RASLS: Reinforcement Learning Active SLAM Approach with Layout Semantic

Changhang Tian[1,2], Shasha Tian[1,3,*], Yilin Kang[1], Hong Wang[1,2] Jun Tie[1,3] Shenhao Xu[1]
[1]*School of Computer Science, South-Central Minzu University, Wuhan, China*
[2]*Hubei Provincial Engineering Research Center for Intelligent Management of Manufacturing Enterprise, Wuhan, China*
[3]*Hubei Provincial Engineering Research Center of Agricultural Blockchain and Intelligent Management, Wuhan, China*

***Abstract*—Active SLAM plays a crucial role in applications of embodied intelligence. Previous learning-based methods struggle to sufficiently leverage semantic information in the environment, while frontier-based algorithms face challenges in mitigating myopic decision-making issues. Considering that humans experientially use observed information while exploring environments, we propose a deep reinforcement learning approach incorporating object semantic information and design a reward-matching mechanism based on the prior object layout. To tackle the instability in exploration gains caused by error optimization in the mapping part of the SLAM system, we introduce a method for differential map uncertainty confidence filtering. We conduct reinforcement learning training using Gazebo in office scenarios based on 3DGEMS and perform comparisons in a new scenario. Through ablation analysis, we demonstrate the effectiveness of layout semantic information. Compared to the latest reinforcement learning baseline, experimental results indicate that our method achieves a higher success rate with a shorter average execution time and path length. Our code and training world is available at https://github.com/Moresweet/DRL_ARE_Graph_Gazebo.**

***Index Terms*—Active SLAM, Semantic, Deep Reinforcement Learning, Differential Map, Embodied Intelligence**

I. Introduction

Active SLAM [1], [2] is a crucial decision-making process implemented in embodied intelligence [3]. Active SLAM refers to the paradigm where a robot can create the most accurate and complete mapping of an unknown environment by controlling its motion. Autonomous robot exploration (ARE) is typically the key point of active SLAM. The robot's active SLAM behavior in a given environment relies on its perceptual information about the surroundings. Frontier-based algorithms [4] are classical approaches to autonomous exploration problems. Researchers focus on designing frontier utility and information gain [5], [6] to determine the Next Best Viewpoint (NBV) [7]. However, frontier-based methods can become inefficient and ambiguous in scenarios with a large number of frontiers. Additionally, these methods inevitably need to be revised to avoid myopic planning. Recently, deep reinforcement learning has achieved excellent results in autonomous navigation. The interactive learning approach aligns with the needs of embodied intelligence. Robotic agents equipped with cameras, radar and experience replay memory [8] mimic human visual and auditory organs and brain memory. Human-like embodied perception data input through the observation space of reinforcement learning and trained for decision policy are vital for achieving embodied intelligence.

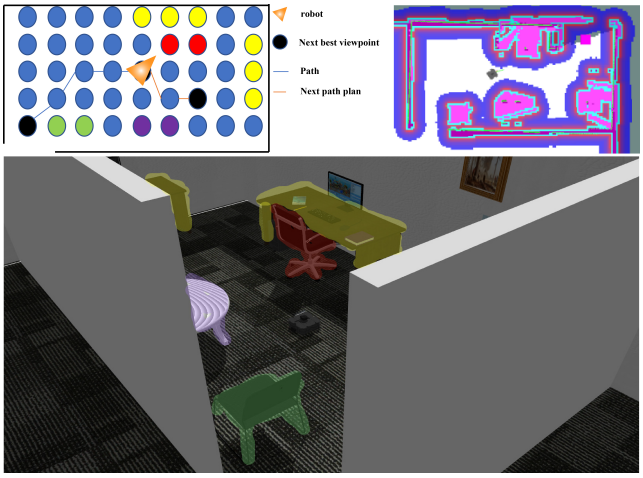


Fig. 1. **Illustration of semantic exploration of RASLS.** We represent the decision-making environment in the form of an augmented graph with nodes incorporating semantic occupancy information of objects (each colored node corresponds to a object of the respective color in the environment). Semantic nodes enables reinforcement learning method to learn the benefits of layout information for autonomous exploration tasks, encouraging better decision waypoints in the reinforcement learning process.

Prior information about maps holds significant inspiration for the active SLAM process. The active SLAM process involves measurement and decision-making in unknown areas. Therefore, the more prior information an agent utilizes during exploration, the greater the information entropy we can get for the agent's decision-making. Semantic maps [9], [10] allow robotic agents to understand the environment during action cognitively. In active visual exploration [11], semantic maps abstract sensor data to human-level concepts, aiding agents in semantic-driven exploration [12]. However, traditional active visual techniques do not truly understand semantic maps, often treating objects as obstacles for collision avoidance [13]. Recent work in semantic map-based environment search

This work is supported by the Special Project on Regional Collaborative Innovation in Xinjiang Uygur Autonomous Region (Science and Technology Aid Program) [grant number 2022E02035]; the Hubei Provincial Administration of Traditional Chinese Medicine Research Project on Traditional Chinese Medicine [grant number ZY2023M064]; the Hubei Province Key Research and Development Special Project of Science and Technology Innovation Plan [grant number 2023BAB087]; Wuhan Key Research and Development Projects[grant number2023010402010614]; and the Wuhan knowledge innovation special Dawn project [grant number 2023010201020465].

*Corresponding author: Shasha Tian(shashatian77@mail.scuec.edu.cn).

has emerged in visual target navigation [14]. However, most semantic maps use representations related to the number of detected categories, such as multi-layered maps, making the handling of multidimensional input data complex. In our work, we employ an augmented graph representation of the map, associating regions occupied by detected semantic categories with nodes in the graph to provide more specific environmental semantic prior information for the active SLAM process. Since nodes inherently represent positions, the semantic observation of object layout in the map provides valuable insights for agent learning. We decouple the input dimensions of semantic encoding from the number of detected categories to reduce the input dimensions of the reinforcement learning algorithm. Specifically, we unify the semantic input dimensions through prime numbers and one-hot vector conversion. Additionally, in reward design, we encourage new area discovery behavior based on prior layout patterns.

Due to the interactive nature and partial observability [15] of environmental interactions, deep reinforcement learning methods are sensitive to the environment. In our experiments, we found that the SLAM system's backend optimization [16] introduces significant variance into the observation space. To avoid propagating errors, we cannot directly use the mapping results of the SLAM system as the reward basis since the pose and mapping are not always robust. In our method, we design a differential map morphology filter processing method based on uncertainty measurement to obtain stable mapping gains in new areas. As the number of nodes in the augmented graph is downsampled from grid maps in SLAM, minor pose estimation fluctuations do not affect the positions of the nodes in the augmented graph, ensuring reliable learning of the agent's policy. The main contributions of this paper are as follows:

1. We employ an augmented semantic map to represent environmental object layout relationships. Simultaneously, we unify semantic input dimensions through prime numbers and one-hot vector conversion.

2. We design an uncertainty measurement metric to evaluate region contours. We then obtain stable exploration gains through filtering and morphological operations on the confidence intervals of differential map images.

3. Through high-fidelity experiments on the ROS Gazebo platform, we demonstrate that introducing object layout semantic information can enhance exploration efficiency. Additionally, it performs well in a new scenario with structured layouts.

II. Related Works

Active Mapping Strategies. Frontier-based algorithms, initially proposed in [4], are intuitively practical. Although recent algorithms for frontier selection always cover the workspace, zigzag paths are generated frequently because of greedily selecting the nearest frontier. To make frontier selection more reasonable, researchers have combined sampling-based search algorithms, such as Rapidly Exploring Random Trees (RRT), with frontier-based algorithms [17], [18]. However, frontier-based methods are effective in short-term path planning, and greedy strategies inevitably lead to myopic decision. Due to the success of deep reinforcement learning in autonomous navigation, frontier-based deep reinforcement learning method is first proposed in [19]. Some approaches generate end-to-end instructions without planning and navigation [20], [21], while others directly generate viewpoints for the agent to access, leaving navigation and planning to the system [22]–[24]. Recently, [25] proposed a reinforcement learning exploration method with multi-scale attention using image simulation training. However, ideal observation data is unavailable in real active SLAM systems. And the method using pictures data simulation may deviate from the constraints of the actual physical environment.

Semantic Information Mapping. The construction of obstacle maps using semantic mapping has been extensively researched in the SLAM field [26]. Most work is focused on visual SLAM (VSLAM), with limited applications in active SLAM. Recently, [27], [28] designed a multi-category semantic OctoMap for active SLAM, considering Shannon mutual information (MI) of observed categories when selecting the robot's trajectory. With the development of computer vision, incorporating computer vision models into semantic maps [29], [30] has emerged. Chaplot D S et al. [31] employed a differentiable projection operation to map visually detected objects into a multi-channel matrix based on the number of categories. However, this mapping is tightly coupled with the detected categories, lacking a unified representation for the input of semantic maps.

Uncertainty Indicator in Reward Design. Agents performing active SLAM activities in complex scenarios rely on environmental data obtained from the SLAM system. Due to the uncertainty in probability estimation, the pose and mapping provided by the SLAM system are not always stable and correct. Niroui et al. [19] and Chen et al. [24] considered historical poses and frontiers in the reinforcement learning observations and adopted a structured graph linking optimal candidates. Recent works on policy learning with separated motion planning [25], [32] used nearby position sampling in simulated experiments in ideal environments. However, errors are inherent in embodied intelligence scenarios. Chaplot et al. [31] used a hierarchical trajectory planner to separate viewpoint decision-making based on frontiers and motion control targets.

III. Problem Formulation

A two-dimensional graph map P can be modeled as $x \times y$, containing a set of points $V = \{p_{0,0}, p_{0,1}, \dots p_{x,y}\}$. Let P_u represent the unknown region in the map and P_k the known region, such that $P = P_u \cup P_k$. In this paper, we consider the path of active SLAM actions as a sequence of viewpoints. The point chosen in the i-th decision is denoted as μ_i, where $\mu_i \in V$. The sequence of path points for the exploration task is $\mathrm{M} = \{\mu_0, \dots, \mu_i, \dots, \mu_n\}$, where after the n-th point selection, $P = P_k$. Consequently, obtaining the solution space for

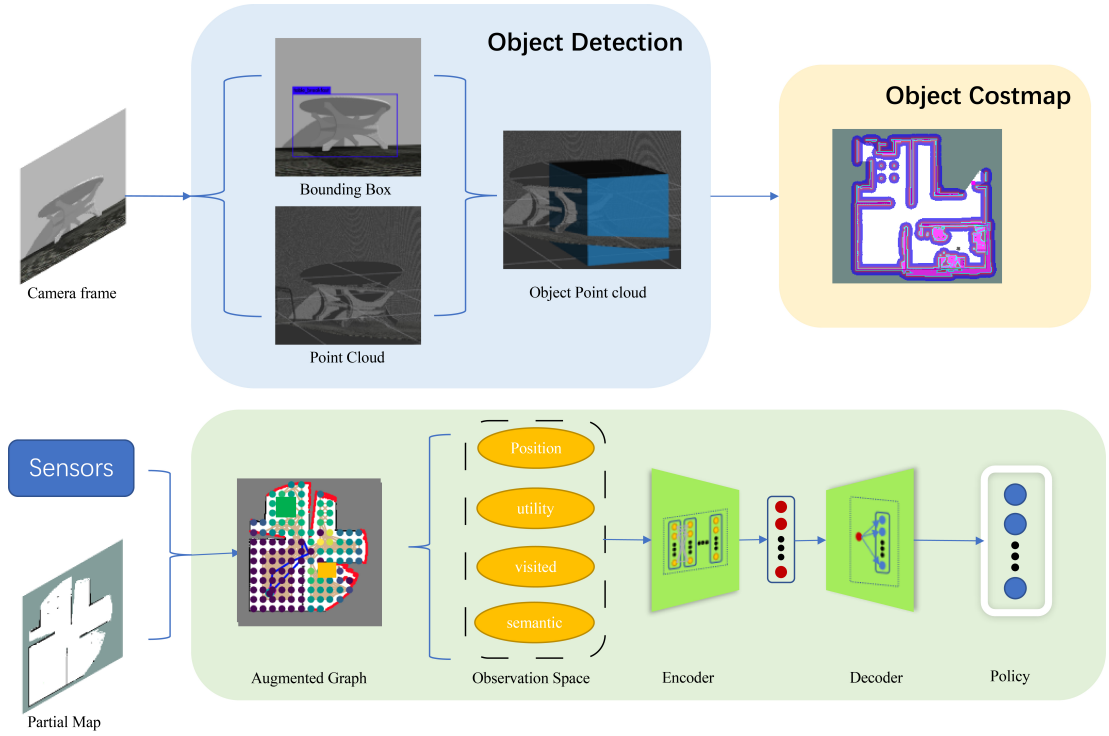


Fig. 2. **An overview of RASLS.** The obstacle avoidance module utilizes depth camera frames to extract object category labels and three-dimensional dimensions, incorporating them into a local obstacle avoidance cost map that serves as a basis for obstacle avoidance. Simultaneously, object detection category labels, sensor data, and local maps are utilized as the foundation of an augmented graph. This augmented graph provides observational data for Soft Actor-Critic (SAC). Through an encoder-decoder structure based on multi-head attention, the system ultimately utilizes a policy network to output a sorted sequence of decision-making waypoints, facilitating the selection of the next viewpoint.

the entire active SLAM active mapping becomes a sequential decision-making problem. Sequential decision problems are typically described using a Markov Decision Process (MDP) [33]. Due to the uncertainty in actions and observations, active SLAM is generally modeled as a Partially Observable Markov Decision Process (POMDP) [1], represented by a seven-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \Omega_{\mathcal{S}}, \Omega_{\mathcal{O}}, r, \gamma)$. The elements of seven-tuple include $\mathcal{S}$ as the agent's state space, $\mathcal{A}$ as the agent's action space, $\mathcal{O}$ as the agent's observation space, $\Omega_{\mathcal{S}}$ as the state transition function, $\Omega_{\mathcal{O}}$ as the observation transition function, r as the reward mapping function influencing the transition from state to action, and γ as the discount factor for long-term returns. Specifically, $\Omega_{\mathcal{O}} : \mathcal{S} \times \mathcal{A} \mapsto \Pi(\mathcal{S})$ represents the joint state transition probability model for $\mathcal{S}$ and $\mathcal{A}$ (where $\Pi(\mathcal{S})$ is the probability density function for $\mathcal{S}$, and $\Omega_{\mathcal{O}} : \mathcal{S} \mapsto \Pi(\mathcal{O})$ represents the $\mathcal{S}$-influenced $\mathcal{O}$ transition probability model (where $\Pi(\mathcal{O})$ is the probability density function for $\mathcal{O}$). $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ maps $\mathcal{S}$ and $\mathcal{A}$ to the real domain of rewards, and $\gamma \in (0, 1) \in \mathbb{R}$ is the discount factor for long-term returns. Due to the influence of system errors, there is uncertainty in the map areas in the observation space. We employ mapping $\mathcal{C}(P) \triangleq \{c : (P_u \times P_k \mapsto \mathbb{R}\}$ to obtain more reliable observations of newly added areas. Finally, under the POMDP framework, we find the optimal policy function $\pi^*(a \mid s)$ that maximizes the long-term return. The optimal policy function includes a sequence of actions M^*:

$$\mathrm{M}^* = \underset{s \in \mathrm{V}}{\operatorname{argmin}} \sum_{i=1}^{t} \Omega_s \left(s_t \mid s_{t-1}, \mu_{t-1}\right) \gamma^t r_t, \quad \text{s.t. } |\mathcal{C}(P)| = |P_g| \tag{1}$$

where P_g is the ground truth of the map. Although the ground truth of the map is unknown in active SLAM, in practice, we can have prior knowledge in the form of labels during training and validation. The trajectory composed of path points in M^* can achieve the entire active SLAM active mapping while minimizing the path's length and time.

IV. METHOD

The overview of our proposed system is illustrated in Figure 2. The SAC network structure employed in our work is based on [25].

A. Map Representation

We model the map as a collision-free graph $G_t = (V_t, E_t)$, where V_t represents path viewpoints uniformly distributed in the free area, and E_t is the edge set of V_t in G_t, representing the connectivity relationship Ξ of sample points in the vertex set V_t. The definition of Ξ is given as $\Xi = \{(v_i, v_j) | v_i, v_j \in V_t, 1 \leq i, j \leq | V_t |, \delta(v_i, v_j) = 1\}$, where $\delta(v_i, v_j)$ is the collision detection function checking whether there is an obstacle between two points. The collision-free graph G_t updates nodes and edge sets as the

map's free area changes, ensuring it consistently represents the latest environmental perception information. Compared to commonly used grid maps, the graph modeling approach can downsample dense grids, reduce the number of sampled points, and still maintain the connectivity between path points. We use YOLOv3-tiny [34] as the object detection algorithm and retain the object detection results as one-hot. By real-time detecting video frames with a depth camera, define obj_i as the semantically observed object for the current node. We use prime numbers greater than or equal to 2 to represent the fused semantics of the object. Our semantic object sequence $\mathbf{t}$ comes from the output of the YOLOv3-tiny algorithm, where t_i is the detection result of the i-th class object, $t_i = 1$ when the corresponding object is detected, and $t_i = 0$ otherwise. The existing prime number sequence $\mathbf{prime} = [2, 3, 5, \dots]$, and the size of the prime number sequence is the same as the one-hot vector $\mathbf{t}$ of the semantic objects to be detected. The transformation relationship from the detected object vector $\mathbf{obj}$ to the prime-number value space is follows:

$$\mathbf{obj}_i = \begin{cases} \prod prime_i, & t_i = 1,\ \forall t_i \in T \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Through the above calculation, we map the one-hot vector of object detection results to a constant term.

Proof. According to the unique prime factorization theorem for positive integers, for any positive integer $\forall n \in N^+, n = \prod_{i=1}^{k} \mathbf{prime}_i$. As for the uniqueness of the factorization, if we have two different prime factorizations, they must be one-to-one. This is because the selection of primes is unique, and their order and quantity are also unique. Assume n has two different prime factorizations is:

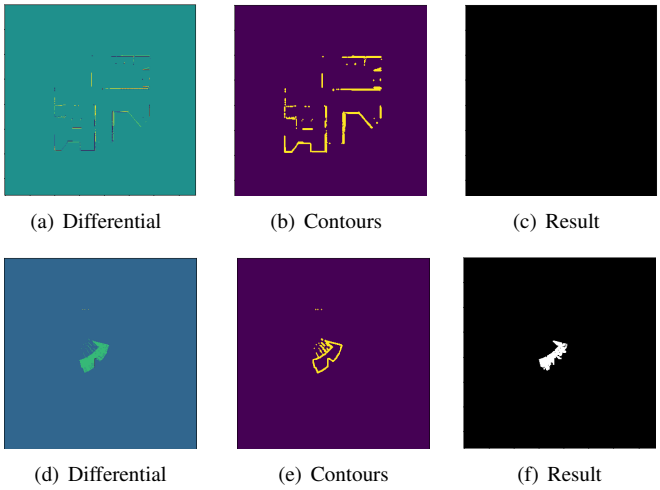
$$\begin{aligned} \mathrm{n} &= \mathrm{p}_1 \times p_2 \times \ldots \times p_k \\ \mathrm{n} &= \mathrm{q}_1 \times q_2 \times \ldots \times q_m \end{aligned} \tag{3}$$

Then, $\frac{n}{n} = \frac{p_1 \times p_2 \times \ldots \times p_k}{q_1 \times q_2 \times \ldots \times q_m} = 1$ According to the unique prime factorization theorem for positive integers, 1 has only one prime factorization, namely 1=1. Therefore, the two factorizations are equivalent, i.e., the two factorization methods are only different in the arrangement of elements.

By inputting the numerical results into the observation space as one-hot, we keep the observation space dimension of semantic information single rather than expanding it to the same size as the target detection category scale. This method significantly reduces computation and avoids the problem of dimension explosion. Our method maps the results of object detection to a constant and adds them to the node weight values of map modeling. The bounding box of the object detection result is mapped from the two-dimensional image pixels to the three-dimensional space. The object's three-dimensional contour is extracted through point clouds, and the nodes containing the occupied area in the two-dimensional map are added to the object semantic information. In this way, the SAC algorithm can introduce more prior information during the training process.

B. Policy Learning

Observation Space. In the SAC algorithm's observation space, information about the robot's current position is crucial, and discovering new frontiers in exploration tasks is vital. Each node has the number of observable frontiers within the range of the laser sensor as utility weights are input to the observation space. To introduce more useful observation information, we further process the augmented graph $G_t' = (V_t', E_t)$ of the graph $G_t = (V_t, E_t)$. Let $F_{o,i}$ be The frontiers observed by the current node can be described as , d_s be the scanning range of the current laser sensor, and $L(v_i, f_j)$ be the straight line connecting the current node to the set of frontier points. V_t' is the enhanced node set, with each enhanced node $v_i' = (x_i, y_i, u_i, b_i, \mathbf{obj}_i)$, where x_i and y_i are the current position of the robot, u_i is the utility value of the current node (i.e., $u_i = |F_{o,i}|, \forall f_j \in F_{o,i}, \|f_j - v_i\| \leq d_s, L(v_i, f_j) \cap (P - P_f) = \emptyset$), b_i indicates whether the agent has visited this node, and including the detected object semantic information $\mathbf{obj}_i$ in the experiment can make the agent's policy discover the relationship between semantics and enhance exploration efficiency. The agent's observation sample is $o_t = (G_t', S_t)$.



(a) Differential (b) Contours (c) Result

(d) Differential (e) Contours (f) Result

Fig. 3. **Differential map filtering process.** The figure illustrates handling differential images for adjacent frames in the incremental map. Figures (a) to (c) depict an example of the SLAM system correcting jitter caused by errors. No new areas occurred in this case. Figures (d) to (f) present a genuine example where the discovery of new areas does occur.

Action Space. In the update learning process, the actions still use the relationship between graph node connections to reflect the relationship between nodes. During training, the policy selected by nodes will be combined with the process with actual motion. A more smooth waypoints policy is learned through reinforcement learning. According to the observation sample o_t, processed by the deep neural network module based on the attention layer, the policy network generates the coordinate information $(x, y) \in \mathrm{M}^*$ of the next viewpoint $\forall \mu_i \in P_f$. The policy is represented as $\pi_\theta(a_t \mid o_t) = \pi_\theta(\mu_{t+1} = v_i, (\mu_t, v_i) \in E_t \mid o_t)$, where θ is the weight of

the neural network. According to the policy output, the robot selects the navigation target point, and the robot's local map structure is synchronized with the change in sensor perception and robot position.

Reward Space. The reward design in reinforcement learning can encourage the exploration process and guide the generation tendency of the policy. In previous work, reward design only focused on discovering the number of new frontiers. However, in the actual process of SLAM mapping, the noise of frontier points can be significant, and the change of frontier points is real-time. Therefore, we use a more reliable reward for discovering new regions, $r_p = |P_k, s_{t+1}| - |P_k, s_t|$. We found that calculating the newly explored area will have a significant impact due to the uncertainty in SLAM loop detection and pose estimation. Therefore, based on the difference map between the previous and current frames, we adopted a method that measures the uncertainty of the edge of the newly explored area and morphological processing, obtaining an area beneficial to reinforcement learning and filtering out a number of noise points. We obtain the belief space before and after the action, take the difference of the belief space matrices, extract the contour of the unknown-to-free area in the representation matrix, and measure the uncertainty of the difference image I_{diff} using the following metric:

$$uncertain_i = \frac{\sum_J \sum_{(x',y')\in N(x,y)} I_{\text{free}}\ (x',y')}{|J|} \tag{4}$$

where I_{free} represents pixel points in the difference image that indicate unchanged, changed from unknown to free, and changed from unknown to the obstacle, and J represents the pixel point set of the contour. $N(x,y)$ is the pixel points in the current pixel neighborhood, $N(x,y) = \{(x',y') \mid x' = x-1, x, x+1; y' = y-1, y, y+1; (x',y') \neq (x,y)\}$. We use a threshold to determine the confidence interval (taking 0.18 in actual experiments) and filter pixels below a certain threshold. For the processed image, we use the opening operation of the image to obtain the final reliable map's newly added area as a reward reference.

We consider the total distance moved as a penalty in the reward space. With the growth of iteration steps, the penalty term will become larger. The action decay reward function is $r_a = -B(p_1, p_2, ..., p_i)$.

We have designed an algorithm for matching semantic information. For each node v_i in the current exploration area V_c' with $\mathbf{obj}_{v_i} \neq 0$, we form different node clusters based on different object semantic values. These clusters are represented as $\mathbf{Clusters} = \{C_1, C_2, \ldots, C_k\}$, where C_i denotes the i-th node cluster. We employ a depth-first strategy for each node cluster C_i, starting from node v_i and expanding to explore other nodes within a 3-layer depth. When encountering a node v_j in the previous exploration area V_p' with $\mathbf{obj}_{v_i} \neq 0$, the algorithm checks the matching relation in the matrix $\mathbf{K}$. If the element in the $\mathbf{obj}_{v_i}$-th row and $\mathbf{obj}_{v_j}$-th column of $\mathbf{K}$ is 1, the traversal is stopped, and the matching count is incremented by 1. This process is repeated for other newly discovered node clusters until all node clusters are processed. Ultimately, the algorithm outputs the total matching count Q_m:

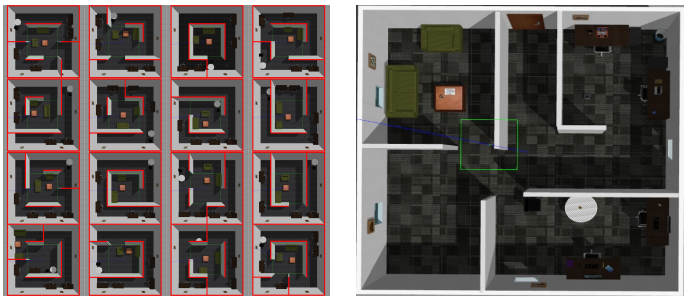
$$Q_m = \sum_{i=1}^{k} \sum_{v\in C_i} \sum_{u\in V_p'} \sigma\left(\text{obj}_u, \text{obj}_v \neq 0\right) \cdot K\left[\text{obj}_v, \text{obj}_u\right] \tag{5}$$

where k denotes the total number of node clusters. The symbol $\sigma(\cdot)$ represents the logical delta function, taking a value of 1 when the condition in the parentheses is true and 0 otherwise. Leveraging the metrics outlined above, the semantic object discovery reward is formally defined as $r_o = \alpha \cdot Q_m$.

Due to obstacles causing occlusion, our local obstacle avoidance strategy often enters blind spots in the depth camera. Therefore, the cost map is not always complete. We have designed the following penalty for planning failures is:

$$r_c = \begin{cases} -20 & \text{if a collision occurred} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The total reward is calculated as $r = r_p + r_a + r_o + r_c$.



(a) Train (b) Test

Fig. 4. **Experiment enviroment.** The figure illustrates our training and validation scenarios. Panel (a) displays 16 distinct scenarios used for training, while panel (b) showcases the scenarios utilized in our validation experiments. These example scenarios feature similar object layouts but with randomized positions. Additionally, the structure of the walls in the rooms differs.

V. Experiment

Our experiments consist of two parts: comparative experiments and ablation experiments. In comparative experiments, we compared the traditional Nearest Frontier Selection method [4], DUDE-based approach [35], and ARiADNE [25] in a new scenario as shown in Figure 4. Comparative experiments, the evaluation metrics include the success rate of exploration, coverage rate at the unit decision step, as well as the mean, extremum, and standard deviation of exploration time and trajectory length.The ablation experiments involved comparing the total exploration area per episode after eliminating the observation of semantic layouts as shown in Figure 7.

A. Train Detail

Our work employed the publicly available small-scale office scenarios provided by 3DGEMS [36] as our training environment. We curated 16 scenarios with predefined semantic layouts and alternated between them after five learning cycles. We manually introduced partitioning walls to impart generalizability, heightening the exploration complexity. In our

TABLE I. **Comparison with baseline exploration methods (30 times each test scenario)** The decision viewpoints is generated from baseline algorithms, while the underlying navigation planning algorithm follows consistent conditions. We integrated A* as the global path planner and DWA as the local path planner using the move_base framework in ROS.

Method	Office								
	Exploration Time (s)				**Path Length (m)**				**Success (%)**
	Min	Max	Avg	Std	Min	Max	Avg	Std	
Frontier	180.05	259.98	222.23	23.32	572.26	990.55	775.18	132.26	30.0
DUDE-based	**142.95**	283.34	213.43	37.13	**482.66**	**869.79**	**742.97**	134.28	36.6
ARIADNE	208.92	273.30	244.02	19.98	884.46	1086.63	985.27	**61.14**	43.3
RASLS	174.33	**237.30**	**206.56**	**18.68**	691.91	978.32	838.43	71.44	**76.6**

comparative experiments, each baseline and proposed method were tested 30 times in entirely new scenarios. Our mapping precision was set at 0.05m/pixel, encompassing a map size mapped onto a graph of 900 uniformly distributed nodes. The map's weights and semantic occupancy information were updated via sensor data interactions. In our experiment, six object categories were detected, including 'office desk,' 'office chair,' 'breakfast table,' 'breakfast chair,' 'sofa,' and 'coffee table.' Simultaneously, we provided a prior layout matching relationship matrix for the six object categories.

Utilizing the turtlebot3 waffle pi robot model, our setup featured veldeny laser sensors and realsense depth cameras. The laser sensors had an 80-unit range and underwent dilation in specific proximity to obstacles, eliminating connectivity between obstacles and closely adjacent nodes, consequently reducing collision probabilities at designated strategy points while enhancing path generation smoothness. Our global path planner employed the A* algorithm, while the local planner adopted the DWA algorithm, with a simulation time set at 3.1. We acquired images of six distinct categories within the scenarios through photographic documentation. Following data annotation, we utilized the YOLOv3-tiny model for training and employed it in the experimental phase for object detection. Subsequently, to delve into the positive implications of semantic objectives on autonomous exploration.

For mapping purposes, we utilized the gmapping [37] algorithm within the SLAM framework. Gazebo served as our simulation tool, within which we engineered a system for reinforcement learning environments. Interactions within the simulation spanned 300 episodes, allowing a maximum of 64 decision steps per episode. Ground truth data for the experimental map relied on manually constructed map information, considering exploration successful upon reaching 95% coverage. Our training workstation boasted an i5-12490F CPU and an NVIDIA GeForce RTX3060 GPU. The entire training process required approximately two days to complete.

B. Experimental Results and Analysis

Comparative experiment results as shown in Table I. Each episode in our experimental algorithm comprised 32 decision steps. We set the learning rate for the value network at 0.00001 and the temperature parameter's learning rate at 0.0001. Through a comparative analysis of experimental results, we observed that in successful cases, DUDE-based method consistently demonstrated optimal performance in terms of path length and time efficiency, consistently achieving the shortest path and minimal time. However, DUDE-based method exhibited relatively high standard deviation in exploration time and path length, indicating a lower stability in novel scenarios.

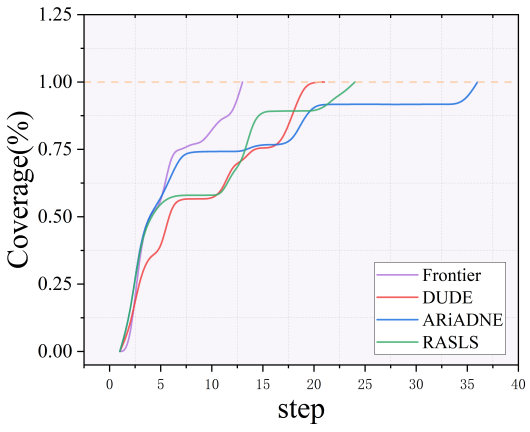
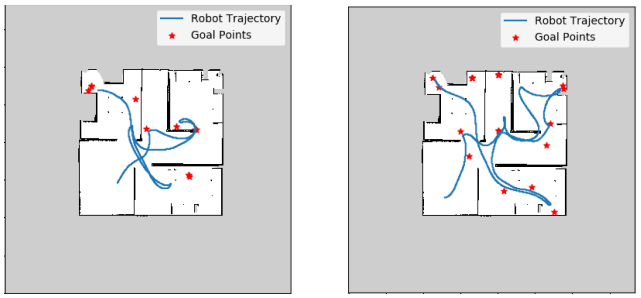


Fig. 5. **Coverage rate of each decision step.** We compared the coverage at each decision step for the four experiments. Different colored solid lines in the graph represent the variation of coverage for each algorithm with respect to decision steps. The x-axis represents the decision steps, while the y-axis indicates the coverage rate.

Traditional nearest frontier selection algorithms showed lower success rates in scenarios with obstacles, lacking consideration of practical environmental factors in decision point selection in crowded environments. In contrast, our baseline method exhibited the lowest standard deviation in overall path length, highlighting the algorithm's stability. However, due to ARiADNE's encouragement of short paths in reward design, it faced challenges in completing exploration tasks within a specified number of steps. The increased frequency of short paths resulted in relatively longer exploration times. In comparison, our method encouraged a penalty that grows with time without imposing specific requirements on path length. Consequently, our approach demonstrated shorter and relatively stable average exploration times.

While our method did not yield the shortest overall path length, it outperformed ARiADNE, another reinforcement learning-based approach, in achieving shorter average paths. Notably, our method exhibited the highest success rate in new



(a) Frontier (b) DUDE-based

(c) ARiADNE (d) RASLS

Fig. 6. **Visual comparision of RASLS with baselines in an example scenario.** The figure depicts the actual trajectories of four algorithms in the case of successful exploration during experiments. Figures (a), (b), (c), and (d) correspond to the Nearest Frontier Selection algorithm, the Dual-space Decomposition (DUDE) [35] based algorithm, ARiADNE, and our proposed algorithm, respectively. The red stars represent the decision target points, and the blue lines depict the actual trajectory routes of the robot.

scenarios with similar object layouts. In trajectory plot as shown in Figure 6, we observed that ARiADNE's approach led to denser decision points, whereas our method involved decision points at longer and shorter distances. The Frontier method, selecting frontier points directly, resulted in widely dispersed decision target points, forming a zigzag path. In contrast, our method consistently completed exploration in local areas before expanding to explore new regions.

In coverage rate plot as shown in Figure 5, we presented coverage-enhanced data with decision steps. In cases of successful exploration, Frontier directly selected frontier points, achieving higher coverage with fewer decision steps. Each decision step contributed to changes in coverage. In contrast, ARiADNE required more decision steps to complete exploration in successful cases. Our method exhibited a coverage evolution similar to DUDE-based in decision steps while maintaining a higher success rate.

C. Ablation Experiment

In our ablation experiments, we removed the high-level object semantic dimension and layout matching reward. Observations from the collected training data reveal a significant reduction in the average exploration area when the goal semantics are absent, compared to our complete method. This discrepancy arises from the fact that, during training, the robot gains additional observations of semantics in failed scenarios, providing more information to the agent.

Collisions are related to the presence of goals in the scenario. Without semantics, the robot needs to learn additional obstacle avoidance strategies when a complete cost map is not established in the blind spot of the depth camera. Moreover, the observations for obstacle avoidance strategies are limited to the utility of frontier points unrelated to collision events, access flags, and the robot's current position. Due to these characteristics, the likelihood of exploration failure increases, thereby impacting the average exploration area per round.

Through these experimental findings, we demonstrate the positive impact of semantic layout on the learning process of the intelligent agent.

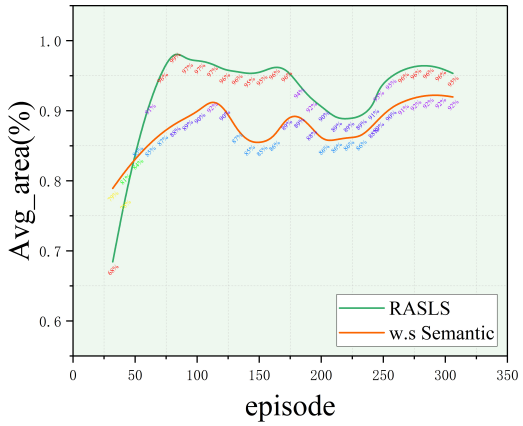


Fig. 7. **Average exploration area of each episode.** After removing semantic observations and rewards, we recorded each episode's average exploration area completion rate during the training process. The solid green line in the graph represents the performance of our method, while the solid orange line indicates the performance without semantic observations and rewards.

VI. Conclusion

In this study, we propose a reinforcement learning-based autonomous exploration method that incorporates semantic information about target layouts. Introducing target detection semantic information enables reinforcement learning methods to leverage prior knowledge more effectively, discovering potential layout placement patterns among objects during exploration. This approach aims to maximize the efficiency of the agent's exploration in scenarios with similar prior layouts. Through testing, we validate the assistance provided by semantic information in updating the agent's strategy. The proposed method exhibits higher exploration success rates and stable path trajectories in a new scenario with similar layouts. Our ablation experiments further demonstrate the enhancement of the agent's policy learning through high-level semantics.
During experimentation, we observed that precise localization significantly influences the mapping effectiveness of SLAM. Accumulated errors may introduce disturbances to the reinforcement learning process. We plan to introduce more accurate localization information to address these challenges

in future work. Additionally, the three-dimensional scale information of our object semantics is currently only utilized in the cost map during obstacle avoidance. Therefore, beyond two-dimensional obstacle avoidance tasks, there remains untapped potential for leveraging three-dimensional scale data. In future research, we aim to fully exploit three-dimensional data to broaden the agent's strategic perspective in the physical world.

REFERENCES

[1] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, L. Carlone, and J. A. Castellanos, "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *IEEE Transactions on Robotics*, 2023.

[2] 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.

[3] J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, "A survey of embodied ai: From simulators to research tasks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 2, pp. 230–244, 2022.

[4] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*. IEEE, 1997, pp. 146–151.

[5] M. Kulich, J. Faigl, and L. Přeučil, "On distance utility in the exploration task," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 4455–4460.

[6] S. Bai, J. Wang, F. Chen, and B. Englot, "Information-theoretic exploration with bayesian optimization," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1816–1822.

[7] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon" next-best-view" planner for 3d exploration," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1462–1468.

[8] S. Zhang and R. S. Sutton, "A deeper look at experience replay," *arXiv preprint arXiv:1712.01275*, 2017.

[9] G. Gemignani, R. Capobianco, E. Bastianelli, D. D. Bloisi, L. Iocchi, and D. Nardi, "Living with robots: Interactive environmental knowledge acquisition," *Robotics and Autonomous Systems*, vol. 78, pp. 1–16, 2016.

[10] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 915–926, 2008.

[11] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza, "A comparison of volumetric information gain metrics for active 3d object reconstruction," *Autonomous Robots*, vol. 42, no. 2, pp. 197–208, 2018.

[12] V. Suriani, S. Kaszuba, S. R. Sabbella, F. Riccio, and D. Nardi, "S-ave: Semantic active vision exploration and mapping of indoor environments for mobile robots," in *2021 European Conference on Mobile Robots (ECMR)*. IEEE, 2021, pp. 1–8.

[13] J. Jiang, J. Xu, J. Zhang, and S. Chen, "Deep reinforcement learning with new-field exploration for navigation in detour environment," in *2021 6th IEEE International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2021, pp. 13–18.

[14] B. Yu, H. Kasaei, and M. Cao, "Frontier semantic exploration for visual target navigation," *arXiv preprint arXiv:2304.05506*, 2023.

[15] M. T. Spaan and N. Spaan, "A point-based pomdp algorithm for robot planning," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 3. IEEE, 2004, pp. 2399–2404.

[16] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[17] Z. Xu, D. Deng, and K. Shimada, "Autonomous uav exploration of dynamic environments via incremental sampling and probabilistic roadmap," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2729–2736, 2021.

[18] T. Dang, M. Tranzatto, S. Khattak, F. Mascarich, K. Alexis, and M. Hutter, "Graph-based subterranean exploration path planning using aerial and legged robots," *Journal of Field Robotics*, vol. 37, no. 8, pp. 1363–1388, 2020.

[19] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 610–617, 2019.

[20] L. Deng, W. Gong, and L. Li, "Multi-robot exploration in unknown environments via multi-agent deep reinforcement learning," in *2022 China Automation Congress (CAC)*. IEEE, 2022, pp. 6898–6902.

[21] R. Cimurs, I. H. Suh, and J. H. Lee, "Goal-driven autonomous exploration through deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 730–737, 2021.

[22] D. Zhu, T. Li, D. Ho, C. Wang, and M. Q.-H. Meng, "Deep reinforcement learning supervised autonomous exploration in office environments," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 7548–7555.

[23] 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, 2019.

[24] F. Chen, J. D. Martin, Y. Huang, J. Wang, and B. Englot, "Autonomous exploration under uncertainty via deep reinforcement learning on graphs," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 6140–6147.

[25] Y. Cao, T. Hou, Y. Wang, X. Yi, and G. Sartoretti, "Ariadne: A reinforcement learning approach using attention-based deep networks for exploration," *arXiv preprint arXiv:2301.11575*, 2023.

[26] Y. Dai, J. Wu, D. Wang *et al.*, "A review of common techniques for visual simultaneous localization and mapping," *Journal of Robotics*, vol. 2023, 2023.

[27] A. Asgharivaskasi and N. Atanasov, "Active bayesian multi-class mapping from range and semantic segmentation observations," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1–7.

[28] A. Asgharivaskasi and N. Atanasov, "Semantic octree mapping and shannon mutual information computation for robot exploration," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1910–1928, 2023.

[29] L. Ma, J. Stückler, C. Kerl, and D. Cremers, "Multi-view deep learning for consistent semantic mapping with rgb-d cameras," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 598–605.

[30] L. Zhang, L. Wei, P. Shen, W. Wei, G. Zhu, and J. Song, "Semantic slam based on object detection and improved octomap," *IEEE Access*, vol. 6, pp. 75 545–75 559, 2018.

[31] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4247–4258, 2020.

[32] M. Lodel, B. Brito, A. Serra-Gómez, L. Ferranti, R. Babuška, and J. Alonso-Mora, "Where to look next: Learning viewpoint recommendations for informative trajectory planning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4466–4472.

[33] J. Zhang, L. Tai, M. Liu, J. Boedecker, and W. Burgard, "Neural slam: Learning to explore with external memory," *arXiv preprint arXiv:1706.09520*, 2017.

[34] J. Redmon, "Darknet: Open source neural networks in c," http://pjreddie.com/darknet/, 2013–2016.

[35] H. Kim, H. Kim, S. Lee, and H. Lee, "Autonomous exploration in a cluttered environment for a mobile robot with 2d-map segmentation and object detection," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6343–6350, 2022.

[36] A. Rasouli and J. K. Tsotsos, "The effect of color space selection on detectability and discriminability of colored objects," 2017.

[37] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.