2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Gas-Source Efficiently Active Searching in Unfamiliar Environments

Yu Zhai and Yanzi Miao

***Abstract*— Searching Gas Source actively and efficiently in unknown hazard environments is an important but challenging issue. Using mobile robots to autonomously search and navigate to gas source location provides a promising way. Existing methods are mostly based on the modularization framework which investigates the gas-source search and robot navigation tasks independently, leading to a decoupled approach that results in higher collision risks and lower navigation efficiency. Moreover, existing robot navigation techniques grapple with the intricacies of navigating through unknown environments. To tackle these complexities, we introduce an integrated framework that merges gas source localization with robot navigation. This unified structure, underpinned by an end-to-end learning approach, resolves the inherent conflicts between gas exploration and collision avoidance. Our approach aggregates the local observations (raw 3D-LiDAR data) and the expert guidance information (gas distribution), and directly generates navigation actions by implementing the reinforcement learning with a novel reward function based on region dynamic guidances, thus effectively addressing the challenges of active gas source searching in unknown environments. Simulation results underscore the adaptability of our method to diverse unknown environments, along with its superior gas source searching capabilities compared to conventional approaches. Finally, we conduct real-world experiments to demonstrate our feasibility.**

I. Introduction

Mobile robots [1] [2] are one of the most widely used groups of intelligent robots and have been extensively used for disaster relief operations [3]–[5]. Substituting humans with mobile robots in search and rescue operations within hazardous areas has arisen as a superior and safer alternative. Robots based solutions have been widely applied in responses to high radiation sources in nuclear power plants [6], gas pipeline leaks in industrial production [7], and post-disaster assistance in catastrophic scenarios. However, the gas diffusion is influenced by the surrounding environment, and there exists a paucity of a prior knowledge regarding the hazardous environment in the event of an abrupt gas leakage incident. Autonomous navigation for active searching of gas sources is still a challenging task.

Over the past decade, a significant number of researchers have devoted their efforts to actively searching gas sources. Their studies have demonstrated that robots can compute a more precise estimation of the gas source's searching by analyzing the distribution of gas concentrations in the surrounding environment [7], [8]. This iterative process is employed to progressively refine the estimation of the gas-source's location. These studies frequently make the assumption that robots are equipped with navigation modules boasting flawless obstacle avoidance capabilities within simulated environments. Consequently, they do not consider potential conflicts between these modules, and ignore the limitations of existing navigation modules concerning their navigation abilities.

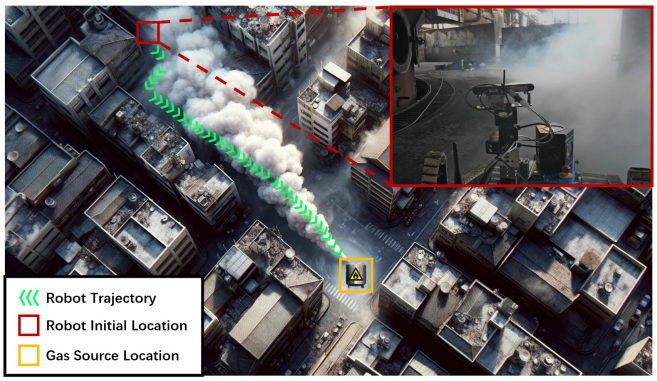

Fig. 1. The robot is tasked with locating gas diffusion sources relying on local odor sensors. Simultaneously, it is required to navigate safely in unfamiliar environments, avoiding collisions based on local perception information.

Traditional autonomous navigation methods [9]–[11] usually require robots to pre-map the navigation environment globally. Learning-based methods [12] employ neural networks to approximate navigation strategies using local map information. These methods handle scenarios where there is no prior environmental knowledge, especially in sudden or unexpected events. Recently, some researchers have tackled the map-dependency issue by employing reinforcement learning (RL) techniques [13] to construct end-to-end policy networks that map raw sensor data to navigation actions directly. However, visual information exhibits poor transferability across different scenes, while 3D LiDAR data is high-dimensional, unstructured, and unordered, resulting in low efficiency. Additionally, the high dimensionality of sensor information often drowns out goal guidance information.

To address the above challenges, We propose an end-to-end navigation framework for active gas-source search. Our main contributions include: 1) We propose a reinforcement-learning based navigation system capable of performing complex gas-source search tasks in unfamiliar environments. 2) A generalized expert guidance information presentation and aggregation method is presented which can be easily scaled to other navigation tasks, and the raw 3D-LiDAR data is directly considered as system input to describe local observations. Then a novel reward function is designed to

This work was supported by the Fundamental Research Funds for the Central Universities (2022XSCX03).

Y. Zhai and Y. Miao are with the School of Information and Control Engineering, China University of Mining and Technology, China. Corresponding author: Y. Miao (myz@cumt.edu.cn).

navigate the robot to the gas source while avoiding any collisions in unknown environments. 3) We demonstrate that our model trained in a simulated environment can be directly transferred to real-world scenarios.

II. Related Work

A. Gas Source Searching by Mobile Robots

Previous studies usually divide this task into two separate modules: gas source localization and robot navigation. Researchers pay more attention to gas source localization which aims to locate the source of a gas leak based on the measurements of gas concentration. Various gas source localization techniques have been proposed, including Silkworm moth algorithm [14], E.coli algorithm [15], and Braitenberg algorithm [16]. In addition to the methods mentioned above, some bio-inspired algorithms have also been applied to gas source localization. Fruit fly algorithm [17] was proposed for gas source localization. This algorithm and its related derivative algorithms have achieved excellent results in gas source localization. However, robot navigation is not a major concern [7] [8], assuming the use of a navigation module with perfect capabilities or a commonly employed navigation method widely used in the field of robotics as a standard solution, for example A* algorithm, dynamic window algorithm [9] and so on.

B. Automatic Navigation for Mobile Robots

Traditional methods include gragh-based approaches (for example dynamic windows approach [9] and A*) and sampling-based approaches (for example RRT and its variant), in which require a long time to build maps for planning a path to avoid conflicts with obstacles. Imitation learning-based navigation methods [12] streamline the intricate computations of traditional methodologies through neural network approximations. Nonetheless, their navigation capabilities are confined by the expert algorithms from which they derive their learning. RL based approaches [13], [18], [19] can guide robots to learn unlimited strategy by exploring environment. As usually the robot can only receive the reward when it reaches the endpoint or collides, this sparse information leads to suboptimal or even ineffective navigation strategies. Some researchers have applied RL techniques to construct end-to-end policy networks that directly map raw data obtained from cameras to navigation actions [20]–[22]. Nevertheless, cameras are highly sensitive to variations in lighting, that renders them less adaptable across various scenarios. 3D LiDAR is another widely used sensor in the field of robotics, which has strong transferability. However, 3D point cloud is unstructured, high-dimensional and unordered. It leads to huge network structure for learning environment feature [23]–[25]. Furthermore, the low-dimensional goal information may be overwhelmed by the multi-dimensional observation data. To address this challenge, multi-layer perceptrons (MLPs) have been employed to augment the dimensionality of low-dimensional information [26]. Nevertheless, this approach losses the spatial relationship between the goal guidance and the local environment structure information, thus leading to high collision risks.

III. Problem Formulation

Environment: We consider gas diffusion occurs in a three-dimensional space $D \in \mathbb{R}^3$, x, y and z are defined as the position in space $[x, y, z] \in D$, t represents the time. $C(x, y, z, t)$ denotes the gas concentration at position x, y, z at time t, and $C_s(x_s, y_s, z_s, t)$ is defined as the gas concentration at the gas source location $p_s = [x_s, y_s, z_s]$ at time t, then we have

$$C_s(x_s, y_s, z_s, t) \geq C(x, y, z, t) > 0, \forall [x, y, z] \in D, \forall t \geq 0$$

Assumption: Assuming the robot can quickly detect the gas concentration $C(t)$ at its current position $p(t) = [x_t, y_t, z_t]$ and also the concentrations $C_e(t) = [C_{e1}(t), C_{e2}(t), \ldots, C_{eN}(t)]$ of N surrounding locations within a distance of L_c from its current position. In addition, although we consider the 3D environment space, however, we assume that the robot moves in a 2D plane and the gas concentration does not change in the z-axis direction.

Action Space: We design four discrete actions as the mobile robot's action space $a \in [a_f, a_l, a_r, a_s]$, in which a_f, a_l, a_r and a_s are moving forward, turning left, turning right and stopping. Each action is a two-dimensional vector composed of the left wheel's speed and the right wheel's speed. In detail, $a_f = [v_{lf}, v_{rf}]$, $a_l = [v_{ll}, v_{rl}]$, $a_r = [v_{lr}, v_{rr}]$ and $a_s = [0, 0]$.

Objective: Inputting the local observations (raw 3D LiDAR data) and the gas concentration values C_m and C_e, our objective is to output a series of actions (from the action space a) to navigate the robot to the source location $[x_s, y_s, z_s]$, in the meantime, avoiding any collisions with the obstacles in the environment D.

IV. Main Approach

In this section, we provide an integrated framework as shown in figure 2, which combines a gas-source localization algorithm with an RL local navigation model. Gas-source localization algorithm provides the expert guidance based on the distribution and trend of gas concentration in the current location. The RL local navigation model considers the expert guidance as well as the local observation information for making searching actions.

A. Goal Information Guidance from Expert Algorithm

The goal information guides the robot to move to the gas source location step by step, which can be generated by using any expert algorithms. In this paper, we utilize the Guided Fruit Fly Optimization Algorithm (GFOA) [27], which is a widely utilized approach for actively locating gas sources due to its quick response and robust local search capabilities. On the foundation of GFOA, we propose the reverse learning and Cauchy probability random learning to enhance the algorithm's capability [28] to address local minima issues. The algorithm computes the locally guided position $g(t) = [x_g, y_g, z_g]$ (x_g, y_g and z_g are the parameters

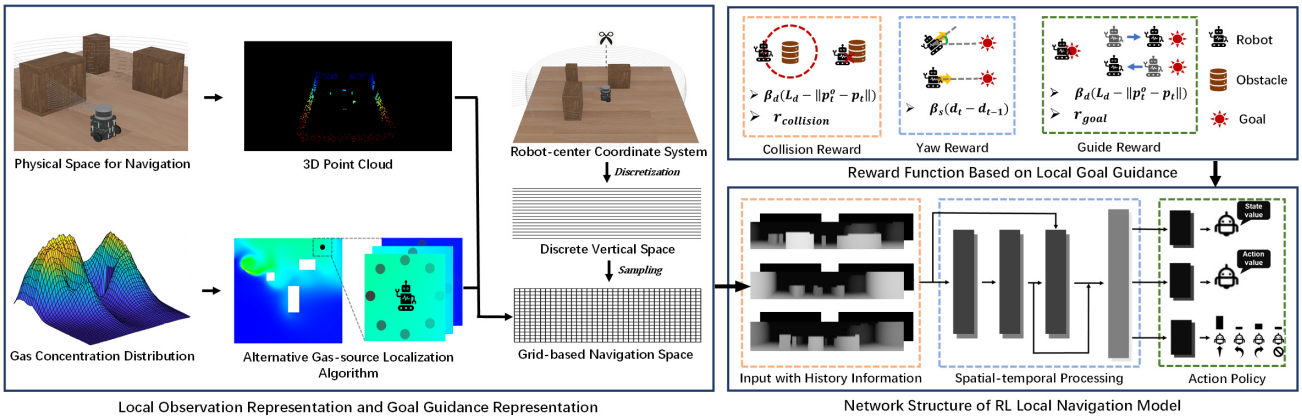

Fig. 2. The structure of our model, which combines a alternative gas-source localization algorithm, local observation representation and goal guidance representation approach and a local RL navigation model. A novel reward function based on local goal guidance is used to improve performance of RL local navigation model.

related time t, while we leave out the subscript t.) based on the gas information at the current location $p(t)$, where

$$g(t) = p(t) + e^{\omega} \cdot cauchy(0,1) \cdot (p_{ei}(t) - p(t)) \quad (1)$$

$$i = \arg\max\{C_{ei}(t) - C(t)\} \quad (2)$$

where $\omega = C(t)/(C(t)) + C(t-1))$ is a learning modulation coefficient guided by gas concentration. $cauchy(0,1)$ is a random number that follows the standard Cauchy distribution.

B. Local Observation Representation and Goal Guidance Representation

As shown in Algorithm 1, we first transform the raw 3D LiDAR data into a fixed-size virtual image E.

A group raw data of 3D LiDAR with N points in the space space D, each represented by Cartesian coordinates (x_i, y_i, z_i), where $i = 1, 2, \ldots, N$ (Point i and location p are not in the same coordinate system, we use the farthest point to complete the missing points). Firstly, we convert the Cartesian coordinates to polar coordinates. For each point i, we define yaw η_i and pitch θ_i, in detail:

$$\eta_i = \text{atan2}(y_i, x_i), \ \theta_i = \text{atan2}(z_i, x_i) \quad (3)$$

Here, $\text{atan2}(z, x)$ and $\text{atan2}(y, x)$ are the arctangent function that returns angles in the range $[-\pi, \pi]$. We map the polar coordinates (η_i, θ_i) to a fixed-size image, which has dimensions $M \times L$, where M represents the resolution of 3D LiDAR in the horizontal direction and L represents the number of lines in 3D LiDAR.

We assume the η_i spans from $\eta_{\min}$ to $\eta_{\max}$ and the θ_i spans from $\theta_{\min}$ to $\theta_{\max}$. We define the step sizes for the range and azimuth as Δr and $\Delta\theta$, respectively:

$$\Delta\eta = \frac{\eta_{\max} - \eta_{\min}}{M}, \ \Delta\theta = \frac{\theta_{\max} - \theta_{\min}}{L} \quad (4)$$

Then, for each point i, we compute its pixel coordinates (h, v) in the image E:

$$h = \left\lfloor \frac{\eta_i - \eta_{\min}}{\Delta\eta} \right\rfloor, \ v = \left\lfloor \frac{\theta_i - \theta_{\min}}{\Delta\theta} \right\rfloor \quad (5)$$

Algorithm 1 Environmental and Knowledge Representation

1: Initialize representation matrix E, K
2: **for** $i = 0$, N **do**
3: $E\ [h, v] = d_i$
4: **end for**
5: **for** $j = 0$, M **do**
6: **if** $max[0, h_c - H_d] <= j <= min[h_c + H_d, M]$ **then**
7: $K\ [j, :] = d_r$
8: **end if**
9: **end for**
10: **return** E, K

We use $d_i = \|x_i, y_i, z_i\|$ as the value of the pixel (h, v) in the image E.

To address the significant dimensional gap between the original expert knowledge and environmental representation data, we transform goal guidance information into image K with the same dimensions as image E. The expert knowledge is converted into normalized forms of relative distance d_r and relative angle d_s.

$$d_r = \|x_g - x_t, \ y_g - y_t, \ z_g - z_t\| \quad (6)$$

$$d_s = \text{atan2}(y_g - y_t, \ x_g - x_t) \quad (7)$$

The horizontal axis of goal guidance on a fixed image is

$$h_c = \frac{d_s - \eta_{\min}}{\Delta\eta} \quad (8)$$

We mark the area with a width of $2 \times H_d$ on the goal guide image. h_c is the horizontal axis of the regional center. The details are demonstrated in steps 5-9 of Algorithm 1.

C. Reward Function Based on Local Goal Guidance

Previous robot navigation methods based on RL are mostly designed with sparse rewards. It causes these methods require an extremely long time to explore space when the local goal is far away from initial position. The dense reward based on potential field method efficiently reduce the explore time. However, it inevitably leads the problem of local optima. To attack above problem, we propose a novel dense reward

function based on the local goal guidance. The robot can avoid potential collisions while it is guided to approach the local goal by dense reward. The detailed structure of the reward function is described as:

$$r_t = {}^{s}r_t + {}^{d}r_t + {}^{g}r_t + {}^{p}r \tag{9}$$

The robot get total reward r_t consisted by ${}^{s}r_t$, ${}^{d}r_t$ and ${}^{f}r_t$ at time t. ${}^{d}r_t$ is the collision reward which guide the robot to efficiently avoid obstacles.

$${}^{d}r_t = \begin{cases} \beta_d(L_d - \|p_t^o - p_t\|), & \text{if } \|p_t^o - p_t\| < L_d \\ r_{collision}, & \text{if } \|p_t^o - p_t\| < L_c \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where p_t^o, L_d and β_d are defined as the position of obstacles, the safe distance and the collision penalty factor (hyperparameters). The robot have more space to avoid obstacles, it gets benefit from the safety distance L_d, which does not strictly ban the robot from enter the area around obstacles. The yaw reward ${}^{s}r_t$ guide the robot to quickly comeback the correct direction after avoiding collision.

$${}^{s}r_t = \beta_s(d_t - d_{t-1}) \tag{11}$$

where d_t is the yaw angle between the current position of the robot and the position of the local goal at time t. β_s is a hyperparameter. The guide reward ${}^{g}r_t$ guides the robot to approach the local goal quickly by dense positive feedback.

$${}^{g}r_t = \begin{cases} r_{goal}, & \text{if } \|g - p_t\| < L_g \\ \beta_g(\|g - p_{t-1}\| - \|g - p_t\|), & \text{otherwise} \end{cases} \tag{12}$$

${}^{p}r$ is single step time penalty parameter.

D. Network Structure of RL Local Navigation Model

We design a RL local navigation model, which inputs the local observation representation information and the goal guidance representation information and outputs an action for searching gas-source. The network structure of RL local navigation model is shown in Fig. 2. First, we design the three-layers convolutional neural networks (CNN) with residual structure for extracting spatial features from the dual-channel representation image. Long short-term memory (LSTM) network is used to extract temporal features form the information of three continued time steps (the current time, the previous time and the time before the previous frame).SAC is a widely used RL method, which has higher training efficiency. We use the SAC algorithm for learning gas-source searching policy. Two group of different MLPs are used as two critic networks (Critic Q, Critic V). They evaluate the state value. Another three-layers MLP is used for building the actor network. The critic networks are only used in retraining to obtain a better performing actor network. They not be used in testing or specific applications. The specific network parameters and training details will be discussed in detail in the next section.

V. Implementation

A. RL Parameters

The parameters in our reward function $\mathcal{R}_t$ are set as $r_{collision} = -5, r_{goal} = 5, {}^{p}r = -0.01, \beta_s = 0.5, \beta_d = 0.35, \beta_t = 0.07, L_d = 0.6, L_c = 0.1$, $H_d = 10$ and $L_g = 0.2$. We set the discount factor as $\gamma = 0.99$ and $\alpha = 0.02$. In action spaces, we set $v_{lf} = 5, v_{rf} = 5, v_{ll} = 2.5, v_{rf} = 0, v_{lr} = 2.5, v_{rr} = 0, v_{ls} = 0 and v_{rs} = 0$. Hyperparameters of SAC $\alpha = 0.02$,$\gamma = 0.99$ and the size of replay buffer is 3000.

The detail of CNN structure is described in Table I, and the following LSTM layer has 1024 units with `ReLU` activation functions. The actor network and critic Q network have same structure, they have three FC layers with 512, 256 and 4 units. The activation functions are `ReLU` and `softmax` in last layer, while none activation function is set in output layer of the critic Q network. Three FC layers with 512 ,256 and 1 units are used in critic V network. Their activation functions are `ReLU`.

TABLE I
THE CNN STRUCTURE OF LOCAL RL NAVIGATION MODEL

	kernel size	stride	channel number
Cov 0	(3,3)	(3,3)	16
Cov 1	(3,3)	(3,3)	32
Cov 2	(16,9)	(3,9)	32

B. Training Parameters

We train our model with an inter i7 9700k CPU and one NVIDIA GTX 2080Ti GPU, and on the python3.8 with Pytorch 1.9 platform. The time of our model computing navigation action is about 2.7ms through CPU. We used accurate gas source location information as guidance to train the network for 1500 epochs. Subsequently, Our network is trained for another 500 epochs when local guidance information provided by the expert algorithm. The learning rate of actor network, v critic network and q critic network is 5×10^{-6}, 5×10^{-4} and 1×10^{-6}. The training optimizer is `Adam` and the batch size is 32.

C. Training Environments and Comparison Methods

We use Webots to create the simulation environment. Webots is a widely used robot simulation software. We uses two-wheeled differential robots (TurtleBot3Burge) with a 3D LiDAR (VLD-16) in the simulation. We set up the following scenes for training and testing:

- *Random Scene (S_r)*: We randomly place rectangular wooden boxes with side lengths of 0.2m x 0.2m x 0.3m, ranging from 40/80/120 in number, as obstacles in the scene with the size of 4m $\times$ 4m or 8m $\times$ 8m. The position of the obstacles changes in each round.
- *Dynamic Scene (S_d)*: We set 10 non-cooperative mobile objects in this scene with the size of 4m $\times$ 4m. Each mobile objects have different positions.
- *Complex Scene (S_c)*: 30 differently sized rectangular wooden boxes with different size are randomly placed in this scene with the size of 4m $\times$ 4m.

- *Mixed Scene (S_m)*: Obstacles consist of 30 differently sized and shaped wooden boxes, cardboard boxes, and oil drums. They are randomly placed in this scene with the size of 4m $\times$ 4m.

We considered the following methods for comparison and ablation study:

- *Method of Raw Data*: We do not use expert algorithm to compute Local goal guidance, while we take the average sampling of gas concentrations around the robot. The concentration of each sampling point is input into the network for computing searching action.
- *Method of Tradition*: The navigation module and expert algorithm module are two completely independent modules. The navigation module strictly follows the direction guidance given by the expert module to move forward. The robot only makes avoidance actions outside of the guidance given by the expert when it will lead to a collision.

VI. Simulation and Real-World Experiments

A. Navigation Ability in Unfamiliar Environments

We fristly demonstrate navigation ability of our approach in unfamiliar environmnets. Specifically,We train our model on the random scene S_r with a size of 4*4, 20 obstacles. And, we train the trained model on the dynamic scene d_y. At the beginning of each round of training, the positions of all obstacles (static and dynamic) are randomly placed. The initial position of the robot and the position of the gas source are also reset in each round.Then, We directly test the trained model on various unfamiliar scenes. The simulation results are shown in Table II. The results show that our model achieve 100% success navigation rate in trained static scene. Although the model is just trained in the scene with small size and low density static obstacles, it still maintains high success navigation rate and stable navigation efficiency (navigaton time) in more complex scene with unfamiliar map size and unfamiliar static obstacles density. Additionally, the results in the dynamic scene S_d demonstrate that our model can avoid dynamic obstacles with unknown moving track. In order to further demonstrate the navigation performance in the scene with unfamiliar environment structures, we test our model in the complex scene S_c and the mixed scene S_m. Our approach has the ability to handle unknown environmental structures and still remain high success navigation rate. Our model provides a guarantee for robots to efficiently complete the active gas source localization task.

B. Navigation Performance in Gas-Source Searching Task

We demonstrate capability of our model in gas-source searching. We randomly set an gas-source in the free space. We employ a Gaussian distribution to simulate the gas diffusion pattern of the gas-source. To compare our approach with the performance of traditionally modular combinations, we design an independent navigation module coupled with an expert module. The expert module generates navigation waypoints for the subsequent phase, followed by the navigation module guiding the robot collision-free towards these waypoints. we established a comparative method using raw sensor data to demonstrate the role of expert guidance and knowledge representation. A task is considered a failure if the robot fails to reach the destination within 200 steps.

The simulation results is shown in Table III, our approach shows distinct advantages over the independent modular combination method and method using raw data in the task of gas source localization. Moreover, as the complexity of the testing scenes gradually increases, our model shows the more obvious advantage. This outcome can be attributed to two factors. First, the expert algorithm does not consider environmental factors such as the distribution of static obstacles while calculating the next local goal guidance based on gas concentration distribution. Consequently, the robot have a higher risk to fall in local optima or cause collision. Our method effectively addresses this issue by the coupled structure can better balance the relationship between following the local guidance and collision avoidance. Second, our approach achieves rapid learning of navigation policy through skipping

TABLE II

Our Generalibility to Various Unfamiliar Scenes

Environment Style		SNR	CR	NT
S_r	$4 \times 4/10$	$100\% \pm 0\%$	$0\% \pm 0\%$	43.1 ± 2.0
	$4 \times 4/20$	$97.9\% \pm 0.7\%$	$2.1\% \pm 0.7\%$	47.3 ± 2.2
	$4 \times 4/30$	$96.5\% \pm 0.9\%$	$3.5\% \pm 0.9\%$	48.7 ± 1.2
	$8 \times 8/40$	$99.4\% \pm 0.2\%$	$0.6\% \pm 0.2\%$	94.3 ± 3.8
	$8 \times 8/80$	$95.4\% \pm 0.8\%$	$4.6\% \pm 0.8\%$	103.2 ± 4.6
	$8 \times 8/120$	$92.2\% \pm 1.2\%$	$7.8\% \pm 1.2\%$	110.8 ± 5.3
S_d	$4 \times 4/10(D)$	$96.4\% \pm 2.0\%$	$3.6\% \pm 2.0\%$	41.3 ± 5.8
S_c	$4 \times 4/30$	$95.5\% \pm 2.5\%$	$4.5\% \pm 2.5\%$	54.3 ± 4.1
S_m	$4 \times 4/30$	$94.6\% \pm 2.4\%$	$5.4\% \pm 2.4\%$	49.7 ± 7.6

We calculate the Successful Navigation Rate (SNR), Collision Rate (CR) and Navigation Time (NT) for evaluating the navigation performance.

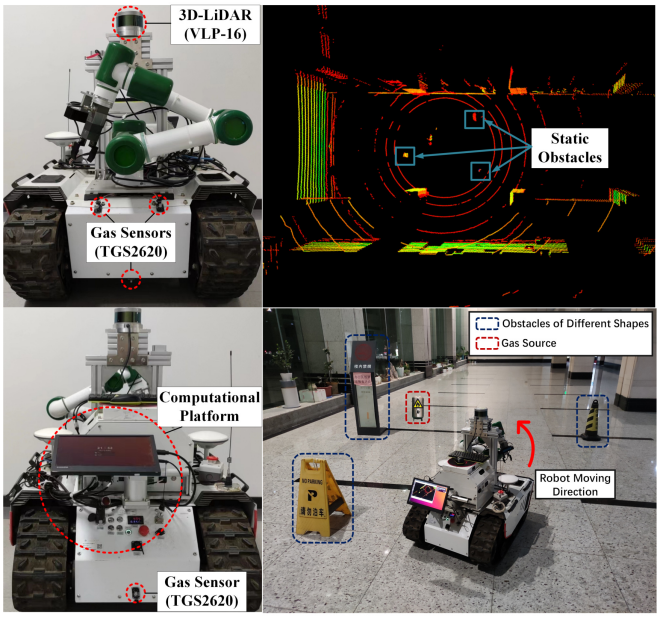

Fig. 3. The left pictures are robot platform used in our physical experiments. The right pictures are real experimental scenes and corresponding 3D point clouds.

TABLE III

THE EXPERIMENTAL RESULTS OF ACTIVE SEARCHING FOR GAS SOURCES

Method		Method of Raw Data			Method of Tradition			Ours		
Environment Style		SNR	CR	NT	SNR	CR	NT	SNR	CR	NT
S_r	$4 \times 4/10$	$89.4 \pm 3.1\%$	$5.8 \pm 1.4\%$	73.6 ± 6.2	$98.4 \pm 1.2\%$	$1.6 \pm 1.2\%$	61.4 ± 4.8	$100 \pm 0\%$	$0 \pm 0\%$	53.4 ± 2.4
	$4 \times 4/20$	$80.1 \pm 4.7\%$	$11.9 \pm 2.5\%$	75.2 ± 6.4	$90.6 \pm 2.8\%$	$10.4 \pm 2.8\%$	66.9 ± 3.7	$96.8 \pm 0.5\%$	$3.2 \pm 0.5\%$	60.3 ± 2.1
	$4 \times 4/30$	$75.8 \pm 6.2\%$	$19.8 \pm 3.6\%$	80.7 ± 8.8	$83.2 \pm 4.2\%$	$16.8 \pm 4.2\%$	69.2 ± 3.8	$93.8 \pm 0.9\%$	$6.2 \pm 0.9\%$	62.1 ± 2.8
	$8 \times 8/40$	$54.8 \pm 4.9\%$	$26.1 \pm 5.4\%$	151.3 ± 7.8	$96.6 \pm 2.4\%$	$3.4 \pm 2.4\%$	94.3 ± 3.8	$99.4 \pm 0.6\%$	$0.6 \pm 0.6\%$	102.1 ± 5.4
	$8 \times 8/80$	$47.1 \pm 5.6\%$	$32.6 \pm 6.3\%$	162.9 ± 8.5	$84.7 \pm 4.3\%$	$15.3 \pm 4.3\%$	103.2 ± 4.6	$92.8 \pm 1.6\%$	$7.2 \pm 1.6\%$	109.6 ± 6.2
	$8 \times 8/120$	$42.6 \pm 6.8\%$	$35.1 \pm 7.1\%$	163.2 ± 9.3	$76.8 \pm 6.6\%$	$23.2 \pm 6.6\%$	123.2 ± 11.3	$89.2 \pm 2.0\%$	$10.8 \pm 2.0\%$	115.3 ± 6.9
S_d	$4 \times 4/10(D)$	$85.2 \pm 4.3\%$	$10.4 \pm 4.3\%$	90.2 ± 8.8	$73.4 \pm 6.8\%$	$24.8 \pm 1.3\%$	63.4 ± 4.8	$95.2\% \pm 2.2\%$	$4.2\% \pm 2.4\%$	52.3 ± 4.8
S_c	$4 \times 4/30$	$77.3 \pm 5.4\%$	$17.3 \pm 3.1\%$	79.5 ± 9.5	$85.4 \pm 3.4\%$	$14.6 \pm 3.4\%$	65.3 ± 4.5	$90.4 \pm 1.2\%$	$9.6 \pm 1.2\%$	57.2 ± 2.2
S_m	$4 \times 4/30$	$76.8 \pm 4.9\%$	$19.1 \pm 4.3\%$	78.2 ± 8.8	$83.9 \pm 2.9\%$	$16.1 \pm 2.9\%$	63.7 ± 5.8	$87.4 \pm 1.8\%$	$12.6 \pm 1.8\%$	54.0 ± 3.0

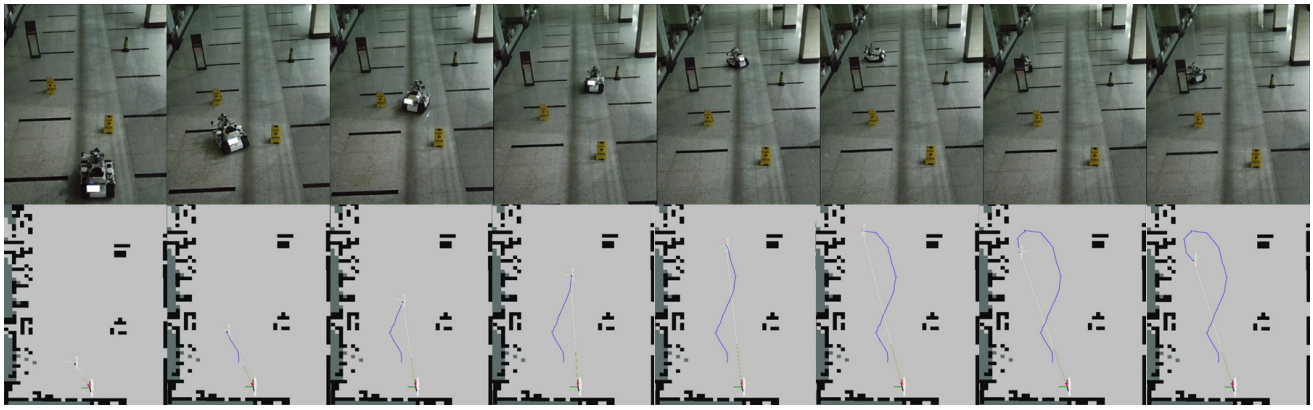

Fig. 4. Real-world experiment environment, where the bule line plots the real robot trajectory, the gray area is the free space where the robot can move. The obstacles placed are not marked in the map. Robots need local observations from their own sensors to avoid them.

inferring the variation pattern of gas concentration.

To closely emulate the gas distribution patterns in real-world environments, we utilized Fluent simulation software to construct a confined space resembling a factory structure. This space features an intake for the gas source, an intake and an exhaust for airflow circulation. Three differently shaped obstacles are introduced within the space to affect the gas distribution. The concentration distribution of gases changes over time. To showcase the robot's gas source localization capability amidst a plume, we selected a navigation scenario characterized by relatively stable gas distribution during the diffusion phase. Notably, even within environments with intricate concentration patterns, our approach successfully executes the task of searching gas-source when expert algorithm generate correct local guidance information.

C. Physical Robotic Experiments

In this section, we present real-world experiments to showcase the effectiveness and practicality of our approach. As shown in the robot illustration, we validated our method on an industrial-grade mobile robot platform. The robot is equipped with a three-dimensional laser sensor (VLP-16) and four gas sensors (TGS2620). These sensors are strategically positioned three at the front and one at the rear of the robot to perceive environmental gas concentrations. The robot incorporates a computational platform housing an Intel Core i5-9600 central processing unit and a low-level motion control system with encoders. As depicted in the environmental representation, we considered a closed indoor environment containing randomly placed static obstacles, such as signs, markers, and barriers. For the gas source, we utilized industrial alcohol. Notably, the robot showcases remarkable agility in navigating around diverse-shaped static obstacles. Importantly, when encountering the gas source located near the rear of an obstacle, the robot employs a strategy to navigate around the obstacle and approach the gas source from the back. This behavior underscores the robot's capacity to comprehend the environment and reasonably plan gas source localization routes based on the current distribution of obstacles, rather than blindly adhering to guidance from the expert module. The experimental outcomes affirm the practicality of our method in both physical robotic systems and real-world scenarios.

VII. CONCLUSION

In this paper, we study an active gas-source search task in various unfamiliar environments with many unknown physical structure static obstacles. Local goal guidance generated by expert algorithm achieves more efficient navigation policy, due to skip skipping the study of gas distribution. Local observation representation and goal guidance representation generated the input of fix-size image, which make model learning the environment information and the guidance information higher-quality and easier. Simulation results show the versatility and transferability of our approach in unfamiliar environments with various structures of obstacles. In future work, we intend to extend our approach to multi-robot collaborative gas source exploration tasks.

REFERENCES

[1] R. R. Murphy, J. Kravitz, S. L. Stover and R. Shoureshi, "Mobile robots in mine rescue and recovery," in IEEE Robotics and Automation Magazine, vol. 16, no. 2, pp. 91-103, June 2009.

[2] Y. Guo, Jiatong Bao and Aiguo Song, "Designed and implementation of a semi-autonomous search robot," 2009 International Conference on Mechatronics and Automation, Changchun, China, 2009, pp. 4621-4626

[3] I. Palunko, P. Cruz and R. Fierro, "Agile Load Transportation : Safe and Efficient Load Manipulation with Aerial Robots," in IEEE Robotics and Automation Magazine, vol. 19, no. 3, pp. 69-79, Sept. 2012.

[4] E. T. Alotaibi, S. S. Alqefari and A. Koubaa, "LSAR: Multi-UAV Collaboration for Search and Rescue Missions," in IEEE Access, vol. 7, pp. 55817-55832, 2019.

[5] F. Negrello et al., "Humanoids at Work: The WALK-MAN Robot in a Postearthquake Scenario," in IEEE Robotics and Automation Magazine, vol. 25, no. 3, pp. 8-22, Sept. 2018.

[6] C. M. Humphrey and J. A. Adams, "Robotic tasks for chemical, biological, radiological, nuclear and explosive incident response,"in Adv. Robot., vol. 23, no. 9, pp. 1217–1232, Jan. 2009.

[7] T. Wiedemann, C. Vlaicu, J. Josifovski and A. Viseras, "Robotic Information Gathering With Reinforcement Learning Assisted by Domain Knowledge: An Application to Gas Source Localization," in IEEE Access, vol. 9, pp. 13159-13172, 2021.

[8] Z. -P. Wang and H. -N. Wu, "Gas Source Localization using Improved Multi-Agent Reinforcement Learning," 2020 Chinese Automation Congress (CAC), Shanghai, China, 2020, pp. 6696-6701

[9] D. Fox, W. Burgard, S. Thrun, "The Dynamic Window Approach to Collision Avoidance, " *IEEE Robotics and Automation Magazine*, 2002, 4(1):23-33.

[10] Patle B K, Chen S L, Singh A, et al, "Optimal trajectory planning of the industrial robot using hybrid S-curve-PSO approach, " *Robotic Intelligence and Automation*, 2023, 43(2): 153-174.

[11] Ul Islam Q, Ibrahim H, Chin P K, et al, "FADM-SLAM: a fast and accurate dynamic intelligent motion SLAM for autonomous robot exploration involving movable objects," *Robotic Intelligence and Automation*, 2023.

[12] Q. Li, F. Gama, A. Ribeiro and A. Prorok, "Graph Neural Networks for Decentralized Multi-Robot Path Planning," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2020, pp. 11785-11792.

[13] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-Robot Interaction: Crowd-aware Robot Navigation with Attention-based Deep Reinforcement Learning," *IEEE International Conference on Robotics and Automation*, 2019, pp. 6015–6022.

[14] Monroy J, Ruiz-Sarmiento J R, Gonzalez-Jimenez J. "An evaluation of plume tracking as a strategy for gas source localization in turbulent wind flows," *2019 IEEE International Symposium on Olfaction and Electronic Nose (ISOEN)*, IEEE, 2019: 1-3.

[15] Ishida H, "Blimp robot for three-dimensional gas distribution mapping in indoor environment," *American Institute of Physics*, 2009, 1137(1): 61-64.

[16] Zhang S Q, Cui R X, Xu D M. "Performance Analysis of information tendency search algorithm in sparse environment," *Robot*, 2013, 35(4): 432-438.

[17] Xing, B., Gao, WJ. "Fruit Fly Optimization Algorithm". in Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms. Intelligent Systems Reference Library, vol 62. Springer, Cham.

[18] C. Chen, S. Hu, P. Nikdel, G. Mori, M. Savva, "Relational Graph Learning for Crowd Navigation," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 10007–10013.

[19] 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 Robotics and Automation Letters*, vol. 5, no. 2, pp. 2754–2761, 2020.

[20] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. L. Cun, "Off-road obstacle avoidance through end-to-end learning, " *Advances in neural information processing systems*, 2006, pp. 739–746.

[21] J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard, "Deep reinforcement learning with successor features for navigation across similar environments, " *arXiv*, 2016, arXiv:1612.05533.

[22] S. Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert, "Learning monocular reactive uav control in cluttered natural environments, " *International Conference on Robotics and Automation*, 2013, pp. 1765–1772.

[23] Z. Liu et al., "LPD-Net: 3D Point Cloud Learning for Large-Scale Place Recognition and Environment Analysis," *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2831-2840.

[24] H W Kang, H Y Zhou, X Wang et al., "Real-Time Fruit Recognition and Grasping Estimation for Robotic Apple Harvesting", in SENSORS, vol. 20, no. 19, 2020.

[25] Y Chen, G L Liu, Y M Xu et al., "PointNet plus plus Network Architecture with Individual Point Level and Global Features on Centroid for ALS Point Cloud Classification", in REMOTE SENSING, vol. 13, no. 3, pp. 472, 2021.

[26] P. Long, T. Fan, X. Liao, W. Liu, H. Zhang and J. Pan, "Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning, " *IEEE International Conference on Robotics and Automation*, 2018, pp. 6252-6259.

[27] L. Chen, Y. Lin, Z. Kang, "Improved bottle sea sheath group algorithm based on attenuation factor and dynamic learning, " *Control Theory & Applications*, 2020, 37(8): 1766-1780.

[28] H. Wang, H. Li, Y. Liu, C. Li and S. Zeng, "Opposition-based particle swarm algorithm with cauchy mutation," *2007 IEEE Congress on Evolutionary Computation*, Singapore, 2007, pp. 4750-4756