdd.

[68] X. Liu, S. Wen, Y. Hu, F. Han, H. Zhang, and H. R. Karimi, "An active SLAM with multi-sensor fusion for snake robots based on deep reinforcement learning," *Mechatronics*, vol. 103, p. 103248, Nov. 2024, doi: https://doi.org/10.1016/j.mechatronics.2024.103248.

[69] Q. Wang, W. Zhuang, L. Wang, and F. Ju, "Lane keeping assist for an autonomous vehicle based on deep reinforcement learning," *SAE Technical Paper*, 0148-7191, 2020.

[70] C. Arvind and J. Senthilnath, "Autonomous vehicle for obstacle detection and avoidance using reinforcement learning," in *Soft Computing for Problem Solving: SocProS 2018, Volume 1*, Springer, 2020, pp. 55–66.

[71] K. Rezaee, P. Yadmellat, and S. Chamorro, "Motion planning for autonomous vehicles in the presence of uncertainty using reinforcement learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 3506–3511.

[72] C. Tang, B. Abbatematteo, J. Hu, R. Chandra, R. Martín-Martín, and P. Stone, "Deep reinforcement learning for robotics: A survey of real-world successes," *arXiv preprint*, arXiv:2408.03539, 2024.

[73] Y. Song, A. Romero, M. Müller, V. Koltun, and D. Scaramuzza, "Reaching the limit in autonomous racing: Optimal control versus reinforcement learning," *Science Robotics*, vol. 8, no. 82, p. eadg1462, 2023.

[74] M. Baziyad, M. Saad, R. Fareh, T. Rabie, and I. Kamel, "Addressing real-time demands for robotic path planning systems: A routing protocol approach," *IEEE Access*, vol. 9, pp. 38132–38143, 2021.

[75] S.-Y. Chen, Q.-F. He, and C.-F. Lai, "Deep reinforcement learning-based robot exploration for constructing map of unknown environment," *Information Systems Frontiers*, vol. 26, no. 1, pp. 63–74, 2024.

[76] W. Ruiqi, "Research on robot path planning based on reinforcement learning," *arXiv preprint*, arXiv:2404.14077, 2024.

[77] S.-Y. Chen, Q.-F. He, and C.-F. Lai, "Deep reinforcement learning-based robot exploration for constructing map of unknown environment," *Information Systems Frontiers*, vol. 26, Nov. 2021, doi: 10.1007/s10796-021-10218-5.

[78] H. Li, Q. Zhang, and D. Zhao, "Deep reinforcement learning-based automatic exploration for navigation in unknown environment," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 2064–2076, 2020, doi: 10.1109/TNNLS.2019.2927869.

[79] A. Iqbal, "Obstacle detection and track detection in autonomous cars," 2020.

[80] S. Wen, Y. Zhao, X. Yuan, Z. Wang, D. Zhang, and L. Manfredi, "Path planning for active SLAM based on deep reinforcement learning under unknown environments," *Intelligent Service Robotics*, vol. 13, pp. 263–272, 2020.

[81] S. Nam, C. Woo, S. Kang, T. A. Nguyen, and D. Min, "iNAV-drlSLAM: An improved indoor self-driving framework for mobile robots using deep reinforcement learning integrated with SLAM," in *2023 15th International Conference on Advanced Computational Intelligence (ICACI)*, May 2023, pp. 1–8, doi: 10.1109/ICACI58115.2023.10146173.

[82] A. R. Fayjie, S. Hossain, D. Oualid, and D. J. Lee, "Driverless car: Autonomous driving using deep reinforcement learning in urban environment," in *2018 15th International Conference on Ubiquitous Robots (UR)*, Jun. 2018, pp. 896–901, doi: 10.1109/URAI.2018.8441797.

[83] S. Arshad and G.-W. Kim, "Role of deep learning in loop closure detection for visual and lidar SLAM: A survey," *Sensors*, vol. 21, no. 4, p. 1243, 2021.

[84] K. A. Tsintotas, L. Bampis, and A. Gasteratos, "The revisiting problem in simultaneous localization and mapping: A survey on visual loop closure detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 19929–19953, 2022.

[85] A. Charroud, K. El Moutaouakil, V. Palade, A. Yahyaouy, U. Onyekpe, and E. U. Eyo, "Localization and mapping for self-driving vehicles: A survey," *Machines*, vol. 12, no. 2, p. 118, 2024. [Online]. Available: https://www.mdpi.com/2075-1702/12/2/118.

[86] A. Iqbal, R. Thapa, and N. R. Gans, "Deep reinforcement learning based loop closure detection," *Journal of Intelligent & Robotic Systems*, vol. 106, no. 2, p. 51, 2022.

[87] A. Iqbal, *Classified Object Localization in SLAM and Loop Closure Through Reinforcement Learning*. The University of Texas at Dallas, 2019.

[88] J. A. Placed and J. A. Castellanos, "A deep reinforcement learning approach for active SLAM," *Applied Sciences*, vol. 10, no. 23, p. 8386, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/23/8386.

[89] Y. Chen, S. Huang, and R. Fitch, "Active SLAM for mobile robots with area coverage and obstacle avoidance," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1182–1192, 2020.

[90] N. Palomeras, M. Carreras, and J. Andrade-Cetto, "Active SLAM for autonomous underwater exploration," *Remote Sensing*, vol. 11, no. 23, p. 2827, 2019.

[91] B. Fang, X. Chen, Z. Pan, and X. Di, "SLAMuZero: Plan and learn to map for joint SLAM and navigation," in *34th International Conference on Automated Planning and Scheduling*.

[92] Z. Pei, S. Piao, M. Quan, M. Z. Qadir, and G. Li, "Active collaboration in relative observation for multi-agent visual simultaneous localization and mapping based on deep Q network," *International Journal of Advanced Robotic Systems*, vol. 17, no. 2, p. 1729881420920216, 2020.

[93] M. Alcalde, M. Ferreira, P. González, F. Andrade, and G. Tejera, "DA-SLAM: Deep active SLAM based on deep reinforcement learning," in *2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE)*. IEEE, 2022, pp. 282–287.

[94] R. Zhao, Y. Li, Y. Fan, F. Gao, M. Tsukada, and Z. Gao, "A survey on recent advancements in autonomous driving using deep reinforcement learning: Applications, challenges, and solutions," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[95] A. Tourani, H. Bavle, J. L. Sanchez-Lopez, and H. Voos, "Visual SLAM: What are the current trends and what to expect?," *Sensors*, vol. 22, no. 23, p. 9297, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/23/9297.

[96] M. N. Favorskaya, "Deep learning for visual SLAM: The state-of-the-art and future trends," *Electronics*, vol. 12, no. 9, p. 2006, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/9/2006.

[97] H. Bavle, J. L. Sanchez-Lopez, C. Cimarelli, A. Tourani, and H. Voos, "From SLAM to situational awareness: Challenges and survey," *Sensors*, vol. 23, no. 10, p. 4849, 2023.

[98] C. Tessler, T. Zahavy, D. Cohen, D. J. Mankowitz, and S. Mannor, "Action assembly: Sparse imitation learning for text-based games with combinatorial action spaces," *arXiv preprint*, arXiv:1905.09700, 2019.

[99] A. Kudriashov, T. Buratowski, M. Giergiel, and P. Małka, *SLAM Techniques Application for Mobile Robot in Rough Terrain*. Springer, 2020.

[100] J. Buckman, D. Hafner, G. Tucker, E. Brevdo, and H. Lee, "Sample-efficient reinforcement learning with stochastic ensemble value expansion," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[101] C. Pang, L. Zhou, and X. Huang, "A low-cost 3D SLAM system integration of autonomous exploration based on fast-ICP enhanced LiDAR-inertial odometry," *Remote Sensing*, vol. 16, no. 11, p. 1979, 2024.

[102] T. Hester and P. Stone, "Texplore: Real-time sample-efficient reinforcement learning for robots," *Machine Learning*, vol. 90, pp. 385–429, 2013.

[103] N. Botteghi, B. Sirmacek, R. Schulte, M. Poel, and C. Brune, "Reinforcement learning helps SLAM: Learning to build maps," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 329–335, 2020.