Efficient Exploration in Crowds by Coupling Navigation Controller and Exploration Planner

Zhuoqi Zheng, Shengfeng He, and Jia Pan, *Senior Member, IEEE*

***Abstract*—Autonomous exploration in scenes with moving pedestrians is critical for deploying autonomous mobile robots in populated places such as malls, airports, and museums. The existence of dynamic obstacles poses challenges on achieving an efficient, safe, and robust exploration system: the robot may get stuck in the pedestrians without making progress in scene coverage; it may collide with humans and hurt them; the human-robot collision will fail the exploration process or cause large drift and artifacts in simultaneous localization and mapping (SLAM). In this work, we propose a framework that can solve these challenges by tightly coupling a reinforcement learned navigation controller and a hierarchical exploration planner enhanced with a recovery planner. The navigation controller provides a value function describing the distribution of crowds around the robot, which will be leveraged by exploration planner and recovery planner to minimize the human-robot interruptions. We evaluate the proposed exploration framework against several methods on a set of indoor benchmarks with pedestrians, verifying the advantages of our method in terms of exploration efficiency, navigation safety, and SLAM quality.**

***Index Terms*—Collision avoidance, motion and path planning.**

I. Introduction

AUTONOMOUS exploration, which aims to drive the robot to traverse unknown environments for mapping, is one fundamental task of mobile robots, and has real-world applications including visual inspection, search and rescue, and sterilization etc.. Traditional exploration approaches are designed for quick coverage of unknown environments by selecting the next visiting frontier via cost-based, sample-based, or potential field-based selection strategies. However, these methods cannot deal with the challenges in real-world scenarios caused by human crowds, such as airports, museums, hospitals, and other populated places.

In populated places, dynamic pedestrians not only degrade SLAM's accuracy in autonomous exploration but also demand more sophisticated navigation control. In particular, occlusions of sensor inputs by moving objects may lead to scan matching errors in SLAM. Most previous research attempts to resolve this issue by pre-processing dynamic obstacles at the SLAM front-end. One typical solution is detecting and removing pedestrians from the laser scan to obtain clean measurements [1]. However, simply neglecting dynamic objects in the planning step of the exploration system will lead to high collision risks preventing the mobile robot from reaching its exploration goal. One alternative is to continuously track and predict pedestrians [2], which benefits the high-level exploration planning and improves SLAM's stability in crowds. However, due to human motion's high uncertainty, the robot has to treat prediction results conservatively, which may make the robot fail to find a valid collision avoidance solution, making safe and efficient exploration in crowds difficult.

Our previous work [3] addresses the problem of safe and efficient exploration in dense crowds by combining a two-layer exploration planner with an reinforcement learning (RL)-based collision avoidance algorithm as the navigation controller, which guarantees the robot to navigate safely through crowds. However, in [3] the exploration planner chooses the target viewpoint by applying the typical next-best-view (NBV) algorithm on the explored map and it does not take the pedestrians into consideration, i.e., the viewpoint selection and the collision avoidance are completely decoupled. In this way, the robot may choose a target location occupied by pedestrians or an exploration direction against the crowd flow. Then the robot has to walk more tortuous paths through the crowd, leading to many collisions. Even worse, the robot can get stuck in crowds and stop making progress in exploration. To avoid this issue, we need to penalize viewpoint candidates with many nearby dynamic obstacles. Our solution is a novel framework where the exploration planner is tightly coupled with the RL-based navigation controller. In particular, the exploration planner is guided by the Critic network as a byproduct of training the RL-based controller. By combining them together, the robot can perceive the distribution of surrounding pedestrians without an explicit model and select a target viewpoint that not only maximizes the map information gain for efficient exploration but also minimizes the human-robot interruption for safe navigation.

In addition, even if the robot could navigate through dense crowds, dynamic obstacles still negatively impact the SLAM module in the exploration system. Since SLAM is generally designed for static scenes, unexpected dynamic objects would corrupt the quality of robot state estimation or even lead to

Manuscript received 7 April 2022; accepted 7 September 2022. Date of publication 10 October 2022; date of current version 13 October 2022. This letter was recommended for publication by Associate Editor H. Soh and Editor H. Kurniawati upon evaluation of the reviewers' comments. This work was supported in part by Hong Kong General Research Fund (GRF) under Grants 11207818 and 11202119, and in part by the Centre for Transformative Garment Production *(Corresponding author: Jia Pan.)*

Zhuoqi Zheng is with the Department of Biomedical Engineering, City University of Hong Kong, Kowloon Tong, Hong Kong, and also with the Department of Computer Science, The University of Hong Kong, Pok Fu Lam, Hong Kong (e-mail: zqzheng3-c@my.cityu.edu.hk).

Shengfeng He is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China (e-mail: shengfenghe7@gmail.com).

Jia Pan is with the Department of Computer Science, The University of Hong Kong, Hong Kong (e-mail: panjia1983@gmail.com).

Digital Object Identifier 10.1109/LRA.2022.3212670

2377-3766 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: National Tsing Hua Univ.. Downloaded on August 28,2025 at 03:40:41 UTC from IEEE Xplore. Restrictions apply.

localization and mapping failure. A typical solution is to add a recovery mode, where the robot performs active re-localization, i.e., approaching a location with sufficient features to save SLAM from degraded quality or failures. But in a scene with dense crowds, the robot may not be able to reach the recovery location easily, especially when the location is occupied by pedestrians moving against the robot.

In this paper, we solve the challenging problem of robotic exploration in scenes with crowds. The exploration framework consists of a global planner determining the global goal location, a local planner selecting a local goal location based on both information gain and the crowd distribution around the robot, and a navigation controller calculating the robot's steering command. In addition, the framework is assisted by a recovery mode that automatically performs re-localization, which allows the robot to actively navigate to recovery locations determined by the explored map and historical path when the quality of the SLAM is inferior. As a summary, our contributions are three-fold:

- A two-layer hierarchical exploration planner that is tightly coupled with a reinforcement learned navigation controller in crowds. The exploration planner allows the robot to determine optimal exploration policy not only maximizing the information gain of the explored map but also minimizing the human-robot interruption.
- A SLAM recovery planner that re-localizes the robot by moving toward a recovery point, which is selected from feature-rich locations in the explored map and the historical trajectory. The selection process is also tightly coupled with the reinforcement learned navigation controller in crowds.
- Our exploration framework coupling navigation controller with exploration and recovery planners demonstrates satisfactory performance in exploration efficiency, navigation safety, and SLAM quality.

II. Related Work

Here we briefly summarize previous works related to robotic exploration in scenarios with pedestrian crowds.

Early exploration methods mostly adopt the frontier-based approach proposed by [4]. For frontier selection, cost-based methods [5] navigate the robot to the nearest frontier and [6] accelerates the exploration efficiency by exploiting background knowledge of the environment. Recent research formulates the exploration as a multi-resolution hierarchy balancing efficiency and robustness in large-scale environments [7]. Besides traditional methods, deep reinforcement learning (DRL) has been applied in robotic exploration. [8] designs an Asynchronous Advantage Actor-Critic (A3C) [9] network, which has the explored map, robot's location, and frontiers' location as its input and has dynamic coefficients of the utility function as output. [10] uses a similar network structure, but changes the network's output to be the robot's next visiting direction. However, these robotic exploration methods are only conducted and tested on static scenes without considering the effect of pedestrians on the robot's behavior.

The study of robot behavior in crowds is mainly about safe navigation in dynamic environments. While the classical model-based Dynamic Window Approach (DWA) [11] uses cost maps to store information about obstacles, many other model-based approaches use human social interaction models, such as social forces [12], [13], Gaussian mixture models (GMMs) [14] and risk map [15], [16] etc.. All these approaches require knowledge of pedestrian kinematic models and often need manual parameter tuning. DRL has also become popular for navigation in dynamic environments. [17] proposes a fully decentralized multi-robot collision avoidance framework with a network policy trained using the Proximity Policy Optimization (PPO) algorithm [18]. Other DRL approaches use self-attention modules to model robot-human interactions [19], [20] or graph convolutional networks (GCN) to encode crowd states [21]. These works demonstrate the importance of pedestrian kinematic information for achieving safe navigation in dynamic environments. However, most of the above-mentioned studies are designed and evaluated on pre-built maps while in robotic exploration in crowds, the robot does not have a priori map and must conduct mapping, navigation, and collision avoidance of pedestrians simultaneously in a partial-observed map that grows incrementally.

However, autonomous exploration in crowds without a priori map has not been well studied yet. Most current research focuses on the problem of SLAM in dynamic environments [22], [23], where dynamic obstacles are treated as outliers or are tracked individually. [1] first filters pedestrians from the 2D laser range data and then the exploration system only needs to deal with localization and mapping with static data. None of these approaches considers the tradeoff between the robot's exploration efficiency and navigation safety in terms of robot-crowd collision avoidance. [3] attempts to solve this issue at the motion planning stage of the exploration system by introducing the advanced collision avoidance algorithm [17] to navigate through crowds. Even showing good performance, in [3] the collision avoidance controller is decoupled with the viewpoint selection for exploration. As a result, this method may select viewpoints which attract the robot to move against the crowd flow rather than along with the flow. In this way, the robot will be trapped in crowds and make very slow progress in scene coverage. As a solution to this issue, in this work we propose a new exploration framework with tightly coupled *planners* (for exploration viewpoint selection and failure recovery) and *controller* (for collision avoidance in crowds), in order to make a good tradeoff between exploration efficiency and safety in dense crowds.

III. Methodology

Our proposed exploration framework, as shown in Fig. 1, consists of two operation modes, the exploration mode and the recovery mode, which switched between each other according to the quality of the explored map evaluated by a mode switch. The exploration mode is implemented as a hierarchical exploration planner, including a global planner and a local planner. The global planner determines a touring order in terms of a coarse next-visiting goal, while the local planner selects an optimal viewpoint that makes a trade-off between maximizing the map

Authorized licensed use limited to: National Tsing Hua Univ.. Downloaded on August 28,2025 at 03:40:41 UTC from IEEE Xplore. Restrictions apply.

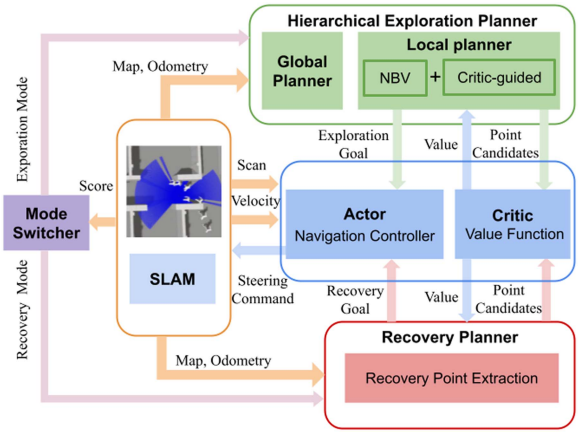

Fig. 1. Our system switches between two modes, the exploration mode (green) and the recovery mode (red), according to the SLAM quality. The RL-based navigation module's Actor network serves as a navigation controller, whose input includes the laser scan, the robot position, and the goal position provided by the exploration or recovery planner. Its Critic network is responsible for the goal point evaluation in exploration and recovery planners. .

information gain and minimizing the interruption between the robot and crowds. The recovery planner chooses a feature-rich recovery point in the mapped region which can help the robot to recover from SLAM failures. The recovery point is selected according to the effort required to reach this point in terms of navigation distance and easiness to move through the crowds. The crowd-related information used in the exploration planner and recovery planner is provided by the navigation controller, which is trained using an actor-critic RL algorithm. In this way, we achieve a tightly-coupled autonomous exploration framework where the navigation controller not only generates an appropriate steering command for collision avoidance, but also guides the exploration planner and recovery planner to prefer regions with fewer pedestrians for minimizing the interruption between the robot and crowds.

Unlike our previous work [3] where navigation controller and exploration planner are completely decoupled, in this work the RL-based navigation controller is tightly coupled with both the exploration planner and recovery planner for viewpoint and recovery point selection. The recovery planner also leverages the robot's historical trajectory and the explored map for re-localization, to prevent the SLAM module from being crashed in dense crowds. The design of each module in our proposed exploration system will be described in detail as follows.

A. TSP-Based Global Exploration Planner

The TSP-based global planner optimizes a touring order of all frontiers to determine a global goal $\mathbf{g}_{\text{global}}$. To optimize the touring order, we solve the Travelling Salesman Problem (TSP) on a graph $G = (V, E)$. In $G = (V, E)$, the set of nodes $V = v_0, \ldots, v_n$ consists of the current robot position $\mathbf{p}$ and all frontiers, and the set of undirected edges $E \subset V \times V$ describes whether the robot can traverse between these two nodes. The weight of an edge in E is defined as the length of the shortest path between its two end nodes, which is computed by running the A^* algorithm over a low-resolution map to reduce the computational cost. After the graph G is built, we use the TSP solver in Google OR-Tools [24] to find the optimal touring order $\mathcal{T}$ of all frontiers. In the resulting $\mathcal{T}$, the frontier right after the robot position node will be sent to the local exploration planner as the global goal $\mathbf{g}_{\text{global}}$.

B. Critic-Guided Local Exploration Planner

In our previous work [3], an NBV-based algorithm is used to refine a local viewpoint $\mathbf{g}_{\text{local}}$ with the highest map information gain from a global goal $\mathbf{g}_{\text{global}}$ for maximizing the exploration efficiency. However, since the information gain does not take into account the distribution of surrounding pedestrian crowds, driving the robot toward a region with highest information gain may make it move against the pedestrian flow, resulting in low exploration progress and high risk of collisions. To select a viewpoint appropriately balancing the low pedestrian density and high information gain, we leverage the information provided by the Critic network of the RL-based navigation controller. This value function describes how the crowd distribution in the robot's neighborhood influences its effort to move in different directions, as to be introduced in Section III-C. In this way, the local exploration planner will prefer robot movement direction toward viewpoint with fewer pedestrians or consistent with the crowd flow.

In particular, given the global goal $\mathbf{g}_{\text{global}}$ from the global planner, the local planner computes a viewpoint location within a sub-map M_{local} of the explored map M. The local map $M_{\text{local}} = M_{\text{sensor}} \cup M_{\text{goal}}$ covers two parts. The first is the sub-map M_{sensor}, which is the region within the range of the laser sensor. The second sub-map M_{goal} is a minimal rectangular area covering the robot position $\mathbf{p}$ and the global goal $\mathbf{g}_{\text{global}}$. In the extracted local map, we first randomly select multiple exploration viewpoint candidates $\mathbf{p}_{\text{local}}^{1:k}$ around $\mathbf{g}_{\text{global}}$, which will then be evaluated according to two different criteria to be introduced below.

1) Information Gain-Based Evaluation: We score k sampled viewpoint candidates $\mathbf{p}_{\text{local}}^{1:k}$ according to the evaluation function $V_{\text{nbv}}(\mathbf{p}_{\text{local}})$, which rewards the normalized information gain $-\frac{1}{|\mathcal{Z}|}\sum_{i=1}^{k} p(z_i \,|\, \mathbf{p}_{\text{local}}) \log p(z_i \,|\, \mathbf{p}_{\text{local}})$ [25] and penalizes the distance cost $\frac{L(\mathbf{p}_{\text{local}})}{L_{\max}}$:

$$V_{\text{nbv}}(\mathbf{p}_{\text{local}}) = -\frac{1}{|\mathcal{Z}|}\sum_{i=1}^{k} p\left(z_i \,|\, \mathbf{p}_{\text{local}}\right) \log p\left(z_i \,|\, \mathbf{p}_{\text{local}}\right) - \alpha \frac{L(\mathbf{p}_{\text{local}})}{L_{\max}},$$

where the normalization factor $|Z|$ is the number of map grids within the range of the laser ranger sensor and $p(z_i|\mathbf{p}_{\text{local}})$ is a uniform distribution modeling the belief of map grid i with measurement z_i. Usually, $p(z_i|\mathbf{p}_{\text{local}}) = 0.5$ for unexplored grid and $p(z_i|\mathbf{p}_{\text{local}}) = 1$ once the grid is perceived by the laser scanner. $L(\mathbf{p}_{\text{local}})$ is the path length between the viewpoint $\mathbf{p}_{\text{local}}$ and the robot's current position $\mathbf{p}$, and $L_{\max}$ is the maximal path length among all k viewpoints $L(\mathbf{p}_{\text{local}}^{1:k})$.

Authorized licensed use limited to: National Tsing Hua Univ.. Downloaded on August 28,2025 at 03:40:41 UTC from IEEE Xplore. Restrictions apply.

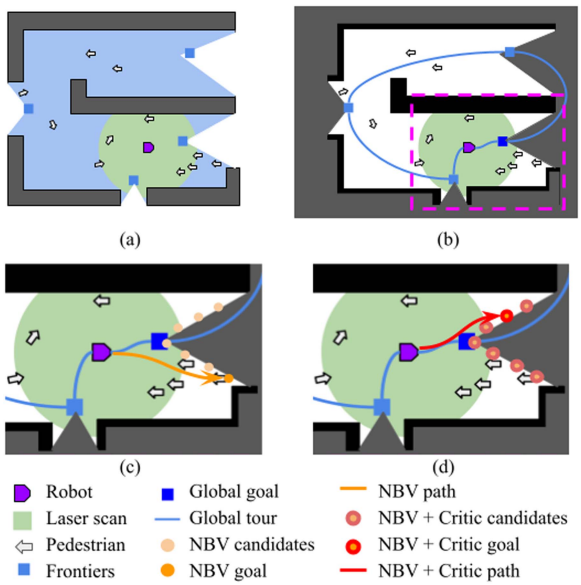

Fig. 2. (a) The crowded scenario to be explored. (b) The global planner provides a touring order for frontiers (blue points and lines) and determines a global goal (the blue square) for next-step action. (c) The local planner in [3] selects an optimal viewpoint (orange point) as the robot's intermediate goal. (d) Our proposed local planner notices the necessity of tight coupling between exploration and navigation controller in dense crowds, and selects a viewpoint with fewer nearby dynamic obstacles according to the guidance provided by an RL-based Critic network.

2) Critic Network-Based Evaluation: To drive the robot to a viewpoint with fewer pedestrians, we evaluate the reachability of the region around the robot using a Critic network provided by the Actor-Critic RL framework which is to be introduced in Section III-C. The Actor network serves as a navigation controller and the Critic network guides the gradient update of the Actor. During the RL training process, the agent is rewarded when reaching its goal and the Critic learns to score higher in those situations where the robot is easier to reach the target point [26]. In other words, the Critic network incorporates inputs of the robot position, goal position, and the laser scan data to build an implicit model that estimates the chances of reaching a given destination through the surrounding dynamic obstacles. For instance, along the direction without crowds or consistent with the flow, the Critic value would be higher; while along the direction against the crowd flow, the Critic value would be lower. Thus, the Critic network's output can guide the robot toward the most accessible viewpoint $\mathbf{p}_{\text{local}}$ that maximizes:

$$V_{\text{rl}}(\mathbf{p}_{\text{local}}) = V_{\pi}(\mathbf{s}_{\text{scan}}, \mathbf{s}_{\text{vel}}, \mathbf{p}_{\text{local}}), \tag{1}$$

where V_{π} is the value function of the Critic network.

Next, the candidate viewpoint $\mathbf{g}_{\text{explore}}$ with the highest score is selected as the next destination:

$$\mathbf{g}_{\text{explore}} = \arg\max_{\mathbf{p}_{\text{local}}^{1:k}} \left(\omega_{\text{nbv}} \cdot V_{\text{nbv}}(\mathbf{p}_{\text{local}}) + \omega_{\text{rl}} \cdot V_{\text{rl}}(\mathbf{p}_{\text{local}})\right),$$

where ω_{nbv} and ω_{rl} are hyperparameters weighting the importance of information gain and compliance with the crowds. One example of the selected viewpoint is shown in Fig. 2 as the red point. Finally, the selected viewpoint $\mathbf{g}_{\text{explore}}$ and the corresponding path connecting the robot's position $\mathbf{p}$ and $\mathbf{g}_{\text{local}}$ are passed to RL-based navigation controller for steering command generation.

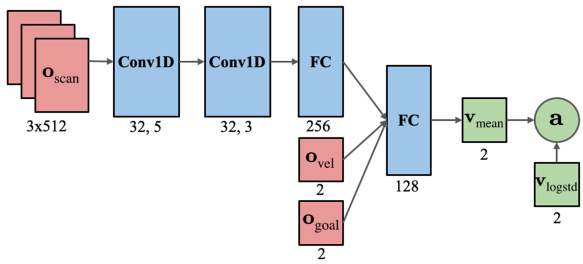

Fig. 3. The architecture of the RL-based navigation controller.

C. RL-Based Navigation Controller

We incorporate an RL-based reactive collision avoidance algorithm as the navigation controller. Compared to traditional navigation method, this approach can significantly improve the robot's collision avoidance performance in a dense crowd, thus ensuring safe navigation in environments with pedestrians.

Similar to our previous work [3] [26], we use an Actor-Critic-based PPO algorithm [18] to train the collision avoidance policy. As shown in Fig. 3, the Actor and Critic have the same network architecture as well as inputs, which contain three latest consecutive laser scans $\mathbf{o}_{scan}$, the robot's current velocity $\mathbf{o}_{vel}$ and the relative goal position $\mathbf{o}_{goal}$. The Actor network, which outputs action $\mathbf{a} = [v, \omega]$ sampled from a Gaussian distribution constructed by the mean $\mathbf{v}_{mean}$ with a log standard deviation vector $\mathbf{v}_{logstd}$, are serves as the controller. And the Critic network, which outputs a scalar value representing V_{π}, is used to guide the gradient update of the Actor during training.

The RL-based collision avoidance was trained using Actor-Critic based PPO algorithm [18] on a set of scenarios with simple static obstacles and small crowds with randomized initial goal positions for each agent with 2 stages [17]. Each episode is terminated when the robot reach its goal or when the maximum time step is reached. The reward function of the RL network is designed to encourage the robot to avoid collisions and reach the destination as fast as possible, including three terms: $r = r_a + r_c + r_\omega$ where r_a scores when the robot arrive its goad, r_c penalizes collisions, and r_ω penalizes excessive rotational velocity. For more details about the RL-based navigation controller, please refer to [17] and Fig. 4. For network deployment, after obtaining the path $\mathcal{P}$ from the local exploration planner, the goal location s_{goal} on that path is converted to o_{goal} as the input to the RL-based controller. The Actor network in the RL-based controller then outputs an appropriate steering command to move the robot through the crowd safely and efficiently toward s_{goal}.

In our proposed autonomous exploration system, the collision avoidance policy is tightly coupled with exploration optimization, by integrating the trained collision avoidance network with the hierarchical planners as described above. In particular, the

Authorized licensed use limited to: National Tsing Hua Univ.. Downloaded on August 28,2025 at 03:40:41 UTC from IEEE Xplore. Restrictions apply.

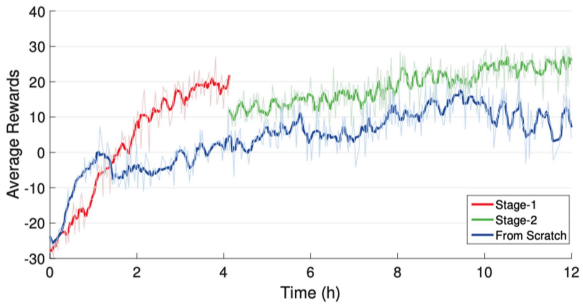

Fig. 4. Average rewards shown during the training process. Stage-1: crowded scenarios without static obstacles; Stage-2: crowded scenarios with static obstacles. From scratch: with both pedestrians and static obstacles.

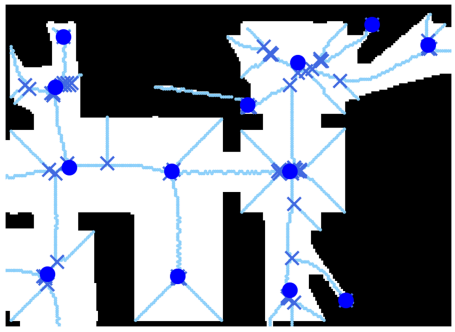

Fig. 5. Map-based recovery point extraction: the light blue line is the skeletonization result within the explored area, $\times$ marks the junction points in the skeletonization result, and the blue dots are the map-based recovery after clustering.

Actor network acts as the controller, while the Critic network not only directs the gradient update of the actor module by minimizing the agent loss but also attracts the robot to the most accessible direction in the current distribution of crowds. Compared to [3], our proposed exploration system takes full advantage of this Actor-Critic framework, i.e., in addition to using the Actor network as the robot's navigation controller, latent information from the Critic network is also leveraged to assess the dynamic states around the robot to prevent getting stuck in dense crowds.

D. Recovery Planner

To improve the robustness of the exploration system in dense crowds, we also add a recovery planner to help the robot recover from the potential crash of the SLAM module, which may happen when dense crowds occlude all informative landmarks important for localization, or when the robot is hijacked e.g., after its collision with pedestrians. In particular, in our exploration framework, the robot has two operation modes: the first mode is the normal exploration mode, in which the robot explores unknown regions using the hierarchical planners described above; the second mode is the recovery mode, in which the robot automatically navigates to a region with rich features in the explored map for active re-localization. These two modes are switched according to a simple and effective trigger condition as explained below.

1) Trigger Conditions: For robust autonomous exploration, the robot should switch to active recovery mode when the performance of the SLAM module degrades. Thus, we design the trigger condition as a metric measuring the quality of the SLAM output in terms of the covariance of the localization result. In our implementation, we choose the SLAM toolbox [27] as the SLAM algorithm. It provides a metric V_{slam} which evaluates the quality of the SLAM algorithm via a set of heuristics, including the amount of area overlap, the ratio of overlap to laser measurements, the decay of constraints for candidate laser scans, and the final score of recursive updates. When V_{slam} is lower than a given threshold, the robot will automatically switch to the recovery mode. As shown in Fig. 1, after entering the recovery mode, the robot will move towards a selected recovery point. After the recovery procedure has completed, i.e., when the robot's SLAM quality metric V_{slam} is higher than a given threshold, the robot will return to the normal exploration mode. The recovery procedure consists of three steps: recovery point extraction, recovery point selection, and navigation toward the recovery point.

2) Recovery Point Extraction: We compute the set of recovery point candidates by taking into account both the richness of map features and information embedded in historical trajectories. On the one hand, recovery points should be feature-rich locations on the map so that the robot can achieve active re-localization. For example, in a typical indoor environment, areas with corner point features are beneficial for scan matching, while long corridors are not conducive for scan matching of the SLAM module. On the other hand, we find that locations with high SLAM accuracy in the robot's historical trajectory is also beneficial for robot re-localization. These two types of recovery points are extracted as follows:

i) *Map-based recovery point candidates:* The goal of map-based recovery point extraction is to capture locations with features beneficial for SLAM. For structured indoor environments, these areas generally have corner features while avoiding long corridors. Thus, as shown in Fig. 5, we first skeletonize the explored binarized map, then extract junction points from the resulting skeleton, and finally use DBSCAN [28] algorithm to cluster these junction points to get the map-based recovery points $\mathbf{p}_{\text{map}}^{1:m}$ (blue dots in Fig. 5).

ii) *Trajectory-based recovery point candidates*: Besides utilizing features from the explored map, we also use historical trajectory to determine the recovery point candidates. We believe that areas around the robot's historical trajectory with decreasing entropy or significantly higher V_{slam} values are favorable to the SLAM module and can be used as candidate recovery points. In particular, we first compute the quality of the SLAM output, i.e., V_{slam} for each pose on the historical trajectory. Next, we calculate the difference of this metric between adjacent poses t and $t+1$: $\mathrm{d}V_{\text{slam}} = V_{\text{slam}}^{t+1} - V_{\text{slam}}^{t}$, which measures the impact of the environment on SLAM at pose $t+1$. The higher $\mathrm{d}V_{\text{slam}}$, the more helpful the measurement at pose $t+1$

Authorized licensed use limited to: National Tsing Hua Univ.. Downloaded on August 28,2025 at 03:40:41 UTC from IEEE Xplore. Restrictions apply.

is for SLAM estimation. Finally, all poses with $\mathrm{d}V_{\text{slam}} > T_{\text{slam}}$ will be marked as trajectory-based recovery point candidates $\mathbf{p}_{\text{traj}}^{1:n}$, where T_{slam} is a given threshold.

These two sets of recovery point candidates $\mathbf{p}_{\text{traj}}^{0:n}$ and $\mathbf{p}_{\text{map}}^{0:m}$ will be combined together to form

$$\mathbf{p}_{\text{recover}}^{1:j} = \left[\mathbf{p}_{\text{map}}^{1:m}, \mathbf{p}_{\text{traj}}^{1:n}\right],$$

from which the optimal recovery point will be selected.

3) Recovery Point Selection: We consider two factors when selecting the most suitable recovery point, including the distance of each candidate point to the robot and whether the point can be easily reached through the crowds. In particular, the optimal recovery point $\mathbf{g}_{\text{recover}}$ is computed as:

$$\mathbf{g}_{\text{recover}} = \underset{\mathbf{p}_{\text{recover}}^{1:j}}{\arg\max} \left(\omega_{\text{d}} \cdot V_{\text{d}}(\mathbf{p}_{\text{recover}}) + \omega_{\text{rl}} \cdot V_{\text{rl}}(\mathbf{p}_{\text{recover}})\right),$$

where ω_{d} and ω_{rl} are hyperparameters, V_{d} is the evaluation function measuring the normalized negative distance between the robot position $\mathbf{p}$ and a recovery point candidate $\mathbf{p}_{\text{recover}}$:

$$V_{\text{d}}(\mathbf{p}_{\text{recover}}) = -\frac{\|\mathbf{p} - \mathbf{p}_{\text{recover}}\|_2}{\sum_{i=1}^{j} \|\mathbf{p} - \mathbf{p}_{\text{recover}}^{i}\|_2},$$

and V_{rl} is the evaluation function provided by the Critic network similar to (1):

$$V_{\text{rl}}(\mathbf{p}_{\text{recover}}) = V_{\pi}\left(\mathbf{s}_{\text{scan}}, \mathbf{s}_{\text{vel}}, \mathbf{p}_{\text{recover}}\right).$$

In this way, the optimal recovery point not only provides rich geometric information for re-localization but also can be easily approached by the robot through dense crowds.

IV. Experiments and Results

We now quantitatively evaluate the performance of autonomous exploration in simulated scenes with crowds.

A. Experiment Setup

1) Simulation Setting: We use Gazebo [29] to simulate the robot and the crowded environments in ROS. A Turtlebot2 equipped with an RPLIDAR-A2 Lidar is used as the mobile robot platform. And we set the Lidar's scanning angle to 360°, scanning range to 6 m, and angular resolution to 1°.

For static scene layouts, we design eight benchmarks with different structures and features as shown in Fig. 6, including multiple offices and apartment-like scenarios consisting of multiple rooms and corridors, a topologically one-way environment, a large space with a layout similar to an airport or shopping mall, and a tunnel-like environment. Maps 1–4 are selected from houseexpo [30], a dataset of typical indoor environments such as offices and apartments. Maps 5–8 are manually designed for simulating a wide variety of scenarios.

For crowd simulation, we use Menge [31] to simulate dynamic pedestrians in static scenes. In each scene, three random crowd motion sequences were generated by randomly selecting the initial and target locations of each agent. This allows us to evaluate the robot's behavior in different types of pedestrian crowds.

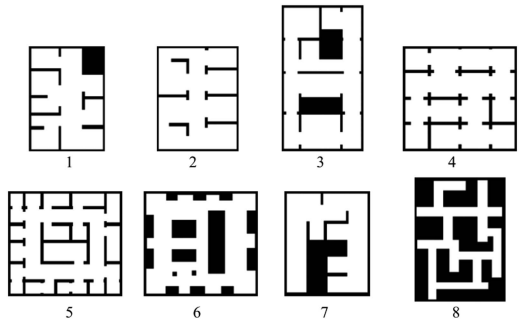

Fig. 6. The floor-plan samples for evaluation.

TABLE I
Methods for Comparison

Exploration Planner	Navigation Planner	Recovery	Name
Nearest	DWA	×	Baseline
TSP + NBV	DWA	×	HEC-MV [3]
TSP + NBV	RL	×	HEC [3]
TSP + NBV + Critic	RL	×	T-HEC
TSP + NBV + Critic	RL	√	RT-HEC

For the SLAM algorithm, we choose the 2D-Lidar-based SLAM toolbox [27]. As we focus on analyzing the impact of planning and control of the mobile robot rather than the SLAM algorithm, we set the resolution of the generated occupancy grid map to 0.1 m to simplify our system. Since this resolution is smaller than the width of a human leg, the SLAM algorithm can correctly distinguish static obstacles from dynamic obstacles in our benchmarks.

2) Methods for Comparison: We perform a comprehensive comparison of five different approaches in all eight test scenarios, as summarized in Table I. The *baseline* approach combines ROS's frontier-based exploration method and *move_base* navigation, where the *move_base* node uses a dynamic windowing approach (DWA) for local planning and Dijkstra's algorithm for global planning. The second approach is the *hierarchical exploration method in crowds with move_base* (HEC-MV) from our previous work [3], which has three components – TSP-based global planner, an NBV-based local planner, and a DWA navigation controller from *move_base*. The third method *hierarchical exploration method in crowds* (HEC) is also proposed in our previous work [3], where its navigation module is replaced by a DRL network rather than *move_base* in HEC-MV. In HEC, the navigation planner is loosely-coupled with the exploration planner. The fourth and fifth methods are proposed in this paper. One is the *tightly-coupled hierarchical exploration in the crowd* (T-HEC), where the local planner is guided by the Critic network from the RL-based collision avoidance. The other is its variant, *recovery-based tightly-coupled hierarchical exploration in the crowd* (RT-HEC), which further adds a recovery planner coupled with RL-based navigation controller.

3) Evaluation Metrics: All exploration algorithms to be compared are evaluated using the following metrics:

Authorized licensed use limited to: National Tsing Hua Univ.. Downloaded on August 28,2025 at 03:40:41 UTC from IEEE Xplore. Restrictions apply.

TABLE II
EVALUATION RESULTS: MEAN (STANDARD DEVIATION)

Method	Efficiency		Safety				SLAM Quality					
	Path Length		Collision		Success Rate		Translation Error		Rotation Error		Mapping Error	
Baseline	151.32	(51.37)	7.12	(1.60)	0.48	(0.10)	1.83	(0.17)	1.04	(0.04)	7.31	(1.63)
HEC-MV	**140.67**	(53.84)	7.15	(1.53)	0.48	(0.12)	1.93	(0.21)	1.12	(0.03)	6.85	(1.34)
HEC	142.83	(50.26)	5.31	(1.53)	0.83	(0.15)	1.48	(0.22)	**0.92**	(0.03)	4.36	(1.24)
T-HEC	143.26	(51.08)	**3.98**	(1.46)	0.89	(0.16)	1.45	(0.20)	0.94	(0.04)	4.38	(1.12)
RT-HEC	149.45	(49.95)	4.21	(1.42)	**0.92**	(0.12)	**1.37**	(0.20)	**0.92**	(0.02)	**3.99**	(1.25)

i) *Safety:* For each algorithm, we attempt T times until completing 3 exploration runs for a given benchmark, and the success rate is defined as $3/T$, which measures the exploration robustness. For each completed benchmark, we record the number of collisions. We define "collision" happens when there exists a pose where the laser scan measurement is shorter than the robot radius 0.5 m.
ii) *Efficiency:* We use the path length that the robot has traversed when finishing the scene coverage to evaluate the exploration efficiency.
iii) *SLAM quality:* Both localization errors and mapping errors are evaluated for each completed exploration task. We calculate the root mean square error (RMSE) of the robot's pose sequences estimated by the SLAM with respect to the ground-truth trajectory of the simulator to evaluate the localization accuracy. For mapping error, an environment without pedestrians was built and the robot is manually controlled to estimate a precise map with the same parameters, which is served as the ground-truth map. We use a 2D SLAM map evaluation method in [32] to compute a normalized distance between two occupancy grid maps.

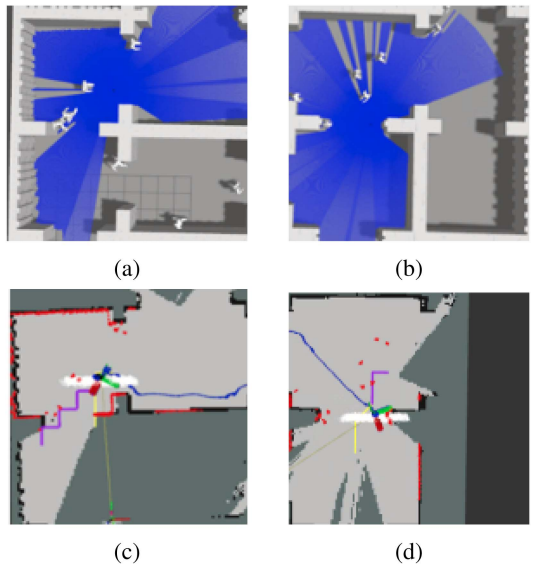

Fig. 7. Demonstration of the Critic-guided local planner. (a) (b) Simulation environments; (c) (d): Corresponding local planner results. Purple: HEC (decoupled local planner); yellow: T-HEC (tightly-coupled planner). Thanks to the Critic network, the robot selects the goal position with fewer pedestrians.

B. Results and Analysis

The comparison results are summarized in Table II, where we repeat 3 times with randomness for each benchmark and then report the mean and standard deviation of the performance estimated over all benchmarks. We can observe that by combining the tightly-coupled exploration and recovery planner with the navigation controller, our proposed autonomous exploration framework RT-HEC has great advantages over other algorithms in terms of making a good balance among exploration efficiency, navigation safety, and SLAM quality.

First, HEC-MV, HEC, T-HEC and RT-HEC all compute exploration paths shorter than that of the baseline. HEC-MV has the shortest exploration path and has lower success rate comparing to HEC, verifying the benefit of hierarchical exploration and RL-based navigation as explained in our previous work [3].

But in dense crowds, following the shortest path can be sub-optimal, because the robot may run into a dense crowd and get stuck. T-HEC and RT-HEC consider additional crowd-related factors, such as avoiding bad movement against the flow and effectively recovering from localization failures. As a result, T-HEC and RT-HEC usually have longer exploration path (due to crowd-avoidance detour as shown in Fig. 2(d) or additional recovery actions as shown in Fig. 8). Fortunately, such overhead is minor, because HEC, T-HEC and RT-HEC all select local planner's viewpoint within a local area M_{local} around the robot, resulting in bounded difference in path length. Second, T-HEC and RT-HEC significantly outperform baseline, HEC-MV and HEC in terms of the exploration safety, because our proposed tightly-coupled local planner allows the robot to avoid crowds in the reversed direction, as shown in Fig. 7. We observe that T-HEC has the lowest number of collisions during exploration, which is much lower than that of baseline, HEC-MV and HEC. RT-HEC needs to take additional movements/actions for recovery, resulting in its higher number of collisions than T-HEC. But such additional collision risk is well deserved, as supported by the RT-HEC's highest success rate of the exploration task. This is because RT-HEC can survive from serious occlusion in the dense crowds while T-HEC cannot.

Last but not least, the quality of maps generated by the SLAM module determines whether the autonomous exploration system actually works in dynamic scenarios. We compare the localization and mapping accuracy provided by all four approaches. We can observe that HEC, T-HEC and RT-HEC have much lower localization error in both translation and rotation than that of

Authorized licensed use limited to: National Tsing Hua Univ.. Downloaded on August 28,2025 at 03:40:41 UTC from IEEE Xplore. Restrictions apply.

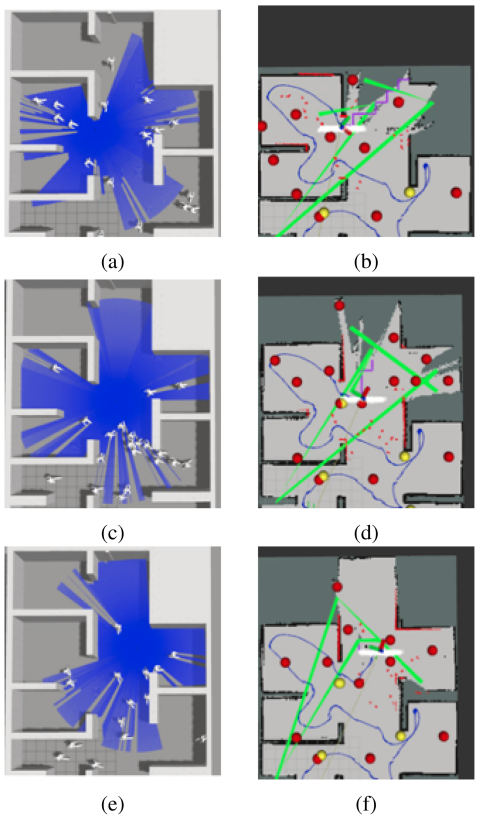

Fig. 8. Switchable exploration strategy. (a), (c), (e) Simulation environments; (b), (d), (f): the corresponding SLAM results. From top to bottom: switching exploration mode to recovery mode and heading toward the recovery point; reaching the recovery point; and switching back to exploration mode.

the baseline and HEC-MV, which demonstrates the importance of minimizing the interruption between the robot and crowds. RT-HEC's mapping error is also significantly smaller than that of HEC and T-HEC, because the recovery mode can search for useful geometric features in the environment to minimize the potential mapping mistakes.

V. Conclusion

In this letter, we present an autonomous exploration system with two modes: exploration mode and recovery mode. In both modes, the exploration or recovery planner is tightly coupled with a reactive navigation controller that actively avoids collisions with dense crowds. Our approach has been demonstrated to be able to provide a good trade-off among exploration efficiency, navigation safety and SLAM quality, by minimizing the interruptions or collisions between the robot and the pedestrian crowds in indoor environments.

References

[1] D. Mammolo, "Active slam in crowded environments," Master's Thesis, Auton. Syst. Lab, ETH Zurich, Zürich, Switzerland, 2019.
[2] M. P. Zapf, M. Kawanabe, and L. Y. M. Saiki, "Pedestrian density prediction for efficient mobile robot exploration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 4615–4622.
[3] Z. Zheng, C. Cao, and J. Pan, "A hierarchical approach for mobile robot exploration in pedestrian crowd," *IEEE Robot. Automat. Lett.*, vol. 7, no. 1, pp. 175–182, Jan. 2022.
[4] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Automat.*, 1997, pp. 146–151.
[5] Y. Mei, Y.-H. Lu, C. G. Lee, and Y. C. Hu, "Energy-efficient mobile robot exploration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2006, pp. 505–511.
[6] S. Oßwald, M. Bennewitz, W. Burgard, and C. Stachniss, "Speeding-up robot exploration by exploiting background information," *IEEE Robot. Automat. Lett.*, vol. 1, no. 2, pp. 716–723, Jul. 2016.
[7] C. Cao, H. C. Hongbiao Zhao, and J. Zhang, "Exploring large and complex environments fast and efficiently," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 7781–7787.
[8] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 610–617, Apr. 2019.
[9] V. Mnih et al., "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.
[10] D. Zhu, T. Li, D. Ho, C. Wang, and M. Q.-H. Meng, "Deep reinforcement learning supervised autonomous exploration in office environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 7548–7555.
[11] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *Proc. IEEE Robot. Automat. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997.
[12] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, 1995, Art. no. 4282.
[13] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1688–1694.
[14] M. Sebastian, S. B. Banisetty, and D. Feil-Seifer, "Socially-aware navigation planner using models of human-human interaction," in *Proc. IEEE Int. Symp. Robot Hum. Interactive Commun.*, 2017, pp. 405–410.
[15] A. Pierson, W. Schwarting, S. Karaman, and D. Rus, "Navigating congested environments with risk level sets," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 5712–5719.
[16] H. Guo et al., "Safe path planning with gaussian process regulated risk map," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 2044–2051.
[17] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *Int. J. Robot. Res.*, vol. 39, no. 7, pp. 856–892, 2020.
[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[19] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 6015–6022.
[20] L. Liu, D. Dugas, G. Cesari, R. Siegwart, and R. Dubé, "Robot navigation in crowded environments using deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5671–5677.
[21] Y. Chen, C. Liu, B. E. Shi, and M. Liu, "Robot navigation in crowds by graph convolutional networks with attention learned from human gaze," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2754–2761, 2020.
[22] G. Huang, A. Rad, and Y. Wong, "Online slam in dynamic environments," in *Proc. Int. Conf. Adv. Robot.*, 2005, pp. 262–267.
[23] M. S. Bahraini, M. Bozorg, and A. B. Rad, "Slam in dynamic environments via ML-RANSAC," *Mechatronics*, vol. 49, pp. 105–118, 2018.
[24] L. Perron and V. Furnon, "Or-tools," Google. [Online]. Available: https://developers.google.com/optimization/
[25] S. Thrun, W. Burgard, and D. Fox, "Exploration," in *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005, ch. 14, pp. 571–585.
[26] T. Fan et al., "Getting robots unfrozen and unlost in dense pedestrian crowds," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1178–1185, 2019.
[27] S. Macenski and I. Jambrecic, "Slam toolbox: Slam for the dynamic world," *J. Open Source Softw.*, vol. 6, no. 61, 2021, Art. no. 2783.
[28] M. Ester et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," *kdd*, vol. 96, no. 34, 1996, pp. 226–231.
[29] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 3, 2004, pp. 2149–2154.
[30] T. Li et al., "HouseExpo: A large-scale 2D indoor layout dataset for learning-based algorithms on mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5839–5846.
[31] S. Curtis, A. Best, and D. Manocha, "Menge: A modular framework for simulating crowd movement," *Collective Dyn.*, vol. 1, pp. 1–40, 2016.
[32] J. M. Santos, D. Portugal, and R. P. Rocha, "An evaluation of 2D slam techniques available in robot operating system," in *Proc. IEEE Int. Symp. Safety, Secur., Rescue Robot.*, 2013, pp. 1–6.

Authorized licensed use limited to: National Tsing Hua Univ.. Downloaded on August 28,2025 at 03:40:41 UTC from IEEE Xplore. Restrictions apply.