2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Deep Reinforcement Learning Supervised Autonomous Exploration in Office Environments

Delong Zhu*, Tingguang Li*, Danny Ho*, Chaoqun Wang, Max Q.-H. Meng

***Abstract*— Exploration region selection is an essential decision making process in autonomous robot exploration task. While a majority of greedy methods are proposed to deal with this problem, few efforts are made to investigate the importance of predicting long-term planning. In this paper, we present an algorithm that utilizes deep reinforcement learning (DRL) to learn exploration knowledge over office blueprints, which enables the agent to predict a long-term visiting order for unexplored subregions. On the basis of this algorithm, we propose an exploration architecture that integrates a DRL model, a next-best-view (NBV) selection approach and a structural integrity measurement to further improve the exploration performance. At the end of this paper, we evaluate the proposed architecture against other methods on several new office maps, showing that the agent can efficiently explore uncertain regions with a shorter path and smarter behaviors.**

I. Introduction

In autonomous exploration task, the agent is expected to find an optimal path with some constraints, such as reducing path cost, increasing the accuracy of map-building or both [1]. However, due to the uncertainty of unexplored regions, it is impossible to formulate an exact global optimization function to solve this problem, so a majority of methods adopt greedy strategy as an alternative solution [2], without any consideration of future planning. Another category of methods regards this problem as a Markov Decision Process (MDP), which seeks to reduce future cost by dynamic decision making. However, suffering from the curse of dimensionality, this type of method shows some significant shortcomings like limited generalization ability, poor convergence property, and high computational cost. In this paper, the potential of MDP based Deep Reinforcement Learning (DRL) method in the context of autonomous exploration is re-examined on account of its significant achievement in AI field [3]. We want to verify whether this technique is applicable to some particular exploration environments, e.g. office buildings, and how it can be applied to these environments in a proper manner.

The whole picture of our idea is shown in Fig.1, which comes from our observation of human behaviors. We notice that when people enter an unknown building, a rough prediction about the layout will first come into their mind according to former experience; directed by this prediction, people attempt to maximize their vision field at each step and gradually refine the prediction step by step, until achieving their goals. Related biological experiments are also conducted by Tolman [4], which illustrates that even rates build cognitive map using experience, i.e. a kind of sophisticated spatial representation that supports memory and guides future action. To model this process, we construct a heuristic graph model, shown in Fig.1(left), where each vertex indicates a subregion and each edge indicates the path cost between two adjacent subregions. With reference to this graph model, we design a novel exploration architecture, which utilizes a DRL model to sequentially predict visiting order for unexplored subregions and a NBV selection algorithm to optimize inter-regional planning. To further improve the architecture's performance in the testing maps, we also define a structural integrity measurement to help evaluate the reasonability of DRL policy.

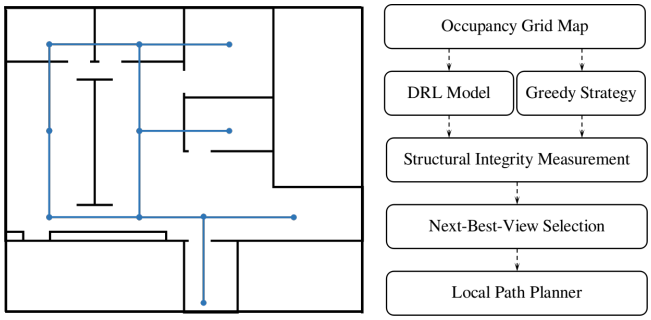


Fig. 1: Heuristic graph model and exploration architecture.

As shown in Fig.1(right), a complete workflow[1] of our method is as follows: 1) according to the current states of the agent and grid-map, the trained DRL model predicts a candidate subregion for the next-step visiting; 2) at the same time, a greed strategy is also adopted to include the nearest subregion as a candidate; 3) the two subregions are scored by the structural integrity measurement; 4) the NBV selection algorithm specifies a waypoint in the winner subregion; 5) the agent employs A* algorithm to plan a local path from its current position to the waypoint. Experiments show that this exploration architecture, which integrates a high-level global planning inference and low-level local optimization, is superior to traditional methods in terms of total path cost and the optimality of global visiting order.

For remaining parts of this article, we first give a brief review of relevant literature in Sec.II, then present the formulation and training process of DRL model in Sec.III. After that, an inter-regional planning method is introduced

* The authors contribute equally to this paper

The authors are with the Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong SAR, China. *email: {dlzhu, tgli, dho, cqwang, qhmeng}@ee.cuhk.edu.hk*

[1]The video demonstration is at *https://www.youtube.com/watch?v=mKJDrRiZN-Q&sns=em*

in Sec.IV, which corresponds to the last three steps of the exploration workflow. At last, related experiments and a conclusion are presented in Sec.V and Sec.VI.

II. Related Work

Frontier-based and information-based strategies are the most popular solutions to exploration problem. In early research, the agent was driven to the nearest frontiers [5] or the most uncertainty regions [6] greedily in an occupancy grip-map. Based on the pioneer work, a lot of improved versions were proposed and adopted in both single-agent and multi-agent systems [2]. For instance, González-Banos *et al.* [7] proposed a NBV selection algorithm, where an utility function was formalized as a criteria to measure the information gain of each candidate waypoint; Vallvé *et al.* [8] adopted an information potential field to indicate the joint path and map entropy, then based on the gradient of this field, an exploration path was efficiently computed. Roughly, we can regard these methods as next-best-view selection or one-step look ahead approaches. Due to the greediness, the performance of this category of methods is limited.

On the basis of these greedy methods, some work tried to optimize the NBV selection process by modeling structure dependency of environments. Jadidi *et al.* [9] proposed to construct a continuous map via Gaussian process regression, with the help of which, the uncertainty of some frontiers that were near obstacles was reduced, and others with high variance were selected as exploration targets; Different with this work, Ruiz *et al.* [10] employed Gaussian process to predict unexplored areas directly on grid map; then combined with a modified A^* algorithm, the grid with high uncertainty was safely selected as next visiting target. These two methods both showed great efficiency in test cases. Souza *et al.* [11] also utilized a Gaussian process to learn terrain characteristics, but a vibration level of the agent was introduced as a physical constraint; in order to make a trade-off between exploration and reducing vibration, the authors introduced the Bayesian optimization to guide next way-point selection and achieved a better compromise than information based methods. Different from these excellent work, we don't model the structure dependency but the next-step reward into Gaussian process at each subregion and a structural integrity measurement is additionally defined to help constrain the reward.

In spite of these effective models, we noticed that little work was conducted to learn and predict a visiting order for unexplored regions. In more recent work, Stefan *et al.* [12] used a global topo-metric prior to guide the exploration task and Georgiou *et al.* [13] proposed to extract an informative prior map from architecture drawings and floor plans, which both achieved a tremendous improvement, showing the power of prior knowledge. Nevertheless, the prior used in these two methods were human-specified, thus lack of generality. Some reinforcement learning based methods were also reported in [14], [15] and [16], but they were all target-driven explorations in the same or similar environments. As far as we know, our work presented in this paper is the first attempt to learn the prior knowledge for global visiting order inference.

III. Frontier-guided DRL Model

In exploration problem, traditional approaches usually focus on local strategy design. Because no global topological information is exploited, there always exists redundancy in exploration path. Without the understanding of basic indoor layout, the agent tends to pick an inefficient visiting order, thus increasing the overall cost. Generally, the layout of office environments are well structured, mainly composed of corridors and rectangular rooms. Although each office has a distinct layout, the latent spatial arrangement is similar. Hence, if the common knowledge about the topological information is extracted and applied to guide the agent's planning, the exploration performance can be further improved.

A. Model Formulation

During training phase, the goal of the exploration problem is to find an optimal path ξ^* to cover the whole target area $\mathbf{m}$. For grid-map based planning, the path ξ is decomposed into a set of way points $F{=}\{\mathbf{x}_1, ..., \mathbf{x}_T\}$ and the goal is thus transformed into finding an optimal point set F^*. This optimization problem can be formalized as follows

$$F^* = \arg\min_{\mathbf{x}_{1:T}} \sum_{i=1}^{T} L(\mathbf{x}_i|\hat{\mathbf{m}}_i) \tag{1}$$

where L is the length of a path segment between way points $\mathbf{x}_i$ and $\mathbf{x}_{i-1}$, and $\hat{\mathbf{m}}_i$ is the current explored map at step i.

In the context of RL, this problem can be regarded as a sequential decision making process, where the agent chooses a sequence of actions to maximize the accumulated future reward. Although there exists a variety of RL algorithms for video games [3], directly applying these methods to exploration problem will lead to the curse of dimensionality problem. For instance, if given a $M \times N$ grid-map, an agent with $|A|$ control orders and an optimal path with K grid steps, the problem space will become a huge space with a scale of $O(|A|^{MN})$ compared with the solution space $O(|A|^K)$. Another issue is that, because there usually exist some extra constraints, e.g. shortest path, finding an optimal exploration path in a grid-map is indeed much more difficult than finding a goal in Maze games, especially when we expect the model to have generalization ability. Due to these challenges, the existing exploration applications are mainly goal-searching tasks [14][15] or obstacle avoidance tasks [16]. In this section, we propose to formalize our MDP problem on a high-level heuristic graph to help reduce the searching difficulties. We denote the MDP model in this paper as $M{=}\{S, A, T, R, \gamma\}$ and the details are as follows:

State: $S{=}\{\mathbf{m}_1, ..., \mathbf{m}_k\}$ is a finite set, representing the environment. In our model, $\mathbf{m}_i$ is a fixed-size (100×100) state map, which is down-sampled from the occupancy grid-map with an indicator of the agent drawn on it. Different from work [14][16] which formulates sensor measurement into state space, we formulate the map and the agent's

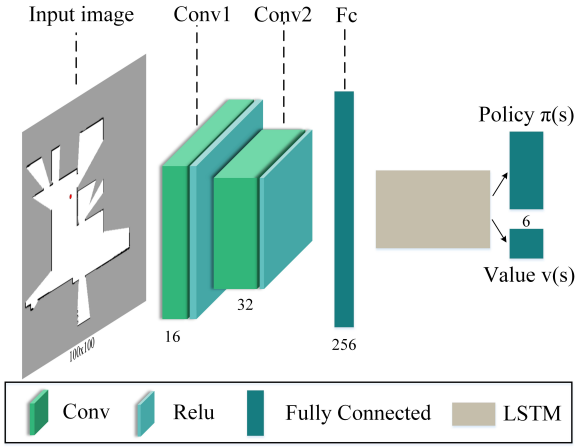


Fig. 2: Network structure. There are two convolutional layers for feature extraction, one LSTM layer with 256 cells for temporal dependency and two fully connected layers to generate policy and value.

location and orientation into the state space, which is one of the key factors to improve the model's generalization ability.

Aciton: $A=\{a_1, ..., a_m\}$ is a finite set of actions that indicates the next visiting directions. Here, the space around the agent is divided into six sectors equally, each sector s_i is indicated by an action a_i. At each step, the agent firstly chooses a sector according to a_i, then the nearest unexplored subregion within s_i as the next visiting target. Once a subregion is selected, the agent utilizes a NBV algorithm to select a waypoint in this subregion, then the A^* algorithm is used to plan a feasible path and move to that waypoint. To accelerate the training process, we adopt the nearest point of the selected subregion as waypoint. During testing phase, we utilize an optimized approach (Sec.IV-A) to conduct the selection process.

In each episode, the agent actually samples a F with the guidance of frontiers in the state map, thus greatly reducing the problem space. Besides, based on the observation that when exploring an unknown environment, human might change their target direction if other new space is discovered, the agent is also designed mimic this behavior in our model, which means that if the increment of explored regions along the planned path exceeds a predefined threshold, new (s_t, a_t, r_t) will be sampled into that episode.

Reward: R is the reward function that can guide the robot to explore the entire area efficiently. To minimize the path length of covering the whole area, for each step we set the reward to be $r_i = -cp_i$, where c is a positive constant number and p_i is the path length between $\mathbf{x}_i$ and $\mathbf{x}_{i-1}$. Equivalently, when the reward comes to its maximum, the shortest path will be found out. In this model, the coefficient c is set to be 0.001.

B. Model Approximation Based on A3C

For the purpose of leading the agent to find an optimal policy π^* that maximizes the expected total discounted reward, an actor-critic architecture is utilized here. The policy π approximated by a policy network is the actor and the value $V(s_t)$ approximated by a value network can be regarded as the critic. The policy gradient is scaled by the advantage of action a_t in state s_t, or $A(a_t, s_t) = R_t - V(s_t)$, where the discounted return R_t is an estimate of $Q^\pi(a_t, s_t)$. In this way, the variance of estimate of the policy gradient is reduced significantly [17].

In this paper, a more advanced version, the Asynchronous Advantage Actor Critic (A3C) network [18], is utilized to approximate our model. The network structure is shown in Fig.2, where the input image is 100×100 state map, the kernel size of *Conv1* is 8×8 with a stride of 4 and the kernel size of *Conv2* is 4×4 with a stride of 2; then the 704-D outputs of convolutional layers are then processed by a fully connected layer and further passed through a LSTM layer; the final layers output possibilities of six actions and the value $V(s_t)$.

The details of our algorithm based on A3C are presented in Alg.1. Because this algorithm is constructed on the heuristic graph, it learns more spatial structure, that contains abundant topological information, rather than the local details of the environments. In next section, we will introduce a local strategy to deal with the planning problem between adjacent subregions.

IV. Inter-Regional Planning

A. Bayesian Optimization Based Sampling

At each step, DRL model can predict a visiting order for unknown subregions according to its learned knowledge, but it cannot propose a waypoint or specify a path for the agent. A common choice is to utilize NBV selection algorithm introduced in Sec.II to select a waypoint, then employ A^* to plan a feasible path to that waypoint. Here we refer this process as inter-regional planning. In view of the time inefficiency problem in the NBV selection process [7][19], we introduce a new optimization strategy based on Bayesian Optimization (BO) [20].

Typical NBV selection methods [6][7] are composed of three steps. Firstly, a number of waypoint candidates $\mathbf{x}$, indicated by '+' in Fig.3(a), are randomly sampled in a specific subregion S; then each candidate is assigned with a score according to some evaluation functions; finally, the candidate with the highest score (the red '+') is selected as the next-step destination. Mathematically, this process can be formulated as an optimization problem:

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in S} f(\mathbf{x}) \tag{2}$$

$$f(\mathbf{x}) = -\frac{1}{|\mathcal{Z}|} \sum_{i=1}^{n} p(z_i|\mathbf{x}) \log p(z_i|\mathbf{x}) - \alpha \frac{L(\mathbf{x})}{L_{max}} \tag{3}$$

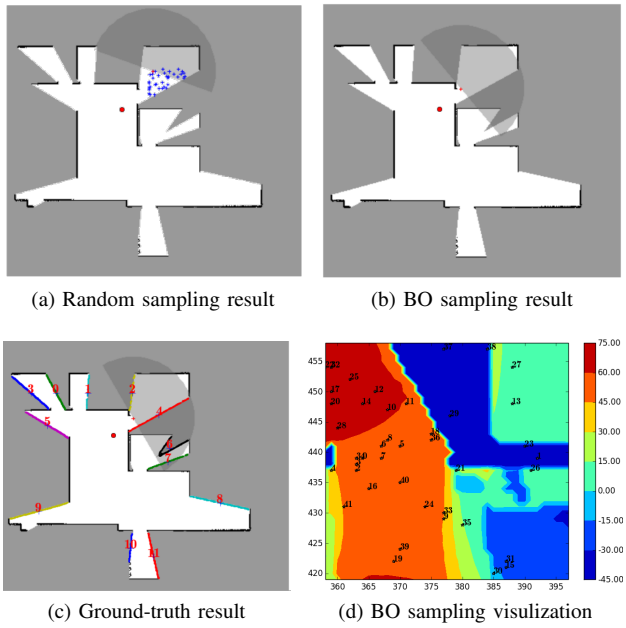
where f is the evaluation function, S indicates a suggested subregion by DRL or greedy policy; $|\mathcal{Z}|$ is a weight coefficient, usually the number of grids within the scope of range-scanner (the shadow sector in Fig.3(a)); $L(\mathbf{x})$ is the path length between $\mathbf{x}$ and the agent's current position (the red point in Fig.3(a)); $p(z_i|\mathbf{x})$ is a uniform distribution that represents a measurement belief over grid i, usually

Algorithm 1 Frontier-Guided Deep Reinforcement Learning

Initialize the global shared weights θ, θ_v
Initialize the weights $\theta^{'}$, $\theta_v^{'}$ for every thread
// For every thread
for episode = $1, M$ **do**
  Synchronize thread-specific weights $\theta^{'} = \theta, \theta_v^{'} = \theta_v$
  Reset gradient $d\theta = 0$, $d\theta_v = 0$
  Set the robot to the start position
  Reset occupancy grid-map.
  $t \leftarrow 1$
  Get state s_t
  repeat
    Choose a sector a_t according to policy $\pi(a_t|s_t;\theta^{'})$
    if no frontiers in a_t **then**
      continue
    end if
    Get the position of nearest frontier in a_t
    Plan a path ξ_t to that frontier using A^* algorithm
    while increased map area $<$ threshold **do**
      Move forward one step according to ξ_t
      Update constructed map
    end while
    Receive reward r_t and new state s_{t+1}
    $t \leftarrow t+1$
  until the entire area is explored
  Define $R = \begin{cases} 0 & s_t \text{ = terminal} \\ V(s_t;\theta_v^{'}) & s_t \text{ = otherwise} \end{cases}$
  for i = t-1,1 **do**
    $R \leftarrow r_i + \gamma R$, where
    $d\theta \leftarrow d\theta + \nabla_{\theta'} log\pi(a_i|s_i;\theta^{'})(R - V(s_i;\theta_v^{'})) + \beta\nabla_{\theta'} H(\pi(s_i;\theta^{'}))$
    $d\theta_v \leftarrow d\theta_v + \partial(R - V(s_i;\theta_v^{'}))^2/\partial\theta_v^{'}$
  end for
  Perform asynchronous update of θ and θ_v using $d\theta$ and $d\theta_v$.
end for

$p(z_i|\mathbf{x})$=0.5 for unexplored grid and $p(z_i|\mathbf{x})$=1 once the grid is perceived by range-scanner. The first term of Eq.(3) can be regarded as a next-step normalized Information Gain [1].

Because brute-force searching based solutions to Eq.(2) are time-consuming, random sampling based methods become a popular choice [7]. However, there are some obvious shortcomings of random sampling: 1) the high-score points may even not be included in candidate set; 2) it is difficult to determine the sampling numbers to balance the efficiency and accuracy. In order to deal with these deficiencies, we propose an intelligent sampling algorithm inspired by BO. As shown in Alg.2, the philosophy of our method as well as BO are as follows: we first assume a Gaussian process prior, from which the function $f(\mathbf{x})$ is sampled; then we use observations $[f(\mathbf{x}_t), f(\mathbf{x}_{t+1}), ..., f(\mathbf{x}_{t+n})]$ to update the



(a) Random sampling result (b) BO sampling result

(c) Ground-truth result (d) BO sampling visulization

Fig. 3: Demonstration of Bayesian optimization based sampling

Gaussian process[2] iteratively; at each iteration, we do not sample f using randomly selected $\mathbf{x}$ but utilize an *acquisition function* to guide the selection process, where the target $\mathbf{x}_i^*$ that mostly improves current optimal value is selected as next sample position. The acquisition function we used here is called *Expected Improvement* [21] and the formula that accommodates to our case is defined as Eq.(4), where $f_{max}(\mathbf{x})$ indicates current maximum of f.

$$\mathcal{A} = E[\, max\{0,\ f(\mathbf{x}_{t+1}) - f_{max}(\mathbf{x})\}\,] \tag{4}$$

To demonstrate the sampling process, we firstly detect a gate point, indicated by '2' in Fig.4(b); then taking this gate point as center, we specify a 40×40 sampling subregion and run the BO sampling algorithm. The running result is visualized in Fig.3(d), where the heat map offers a ground-truth result of f by brute-force searching, and the black numbers indicate BO sampling steps. It can be seen that the algorithm intelligently leads the sampling target into high-score regions, while keeping an exploration ability to search unsampled areas. This provide us with a possible solution to help overcome the shortcomings of random sampling method. We also visualize the final NBV selection results in Fig.3(a-c) for further comparison and a more detailed evaluation will be presented in Sec.V.

B. Structural Dependency Modeling

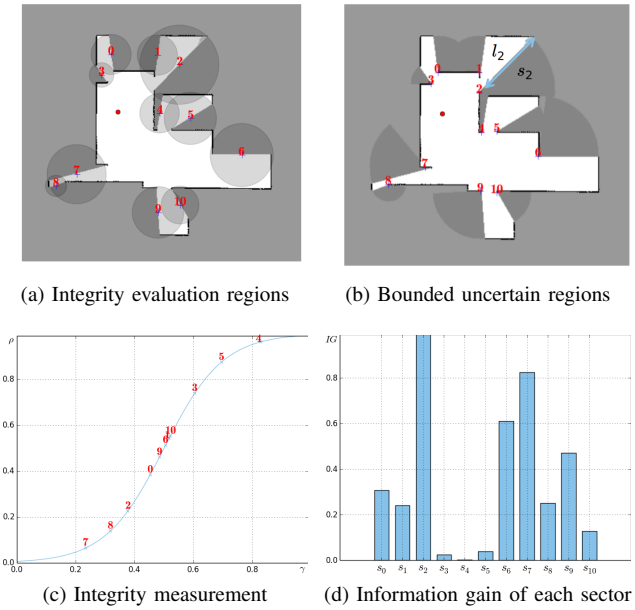
The sampling accuracy of f in Alg.2 has a great effect on BO. Since traditional NBV selection methods do not consider the structure dependency between adjacent map grids, much bias is introduced into f. To model the structural dependency, a common solution is to apply regression approach [9][10],

[2]A Matern kernel with default parameters is used to initialize the Gaussian process *http://scikit-learn.org/stable/modules/gaussian_process.html*

Algorithm 2 Bayesian Optimization Based Sampling

Input: S-evaluation region, f-reward function, K-kernel function, $\mathcal{A}$-acquisition function
Output: $(\mathbf{x}^*, f(\mathbf{x}^*))$-optimal solution

1: initialize a Gaussian process prior $p(f)$ with the kernel K
   $p(f) = \mathcal{GP}(f; \mu, \sigma^2)$
2: randomly sample a small subset $\mathcal{D} \subset \{\mathbf{x}, f(\mathbf{x}) | \mathbf{x} \in S\}$ to fit a posterior distribution $p(f|\mathcal{D})$
   $p(f|\mathcal{D}) = \mathcal{GP}(f; \mu_{f|\mathcal{D}}, \sigma^2_{f|\mathcal{D}})$
3: **for** i = 1:N **do**
4: optimize the *acquisition function* based on $p(f|\mathcal{D})$
   $\mathbf{x}_i^* = \arg\max_{\mathbf{x} \in S} \mathcal{A}(\mathbf{x}; p(f|\mathcal{D}))$
5: sample the evaluation function Eq.(3) and add the sampling result into $\mathcal{D}$
   $\mathcal{D} = \mathcal{D} \cup \{(\mathbf{x}_i^*, f(\mathbf{x}_i^*))\}$
6: update the posterior distribution $p(f|\mathcal{D})$ with current $\mathcal{D}$
7: **end for**
8: find the optimal solution $\mathbf{x}^*$ suggested by current $p(f|\mathcal{D})$
9: **return** $(\mathbf{x}^*, f(\mathbf{x}^*))$



(a) Integrity evaluation regions (b) Bounded uncertain regions

(c) Integrity measurement (d) Information gain of each sector

Fig. 4: Demonstration of structural dependency modeling

where the uncertainty of unexplored regions is modeled to be correlated with the distance to explored structures. Moreover, we also claim that the occurrence of a certain structure in unexplored regions, e.g. office wall, is not only dependent on the distance to their explored neighbors, but also restricted by the integrity of that structure. Based on this observation, we introduce the structural integrity measurement to improve the regression model. Hence the formula of original $p(z_i|\mathbf{x})$ in our model is turned into

$$p(z_i|\mathbf{x}, d, g) = p(z_i|\mathbf{x}) \cdot p(z_i|d) \cdot p(z_i|g) \tag{5}$$

where g indicates the structural integrity, d indicates the distance between z_i and gate points, indicated by red numbers in Fig.4(b); $d, g, \mathbf{x}$ are assumed to be independent of one another. To make the sampling process of f as efficient as possible, we adopt Bernoulli distribution to model $p(z_i|d)$ and $p(z_i|g)$. The modeling procedure can be decomposed into two steps:

Bound the Uncertain Region. According to Eq.(3), f is directly determined by the number of unexplored grids that are located within the scope of range-scanner (Fig.3(a)). Apparently, this evaluation function encourages the NBV algorithm to select waypoints close to unexplored regions, which introduces a severe bias into f. Another consideration is that, distant grids are with a high probability of not being perceived due to the existence of walls, thus assigning $p(z_i|d \geq l_i)$ with a big value will much decrease the sampling accuracy. Based on these intuitions, we propose to bound the interest regions within a sector area, shown in Fig.4(b). Correspondingly, we define

$$p(z_i|d) = \begin{cases} 1-\delta & d < l_i \\ \delta & \text{otherwise} \end{cases} \tag{6}$$

where l_i is the boundary length. As an example, we calculate the information gain of each sector in Fig.4(b) and visualize the result in Fig.4(d).

Measure the Integrity of Local Structure. We first sample a circle area that takes *boundary-i* as its diameter, as shown in Fig.4(a); then the ratio r of explored area in this circle is calculated; after that, a *sigmoid function* is utilized to map r to the integrity measurement g. Mathematically, The formula of g is defined by

$$g = \frac{1}{1 + e^{-k(r-0.5)}} \tag{7}$$

where k controls its curvature. As an example, we plot the integrity of subregions in Fig.4(a) and their corresponding measurement in Fig.4(c). With reference to Fig.1, we notice that the integrity of partially explored office rooms can be reasonably represented by the measurement we defined. Considering the truth that the uncertainty of z_i is negatively correlated with g, here we directly model $p(z_i|g)=1-g$. Apart from serving as a component to help improve the sampling accuracy of f, g also provides a criteria for judging the reasonability of DRL policy since it can better characterize individual features of a specific office environment.

C. Exploration Architecture

So far, we have introduced the DRL model for visiting order suggestion, and the inter-regional planning for waypoint selection and local path generation. In this section, we will present the integration of these two components, as well as some useful tricks. During training phase, we sequentially sample waypoints from target grid-maps according to the suggestion of DRL model, then connect these waypoints using A* algorithm to generate a feasible trajectory. After this forward process, the trajectory length is then used to reinforce the DRL model. However, directly applying the

TABLE I: Comparison of exploration trajectory length between Nearest Frontier (NF), Nearest Bayesian Optimization (NBO), Random action Bayesian Optimization (RBO), Best Integrity Measurement (BIM), and Reinforcement Learning supervised Bayesian Optimization (RLBO)

Office	NF	NBO	RBO	BIM	RLBO	DRL action	greedy action	action ratio
1	499	503	501	691	**479**	24.2	5.6	81.2%
2	**522**	569	746	718	739	38.0	6.6	85.2%
3	**508**	623	574	528	513	27.3	7.3	78.9%
4	**649**	752	767	836	696	28.4	5.2	84.5%
5	**789**	856	875	964	805	29.0	8.5	77.3%
6	729	614	654	902	**572**	23.0	10.5	68.7%
7	740	920	718	711	**696**	24.2	14.6	62.4%
8	1147	991	978	1521	**969**	32.6	9.4	77.6%
9	1174	1357	938	1211	**790**	31.7	7.5	80.8%
10	1124	985	**852**	996	853	31.6	9.8	76.3%

forward process to testing maps doesn't yield a good performance. This is because the DRL model is trained to predict global visiting order rather than step-by-step planning, the greedy waypoint selection method in training process and new structures of testing maps have greatly reduced its advantages.

Based on this analysis, we additionally develop some optimization strategies (Sec.IV-A and Sec.IV-B) to suppress the influence of inter-regional planning. Combining with these strategies, a complete workflow during testing phase is depicted in Fig.1(right). The agent firstly evaluates the reasonability of DRL policy for a specific office structure. If it is reasonable enough, a BO based NBV selection algorithm is then utilized to help select the next waypoint; otherwise, the agent simply takes the nearest frontier (a type of greedy strategy) as the next waypoint. In practice, we directly calculate the structural integrity of DRL policy and greedy policy, then choose a waypoint from the winner's suggested subregion. Once the waypoint is specified, the agent will be able to plan a local path and move there. With these simple modifications, the DRL model shows an superior performance compared with the nearest-frontier method.

V. Experiments and Results

A. Evaluation of the Exploration Architecture and DRL model

To comprehensively evaluate the performance of the exploration architecture and the DRL model, five comparative experiments are conducted on ten new office plans, as shown in TABLE I. The experiments on each office environment are repeated for 20 times and the performance is measured by the average trajectory length. The models used in the experiment are nearest frontier (NF) method [5] which serves as a baseline, the nearest frontier optimized by BO sampling (NBO), deep reinforcement learning supervised BO (RLBO) where action policy is generated by network, random action BO (RBO) where action policy is generated randomly, and best integrity measurement (BIM) based algorithm where the point with highest integrity measurement score is chosen as waypoint at each step.

It can be seen that the RLBO model performs quite well and has the shortest trajectory length in five testing environments out of ten and has second shortest trajectory in another 4 maps, i.e. Office 3-5 and 10. For some cases, the trajectory length is greatly reduced, e.g. the length of RLBO method is shorter than that of NF in Office 6 and Office 9 by 21.5% and 32.7%, respectively. This shows that the RLBO model indeed has a generalization ability. Besides, by observing the last three columns, we find that around 70% to 80% actions (subregion suggestions) the agent takes are from DRL network, which indicates that the global topological information is exploited and contributes to the system performance.

To further evaluate our DRL model, a comparative experiment is conducted between RBO and RLBO. Out of 10 testing offices, 9 result trajectories with RLBO are shorter than that with RBO, and only one, i.e. Office 10, has similar performance. It justifies that the DRL model is one of the great contributor to the model performance and indeed greatly improves the overall performance.

Furthermore, to evaluate the influence of structure integrity measurement and BO sampling, another comparative experiment is implemented among NBO, BIM, RLBO. It can be seen in TABLE I that except for Office 2, RLBO outperforms both NBO and BIM, which indicates that using structure integrity measurement or BO sampling alone cannot help to improve the system performance. When they work together as a system, the trajectory length is significantly reduced, which also directly gives a strong support to our analysis in Sec.IV-C.

Fig.5 shows the trajectories of Office 2, 6 and 8 with different methods. Comparing with NF baselines, RLBO trajectories are shorter and more reasonable. To fully analyze the performance of RL, we deliberately select Office 2, where the NF trajectories are much shorter than other methods. As we can see, NF strategy is much dependent on the layout of office and is more suitable for those structures that contain few choices during exploration, e.g. Office 2 where rooms are closely located to the walls. For simple maps like Office 2, the optimal planning is straightforward and can be easily

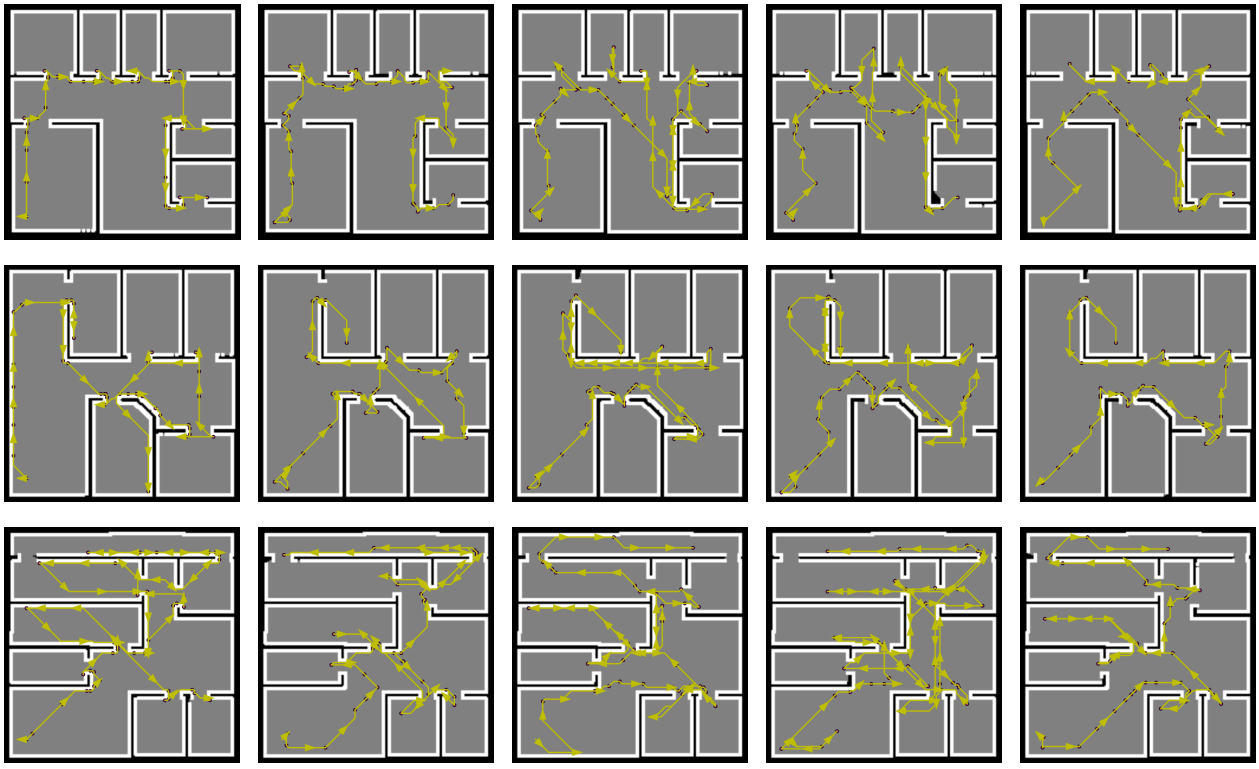


Fig. 5: Trajectories of Office 2, 6 and 8 of NF, NBO, RBO, BIM, RLBO.

TABLE II: Comparison of sampling time and accuracy between sampling methods.

Scene	**BO Sampling**		**Random Sampling**		**Brute-Force**	
	time(s)	$f(\mathbf{x}^*)$	time(s)	$f(\mathbf{x}^*)$	time(s)	$f(\mathbf{x}^*)$
0	2.1	0.655	1.8	0.676	14.1	0.696
1	2.4	0.706	2.6	0.616	31.7	0.886
2	3.6	0.770	1.5	0.647	15.1	0.869
3	2.6	0.431	4.6	0.397	17.5	0.481
4	3.5	0.390	6.1	0.382	41.8	0.394
5	2.9	0.370	6.7	0.380	61.2	0.384
6	1.6	0.804	1.2	0.642	34.0	0.873

obtained using greedy strategies like NF. In contrast, for complicated maps like Office 6 in Fig.5, the NF performance plummets while the RLBO model is still able to find a short trajectory, showing that RLBO is capable of utilizing global spatial information for planning.

The DRL model used in this paper is trained on 20 maps with 40000 episodes per map and each episode contains around 40 steps. The network is asynchronously trained with 4 threads on Intel Core i5-6500 at a frequency of 3.2 GHz without any GPU acceleration. The value loss and policy loss are equally weighted by 0.5, with an entropy weight of 0.001. If with fine-tunning operations, the performance of DRL model can be further improved.

B. Evaluation of BO Based Sampling

In order to evaluate the performance of our proposed BO sampling method, we randomly generate seven scenes as a testing set, similar to Fig.3(c), from several half-explored floor plans. In each case, we define the sampling region as a 20×20 square around the gate points. For BO sampling, the iteration number N in Alg.2 is set to be 25 with two initialization observations. For random sampling, the sample number is set to be $1/5$ of the sampling regions. For brute-force searching, a two-step method is employed, where the agent first selects the optimal position then determines its optimal heading by a 360° searching. Each method is evaluated for 20 times in every scene, then an average performance is reported in TABLE II, where $\mathbf{x}^*$ represents the optimal waypoint that are selected by different methods, $f(\mathbf{x}^*)$ is the corresponding score according to Eq.(3). Since $f(\mathbf{x}^*)$ plays a more important role during the waypoint selection process (Sec.IV-A), we compare the three methods in terms of $f(\mathbf{x}^*)$. As we can see, the BO sampling performs more stably and accurately. The random sampling method sometimes runs faster, but its performance is not as stable as BO sampling and severely dependent on the scale of sampling regions. The brute-force searching method provides the ground truth of $f(\mathbf{x}^*)$, which is the optimal result of NBV selection methods. Apparently, compared with random sampling, $f(\mathbf{x}^*)$ of BO sampling is much closer to the ground-truth result, which verifies the BO sampling method is more accurate. The time cost reported here is based on our initial implementation

of the BO sampling algorithm, there is still a lot of space to optimize, thus the performance of BO sampling can be further improved.

VI. Conclusion

In this paper, we presented a new idea to combine the advantages of high-level global visiting order inference and low-level inter-regional planning. Based on this idea, we utilized the DRL model to learn the common knowledge in office floor plans, and successfully generalized the model to new cases via defining a MDP over the heuristic graph. Comparative experiments showed that exploration performance was improved with the help of a global visiting order suggested by the DRL model. Besides, we presented a NBV algorithm as a waypoint selection method that could be integrated into the exploration architecture. This algorithm showed great efficiency and accuracy compared with traditional random sampling method.

Since the algorithms presented in this paper are the preliminary implementation of our ideas, a lot of future work needs to be done. Firstly, the formulation of MDP and training process of A3C can be further improved, e.g. applying inverse reinforcement learning technique; the data scale and network capacity also need to be increased. Secondly, the time efficiency of the exploration architecture should be greatly improved as well, which actually leaves us with a lot of space for further optimization.

Acknowledgment

This project is partially supported by the Hong Kong RGC GRF grants #14205914 and Shenzhen Science and Technology Innovation projects c.02.17.00601 awarded to Max Q.-H. Meng.

References

[1] S. Thrun, "Probabilistic robotics," *Communications of the Acm*, vol. 45, no. 3, pp. 569–573, 2005.

[2] M. Juliá, A. Gil, and O. Reinoso, "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments," *Autonomous Robots*, vol. 33, no. 4, pp. 427–444, 2012.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G. Ostrovski, "Human-level control through deep reinforcement learning." *Nature*, vol. 518, no. 7540, p. 529, 2015.

[4] E. C. Tolman, "Cognitive maps in rats and men." *Psychological Review*, vol. 55, no. 4, p. 189, 1948.

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

[6] P. Whaite and F. P. Ferrie, "Autonomous exploration: Driven by uncertainty," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 193–205, 1997.

[7] H. H. González-Banos and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 829–848, 2002.

[8] J. Vallvé and J. Andrade-Cetto, "Potential information fields for mobile robot exploration," *Robotics and Autonomous Systems*, vol. 69, pp. 68–79, 2015.

[9] M. G. Jadidi, J. V. Miró, R. Valencia, and J. Andrade-Cetto, "Exploration on continuous gaussian process frontier maps," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6077–6082.

[10] A. V. Ruiz and C. Olariu, "A general algorithm for exploration with gaussian processes in complex, unknown environments," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3388–3393.

[11] J. R. Souza, R. Marchant, L. Ott, D. F. Wolf, and F. Ramos, "Bayesian optimisation for active perception and smooth navigation," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4081–4087.

[12] S. Oßwald, M. Bennewitz, W. Burgard, and C. Stachniss, "Speeding-up robot exploration by exploiting background information," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 716–723, 2016.

[13] C. Georgiou, S. Anderson, and T. Dodd, "Constructing informative bayesian map priors: A multi-objective optimisation approach applied to indoor occupancy grid mapping," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 274–291, 2017.

[14] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3357–3364.

[15] B. Bakker, V. Zhumatiy, G. Gruener, and J. Schmidhuber, "Quasi-online reinforcement learning for robots," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 2997–3002.

[16] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," *arXiv preprint arXiv:1703.00420*, 2017.

[17] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[18] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016, pp. 1928–1937.

[19] C. Wang, L. Meng, T. Li, C. W. De Silva, and M. Q.-H. Meng, "Towards autonomous exploration with information potential field in 3d environments," in *Advanced Robotics (ICAR), 2017 18th International Conference on*. IEEE, 2017, pp. 340–345.

[20] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, 2012, pp. 2951–2959.

[21] J. Močkus, "On bayesian methods for seeking the extremum," in *Optimization Techniques IFIP Technical Conference*. Springer, 1975, pp. 400–404.