IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

# A Policy Model Based Efficient and Accurate Scene Recognition Method for Service Robot

Shaopeng Liu, *Member, IEEE*, Guanzhong Zhou, and Chao Huang, *Senior Member, IEEE*

***Abstract*—In domestic environments, assigning scene semantic labels (scene recognition) to each node of a topological semantic map is an important task. Given the limitations of current scene recognition methods in efficiency, and accuracy for service robot, this letter proposes a scene recognition method based on a policy model. Considering the similarity of images captured from the adjacent nodes and the low-quality image caused by the uncertain node position and observation direction of the robot, we develop a policy model using a deep Q-learning network (DQN). This model enhances accuracy and efficiency by deciding whether to (1) inherit the scene type from the preceding node without re-recognition or (2) adjust the robot's observation angle to capture a more informative image. A rule-based reward function integrated with a scene score model enables simultaneous learning of similarity assessment and viewpoint adjustment policies. Furthermore, a training strategy based on generated path is proposed to provide sufficient data for training the policy model. Extensive comparative experiments in simulated environments demonstrate that our method surpasses state-of-the-art approaches in both recognition accuracy and efficiency. Deployment on a mobile robot confirms its practical efficacy, achieving precise and efficient scene recognition across diverse real-world environments.**

***Index Terms*—Scene recognition, topological semantic map, robot active vision, DQN, service robot.**

## I. Introduction

SEMANTIC maps integrating scene- and object-level semantic labels into 2D topological representations [1], [2], [3], help robots for robust environmental understanding, particularly distinguishing functional spaces like kitchens from other rooms [4]. For service robots in domestic environments, scene semantics (e.g., bedroom, kitchen) constitute a critical component of a topological semantic map [5], [6]. The task of assigning semantic labels to each node in a topological semantic map is termed scene recognition [4], [7].

Rooted in deep learning paradigms, training deep network models on large-scale scene datasets (e.g., MIT-67 [8], SUN397 [9]) enables single-image scene recognition [10], which assumes high-quality input images with comprehensive scene information. However, low-quality images containing occlusions or incomplete fields of view may be captured due to uncertain topological node positioning and robot viewing angles, causing low scene recognition accuracy [11]. Although recognizing multiple images captured from various nodes [4], [12] or multi-view images from one node [13] improves the recognition accuracy to some extent, ignoring visual similarity between nodes leads to extensive and redundant image recognizing, which incurs inefficient scene recognition.

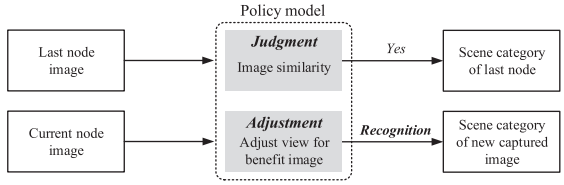


Fig. 1. The framework of proposed scene recognition method based on a policy model. Unlike conventional approaches, our framework decomposes the scene recognition task for each node into two sequential phases: (1) *judgment and adjustment*, and (2) *recognition*. *Judgment and adjustment* phase serves: (1) evaluating whether the image captured at the current node exhibits similarity to the last node's image to improve the efficiency of scene recognition and (2) adjusting the viewing angle based on observational context for more accurate scene recognition.

To address the problems of low accuracy and inefficient scene recognition, these challenges, we propose a scene recognition method based on a policy model (see in Fig. 1), which decomposes the scene recognition task for each node into two sequential phases: (1) *judgment and adjustment*, and (2) *recognition*. Based on the outcomes of the first phase, the second phase either employs a deep learning-based model for recognition or adopts an inheritance mechanism. Specifically, if image similarity is detected, the recognition result from the last node is inherited for the current node, thereby bypassing redundant model execution. Else, the image captured from the current node is recognized. When the current image is low-quality, viewpoint adjustment will occur and the modified image will be recognized to generate the scene recognition result for the current node.

Since the second phase constitutes a conventional image recognition task that is relatively straightforward to address, this paper primarily focuses on optimizing the first phase. The initial phase involves two interdependent tasks—*judgment* (similarity evaluation) and *adjustment* (view optimization)—which traditionally require separate network models. Such an approach, however, introduces redundant parameters and computational overhead. To resolve this inefficiency, we propose an unified policy model capable of jointly executing similarity judgment and view adjustment. This model, built on a deep Q-Learning

Received 2 July 2025; accepted 20 September 2025. Date of publication 29 September 2025; date of current version 14 October 2025. This article was recommended for publication by Associate Editor T. Horii and Editor T. Ogata upon evaluation of the reviewers' comments. *(Corresponding author: Chao Huang.)*

Shaopeng Liu is with the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China (e-mail: shaopeng.liu@polyu.edu.hk).

Guanzhong Zhou is with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China (e-mail: guanzhong.zhou@connect.polyu.hk).

Chao Huang is with the School of Electrical and Mechanical Engineering, The University of Adelaide, Adelaide, SA 5005, Australia (e-mail: chao.huang@adelaide.edu.au).

Digital Object Identifier 10.1109/LRA.2025.3615525

network (DQN), accepts paired inputs comprising images from the current and preceding nodes. Its outputs are discrete actions encoding similarity result and angular adjustment values. A scene score model and rules-based reward function is designed to provide action-specific feedback, guiding the policy model toward optimal action selection during training. Additionally, due to the absence of predefined training paths with nodes, we introduce a generative path-based training strategy to iteratively refine the policy model. The principal contributions of our work can be summarized as follows.

1) A DQN-based policy model is proposed, which simultaneously evaluates inter-node image similarity and adjusts view angle to enhance the efficiency and accuracy of scene recognition.
2) A rule-based reward function integrated with a scene score model and a training strategy based on generative path are designed to enable effective co-learning of judgment and adjustment policies of the policy model.
3) The proposed method is compared with the state-of-the-art methods in simulated environments, demonstrating its superior efficiency and accuracy over state-of-the-art methods. In addition, our method is deployed on a mobile robot in real-world environments, and its effectiveness is verified.

## II. Related Work

### A. Scene Recognition

Early methods of scene recognition [14], [15] focus on manually extracted features. Along with the emergence of Convolutional Neural Networks (CNNs) (e.g., ResNet [16], densenet [17], etc.) and large-scale scene datasets (e.g., MIT-67 [8], SUN397 [9]), deep learning based approaches [18], [19] have been a popular mode to address scene recognition problem. As a multi-class classification problem, indoor scene recognition [20] aims to label various functional areas by semantic information (e.g., dining room, kitchen). Several methods were developed based on object relationship [21], [22] or object-scene relationship [23] in an image to recognize scene. What they have in common is using single image for indoor scene recognition. However, during semantic mapping, a single image may contain occlusions or incomplete fields of view, leading to wrong scene recognition.

To improve the scene recognition accuracy, utilizing multiple images for indoor scene recognition can be combined with semantic mapping [4], [12], [13]. Niko et al. [12] applied CNN to process sequential multi-image inputs during mapping, refining results via Bayesian filtering. Similarly, Ygor et al. [4] extracted the deep visual features from multi-view images to recognize the scene. In addition, viewpoint optimization strategy, such as the reinforcement learning-based method by Jose et al. [13] to prioritize informative perspectives, enhance recognition accuracy during mapping. But it needs extra information (geometric knowledge, semantic knowledge, and laser data). Although these multi-image based approaches can improve the scene recognition accuracy to a certain extent, they ignore the similarity of the images captured from the adjacent nodes, reducing the recognition efficiency. Therefore, we propose a policy model based scene recognition method to enhance both the accuracy and efficiency of scene recognition.

### B. Deep Q-Learning Network

Deep Q-Learning Network (DQN), one of the most popular algorithms in reinforcement learning, was developed for single-agent environments by Google DeepMind [24]. Compared with basic Q-learning [25], DQN utilizes Convolution Neural Networks (CNN) as an approximation to fit the value function for each state. Combining the experience replay with target Q-network [26], DQN can eliminates correlations.

DQN has been a popular tool in robotics field, such as multi-robot management [27], [28], path planning [29], [30], and manipulation [31], [32]. However, the DQN-based models that can judge image similarity and adjust view simultaneously are less involved. Therefore, we need to design a novel DQN-based policy model for these two tasks as well as a suitable reward function. In addition, a training strategy based on generated path is necessary for the DQN-based policy model.

## III. Proposed Approach

### A. Problem Formulation

In a domestic environment, a robot traverses a predefined path $p$ comprising discrete nodes $O = \{o_1, o_2, \ldots, o_n\}$, where nodes are determined based on fixed time or distance intervals during the robot's motion. Image $i_x$ captured at $o_x$ is assigned a semantic label $Y$ (e.g., bathroom, bedroom, etc.) via a deep learning-based scene recognition model:

$$f_\theta(i_x) = Y, \tag{1}$$

where $\theta$ represents the model parameters.

To mitigate redundancy, the current node $o_x$ and its previous node $o_{x-1}$ may inherit labels if their images $i_x$ and $i_{x-1}$ exhibit similarity. A binary similarity judgment strategy

$$\pi_s(i_{x-1}, i_x) = \{yes, no\} \tag{2}$$

is defined. If $\pi_s$ returns $no$ at $o_x$, $i_x$ is recognized by $f_\theta$. Otherwise, $o_x$ inherit the label of $o_{x-1}$.

To address suboptimal image quality (e.g., occlusions, incomplete fields of view) caused by the uncertain node position and observation direction of the robot, a view adjustment strategy $\pi_v$ optimizes camera angles to capture discriminative images. Considering that the images captured from the similar angles have little differences, a discretized action space

$$A_v = \{-60^\circ, -30^\circ, 0^\circ, 30^\circ, 60^\circ\} \tag{3}$$

is set for $\pi_v$, where negative angles indicate counterclockwise rotation and positive angles denote clockwise rotation. After executing an action $a_x \in A_v$ at $o_x$, the new captured image $i'_x$ is obtained, expressed by a transition function:

$$i'_x = T(o_x, a_x). \tag{4}$$

The recognition result of feeding $i'_x$ into $f_\theta$ is the label of $o_x$.

Since separating implementations of $\pi_s$ and $\pi_v$ using distinct neural networks introduce parameter redundancy and training complexity, we unify them into a single action strategy:

$$\pi(i_{x-1}, i_x) = \{-60^\circ, -30^\circ, 0^\circ, 30^\circ, 60^\circ, yes\} \tag{5}$$

approximated by a DQN-based model $Q_\beta$. This model optimizes an action-value function

$$Q_\beta(i_{x-1}, i_x) \rightarrow \pi(i_{x-1}, i_x), \tag{6}$$

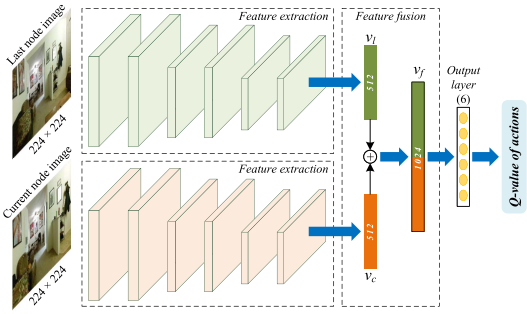


Fig. 2. The architecture of DQN-based policy model with the input $224 \times 224$ images. The two feature extraction module have the same structure based on ResNet-18 backbone [16], excluding its final fully connected layer. This module outputs two 512-dimensional feature vectors $v_l$ (last node) and $v_c$ (current node). These vectors are concatenated to form a 1,024-dimensional fusion feature $v_f$. The output layer comprises six neurons to generate Q-values for the corresponding actions.

where $\beta$ denotes the parameters learned through reinforcement learning. The final scene label $\phi(o_x)$ is determined as

$$\phi(o_x) = \begin{cases} f_\theta(T(o_x, Q_\beta(i_{x-1}, i_x))), & Q_\beta(i_{x-1}, i_x) \neq yes \\ \phi(o_{x-1}), & Q_\beta(i_{x-1}, i_x) = yes \end{cases}. \tag{7}$$

### B. DQN-Based Policy Model

*1) Model Architecture:* As shown in Fig. 2, the DQN-based policy model implements $Q_\beta$, accepting two RGB images captured at the last and current nodes as inputs. The policy model will generate Q-values for the six discrete actions defined in (5).

*2) Reward Function:* A novel reward function is proposed for training the DQN-based policy model in the Active Vision Dataset Benchmark (AVDB) [33], which is summarized in Algorithm 1. The reward function integrates a scene score model (SSM) $f_s(i)$ with heuristic rules to incentivize label inheritance from the preceding node when appropriate. The SSM architecture employs a ResNet-18 pre-trained on ImageNet dataset [34], which has a stronger discrimination ability for images. As ImageNet dataset contains 1,000 categories of objects, the ResNet-18 can only recognize object images rather than scene images. Therefore, to recognize the needed five categories of scenes (bathroom, bedroom, kitchen, living room, dining room) in AVDB, SSM is fine-tuned using corresponding scene image subsets from MIT-67 [8] and SUN397 [9]. The trained SSM generates scene category score, where higher values reflect greater prediction confidence.

As outlined in Algorithm 1 (lines 1-4), if the preceding node's label matches the current node's ground truth and the preceding node's score exceeds 0.5, the agent receives a positive reward of 2 for selecting action $yes$ to inherit label $l_{x-1}$. Otherwise, the agent executes view adjustment actions to capture a modified image $i'_x$ (line 6). Then $i'_x$ is processed by SSM to yield the scene category $l_x$ and score $s_x$ for the current node. Even if $l_x$ is correct (lines 8-9), the agent receives half the maximum reward (1) to reflect suboptimal action selection compared to $yes$. When the adjacent nodes have differing scene types or the preceding node's confidence is below 0.5 (lines 12-25), the agent should prioritize scene recognition over inheriting potentially inaccurate labels. Even if the agent achieves a correct recognition by selecting action $yes$, it only obtains a reduced reward (1). A full reward of 2 is awarded only when view adjustments yield correct recognition.

**Algorithm 1:** Reward Function.

**Input:** Scene score model $f_s(i)$, current node image $i_x$, ground truth of current node $\hat{l_x}$, last node label $l_{x-1}$, scene score $s_{x-1}$ of $l_{x-1}$, and current node action $a_x$.
**Output:** Reward $r_{a_x}$ of $a_x$.

```
1  if l_{x-1} = l̂_x and s_{x-1} ≧ 0.5 then
2      if a_x = yes then
3          l_x = l_{x-1} // Scene recognition result of current node
4          r_{a_x} = 2
5      else
6          i'_x = T(i_x, a_x)
7          l_x, s_x = f_s(i'_x), s_{x-1} = s_x
8          if l_x = l̂_x then
9              r_{a_x} = 1
10         else
11             r_{a_x} = 0
12 else
13     if a_x = yes then
14         l_x = l_{x-1}
15         if l_x = l̂_x then
16             r_{a_x} = 1
17         else
18             r_{a_x} = 0
19     else
20         i'_x = T(i_x, a_x)
21         l_x, s_x = f_s(i'_x), s_{x-1} = s_x
22         if l_x = l̂_x then
23             r_{a_x} = 2
24         else
25             r_{a_x} = 0
26 return r_{a_x}
```

### C. Training Strategy Based on Generated Path

Training the DQN-based policy model requires paths containing scene recognition nodes. However, existing datasets lack such paths. To resolve this, we utilize AVDB [33] to synthetically generate training paths. AVDB provides numerous image-scanned points across diverse domestic environments. Each point comprises 12 images captured at 30° intervals, simulating robotic view adjustments for image acquisition. Starting from a selected image-scanned point, paths are constructed by sequentially connecting adjacent points. These points within a path are designated as scene recognition nodes. Paths containing at least four nodes are employed to train the policy model via the propose training strategy detailed in Algorithm 2. In each epoch (lines 4-24), the agent goes through the nodes from all

**Algorithm 2:** Training Strategy Based on Generated Path.

**Input:** The generated $P = \{p_1, p_2, .., p_i\}$, scene score model $f_s(i)$, random number generator $Random(0, 1)$ from 0 to 1, and empty buffer $\mathbb{S}$.
**Output:** Well trained DQN-based policy model $Q_\beta$.

1 Initialise a evaluation $Q^e_\beta$ and a target $Q^t_\beta$ with random parameters
2 $N_s = 0$ // step count
3 **for** $m \leftarrow 1$ *to* $M$ *epoch* **do**
4  **for** $p$ *in* $P$ **do**
5  $p = \{o_1, o_2, ..., o_n\}$ // nodes in $p$
6  $N_p = 0$ // node count in $p$
7  **for** $o_x$ *in* $p$ **do**
8  $N_p = N_p + 1, N_s = N_s + 1$
9  The initial image $i_x$ of $o_x$
10  **if** $N_p = 1$ **then**
11  $l_{o_{x\ 1}}, s_{o_{x\ 1}} = f_s(i_x)$ // previous node label $l_{o_{x\ 1}}$ and previous scene score $s_{o_{x\ 1}}$
12  **if** $N_s > 200$ *and* $Random(0, 1) < 0.9$ **then**
13  $a_{o_x} = Q^e_\beta(i_{x-1}, i_x)$
14  **else**
15  Randomly choose an action $a_{o_x}$ from $\{-60°, -30°, 0°, 30°, 60°, yes\}$
16  $i'_x = T(i_x, a_{o_x})$
17  Get reward $r_{a_{o_x}}$ based on Algorithm 1
18  Store $\{i_{x-1}, i_x, a_{o_x}, r_{a_{o_x}}, i'_x\}$ in $\mathbb{S}$
19  **if** $N_s > 100$ **then**
20  Sample batch $b_s$ from $\mathbb{S}$ randomly
21  Use $b_s$ to train $Q^e_\beta$ using Adam optimizer
22  **if** $N_s \% 400 = 0$ **then**
23  $Q^t_\beta = Q^e_\beta$
24 return $Q^t_\beta$

the generated paths. For the first node, the label and scene score of previous node is the same as the current node (lines 11-12). The agent tries an action to obtain a reward based on Algorithm 1, forming the five elements $\{i_{x-1}, i_x, a_{o_x}, r_{a_{o_x}}, i'_x\}$ stored in the buffer $\mathbb{S}$ (lines 13-19). When step $N_s > 100$, sample batch $b_s$ from $\mathbb{S}$ randomly to train $Q^e_\beta$ (lines 20-22). Experience replay is introduced every 400 steps (lines 23-24).

## IV. Experimental Setup

### A. Simulated Environments and Setup

We employ AVDB [33] as simulated environments to generate the path for training and testing the proposed method and baseline approaches. AVDB comprises five distinct scene types: bathroom, bedroom, kitchen, living room, and dining room. To enable consistent evaluation of recognition performance across all scene categories, we ensure both training and testing environment groups contain examples from all five scenes. Specifically, $Home\_2, 4, 6, 10, 16$ are selected to generate training paths, while $Home\_5, 7, 8, 14, 15$ serve as testing environments. The quantitative distribution of paths across the environments is illustrated in Fig. 3.

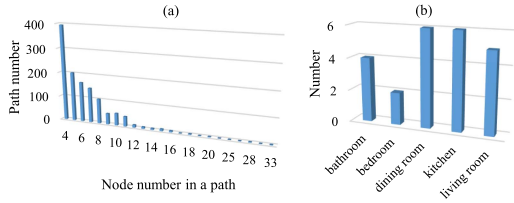


Fig. 3. Quantitative distribution of paths: (a) count of path that contains different node number, and (b) number of various scenes.

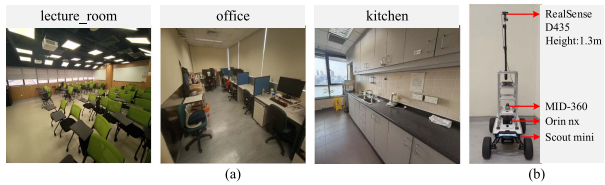


Fig. 4. Three real-world experimental environments: lecture_room, office, and kitchen (a) and the mobile robot: Scout mini.

The simulation experiments were conducted on a server equipped with an Intel Core i7-13700 K CPU, 64 GB memory, 1 TB SSD storage, and an NVIDIA GeForce RTX 4060 Ti GPU. Our method[1] was implemented in Python using PyTorch deep learning framework, running on a Linux-based Ubuntu 18.04 operating system.

### B. Real-World Environments and Setup

As illustrated in Fig. 4(a), the experiments were conducted in three real-world environments—a lecture_room, an office, and a kitchen—to validate the practical performance of our method on topological semantic mapping. The mobile robot platform, depicted in Fig. 4(b), integrates the following components: an Intel RealSense D435 RGB-D camera, a MID-360 3D LiDAR, an NVIDIA Jetson Orin NX onboard computer, and a Scout Mini mobile chassis. The D435 camera, mounted at 1.3 m height, captures RGB images at 224 × 224 pixel resolution. The MID-360 LiDAR generates 3D environmental maps annotated with nodes of scene labels. The policy model and scene recognition model, implemented in Python, are executed on the Jetson Orin NX using Ubuntu 20.04, PyTorch, and ROS 2 Humble. Notably, the policy model architecture remains consistent with that used in simulation experiments. To enhance method extensibility, we employ the Places365-ResNet50 model [35], enabling recognition across broader scene categories. The Scout Mini chassis dynamically adjusted its orientation based on angular outputs from the policy model.

Fig. 9 demonstrates the experimental configuration, where each environment contains an evaluation path comprising multiple scene recognition nodes. Node counts scales with environment size, yielding 17 total evaluation nodes across all environments. All paths originates at node ①. At each node, the robot executes the policy model to determine subsequent movements based on initial visual observations from the RGB-D camera. The recognition result of each node is labeled on the 3D LiDAR map to form a topological semantic map.

[1]https://sites.google.com/view/asrmtsm/

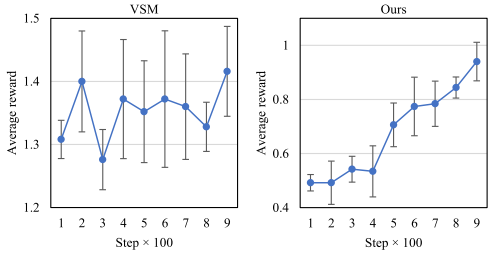


Fig. 5. The average reward curves of VSM and Ours using five different seeds, where the line denotes the average reward of every 100 steps and the error bar refers to the standard deviation.

TABLE I
THE TRAINING PARAMETERS OF SRM, VSM AND OURS

| Method | SRM | VSM | Ours |
|---|---|---|---|
| Image size | $224 \times 224$ | $224 \times 224$ | $224 \times 224$ |
| Batch size | 32 | 16 | 16 |
| Learning rate | $3^{-5}$ | $2.5^{-4}$ | $2.5^{-4}$ |
| Memory size | - | 200 | 200 |
| Replace target iterations | - | 400 | 400 |
| Optimizer | $Adam$ | $Adam$ | $Adam$ |

### C. Evaluation Metrics

We evaluate method performance using two quantitative metrics. First, standard recognition accuracy ($acc$) as defined in prior work [12], [13]:

$$acc = \frac{n_c}{n_{all}}, \tag{8}$$

where $n_c$ denotes correctly recognized nodes, and $n_{all}$ represents the total node count. Second, to quantificationally express how often the scene recognition model is used, we introduce an efficiency metric called recognition rate ($re$):

$$re = \frac{Num(f_\theta(i_x))}{n_c}, \tag{9}$$

where $Num(f_\theta(i_n))$ quantifies the number of scene recognition model executions among $n_c$. $re < 1$ indicates that the execution of scene recognition model is not required for every node. Lower $re$ corresponds to greater efficiency, as fewer model executions achieve equivalent recognition performance. In addition, model computational time at each node ($mct$) is recorded as a metric of scene recognition efficiency.

### D. Baseline Methods and Implementation Details

We evaluate our method against three baseline methods, implemented as follows:

1) *SRM:* The common solution is to train a deep network on scene datasets to recognize scene. A scene recognition model (SRM) based on Vision Transformer [36], initially pretrained on ImageNet [34], serves as the baseline scene classifier. SRM is fine-tuned on 8,711 images from MIT-67 [8] and SUN397 [9] to recognize our five target scene categories using the parameters in Table. I. During training, SRM is saved and tested every epoch. The model achieving peak testing accuracy is selected as the baseline result of SRM.
2) *SRM-Bayes:* We extend the idea of [12] that uses a Bayesian filter to process the recognition results of the scene images from the adjacent nodes for determining the final scene category. Considering that the scene recognition model used in [12] is AlexNet, an early CNN, we leverage the same SRM to replace the AlexNet for a fair comparison.
3) *VSM:* Following the tactics presented in [13], a view selection model (VSM) based on DQN is used as a contrast. At each node, VSM can rotate view for better recognition. SRM is leveraged to recognize the initial image and the new captured image. Then the recognition results of these

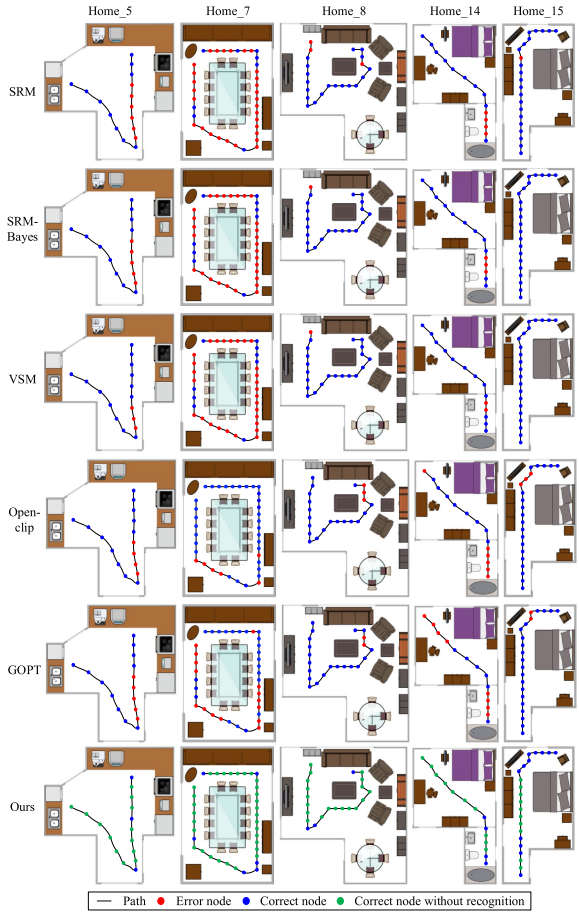


Fig. 6. Some path instances from testing environments, including the scene recognition result of each node by baseline methods and ours. The error node means the wrong result obtain by the scene recognition model. The correct node indicates the scene recognition model outputs right scene category of the node. The correct node without recognition denotes the right result of the node obtained by inheriting from the preceding node instead of running the scene recognition model.

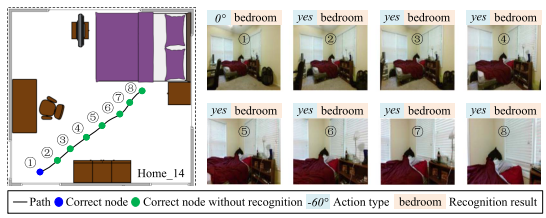


Fig. 7. Demo of scene recognition processes using our proposed method. The left part is the paths with nodes. The right part shows the initial observed image of each node, the output of the policy model, the captured image after performing the action, and the scene recognition result of every node.

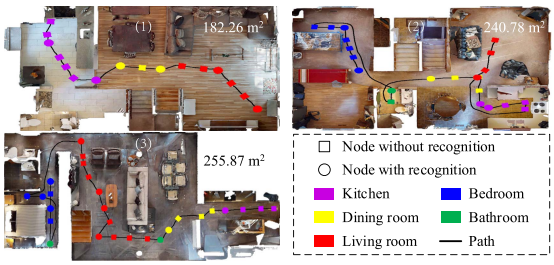


Fig. 8. Demos of scene recognition in three large and complex indoor environments: (1) 00847, (2) 00877, and (3) 00843 from HM3D using our method, including floor space, whether the scene recognition model is used and the recognition results.

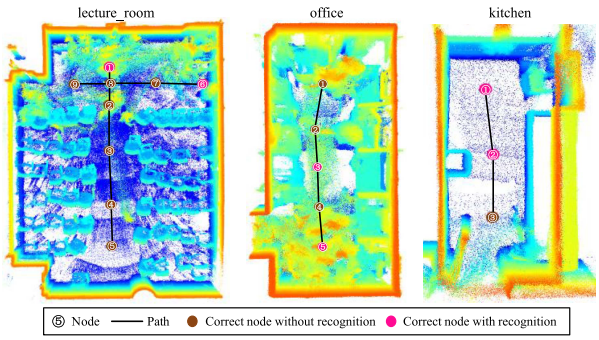


Fig. 9. The built topological semantic maps of lecture_room, office, and kitchen using three-dimensional laser, including nodes of scene recognition, path, and recognition results.

two images are fused by the same Bayesian filter in [12] to output the scene category of the node. We utilize an 18-layer ResNet [16] to build VSM. As there are five actions $\{-30°, -15°, 0°, 15°, 30°\}$ in [13], a new fully connected layer containing five neurons is added in VSM. The negative values indicate counterclockwise rotation and positive values are clockwise rotation. The reward function is based on the recognition result of new captured image after rotation. If the recognition result is correct, the reward is 2. Else, the reward is 0. VSM is trained five times using the parameters in Table I with different random seeds (5, 10, 15, 20, 25). The average reward curve is shown in Fig. 5. During the training, we save checkpoints every 25 steps and select the highest $acc$ as the result of VSM.

4) *VLM:* Well trained vision-language model (VLM) can output the semantic information of the image including scene category by zero-shot way. We leverage two state-of-the-art VLMs (Open-vlip [37] and GOPT [38]) to recognize scene by inputting both an image and a prompt of the five scene labels. Then Open-vlip and GOPT can output the scene recognition result.

Our proposed method (Ours) includes the DQN-based policy model and a scene recognition model to recognize scene at each node. This scene recognition model is replaceable according to the category of the target scene. As AVDB has five types of scenes, ours uses the same model as SRM for scene recognition. When recognizing more types of scenes, we will employ a well-trained Places365-ResNet50 model [35] that can recognize 365 categories of scenes. The DQN-based policy model is trained under identical seed conditions as VSM. The training parameters are given in Table I. For accelerating network convergence, all network parameters including those in the feature extraction modules are not fixed during training. The training progression (see in Fig. 5) demonstrates stable reward convergence. We select the optimal checkpoint (every 25 steps) through joint evaluation of the policy model and SRM performance as the final result of ours.

TABLE II
THE TESTING RESULTS OF THE BASELINE METHODS AND OUR APPROACH

| Model | Home $acc$ | | | | | Avg | Avg | Avg |
|---|---|---|---|---|---|---|---|---|
| | 5 | 7 | 8 | 14 | 15 | $acc$ | $re$ | $mct(ms)$ |
| SRM | 0.7725 | 0.2878 | 0.7794 | 0.7700 | 0.8383 | 0.7329 | 1.0000 | 5.3735 |
| SRM-Bayes | 0.8188 | 0.3284 | 0.8277 | 0.8217 | 0.8717 | 0.7769 | 1.0000 | 5.3743 |
| VSM | 0.8957 | 0.4059 | 0.8655 | 0.8553 | 0.8643 | 0.8196 | 1.1772 | 8.2897 |
| Open-clip | 0.8638 | 0.7159 | 0.8803 | 0.8502 | 0.9498 | 0.8675 | 1.0000 | 231.41 |
| GOPT | 0.7203 | 0.5129 | 0.8278 | 0.6305 | 0.8086 | 0.7235 | 1.0000 | 130.17 |
| Ours | 0.8942 | 0.8339 | 0.8992 | 0.8372 | 0.9294 | **0.8870** | **0.4018** | **5.0752** |

## V. RESULTS AND ANALYSIS

### A. Results in Simulated Environments

*1) Quantitative Results and Analysis:* The comparative results of baseline methods and our proposed approach are summarized in Table II. Given the inherent uncertainty in node image quality, sole reliance on single-image recognition (SRM) yields the low $acc$ across the test environments, with particularly severe degradation in $Home\_7$ (0.2878). SRM-Bayes demonstrates 4.4% accuracy improvements over SRM through Bayesian integration of adjacent node information, confirming the utility of node context in scene recognition. While VSM achieves competitive accuracy (average $acc$ 0.8196) via active view adjustment, its elevated $re > 1.1$ reveals substantial computational overhead from multi-image processing, resulting in low efficiency of scene recognition. Although Open-clip and GOPT can recognize scenes directly without any extra training, they obtain lower $acc$ than ours. Especially, Open-clip and GOPT including too many network parameters leads to bigger $mct$ as well as inefficient scene recognition.

Our method establishes new state-of-the-art performance, surpassing all baselines in average $acc$. The consistent superiority across environments ($acc > 0.83$) demonstrates robust adaptability, contrasting with baseline methods' significant performance variance (e.g., SRM's 0.55 $acc$ gap between $Home\_15$ and $Home\_7$). The baseline methods need to use the scene recognition model as least once at each node. In comparison, our approach maintains computational efficiency ($re$ only 0.4018) through selective execution of scene recognition model, requiring less computational load but more accuracy. This is also reflected in the aspect of model latency. Our method achieves the smallest average $mct$ (5.0752 ms) in Table II, proving its advantages on scene recognition efficiency.

*2) Qualitative Results and Analysis:* Fig. 6 illustrates the superior performance of our method through representative path instances. SRM exhibits the highest error node count across all environments, reflecting its limited scene discrimination capability. While SRM-Bayes and VSM demonstrate moderate improvements in $Home\_5, 8, 14, 15$, both fail to recognize plenty of nodes in $Home\_7$. In contrast, our method achieves consistent accuracy across simple ($Home\_15$) and complex ($Home\_7$) paths. Notably, there are only small number of scene recognition model executions among correct nodes achieved

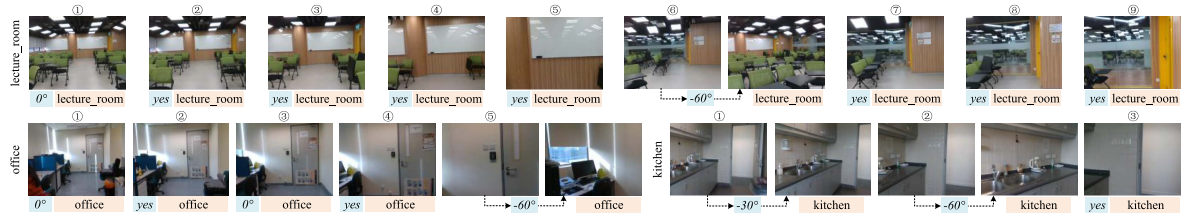


Fig. 10. The scene recognition process of each node, including the initial observed image, output action (aqua block), and scene result (orange block).

TABLE III
ABLATION STUDY RESULTS ABOUT POLICY MODEL AND SCENE RECOGNITION MODEL

| Type | Home $acc$ | | | | | Avg | Avg |
|---|---|---|---|---|---|---|---|
| | 5 | 7 | 8 | 14 | 15 | $acc$ | $re$ |
| No action + SRM | 0.7725 | 0.2878 | 0.7794 | 0.7700 | 0.8383 | 0.7329 | 1.0000 |
| Random action + SRM | 0.7681 | 0.3579 | 0.7332 | 0.6822 | 0.7751 | 0.7015 | 0.8429 |
| PM + Places365-ResNet50 | 0.8710 | 0.8745 | 0.8655 | 0.8269 | 0.8197 | 0.8514 | 0.4018 |
| Ours (PM + SRM) | 0.8942 | 0.8339 | 0.8992 | 0.8372 | 0.9294 | **0.8870** | **0.4018** |

by our approach. The designed policy model generating the inheriting strategy improves the efficiency.

To demonstrate the scene recognition process of the proposed method concretely, Fig. 7 gives a demos. The first path is in a bedroom. At node ①, the policy model retains the original view (0°) due to sufficient visual cues (bed, nightstand). For node ②, spatial similarity to node ① triggers label inheritance without scene recognition model execution. Subsequent nodes (③–⑧) employ the same recognition strategies, achieving correct results.

Moreover, our method has been evaluated on Habitat-Matterport 3D Research Dataset (HM3D) [39] that contains houses with many rooms and large venues. Places365-ResNet50 [35] is employed to replace SRM since the scene category in HM3D is beyond the five scenes in AVDB. Among the 64 nodes in Fig. 8, our method obtains 0.9063 $acc$ and 0.4138 $re$, indicting its scalability and broader applicability in large and complex environments.

*3) Ablation Study:* To justify the performance of our method, the ablation study about the policy model and the scene recognition model has been done in AVDB. We use two action strategies (no action and random action) to replace our DQN-based policy model (PM) for testing respectively. In addition, we employ Places365-ResNet50 model instead of SRM for scene recognition.

The results of ablation study are shown in Table III. Compared with no action and random action, our method including PM achieves the highest $acc$ and lowest $re$ in Table III, indicating that PM helps to improve the accuracy and efficiency of scene recognition. When assessing our method in AVDB, choosing SRM rather than Places365-ResNet50 for scene recognition can obtain a higher $acc$ (see in Table III) since SRM has better recognition ability on the five types of scenes in AVDB. Therefore, the design of our method is effective and reasonable.

### B. Results in Real-World Environments

What stands out from Fig. 9 is that our method successfully recognize the scene of each node and achieves 0.3529 $re$, underscoring its accuracy and computational efficacy for topological semantic mapping. More importantly, the policy model—trained exclusively in simulation environments lacking the three target scene types (lecture_room, office, kitchen)—performs effectively in real-world settings, highlighting the method's robust transferability.

To show how the robot accomplishes the scene recognition at each node, detailed processes are illustrated in Fig. 10. At node ① in the lecture room, the initial observation (image ①) contains sufficient discriminative features, enabling the robot to classify the scene immediately via a 0° rotation. At node ②, the policy model leverages visual similarity between image ① and image ② to inherit the prior node's result, bypassing redundant scene recognition computations while preserving accuracy. This inheritance strategy persists at node ③-⑤ and ⑥-⑨. Crucially, when significant visual discrepancies arise between adjacent nodes (e.g., images ⑤ and ⑥), the policy model triggers re-identification through adaptive rotational adjustments (e.g., $-60°$), ensuring robust scene recognition.

Overall, these experiments confirm that our simulation-trained model transitions seamlessly to real-world deployment without fine-tuning. By balancing accuracy, computational efficiency, and adaptability to environmental variations, our method proves both practically viable and architecturally portable for topological semantic mapping applications.

## VI. CONCLUSION AND DISCUSSION

In this study, we proposed a policy model based scene recognition method for service robots to enhance topological semantic mapping through efficient and accurate node-level scene recognition. We developed a DQN-based policy model to dynamically determine whether to inherit the scene label from the prior node or adjust the robot's observation angle to capture more discriminative image for recognition. A dual-purpose reward function—combining a scene score model with rules—enabled simultaneous learning of visual similarity assessment and view adjustment. We implemented a path-generation approach and training strategy that synthesizes diverse trajectories including sufficient data to train the policy model. Extensive comparative experiments validated our method's superiority over state-of-the-art approaches in terms of accuracy and efficiency. Crucially, real-world deployments on a mobile robot confirmed the practicality of the proposed method for constructing topological semantic maps.

While our method exhibits strong overall performance, minor limitations emerge in spatially constrained environments. For example, in the office (node ③ of Fig. 10), the model prioritizes accuracy over computational efficiency by executing fresh recognition rather than inheriting prior results, marginally increasing $re$. Our future work will focus on further enhancing the scene recognition efficiency by enabling the policy model to predict the scene label of next node in advance.

## REFERENCES


[1] K. Zheng, A. Pronobis, and R. Rao, "Learning graph-structured sum-product networks for probabilistic semantic maps," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 4547–4555.

[2] C. Gomez et al., "Hybrid topological and 3D dense mapping through autonomous exploration for large indoor environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 9673–9679.

[3] A. Taniguchi, S. Ito, and T. Taniguchi, "Hierarchical path planning from speech instructions with spatial concept-based topometric semantic mapping," *Front. Robot. AI*, vol. 11, 2024, Art. no. 1291426.

[4] Y. C. N. Sousa and H. F. Bassani, "Topological semantic mapping by consolidation of deep visual features," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4110–4117, Apr. 2022.

[5] J. R. Ruiz-Sarmiento, C. Galindo, and J. González-Jiménez, "Robot@home, a robotic dataset for semantic mapping of home environments," *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 131–141, 2017.

[6] A. C. Hernandez, C. Gomez, R. Barber, and O. M. Mozos, "Exploiting the confusions of semantic places to improve service robotic tasks in indoor environments," *Robot. Auton. Syst.*, vol. 159, 2023, Art. no. 104290.

[7] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva, "SUN database: Exploring a large collection of scene categories," *Int. J. Comput. Vis.*, vol. 119, pp. 3–22, 2016.

[8] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 413–420.

[9] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to Zoo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3485–3492.

[10] L. Xie, F. Lee, L. Liu, K. Kotani, and Q. Chen, "Scene recognition: A comprehensive survey," *Pattern Recognit.*, vol. 102, 2020, Art. no. 107205.

[11] S. Liu, G. Tian, Y. Zhang, and P. Duan, "Scene recognition mechanism for service robot adapting various families: A CNN-based approach using multi-type cameras," *IEEE Trans. Multimedia*, vol. 24, pp. 2392–2406, 2022.

[12] N. Sünderhauf et al., "Place categorization and semantic mapping on a mobile robot," in *Proc. IEEE Int. Conf. Robot. Automat*, 2016, pp. 5729–5736.

[13] J. L. Matez-Bandera, J. Monroy, and J. Gonzalez-Jimenez, "Efficient semantic place categorization by a robot through active line-of-sight selection," *Knowl.-Based Syst.*, vol. 240, 2022, Art. no. 108022.

[14] E. Fazl-Ersi and J. K. Tsotsos, "Histogram of oriented uniform patterns for robust place recognition and categorization," *Int. J. Robot. Res.*, vol. 31, no. 4, pp. 468–483, 2012.

[15] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[17] G. Huang, Z. Liu, V. Van Der Laurens, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.

[18] M. Dixit, Y. Li, and N. Vasconcelos, "Semantic fisher scores for task transfer: Using objects to classify scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 12, pp. 3102–3118, Dec. 2020.

[19] Q. Wang, F. Zhu, G. Wu, P. Zhao, J. Wang, and X. Li, "Object-level and scene-level feature aggregation with CLIP for scene recognition," *Inf. Fusion*, vol. 120, 2025, Art. no. 103118.

[20] P. Uršič, R. Mandeljc, A. Leonardis, and M. Kristan, "Part-based room categorization for household service robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 2287–2294.

[21] R. Pereira, L. Garrote, T. Barros, A. Lopes, and U. J. Nunes, "A deep learning-based indoor scene classification approach enhanced with inter-object distance semantic features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 32–38.

[22] S. Choe, H. Seong, and E. Kim, "Indoor place category recognition for a cleaning robot by fusing a probabilistic approach and deep learning," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 7265–7276, Aug. 2022.

[23] B. Zhu, X. Fan, X. Gao, G. Xu, and J. Xie, "A heterogeneous attention fusion mechanism for the cross-environment scene classification of the home service robot," *Robot. Auton. Syst.*, vol. 173, 2024, Art. no. 104619.

[24] T. Hester et al., "Deep Q-learning from demonstrations," in *Proc. AAAI Conf. Artif. Intell.*, pp. 3223–3230.

[25] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, pp. 279–292, 1992.

[26] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[27] P. Gautier, J. Laurent, and J.-P. Diguet, "Deep Q-learning-based dynamic management of a robotic cluster," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 4, pp. 2503–2515, Oct. 2023.

[28] E. Latif and R. Parasuraman, "Communication-efficient multi-robot exploration using coverage-biased distributed Q-learning," *IEEE Robot. Automat. Lett.*, vol. 9, no. 3, pp. 2622–2629, Mar. 2024.

[29] S. Liu, G. Tian, Y. Zhang, M. Zhang, and S. Liu, "Active object detection based on a novel deep Q-learning network and long-term learning strategy for the service robot," *IEEE Trans. Ind. Electron.*, vol. 69, no. 6, pp. 5984–5993, Jun. 2022.

[30] A. Kumar et al., "DSQN: Robust path planning of mobile robot based on deep spiking Q-network," *Neurocomputing*, vol. 634, 2025, Art. no. 129916.

[31] S. Joshi, S. Kumra, and F. Sahin, "Robotic grasping using deep reinforcement learning," in *Proc. Int. Conf. Automat. Sci. Eng.*, 2020, pp. 1461–1466.

[32] I. Sarantopoulos, M. Kiatos, Z. Doulgeri, and S. Malassiotis, "Split deep q-learning for robust object singulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 6225–6231.

[33] P. Ammirato, P. Poirson, E. Park, J. Košecká, and A. C. Berg, "A dataset for developing and benchmarking active vision," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 1378–1385.

[34] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, pp. 211–252, 2015.

[35] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.

[36] A. Dosovitskiy et al., "An image is worth 16 x 16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021.

[37] D. Bolya et al., "Perception encoder: The best visual embeddings are not at the output of the network," 2025, *arXiv:2504.13181*.

[38] M. Tschannen et al., "SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features," 2025, *arXiv:2502.14786*.

[39] S. K. Ramakrishnan et al., "Habitat-Matterport 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI," 2021, *arXiv:2109.08238*.