ScanBot: Autonomous Reconstruction via Deep Reinforcement Learning

HEZHI CAO* and XI XIA*, University of Science and Technology of China, China
GUAN WU, University of Science and Technology of China, China
RUIZHEN HU†, Shenzhen University, China
LIGANG LIU, University of Science and Technology of China, China

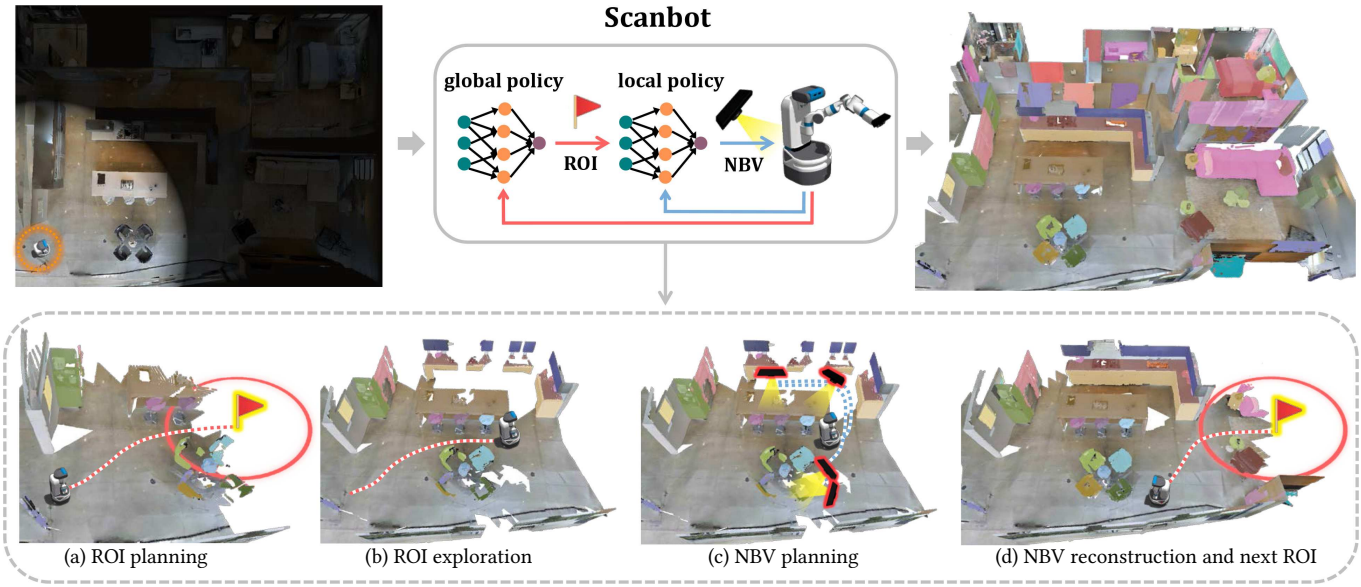


Fig. 1. Top: When entering an unknown environment, our proposed ScanBot (circled in orange, left figure) relies on a global policy and a local policy to alternatively generate the region-of-interest (ROI, denoted as a red flag with the red circle) and a series of next-best-views (NBVs, denoted as black Kinect sensors) to automatically reconstruct the scene with object semantics (in different colors, right figure). Bottom: We demonstrate one iteration of the global and local scanning procedures and show their navigation paths using red (global) and blue (local) dotted lines, respectively. ScanBot first selects an ROI by the global policy (a), and then navigates to the goal position (b) to select a sequence of NBVs nearby guided by the local policy (c). When the local detailed reconstruction is finished following the planned NBVs, ScanBot enters the next iteration by selecting a new ROI (d).

Autoscanning of an unknown environment is the key to many AR/VR and robotic applications. However, autonomous reconstruction with both high efficiency and quality remains a challenging problem. In this work, we propose a reconstruction-oriented autoscanning approach, called ScanBot, which utilizes hierarchical deep reinforcement learning techniques for global *region-of-interest* (ROI) planning to improve the scanning efficiency and local *next-best-view* (NBV) planning to enhance the reconstruction quality. Given the partially reconstructed scene, the global policy designates an ROI with insufficient exploration or reconstruction. The local policy is then applied to refine the reconstruction quality of objects in this region by planning and scanning a series of NBVs. A novel mixed 2D-3D representation is designed for these policies, where a 2D quality map with tailored quality channels encoding the scanning progress is consumed by the global policy, and a coarse-to-fine 3D volumetric representation that embodies both local environment and object completeness is fed to the local policy. These two policies iterate until the whole scene has been completely explored and scanned. To speed up the learning of complex environmental dynamics and enhance the agent's memory for spatial-temporal inference, we further introduce two novel auxiliary learning tasks to guide the training of our global policy. Thorough evaluations and comparisons are carried out to show the feasibility of our proposed approach and its advantages over previous methods. Code and data are available at https://github.com/HezhiCao/Scanbot.

CCS Concepts: • **Computing methodologies**; • **Shape analysis**;

Additional Key Words and Phrases: Indoor scene reconstruction, autonomous reconstruction, hierarchical reinforcement learning, auxiliary learning tasks

ACM Reference Format:
Hezhi Cao, Xi Xia, Guan Wu, Ruizhen Hu, and Ligang Liu. 2018. ScanBot: Autonomous Reconstruction via Deep Reinforcement Learning. *ACM Trans. Graph.* 37, 4, Article 111 (August 2018), 16 pages. https://doi.org/XXXXXXX.XXXXXXX

*Hezhi Cao and Xi Xia are joint first authors.
†Corresponding author: Ruizhen Hu (ruizhen.hu@gmail.com)

Authors' addresses: Hezhi Cao; Xi Xia, University of Science and Technology of China, Hefei, China; Guan Wu, University of Science and Technology of China, Hefei, China; Ruizhen Hu, Shenzhen University, Shenzhen, China; Ligang Liu, University of Science and Technology of China, Hefei, China.

0730-0301/2018/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

With the rapid progress of ubiquitous RGBD depth sensors, autonomous 3D scanning (i.e., autoscanning) of indoor scenes by mobile robots equipped with these sensors has attracted growing attention in both computer graphics and robotics communities [Charrow et al. 2015; Xu et al. 2017]. Due to different applications, they usually emphasize different aspects of the scanning process. For example, when the goal is to navigate to some specific positions or objects, quick exploration is the key, and there is always no need to reconstruct the whole scene with fine details. In contrast, if the goal is to build a realistic digital copy of the entire scene, without a pre-determined purpose, then both exploration coverage and reconstruction quality become essential. In this work, we focus on later, i.e., autonomous reconstruction of an unknown environment.

Previous works on autonomous reconstruction [Charrow et al. 2015; Liu et al. 2018; Xu et al. 2017; Zheng et al. 2019] usually adopt a hierarchical strategy with global path planning to explore the unknown area and local view planning to control the delicate camera movements for the careful reconstruction of discovered objects, which leads to high reconstruction quality. However, the underlying scanning strategy in either global path planning or local view planning is usually hand-crafted and rule-based, which decouples the semantic information extraction and the scanning decision-making. This decoupling makes it harder for the agent to leverage common patterns of indoor scenes, limiting its generality to different environments and robustness to the error in state estimations. Meanwhile, recent works on exploration or navigation-oriented autoscanning [Chaplot et al. 2021, 2020b,a] utilize learning-based methods to improve exploration efficiency and flexibility by integrating feature extraction and decision-making modules. However, they focus more on exploration and navigation without the detailed view planning driven by the completeness of objects, which usually results in poor reconstruction quality. Our goal is to study the possibility of using more advanced learning-based techniques to conflate the exploration efficiency and scanning quality together for the autonomous reconstruction of an unknown environment.

Based on the key observation that autonomous reconstruction, by its nature, is a *sequential decision-making process*, which consists of two integrated levels of sequential planning [Liu et al. 2018], we propose a novel autonomous reconstruction system, called *ScanBot*, which employs a hierarchical *deep reinforcement learning* (DRL) framework to combine the superiority of learning-based exploration methods and detailed view planning to achieve both high efficiency and high quality. Specifically, a global scanning policy is proposed first to plan a *region-of-interest* (ROI) with insufficient exploration or multiple incomplete objects for further investment. Then the agent travels to the planned ROI by an analytic pathfinder and utilizes the local policy to improve the reconstruction quality. The local policy plans and scans along a sequence of *next-best-view*s (NBVs) of incomplete objects in the corresponding region. Once local scanning in the planned ROI is completed, the global policy designates a new ROI for further improvement. These two plannings iteratively alternate until the whole scene has been completely explored and scanned.

Different from works on exploration or navigation-oriented autoscanning [Chaplot et al. 2021, 2020b,a], where usually a 2D occupancy grid with/without semantics is enough as the observation, the completeness of objects cared in the reconstruction task can hardly be derived from their representations. On the other hand, although pure 3D representation is ideal for indicating the completeness of objects, using such a heavy representation is infeasible in practice considering the expensive computational cost of DRL. To overcome the above challenges, we introduce a mixed 2D-3D representation to provide suitable information needed for each step, where a *2D quality map* is utilized in global policy, and a detailed *3D voxel representation* is adopted by the local policy for delicate camera control. Our key insight is that a well-designed 2D representation that encodes the reconstruction quality is sufficient to provide an overall structure for the global policy to decide the next target region, while delicate camera control using local policy for high-quality scanning only needs 3D representation on a smaller scale. By using this hybrid representation, our global-to-local hierarchical framework can be implemented with reasonable computing resources and achieve efficient, complete, and accurate scene reconstruction.

To further alleviate the sample inefficiency of DRL, we propose two dedicated auxiliary learning tasks to guide the training of the global policy by providing additional complementary objectives. By accessing the designated goal point by a path planning method, the global policy can be released from low-level decision-making, but it also makes the agent unconscious of the path it will go through. To improve the path representation and the learning of the environment's dynamics, we propose a Path Complexity Prediction (PCP) task to help the agent sense the complexity of the path and produce a reachable goal point. Moreover, to enhance the information maintained in agent memory, a Scanning Progress Recollection (SPR) task is designed to memorize the scanned objects during the previous action where the quality reward comes from.

To the best of our knowledge, this work is the first DRL-based approach to automatically accomplish the exploration and reconstruction of large-scale indoor environments concurrently in one navigation pass. Our method successfully learns to strategically switch between exploration-oriented and reconstruction-oriented goals and filter out infeasible viewpoints given the current partial observations. The overall system, as well as individual components, are verified through extensive experiments to show the feasibility of the proposed approach and the superiority of our method over previous methods. To summarize, our main contributions include:

- An autoscanning system for high-quality scene reconstruction based on hierarchical DRL techniques with a global policy for ROI planning and a local policy for sequential NBV planning.
- A mixed 2D-3D representation of current perceived scene structures, where novel quality channels are added to a 2D quality map to provide reconstruction-aware information for ROI planning and detailed 3D volumetric representations are used for NBV planning.
- Two new auxiliary learning tasks designed for the global policy, i.e., Path Complexity Prediction (PCP) and Scanning Progress Recollection (SPR), to accelerate the training process and promote understanding of spatial and temporal relations.

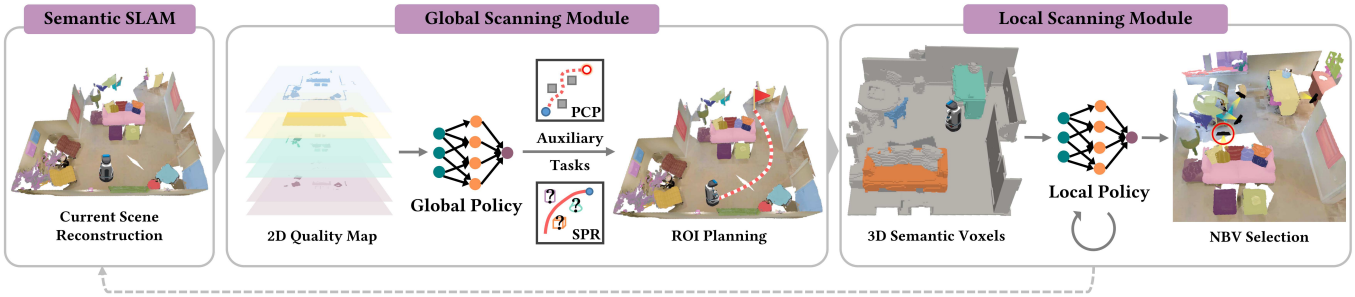


Fig. 2. Overview of our DRL-based autoscanning system with global exploratory scanning module and local NBV scanning module. A 2D quality map is constructed from current scene reconstruction and taken as the input to the global policy to generate a ROI to guide the exploration of regions that need finer reconstruction, where the training of the global policy is accelerated by adding two auxiliary tasks. The local policy then progressively selects and scans objects within the ROI by determining a sequence of NBVs. Once all seen objects are reconstructed, the whole process enters the next iteration.

2 RELATED WORK

2.1 Autonomous 3D reconstruction

The maturity of both online RGB-D reconstruction [Dai et al. 2017; Newcombe et al. 2011; Nießner et al. 2013; Whelan et al. 2016] and mobile robots advances the advent of autonomous reconstruction systems [Charrow et al. 2015; Krainin et al. 2011; Xu et al. 2017]. In early attempts, research effort has been devoted to automatically scanning individual objects with fine-grained geometric details by scrupulously choosing a next-best-scan [Krainin et al. 2011; Kriegel et al. 2012; Wu et al. 2014]. A small region with several objects has also been tackled by either physical interactions [Xu et al. 2015] or 3D attention-based object recognition [Xu et al. 2016]. One step further, [Charrow et al. 2015; Xu et al. 2017] propose to construct dense 3D maps of the entire scene driven by the uncertainty that is commonly computed from mutual information or tensor field. To improve the speed of scene reconstruction, a multi-robot system can be deployed and driven by dynamic task assignment using Optimal Mass Transport optimization [Dong et al. 2019].

Regardless of the significant progress being made, only in recent years have the semantic cues from single objects been used in online scene reconstruction. Liu et al. [2018] first conceive the idea of an objectness-driven autoscanning scheme enabling exploration, reconstruction, and object extraction of an unknown scene in one navigation pass. Besides, Zheng et al. [2019] use voxel-based semantic labeling to generate a viewing score field as an alternative way for online semantic reconstruction. Distinctly, we develop a novel autonomous reconstruction approach to merge the semantic information extraction with the scanning action decisions, making it easier for the robot to leverage learnable patterns in common indoor scenes as its high-level semantic priors.

2.2 Autonomous exploration and navigation

Automatically exploring and navigating in an unknown environment has been a long-standing research topic in robotics [Gasparetto et al. 2015; Yamauchi 1997]. Most existing methods rely on the frontiers of an occupancy map to determine a global goal for broadening the boundary of the explored region [Bourgault et al. 2002; Stachniss et al. 2005; Umari and Mukhopadhyay 2017]. However, frontier-based methods are usually stacked at corners or small areas behind furniture [Chaplot et al. 2020b]. Instead of using frontiers directly, [Chaplot et al. 2020c] introduces graph nodes as landmarks and conducts navigation by selecting nodes toward the target via supervised learning.

To effectively leverage structural regularities of real-world environments, prior works on autonomous navigation-oriented exploration use a DRL policy together with an explicit world map that is either directly learned from first-person images [Chaplot et al. 2020b; Gupta et al. 2017] or constructed by 3D projection [Chaplot et al. 2020a; Chen et al. 2019]. However, the absence of object completeness in these maps makes the agent unconscious of the reconstruction progress, which may lead to the oversight of the area that has been explored but is still occupied with incomplete objects. Recently, a 3D semantic map is introduced in [Chaplot et al. 2021] to actively gather observations and refine a pretrained instance segmentation network using spatio-temporal consistency of semantic labels, but this representation cannot scale to large scenes since the computation and memory costs cubically grow with the scene size.

2.3 Deep reinforcement learning

Reinforcement learning is a powerful framework for dealing with the sequential decision problem, generally formulated as the Markov decision process (MDP). Bolstered by deep neural networks, it has unleashed its potential for solving a wide range of real-world tasks such as games [Hessel et al. 2018; Silver et al. 2018] and robot control [Akkaya et al. 2019; Gu et al. 2017; Peng et al. 2018b].

The usage of DRL also extends to computer graphics. Peng et al. [2018a] adopt a combined objective of motion-imitation and user-specified tasks to learn highly dynamic skills for character animation. Other endeavors have been made to train a DRL agent to generate move plans for scene arrangement based on Monte Carlo tree search [Wang et al. 2020]. Hu et al. [2020] introduce a transport-and-pack problem and solve it with an RL-based box selection strategy interleaved with a heuristic-based box packing.

Hierarchical RL is a popular strategy to split long-horizon tasks by incorporating temporal abstraction [Sutton et al. 1999]. In the most common situation, the high-level policy directs the low-level policy through subgoals (specific states that the agent should reach) either in the original state space [Li et al. 2020; Nachum et al. 2019]

or in an embedding space [Jain et al. 2019; Vezhnevets et al. 2017]. To automatically find such abstraction, Bacon et al. [2017] propose an option-critic architecture that learns both the internal policies and their termination conditions, together with the policy over them.

In this work, we gain the power of hierarchical RL by decoupling end-to-end learning across different spatial scales with both global and local policies tailed for our autonomous reconstruction task. Note that the goal point generated by the global policy in previous works [Chaplot et al. 2021, 2020b,a] mainly indicates the general direction toward the target point or object to guide the local planning, and thus the agent usually does not need to actually arrive at such a goal point. In contrast, the goal point generated by our global policy represents the ROI aiming for either exploration or reconstruction, and the agent needs to arrive at that point first and then start making local decisions based on the current observation. Besides, their local policy is used to generate low-level actions with the goal of promoting exploration efficiency while navigating to the global point. However, we directly go to the global goal point following a simple pathfinder but focus more on the ROI goal point selection and local path planning inside the ROI region, aiming to get a more complete reconstruction while exploring the scene. Therefore, although adopting a similar hierarchical structure, both the global and local policies are carefully designed to better serve our task with quite different operating mechanisms compared with the previous method.

2.4 Auxiliary learning tasks

Using auxiliary learning tasks in DRL is a well-proven technique [Jaderberg et al. 2016; Mirowski et al. 2016; Ye et al. 2021] to improve sample efficiency or even final performance on the actual task. Supervised auxiliary tasks may prepare additional labeled data for the agent to endow it with preferred abilities, e.g., depth prediction [Mirowski et al. 2016] or object key points prediction [Matas et al. 2018]. Alternatively, self-supervised auxiliary tasks collect inputs and surrogate annotations from the agent's own experience, such as next-step feature [Pathak et al. 2017], temporal distance to episode termination [Kartal et al. 2019], and immediate rewards [Jaderberg et al. 2016]. In our work, we concentrate on self-supervised tasks to not restrict the application of our method only in simulation or annotation-rich environments. The proposed auxiliary tasks also integrate intimately with the specifying of distant goals and scanning progress representation, resulting in a fruitful speeding up of the policy training.

3 OVERVIEW

Figure 2 shows an overview of our proposed DRL-based autoscanning method for unknown scene reconstruction. Our method consists of two modules: a global scanning module for ROI planning and a local scanning module for NBV planning. An underlying semantic SLAM framework is also integrated to estimate the camera pose and reconstruct the scene on the fly. Given the current scene reconstruction from the SLAM module, the global policy will select a ROI for scene exploration or reconstruction based on a 2D quality map, and then the local policy will further select a sequence of sensor viewpoints to guide a detailed scanning of the ROI region. The whole process iterates with the updated scene reconstruction until the entire scene has been completely explored, understood, and reconstructed. Note that our method focuses on higher-level decision-making, the actual collision-free path to the selected ROI or NBV is generated using matured pathfinding algorithms like D* lite [Koenig and Likhachev 2002].

Global scanning module. To get an overall structure of the current scene, our method projects the integrated 3D semantic point cloud S onto the floor plane to get the 2D quality map, which encodes not only the occupancy and semantic information as in [Chaplot et al. 2020a] but also the reconstruction quality and surface completeness information tailored for the autonomous reconstruction task. The global policy then takes this 2D quality map as input and outputs a 2D goal position specifying the subsequent ROI to guide the exploration of unknown or incomplete areas. To further accelerate the training process of the global policy, two dedicated auxiliary learning tasks (PCP and SPR) are used to promote the understanding of environmental dynamics and relations between robot movement and the scanning progress. More details about the global scanning module are given in Section 4.

Local scanning module. After the robot reaches the ROI, our method starts planning a sequence of NBVs to reconstruct the local region using 3D volumetric representation of both objects and environment from the current instance segmented point cloud. A set of candidate viewpoints is first selected for each object. Then by inspecting the completeness of each object and the feasibility of corresponding viewpoints through a multi-branch encoder, an appropriate scanning viewpoint (indicated by the red circle) is dictated by our learning-based local NBV policy after filtering out the unsuitable ones due to the blocking of the wall or obstruction caused by other furniture. This view selection operation repeats to gradually increase the reconstruction quality of objects within this region until all seen objects are scanned. More details about the local scanning module are provided in Section 5.

4 GLOBAL SCANNING MODULE

4.1 Global policy

2D quality map. To alleviate the inevitable information loss when projecting a 3D scene onto the image plane, we specially design a 2D quality map for the global scanning module to preserve as many reconstruction-related clues as possible. The 2D quality map $\mathcal{M}_t$ is represented by a multi-channel 2D image with resolution $M \times M$ and C channels and each cell belongs to $[0, 1]$, where $C = C_e + C_s + C_q$ with C_e corresponding to three commonly used exploration, obstacle, and trajectory channels to represent the explored, occupied area, and past robot positions, C_s indicating the number of semantic channels with useful object categories and their spatial distribution information [Chaplot et al. 2020a], and C_q referring to the new quality channels we designed to encode how well each region has been reconstructed.

Based on the key observation that the 3D scanner should be placed within a fixed range of scanning distances and scanning angles to obtain a more satisfying reconstruction result, we discretize the

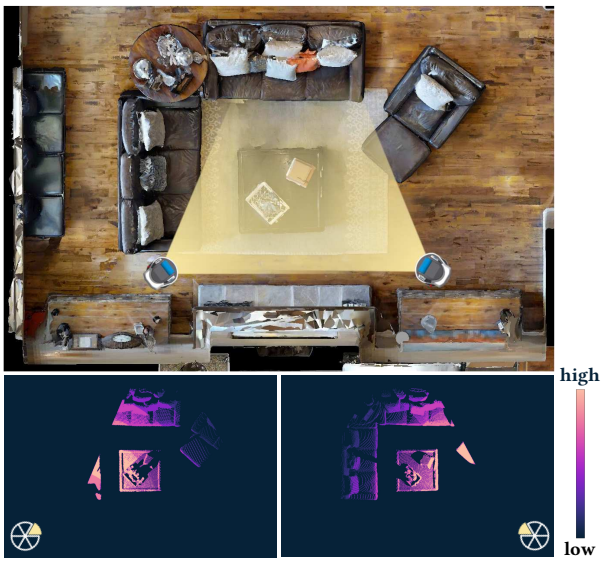


Fig. 3. Examples of two quality channels corresponding to different sensor directions, where the object points scanned with the distance closer to the predefined best scanning distance gets higher quality score.

scanning direction into C_q ranges $\{\phi_k\}_{k=1}^{C_q}$ and calculate the scanning quality within each range to constitute C_q quality channels. For channel k, the quality score q of each object point is defined as follows:

$$q = \begin{cases} (d_{\max} - |d - d_{\text{best}}|)/d_{\max} & if\ \mathbf{d}_{ro} \in \phi_k \\ 0 & otherwise \end{cases} \quad (1)$$

where d_{best} and $d_{\max}$ are the best and maximum valid range of the 3D scanner, respectively, d is the distance between the object point and the scanner, and $\mathbf{d}_{ro}$ is the direction of the vector from the scanner to the object point. Intuitively, these quality channels indicate along which direction and how far each object point is scanned by the robot. Figure 3 visualizes two exemplar quality channels corresponding to two scanning direction ranges.

Network architecture. Given the robot's current estimated position p_t^r, we adopt a coarse-to-fine strategy to apply two other transformations to $\mathcal{M}_t$ to get more localized information around p_t^r. An egocentric map of size $M' \times M' \times C$ is first cropped from the original top-down map based on p_t^r, and then a max-pooling operation is performed on the original allocentric raw map to get the same $M' \times M' \times C$ sized map. By stacking these two maps together, the observation $o_t \in [0,1]^{M' \times M' \times 2C}$ between the current state contains localized and holistic eyesight of the environment. This observation together with the current position p_t^r and previous goal position $a_{t-1}^{(g)}$ form the triplet input $\{o_t, p_t^r, a_{t-1}^{(g)}\}$ to the global policy. By taking p_t^r as input, the misalignment of the egocentric map and holistic map can be eliminated. The actor will then output the mean $\mu(s_t)$ and standard deviation $\sigma(s_t)$ of a Gaussian distribution to control the selection probability of points on the map,

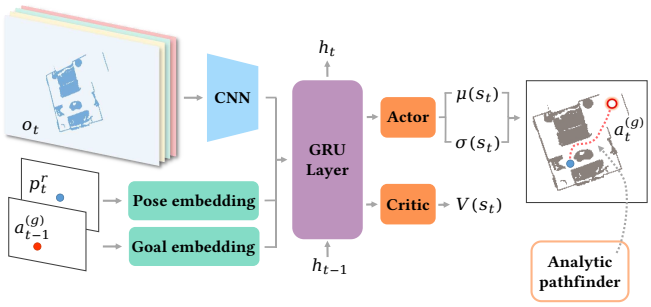


Fig. 4. The architecture of our global policy. The network takes the current observation o_t, current robot position p_t^r, and previous goal position $a_{t-1}^{(g)}$ as inputs. Processed by a series of embedding, convolutional, and GRU layers, a Gaussian distribution is outputted by the actor to guide the sampling of next goal position $a_t^{(g)}$, the path to which is dynamically planned using the analytic pathfinder.

and the subsequent ROI designated by a 2D point $a_t^{(g)}$ is randomly sampled according to the probability map.

Figure 4 shows the network architecture of our global policy. The actor and critic share the same encoder for extracting common features useful to both of them. In our implementation, o_t is encoded by a convolutional neural network (CNN) and then concatenated with the other two inputs processed by separate embedding layers. In addition, to make the agent aware of the current progress and thus avoid repetitive effort, we add an extra gated recurrent unit (GRU) layer to augment the network with memory, i.e., the hidden state h_t of the GRU at time step t.

With the subsequent ROI available, D* lite [Koenig and Likhachev 2002] algorithm is used to guide the robot to navigate to the specified goal while subjecting to the collision avoidance constraint. Note that D* lite is capable of fast replanning by making full use of the previous search result to generate a dynamic path when the goal position is pointed at an unexplored region and some unexpected obstacles are encountered.

4.2 Reward function

To train our global policy, we define the reward function as the combination of the following terms:

$$R^{(g)} = w_q * Q + w_e * E + w_g * G + w_s * S \quad (2)$$

where Q is the scanning quality, E is the exploration coverage, G is the goal reachability, and S is the execution slackness. The first two reward terms encourage the agent to explore the unknown area and reconstruct incomplete objects as much as possible, and the other two reward terms serve as penalties for unsatisfactory scanning strategies. Visual explanations of these terms are shown in Figure 5.

Scanning quality term. The scanning quality Q is defined as the increase of total scores of all quality channels:

$$Q = \sum_{0 \le i,j < M, k \in I_Q} m_{t+1}^{ijk} - m_t^{ijk} \quad (3)$$

where $m_t^{ijk} = max(m_{t-1}^{ijk}, q_t^{jlk})$ is the maximum quality score recorded in position i, j of the k-th channel, and I_Q is the index set of quality

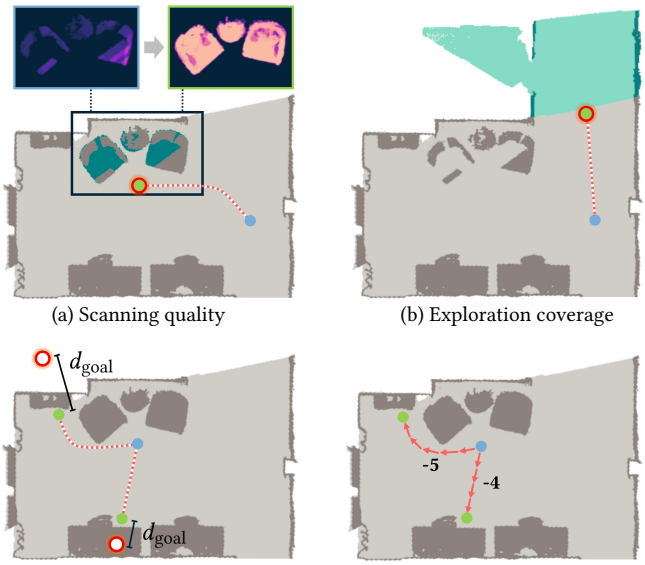


Fig. 5. Global policy reward terms. In each subfigure, the robot starts from the blue point to the red circle (goal position) and eventually arrives at the green point, with the traveling path illustrated by the dotted line. (a) Scanning quality term guides the robot to improve the reconstruction quality of those incomplete chairs, with the top two subfigures demonstrating the change of quality map before and after scanning.(b) Exploration coverage term encourages the robot to explore the unknown area, and the green area indicates the newly explored region after traveling to the goal point. (c) Goal reachability term prefers reachable goal positions by adding a penalty based on the distance d_{goal} between the goal position and the final arrived position. (d) Execution slackness term is a linear penalty proportional to the number of primitive movements that the agent takes to encourage an efficient trajectory.

channels. Intuitively, this reward term encourages the robot to scan an incomplete object from all directions, and thus it can be an indicator of the reconstruction progress and guide the robot to finish its ultimate goal — high-quality reconstruction.

Exploration coverage term. The exploration coverage E measures the increase of the total area in the map that is known to be free to traverse or occupied by an obstacle. We simply accumulate all the values in the exploration channel of the 2D quality map and then calculate the difference between two time steps:

$$E = \sum_{0 \le i,j < M, k = I_E} m_{t+1}^{ijk} - m_t^{ijk} \tag{4}$$

where I_E is the index of the exploration channel.

Goal reachability term. As a collaborative design of our goal-driven global policy, we introduce the goal reachability $G = \|\mathcal{P}_t^{-1} - a_t^{(g)}\|_2$ to measure the Euclidean distance between the location that the robot arrived at and the specified target, where $\mathcal{P}_t^{-1}$ is the last path waypoint that the agent has arrived at time step t. With the help of this regularization, the agent can learn to balance the trade-off between spotting the potentially valuable region and the waste of scanning effort into a blind alley.

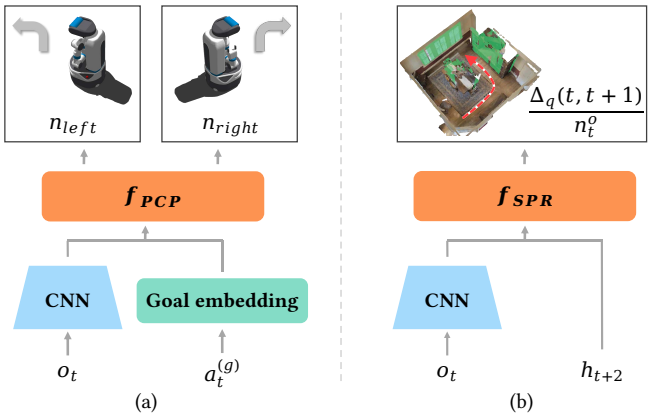


Fig. 6. Illustration of the proposed auxiliary learning tasks. (a) Path complexity prediction: the robot is asked to predict how many left turns n_{left} and right turns n_{right} it needs to navigate to the given goal point $a_t^{(g)}$ with the assistance of the current 2D quality map o_t. (b) Scanning progress recollection: the robot needs to recollect how many object surfaces it has scanned better during the previous step (highlighted in green) from the 2D quality map o_t and GRU hidden state h_{t+2}.

Execution slackness term. The execution slackness S gives $-n_p$ reward for the path execution sequence $\{a_l^{(p)}\}_{l=1}^{n_p}$, where $a_l^{(p)}$ denotes a primitive motion action such as moving forward, turning left and right, which encourages fast scene exploration and reconstruction in general.

4.3 Training with auxiliary learning tasks

Learning with auxiliary tasks can provide supplementary objectives for the agent to improve learning efficiency and transfer prior knowledge into the policy. Inspired by this, we incorporate two novel scanning-centric self-supervised tasks to boost the already acceptable global scanning policy further. The proposed tasks help train both quality map CNN and GRU part after it. In the following discussion, we denote f_{map} as the quality map encoder, and f_{goal} as the goal embedding mapping.

Path Complexity Prediction (PCP).. The incorporation of the analytic pathfinding module releases the network from learning primitive steps for navigation and obstacle avoidance. Nonetheless, the actual path that the robot will follow now becomes part of the environment's dynamics (i.e., environmental factors that are indirectly affected by the agent's actions), which impedes the agent learning the environment model and results in increased sample complexity. To make the robot attuned to these uncontrolled dynamics, we design an auxiliary goal-related task that can derive supervision from the robot's own navigation experience. As shown in Figure 6 (a), given the 2D quality map o_t and the corresponding generated goal point $a_t^{(g)}$, the agent is asked to predict how many left and right turns are needed when approaching it. Since the robot has to veer frequently when there are obstacles in its way, the number of turns is a proper and convenient estimate of the complexity of the planned path. Such estimation can also regularize the policy to output a more navigable target.

Technically, we preprocess o_t and $a_t^{(g)}$ with the same quality map encoder f_{map} and goal embedding layer f_{goal}. After concatenation, the inputs are fed through a fully-connected layer f_{PCP} to directly output logits for n_{left} and n_{right}. We discretize the number of possible turns into several intervals and use a cross-entropy loss with ground truth counts collected from past experience.

$$L_{PCP} = CrossEnt\left(f_{\text{PCP}}\left(f_{\text{map}}\left(o_t\right), f_{\text{goal}}(a_t^{(g)})\right), n_{\text{left}}, n_{\text{right}}\right) \quad (5)$$

Scanning Progress Recollection (SPR).. In the reconstruction-aware 2D representation, we use C_q devoted channels to encode the acquisition state of each observed object. Considering that it may not be straightforward to aggregate information in these channels, we introduce another auxiliary learning task to help the agent master this concept. As shown in Figure 6 (b), the agent is asked to predict the normalized increment of scanning qualities between two time steps, given the quality map o_t and GRU hidden state h_{t+2}. We adopt h_{t+2} rather than h_{t+1} originating from that we want the agent to remember the spots it has visited and build correspondence between the spatial structure of the reconstruction result and the scanning progress in consecutive steps, instead of to remember current observation (h_{t+1} takes o_{t+1} as input) and then calculate the quality increase by f_{SPR} head.

Likewise, a fully-connected layer f_{SPR} is used to predict the relative increment of qualities $\Delta_q(t, t+1)/n_t^o$, where $\Delta_q(t, t+1)$ denotes how many cells have an increased quality while n_t^o denotes all seen object cells during the execution of the actions at step t. We derive the supervisory labels from the rollout data and use MSE as the loss function.

$$L_{SPR} = \frac{1}{2}\left(f_{\text{SPR}}\left(f_{\text{map}}\left(o_t\right), h_{t+2}\right), \frac{\Delta_q\left(t, t+1\right)}{n_t^o}\right)^2 \quad (6)$$

Loss function. During the training of global policy, we jointly optimize the parameters of the quality map encoder, GRU module, and actor-critic heads, altogether θ_g, and auxiliary module parameters θ_a at the same time.

$$\begin{aligned} L^{(g)}(\theta) &= L_{\text{PPO}}(\theta_g) + L_{\text{aux}}(\theta_a) \\ L_{\text{PPO}}(\theta_g) &= L_{\text{actor}} + \alpha_{\text{critic}} L_{\text{critic}} - \alpha_{\text{entropy}} H_{\text{entropy}} \\ L_{\text{aux}}(\theta_a) &= \alpha_{\text{PCP}} L_{\text{PCP}} + \alpha_{\text{SPR}} L_{\text{SPR}} \end{aligned} \quad (7)$$

where L_{PPO} is the original loss of PPO, which is further comprised of surrogate loss from actor (used to estimate policy gradient), regression loss from critic, and entropy loss to encourage exploration in action space, and L_{aux} is the auxiliary task loss.

5 LOCAL SCANNING MODULE

5.1 Local policy

Scanning process. Once the robot reaches the ROI, it starts to improve the reconstruction quality of objects within this region. Since objects in the ROI may not be fully discovered at the current state, it is hard to plan a optimal view path at the beginning, as new objects might be spotted during the scanning process. Hence, we progressively select k partially scanned objects within the region and add newly discovered objects into the unscanned objects bank until all objects in the bank have been scanned. These k selected objects are identified by a heuristic strategy based on distances and object sizes, which favors objects that are larger and closer to current robot position. Then, we uniformly sample n_v points around each object to constitute the viewpoint candidates for NBV selection. We set k to three based on experience to avoid voracious planning with fewer objects and also be able to design a smoother viewpoint sequence between objects. These candidate viewpoints form the action space together with a *STOP* action. Note that the probability of candidate viewpoints of vacant objects is set to zero by the feasibility mask when the number of selected objects is less than k. The quality refinement of the selected objects continues until the robot is satisfied with the reconstruction and selects the *STOP* action or no feasible unscanned candidate viewpoint lefts. Detailed scanning process is provided in the supplementary material.

Coarse-to-fine 3D volumetric representation. After identifying the objects for reconstruction, two types of voxel grids $\mathcal{V} = \{\mathcal{V}^o, \mathcal{V}^r\}$ are constructed as the inputs for the local scanning module. Firstly, points belonging to the selected objects are segmented and discretized individually to obtain k voxel grids each for a single object. We then concatenate them to form a volumetric representation $\mathcal{V}^o \in [0,1]^{L\times L\times L\times k}$. The vacant objects will be occupied with one when the number of remaining objects is less than k. Secondly, environment points encompassing these objects are also extracted as a larger voxel grid $\mathcal{V}^r \in [0,1]^{L'\times L'\times L'}$. These two representations are prepared for the DRL branch to select NBVs for completing the objects and for the feasibility predictor to filter out infeasible viewpoints, respectively. Our insight is that evaluating both the profit and the feasibility of a viewpoint is intractable for a single network. The former focuses on how many new object surfaces can be observed from the given view and only relies on the fine-grained voxel grid of a single object, while the latter involves discarding unreachable views or views that cannot look unobstructed at the object and requires a holistic picture of the region geometry. Thus we develop a masked local scanning policy with a DRL branch and a feasibility predictor to achieve each objective separately, and then collectively determine a succession of optimal scanning views.

Feasibility mask. It often occurs that the viewpoints are blocked by the wall or unreachable when their positions are occupied by other objects. By utilizing the feasibility predictor, the agent can avoid futile attempts to go to the unfeasible viewpoints and make the DRL branch stay focused on improving reconstruction quality. Besides, since the feasibility predictor shares the feature extractor of the objects with the DRL branch, it can further serve as an auxiliary learning task to accelerate the training with strong supervisory signals. The feasibility predictor takes both $\mathcal{V}_t^o$ and $\mathcal{V}_t^r$ as its inputs and outputs a probability map over all candidate viewpoints. Note that the probability of viewpoints of vacant objects is set to zero directly. The NBV action is then sampled by first conducting an element-wise product of the feasibility mask and output of the actor and re-normalizing the probabilities for all actions.

Network architecture. Figure 7 presents the overall structure of the proposed local policy. At time step t, the inputs of the DRL branch are denoted as $\{\mathcal{V}^o_{[t-2:t]}, \{p_i^v\}_{i=1}^{n_v}, a^{(l)}_{[t-3:t-1]}\}$, where $\{p_i^v\}_{i=1}^{n_v}$ represents the relative positions of these candidate viewpoints to the

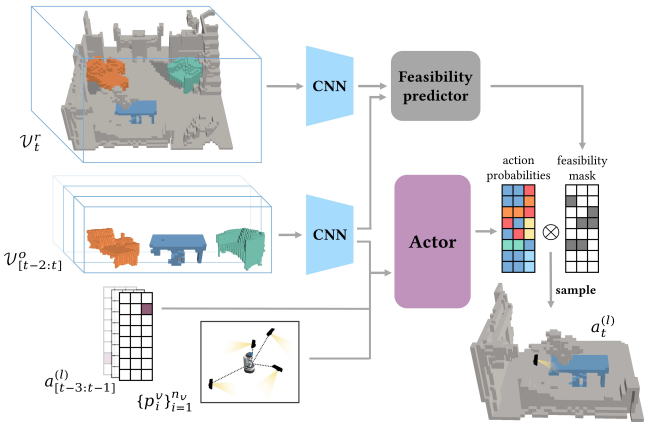


Fig. 7. The architecture of our local policy. When k objects are determined, both the stacked object $\mathcal{V}^o_{[t-2:t]}$ and the surrounding region $\mathcal{V}^r_t$ are voxelized. The DRL part (actor head) and feasibility predictor predict two probability maps over all candidate viewpoints and collectively decide the final probabilities for sampling the NBV action $a^{(l)}_t$.

agent's current location. While $\mathcal{V}^o_{[t-2:t]}$ represents the stacked voxel grids of objects in three consecutive steps providing a form of short-term memory. Likewise, already chosen viewpoints of the past three steps $a^{(l)}_{[t-3:t-1]}$ are stacked and included as well. Compared to the global module, the local scanning episode typically ends after 7 to 9 actions (on average, 2 to 3 views for every single object with $k = 3$). There is not much gain in using GRU in such a short-horizon scenario but incurs much higher learning difficulty. Thus, we stick to this stacked design in our following experiments.

5.2 Reward function and network training

When it comes to local policy training, we include three terms in its reward function:

$$R^{(l)} = w_v * V + w_d * D + w_s * S \tag{8}$$

where V is the reconstruction coverage, S is similar as before, the viewpoint slackness, and D is the moving distance.

Reconstruction coverage term. In local NBV scanning, the primary goal is to reconstruct the objects as completely as possible. So we only include a single positive reward V that is defined as the growth in occupied voxels for all visible objects.

$$V = \sum_{0 \le i,j,l < L, 0 \le o < k} v^{ijl,o}_{t+1} - v^{ijl,o}_t \tag{9}$$

where $v^{ijl,o}_t$ is the 0-1 value in the corresponding spatial position and object of $\mathcal{V}^o_t$.

Viewpoint slackness and moving distance terms. With the same purpose as in the global module, we also add the viewpoint slackness reward S to prevent the agent from being sluggish. Moreover, another negative reward is introduced to spur smooth view trajectories by punishing the agent for moving back and forth. That is, we define moving distance reward as the Euclidean distance between two consecutive NBVs $D = \|a^{(l)}_{t+1} - a^{(l)}_t\|_2$.

Network training. One of the essential components of the local scanning module is the feasibility mask predictor. To train it, we simply resort to the simulator oracle to tell us whether a given viewpoint is navigable and whether an observation containing the selected object can be obtained. These ground truth labels are used to train the feasibility mask predictor as a multi-label classification task. When training the local DRL network, we use the ground truth feasibility mask to modify the output probabilities of the actions and only deploy the predictor at test time.

The final loss of the local policy can be concluded as follows, with θ_o and θ_r representing the parameters of object and region encoders, respectively, and θ_l and θ_m represent the parameters of local policy and mask predictor's heads.

$$L^{(l)}(\theta) = L_{\text{PPO}}(\theta_o, \theta_l) + \alpha_{\text{mask}} L_{\text{mask}}(\theta_o, \theta_r, \theta_m) \tag{10}$$

where L_{mask} is the cross-entropy loss.

6 RESULTS AND EVALUATION

6.1 Experiment setup

Simulation system. We develop our system based on the Habitat platform [Savva et al. 2019], a high-performance 3D simulator wrapped within an extendable high-level API. It ships with configurable sensors (e.g., RGB, depth, semantic) and several predefined tasks (e.g., navigation, instruction following, question answering) for training and benchmarking embodied AI algorithms. Because of the novelty of our setup, there is no existing task and corresponding configuration for the autoscanning agent. We thereupon develop two new tasks (i.e., global exploratory scanning and local object reconstruction) into the Habitat and conduct both training and evaluation based on them. Note that the ground truth camera poses and instance label of objects from this simulated environment is replaced by the estimations from the SLAM module at evaluation phase and the end of training phase for fine-tuning.

Dataset. Our experiments are carried out on Gibson [Xia et al. 2018] and Matterport3D (MP3D) [Chang et al. 2017] datasets, consisting of a diverse set of 3D reconstructed real-world building-scale scenes. The consequent collection contains a total of 111 scenes (90 MP3D and 21 Gibson), which are further divided into 78 scenes (61 MP3D and 17 Gibson) for the training set and 33 scenes (29 MP3D and 4 Gibson) for the test set. These houses have an average indoor area of 275.2 m^2 and 22.3 regions, some of which have exterior structures such as balconies, gardens, fences, etc. Under our problem assumption, there is a unique ground plane for the target scene. Accordingly, we regenerate all navmeshes that define which area the agent can pass through to inhibit the agent from accidentally climbing up the stairs. Apart from this, all scenes are left unchanged even if there still exist detrimental factors for training, such as data missing or uneven ground.

As the difficulty of accomplishing effective exploration and reconstruction generally increases with the number of rooms and accessible areas, we split our test set into four parts according to the navigable area of the scene: *small* (9 scenes with 17-97 m^2),

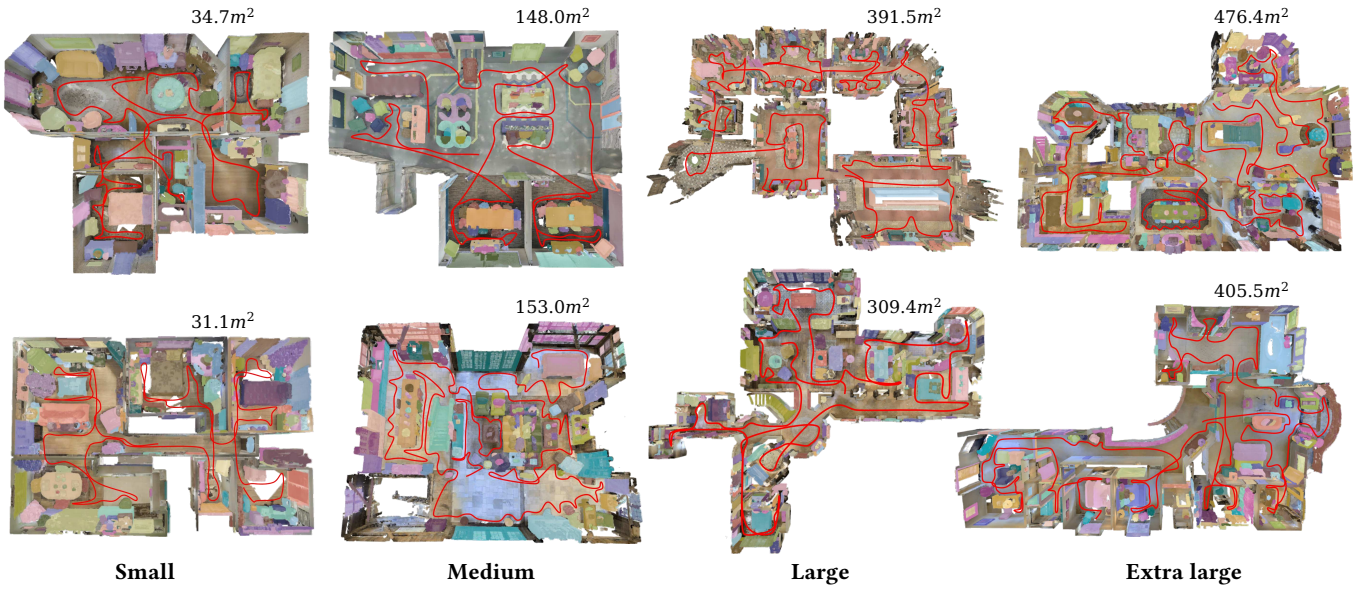


Fig. 8. Visual results of simulated scanning in different splits of the test set.

medium (10 scenes with 100-224 m^2), *large* (8 scenes with 264-391 m^2) and *extra large* (6 scenes with 405-1371 m^2), to fully reflect the performance of the proposed agent.

6.2 Qualitative results

Figure 8 shows a gallery of visual results of the proposed approach for simulated scanning. We pick two scenes from each split of our test set to demonstrate the scalability of our method. For each scene, we show a birds-eye view of the reconstructed scene, recognized objects in different colors, and the final path traveled by the agent. With the proposed system, different types of indoor scenes with real-world complexity can be completely and efficiently reconstructed and understood in a fully automatic manner, thanks to the awareness and utilization of the global structure with the global policy as well as the effective view selections with the local policy.

Figure 9 illustrates two typical infeasible viewpoints masked out by our feasibility prediction. Since the objects are mainly placed near the wall or other objects, it is pretty common that these objects have one or more viewpoints blocked by the wall or occupied by other objects. By utilizing the feasibility predictor, the agent can avoid futile attempts to go to the unfeasible viewpoints and make the DRL branch stay focused on improving reconstruction quality.

Figure 10 demonstrates some emergent scanning strategies learned by the proposed global policy. During the scanning, the global policy selectively generates either exploration-oriented goals (Figure 10(a)) or reconstruction-oriented goals (Figure 10(b)). Compared to frontier-based exploration methods, our method can predict the layout of the unknown regions, therefore, yielding more efficient goal positions and expanding the explored boundary rapidly. On the other hand, by specifying goals near incomplete objects, the global policy can cooperate with the local policy to effectively increase the object reconstruction quality.

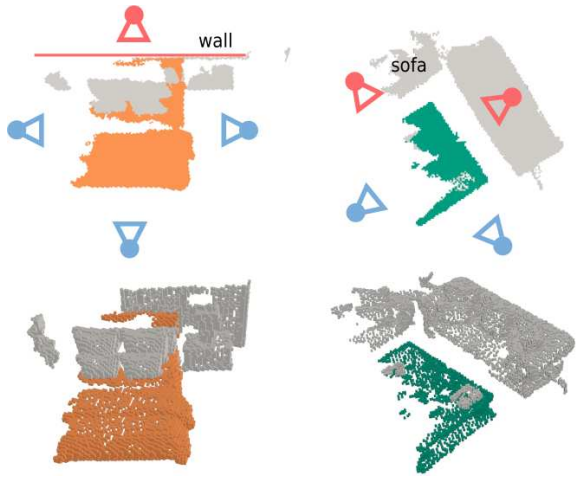
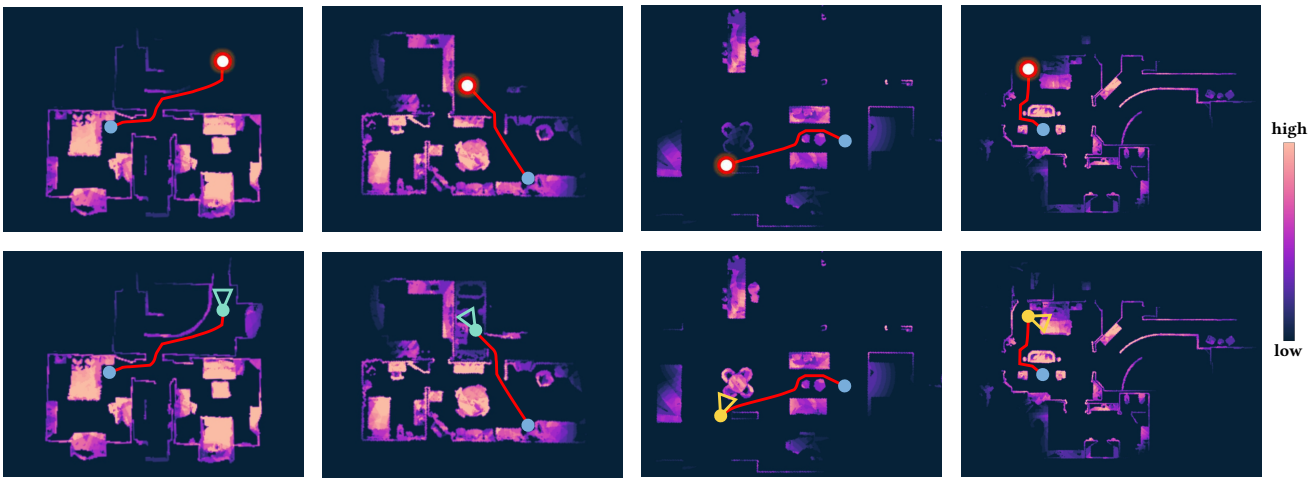


Fig. 9. Typical infeasible viewpoints masked out by our feasibility prediction. The infeasible viewpoints are colored in red while the feasible ones are colored in blue. For each example, we show the 3D volumetric representation on the bottom and the corresponding 2D top-down view on the top to see the viewpoint distribution more clearly. (a) Viewpoint that is reachable but invisible due to the blocking of the wall. (b) Viewpoints that are visible to the object but inaccessible as their positions are occupied by two sofas.

Statistics. In Table 1, we report the assorted statistics of our method and the scene splits. The results are measured and averaged over all the test scenes within each split. The majority of the time is spent on robot navigation and is roughly proportional to the scale and complexity of the scene accordingly. The number of planned ROIs is recorded when the scene reconstruction rate exceeds a given



(a) Exploration-oriented goals **(b) Reconstruction-oriented goals**

Fig. 10. Example goal points (red circles) generated by our global policy. For each example, the input quality map with blue dot for start point and red line for navigation path is shown on the top, and the one obtained after the ScanBot (cyan/yellow camera icons) arrives the ROI is shown on the bottom. (a) Two exploration-oriented goals that help explore unknown area and expand the boundary efficiently. (b) Two reconstruction-oriented goals that help increase the reconstruction quality of discovered objects.

Table 1. Statistics of our method on various test set splits. For each split, we report the navigable area and number of regions per floor, number of ROIs designated by the global policy, number of NBVs selected by the local policy, total time to finish the scanning, and total travel distance of the agent.

Scene	Area	Region	ROI	NBV	Time	Distance
Small	$35.2m^2$	7.8	8.4	60.1	1.4 *min*	13.6 *m*
Medium	$66.2m^2$	8.3	16.0	100.8	3.3 *min*	31.4 *m*
Large	$199.5m^2$	14.9	39.7	226.3	7.7 *min*	81.4 *m*
Extra large	$271.9m^2$	11.9	47.8	271.6	9.8 *min*	99.9 *m*

threshold or a maximum is reached. Therefore, it is also proportional to the scene complexity.

6.3 Evaluation metrics

The arrangement of the performance evaluations mainly considers three vital aspects of autonomous scene reconstruction: 1) How many unseen regions have been successfully explored? 2) How well have the contained objects been reconstructed? 3) How efficient is our method? To quantitatively evaluate these aspects, we adopt the following metrics:

• *2D map completeness.* The 2D map completeness [Chaplot et al. 2020b] is commonly used to measure the percentage of ground truth map that the robot has explored.

$$MC = \frac{1}{|\Gamma|}\sum_{c\in\Gamma}\delta_{\text{explore}}(c) \tag{11}$$

where Γ is the set of free cells in the ground truth map and δ_{explore} is a Dirac delta function indicating whether the agent has explored the cell c.

• *3D scene completeness.* The scene completeness is designed to assess the final 3D reconstruction coverage of the entire scene. We follow [Dong et al. 2019] to compute an asymmetry distance to account for the completeness of the reconstruction result:

$$SC = \frac{1}{\sum A(s_1)}\sum_{s_1\in\mathcal{S}^*} A(s_1)\min_{s_2\in\mathcal{S}}\|s_1-s_2\|_2^0 \tag{12}$$

where $A(\cdot)$ measures the area of a vertex and $\mathcal{S}^*$ denotes the ground truth scene surface. The l_2^0 norm $\|\mathbf{x}\|_2^0$ is used to differentiate between inliers and outliers via the parameter ϵ:

$$\|\mathbf{x}\|_2^0 = \begin{cases} 0 & if \quad \|\mathbf{x}\|_2 < \epsilon \\ 1 & otherwise \end{cases} \tag{13}$$

• *3D object completeness.* We adopt the metrics from *Object-Aware Scanning Benchmark* (OASC) [Liu et al. 2018] to evaluate the object acquisition after autonomous scanning. The object completeness is computed as:

$$OC = \frac{1}{|\mathcal{V}^*|}\sum_{v\in\mathcal{V}^*}\delta_{\text{detect}}(v)\cdot\delta_{\text{valid}}(v) \tag{14}$$

where $\mathcal{V}^*$ represents visible voxels of all objects of the ground truth reconstruction. δ_{detect} and δ_{valid} are used to differentiate voxels that both belong to the detected objects (by a pretrained Mask-RCNN) and have been scanned within a valid sensory range.

• *3D object quality.* Similar to object completeness, the object quality [Liu et al. 2018] is measured with an extra quality score:

$$\begin{aligned} OQ &= \frac{1}{|\mathcal{V}^*|}\sum_{v\in\mathcal{V}^*}\delta_{\text{detect}}(v)\cdot\delta_{\text{valid}}(v)\cdot q(v) \\ q(v) &= e^{-\frac{\psi^2(\mathbf{n}_v,\mathbf{d}_{cv})}{\eta^2\psi_t}}\cdot e^{-\frac{(d(v,c)-d_{\text{best}})^2}{\eta^2 d_{\max}}} \end{aligned} \tag{15}$$

where $\psi(\mathbf{n}_v,\mathbf{d}_{cv})$ is the angle between the normal at voxel v and the viewing the vector from camera c to v and $d(v,c)$ is the distance between the voxel and camera.

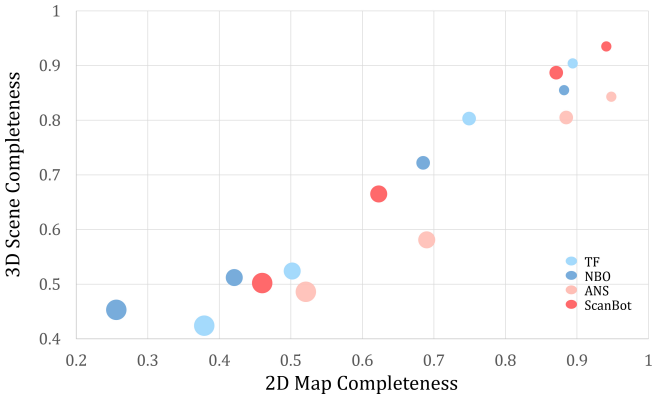


Fig. 11. Comparison on 2D map completeness and 3D scene completeness with all baselines. Different circle sizes denote different splits of the test set, i.e., bigger circle indicates larger scene.

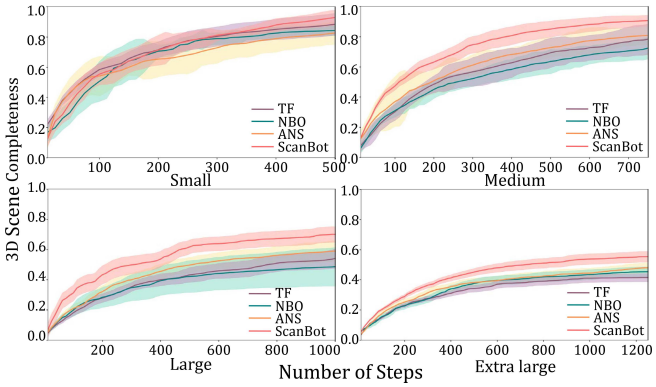


Fig. 12. Comparisons on 3D scene completeness rate with respect to the number of moving steps on four representative scenes with all baselines.

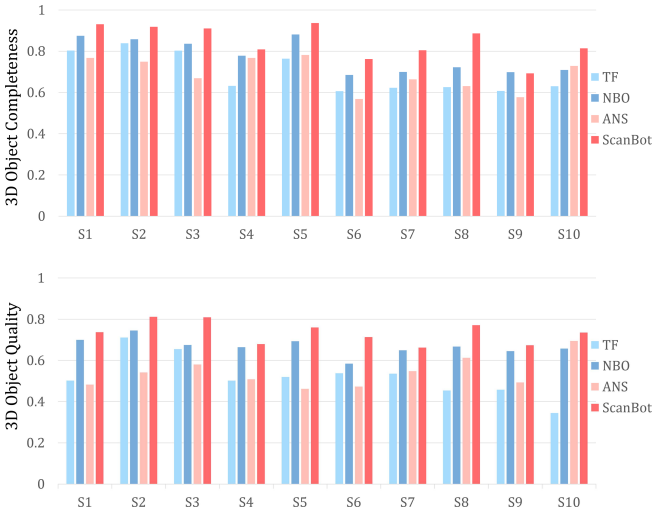


Fig. 13. Comparisons on 3D object completeness and 3D object quality with all baselines.

• *Traveled distance.* To quantify the scanning efficiency, we use the movement distance of the robot in an episode as a measure.
• *Running time.* The running time is recorded with the robot's speed set to 0.25m/s.

6.4 Comparison to state-of-the-art methods

Baseline methods. To quantitatively evaluate the performance, we compare our method with both state-of-the-art rule-based autonomous reconstruction methods and learning-based exploration methods using the metrics defined in Section 6.3. The first work [Xu et al. 2017], referred to as *TF*, is a geometry-based algorithm that drives an autonomous reconstruction by time-varying tensor fields. The second work [Liu et al. 2018], called *NBO*, is a representative method that makes use of object-level semantics and can achieve exploration, reconstruction, and object recognition in one navigation pass. To verify the effect of our reconstruction-oriented designs, we also involve a learning-based exploration approach [Chaplot et al. 2020b], referred to as Active Neural SLAM (*ANS*), which uses similar DRL techniques but aims for a different goal. For a fair comparison, we provide ground-truth camera poses during training for acceleration and use estimations during testing for all the above methods. One thing to note here is that both TF and NBO are reconstruction-orientated methods like ours, thus we all use the SLAM module to get the estimated camera pose, while ANS is exploration-orientation with its own pose estimation module.

Comparison on scene exploration and reconstruction. In Figure 11, we compare against those three baselines in terms of average 2D map completeness and 3D scene completeness on four splits of test sets. As we can see, all four approaches achieve good 2D map completeness in scenes of small sizes, but the performance of the learning-based methods (ANS and ScanBot) drops much slower than the other two when the environments become larger and harder to explore efficiently. Despite the slightly poor performance compared to ANS when purely considering the 2D map completeness, ScanBot performs significantly better than ANS with regard to 3D scene completeness, thanks to the additional scanning progress information encoded in quality channels and the quality reward that advocates proactively object completion.

The average 3D scene completeness of our method is consistently better than baseline methods over all splits. We ascribe this to an essential and valid balance attained by our approach between efficient exploration and high-quality object reconstruction. Figure 12 demonstrates the 3D scene completeness rate over the traveled distance for four representative scenes from different splits. To reduce the bias introduced by the robot's placements, we collect the results from all available episodes and visualize the mean (solid line) and standard deviation (transparent overlay). The plots further validate that our method can reconstruct different types of scenes faster and result in a fuller reconstruction than other approaches.

Comparison on object reconstruction. To remove the disparities in the exploratory abilities of different methods, we choose ten scenes (S1-S10) from small and medium splits to report the result of

Table 2. Performance of different methods on scanning efficiency measured by traveled distance (in meters) and running time (in minutes) in representative scenes.

Scene	Traveled distance				Running time			
	TF	NBO	ANS	ScanBot	TF	NBO	ANS	ScanBot
S1	16.5	19.4	15.8	**14.4**	2.0	10.2	1.7	**1.3**
S2	12.8	18.5	15.1	**12.4**	**1.1**	11.9	1.4	1.2
S3	**14.6**	22.1	17.2	15.2	2.0	12.8	1.8	**1.4**
S4	17.6	21.1	18.3	**15.9**	1.8	18.8	1.9	**1.5**
S5	13.8	16.2	12.3	**10.4**	1.2	14.7	**0.9**	1.0
S6	**34.4**	53.6	39.3	35.8	3.5	23.4	3.6	**3.3**
S7	24.1	34.0	25.8	**21.3**	3.0	16.9	2.7	**2.3**
S8	24.6	26.4	20.0	**16.3**	2.7	13.8	2.3	**1.6**
S9	**38.8**	57.1	47.1	40.8	4.1	28.4	4.9	**3.9**
S10	34.6	47.4	38.9	**32.4**	3.3	23.9	3.5	**2.9**

object reconstruction. It is easy to see in Figure 13 that our object-aware scanning leads to complete and high-quality reconstruction of objects innately. An alternative NBV algorithm is involved in the NBO approach to realize a similar object-centric reconstruction, however, it depends on a synthetic 3D model database to provide the necessary prior knowledge for shape recognition and alignment. In contrast, we use the learning-based methods to directly recognize objects and learn shape priors from raw scanned data and hence achieve robust and accurate reconstruction results in complex and cluttered environments.

Comparison on scanning efficiency. By making the most of high-level semantic clues, we posit that the agent can achieve a thoughtful and efficient scanning plan of the overall environment. To substantiate this idea, we measure the time consumption of the system for achieving fixed scene reconstruction and 2D map completeness rates (0.7 for scene reconstruction and 0.6 for 2D map completeness) and total traveled distance of the robot, respectively, in S1-S10. In Table 2, we report the comparisons of the running time and the movement distance of the robot between our method and three baseline methods. Most of the time, our method can achieve faster and more efficient scanning even compared to the object-agnostic approaches (TF and ANS). The object feature extraction and matching component severely slow down the speed of the NBO, but our method is immune to these due to the fast network inference when testing.

6.5 Ablation Studies

In this section, we perform ablation studies to evaluate the contribution and effectiveness of individual components of our method.

The importance of quality channels for global policy. In our global scanning module, we monitor the scanning process by the quality channels of the map. To evaluate the effectiveness of this quality map, we compare the performance of agents trained with and without quality channels. The results are verified with 2D map completeness and 3D object quality, as shown in Figure 14. The omits of quality channels will not affect the 2D map completeness since the exploration mainly relies on the contour and structure of the scene,

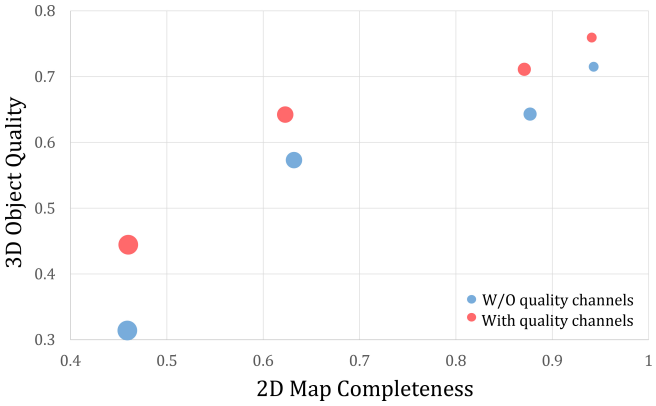


Fig. 14. Ablation study on the effect of quality channels on the performance of 2D map completeness and 3D object quality. Two agents are trained with all quality channels or without quality channels for this experiment. Different circle sizes denotes different size of test set, i.e., bigger circle indicates larger scene.

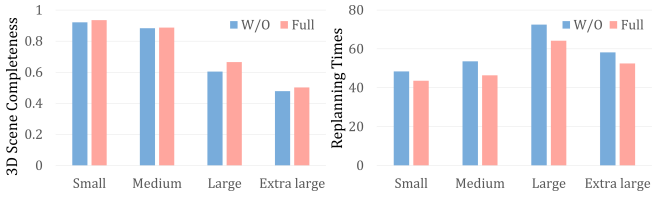


Fig. 15. Impact of goal reachability on 3D scene completeness and replanning time tested on two variants: with and without goal reachability reward term.

which can be derived from the occupancy and semantic channels. However, the 3D object quality significantly drops as the agent is unconscious of the scanning progress and hard to figure out the incomplete objects. This situation exacerbates the growing scene size, leading to the reduced chances of scanning objects.

The importance of reachability reward for global policy. The reachability reward term G of the global policy is introduced to implicitly remove or restrict the unreachable goals in the decision space. To determine its efficacy, we retrain the global policy without G, and compare their performance on 3D scene completeness and a supplementary metric *replanning times* as demonstrated in Figure 15. The variant without reachability reward cannot reach the similar performance obtained by the original method when the environment scale becomes large, in which case an inaccessible goal wastes the robot's energy on finding the way to it. For *replanning times*, we count how many times the path planning module has to regenerate the navigation path due to unexpected obstacles. In reality, an autoscanning robot prefers smooth paths for the ease of both robot control and frame-to-frame registration. Therefore, our method that needs relatively less path replanning is safer and more suitable for practical usage.

The importance of auxiliary tasks for global policy. In the following, we aim to examine the efficiency of individual auxiliary tasks

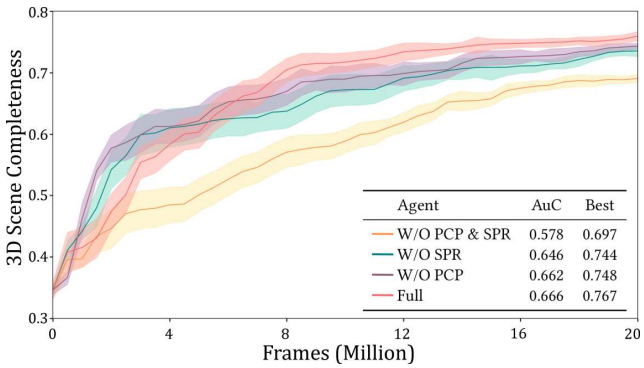


Fig. 16. Learning curves of variants of auxiliary tasks aggregated from three random seeds.

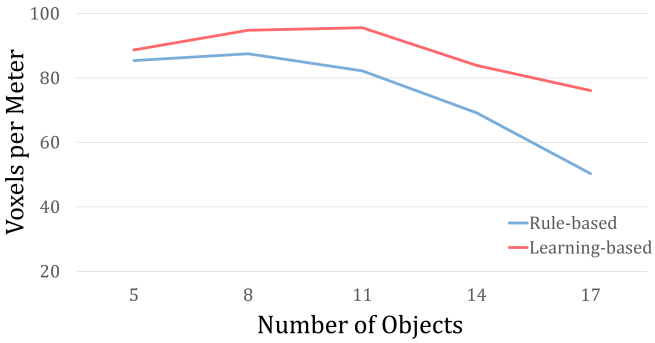


Fig. 17. Effectiveness of learning-based local policy comparing to rule-based method [Liu et al. 2018]. Our learning-based method needs less travel distance to complete the reconstruction of objects within the room, and the difference becomes more clear with the increase of number of objects within the test room.

and their combination. We construct four variants including: only trained with RL reward (W/O PCP & SPR), with RL reward and PCP task (W/O SPR), with RL reward and SPR task (W/O PCP), and with all objectives (Full). Each variant is given a fixed budget of 20M *frames* with three random seeds as its training resource (by a frame, we mean that one global scanning action is finished in the simulation environment), with the hope that a dominant method will achieve a better result when training is completed or achieve the same result faster. To evaluate them, we measure the average scene reconstruction rate on the test set every 0.5M during training and plot the resultant statistics in Figure 16. With the learning curves, we additionally compute the area under them (AuC) together with the best reconstruction rate to reflect the actual disparity.

Both the PCP and SPR tasks boost the training of our global policy. However, these improvements are not quite complementary when we add them together. The multi-task variant loses its performance at the early stage and only catches up after 6M frames. Our hypothesis is that the multi-task training manner interferes with the learning of the primary scanning task. It takes time for the agent to be conditioned on other tasks and finally benefit from them.

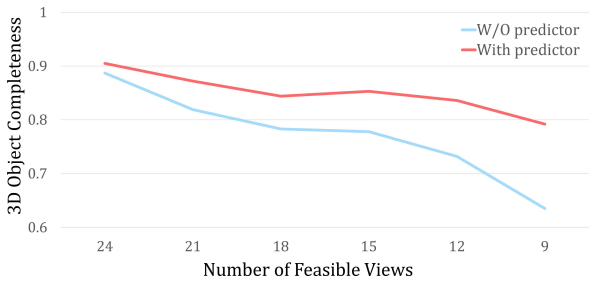


Fig. 18. Effectiveness of the feasibility mask predictor. Finished by taking a similar number of actions in an episode, the local policy with feasibility prediction selects more feasible viewpoints compared to the setting without the predictor, which proves that the feasibility predictor highly increases the scanning efficiency.

The importance of learning-based local policy. The local policy takes shape priors and relationships between objects into consideration for NBV selection rather than relying on the information gain-based strategy. Besides, the agent is also asked to smooth the viewpoint sequence to minimize its movement distance through two negative reward terms. These all contribute significantly to an efficient view selection strategy that helps increase the reconstruction of objects faster, especially in complicated environments. Figure 17 shows the measurement of reconstructed object voxels per meter compared with the NBO method [Liu et al. 2018]. Note that the NBO needs an object shape dataset as a prior to justify the completeness of the object and decide the next NBV while this is not required by our method. We performed this experiment by randomly selecting 50 rooms from our test set and using our learning-based as well as NBO's rule-based local policies to control the agent to scan objects within the given room. From these results, we can see that our method needs less travel distance to finish the reconstruction of the objects within the room, whereas the performance of the NBO saturates in small rooms with fewer objects. When more objects exist, the proposed local policy successfully leverages the spatial relationships among multiple objects to reduce the redundant movements of the robot.

The importance of feasibility prediction for local policy. Since the objects are often placed near the wall or other objects, the feasibility mask predictor is an essential ingredient of the local scanning module to facilitate an effective training of the DRL policy. This idea is confirmed by a comparison of the 2D map completeness between the full method and a variant without the feasibility predictor shown in Figure 18. This alternative version is trained using only the voxel grid of the region as input and is guided by the same reward. As observed from the results, the performance of this variant is disrupted when there exist more infeasible viewpoints around the objects. We also observed obvious instability when training it, which confirms the difficulty for a single network to learn object reconstruction and view feasibility jointly.

6.6 Domain Generalization

Dataset adaption. There are marked variations in the data distribution between Gibson and MP3D datasets. To name a few examples,

Fig. 19. Example results of directly adapting our method for real-world scene reconstruction with blue dots for start points, red circled points for planned ROI, and red lines representing navigation paths.

Table 3. Performance of dataset adaption measured by 2D map completeness (MC), 3D scene completeness (SC), and 3D object completeness (OC).

Agent	Small			Medium		
	MC	SC	OC	MC	SC	OC
Mixed	0.933	0.926	0.842	0.874	0.879	0.766
MP3D-only	0.872	0.903	0.836	0.865	0.834	0.748

Table 4. Performance of scale adaption measured by 3D scene completeness (SC), and 3D object quality (OQ).

Method	Agent	Large		Extra Large	
		SC	SQ	SC	SQ
ScanBot	All	0.665	0.642	0.502	0.444
	Small&Medium	0.603	0.598	0.446	0.403
NBO	All	0.512	0.539	0.453	0.392

scenes in MP3D have some open regions where the agent cannot sense the boundary due to the lack of data while scenes in Gibson are almost closed and bounded; MP3D has some rare environments (e.g., chapel, theater, museum, etc.) while Gibson has principally residential buildings. This inconsistent distribution allows us to investigate the domain generalization performance and scalability of our proposed system by training all models on one domain but evaluating them on the other. This alternative training set consists of 72 scenes from MP3D, and the test set has exclusive access to the 10 scenes from Gibson. We also move 6 training scenes from Gibson into the test set for the original mixed version to maintain a training set of the same size. Note that the test set now only has scenes of small and medium sizes, so we only report the results of scenes with these two scales in the following table.

From the comparison shown in Table 3, the ScanBot trained only on the MP3D dataset exhibits robust generalized behaviors in unseen environments from Gibson. For both scene-level and object-level metrics, the performance only has a small gap between the agents trained on mixed and MP3D-only datasets.

Scale adaption. Since the scale of scenes changes dramatically from small scenes ($< 100m^2$) to extra large scenes ($> 400m^2$) as illustrated in Figure 10, it is interesting to investigate the scalability of our method by training with small and medium scenes but testing with large and extra large scenes. In this experiment, we re-split the collected scenes based on the size of their navigable area, resulting in 50 training scenes (33 MP3D and 17 Gibson) with an average area of 120.9 m^2 and 14 testing scenes (MP3d only) of 512.5 m^2. The comparison of the results of these two models is shown in Table 4, with the performance of NBO [Liu et al. 2018] as a reference. As the layout of objects and structures inside a room or between adjacent rooms is similar, our global policy can still adapt strategy based on the patterns learned from scenes of small and medium sizes to those unseen large and extra-large scenes. Moreover, our local policy focuses on completing objects in the ROI with a fixed range ($5m \times 5m \times 2.5m$ in our experiments), which makes the policy quite robust to the scale of the entire scene as long as there are similar patterns in local object distribution. We can see that both 3D scene completeness and object quality are slightly dropped and still better than the baseline trained with scenes from all sizes, which proves that our method can be extended to large scene scales.

Real-world adaptation. We also test the generality of our method to real scenarios by conducting real-world reconstruction with a Fetch robot equipped with an Azure Kinect DK camera. The Laser and IMU sensor data of Fetch are also incorporated to improve the accuracy and robustness of the camera pose estimation. The reconstruction results of three unknown indoor scenes, including one meeting room, one coffee shop, and one office, are shown in Figure 19. The detailed reconstruction process of the coffee shop is demonstrated in the accompanying video. We can see that our method obtains faithful reconstructions of those three real scenes. Although the holes brought by the transparent and specular materials are inevitable due to the loss of depth data, futile attempts to reconstruct these areas can be avoided thanks to the trajectory channel introduced in the 2D quality map and the memory brought by the GRU unit of global policy. Moreover, we find that the estimated camera pose is sometimes inaccurate due to the noisy inputs from the depth sensor, which introduces misalignment and outliers into the 2D quality map, however, our global policy is still capable of localizing

exploration or reconstruction-oriented ROI as the corresponding regions in the 2D quality map are relatively stable.

7 CONCLUSIONS

We present a hierarchical DRL-based approach for autonomous scanning, including exploration, understanding, and reconstruction of large-scale 3D environments. At the heart of our approach is an elaborate divide-and-conquer scheme that brings the strengths of DRL into both global ROI planning and local NBV planning. To save the computational cost of DRL, we propose a mixed 2D-3D representation with different spatial scales and tailored information. On the one hand, with the help of a reconstruction-aware 2D quality map, the global exploratory scanning module spots a ROI that deserves further exploration and scanning by high-level semantic information and the awareness of current scanning progress. On the other hand, by utilizing region-level and object-level 3D voxel grids, the masked local NBV scanning module plans a series of detailed sensor viewpoints to constantly raise the reconstruction quality of discovered objects. Through comprehensive experiments, we prove both the favorable features of our proposed approach when compared with state-of-the-art alternatives as well as the reasonability and practicability of the design of each component.

Limitations. As a first attempt at incorporating DRL into autoscanning, there remain some limitations of our current solution. Our training strategy relies on a semantically annotated scene dataset to save the already stringent computing resources (the Mask-RCNN semantic segmentation module is only enabled when testing). However, this constraint reduces the number of available scenes for learning and may decrease the final performance as well since the trained DRL policies are oblivious to the segmentation or recognition errors. One possible mitigation is fine-tuning the network in unlabeled scenes with semantics provided by Mask-RCNN. Meanwhile, the training in virtual environments also constrains our adaptation to real-world scenarios with similar layouts and semantics, and constructing larger datasets with more variations of scenes may alleviate this limitation. Moreover, the completeness of objects is roughly encoded in our 2D quality map, and it's interesting to investigate whether there is a more accurate but still efficient representation to further boost the performance. Furthermore, in our current implementation, the 2D quality map has a fixed size of $36m \times 36m$, and the agent is always supposed to be located at the center of the map at the initial step. Although this map construction has spare space for most indoor environments, there are still cases where the scene layout exceeds the boundary of the map, resulting in inadequate exploration and reconstruction.

Future work. Our work demonstrates the practicability and superiority of the DRL-based method when building the robotic autoscanning system. We believe that this proof-of-concept work will inspire future research toward both autonomous scene reconstruction and active scene understanding. First, extending our method to other agents, such as aerial vehicles, is promising by tailoring the SLAM algorithms for corresponding scenarios. Second, it is worth investigating multi-floor or dynamically sized representations for the autonomous reconstruction agent. Integrating them with neural networks is an interesting but challenging problem and merits further study. Third, we plan to study the incorporation of explicit signals to drive the agent to learn meaningful semantic concepts (e.g., types of the room or group structures of similar or functional objects) instead of solely relying on RL rewards. Other semantic-related auxiliary learning tasks or specific network architectures are promising to explore. Last but not least, to increase the scanning efficiency, the autonomous scene reconstruction problem can be extended to the multi-robot setting, where the combination of multi-agent reinforcement learning techniques to form a decentralized collaborative system is another fascinating direction. In this problem setting, the robot has to predict the behaviors of its partners so as to maximize the scanning coverage of the team and minimize redundant efforts. Effective learning of this kind of policy remains unsolved and needs more in-depth research.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments and suggestions. This work is supported by the National Key R&D Program of China (2022YFB3303400), National Natural Science Foundation of China (62025207), Guangdong Natural Science Foundation (2021B1515020085), and Shenzhen Science and Technology Program (RCYX20210609103121030).

REFERENCES

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. 2019. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113* (2019).

Pierre-Luc Bacon, Jean Harb, and Doina Precup. 2017. The option-critic architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

Frederic Bourgault, Alexei A Makarenko, Stefan B Williams, Ben Grocholsky, and Hugh F Durrant-Whyte. 2002. Information based adaptive robotic exploration. In *IEEE/RSJ international conference on intelligent robots and systems*, Vol. 1. IEEE, 540–545.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158* (2017).

Devendra Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. 2020c. Neural Topological SLAM for Visual Navigation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12872–12881.

Devendra Singh Chaplot, Murtaza Dalal, Saurabh Gupta, Jitendra Malik, and Ruslan Salakhutdinov. 2021. SEAL: Self-supervised Embodied Active Learning. In *Advances in Neural Information Processing Systems*.

Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. 2020b. Learning To Explore Using Active Neural SLAM. In *International Conference on Learning Representations (ICLR)*.

Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. 2020a. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems* 33 (2020).

Benjamin Charrow, Gregory Kahn, Sachin Patil, Sikang Liu, Ken Goldberg, Pieter Abbeel, Nathan Michael, and Vijay Kumar. 2015. Information-Theoretic Planning with Trajectory Optimization for Dense 3D Mapping.. In *Robotics: Science and Systems*, Vol. 11. Rome, 3–12.

Tao Chen, Saurabh Gupta, and Abhinav Gupta. 2019. Learning exploration policies for navigation. *arXiv preprint arXiv:1903.01959* (2019).

Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. 2017. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)* 36, 4 (2017), 1.

Siyan Dong, Kai Xu, Qiang Zhou, Andrea Tagliasacchi, Shiqing Xin, Matthias Nießner, and Baoquan Chen. 2019. Multi-Robot Collaborative Dense Scene Reconstruction. *ACM Transactions on Graphics* 38, 4 (2019), 1–16.

Alessandro Gasparetto, Paolo Boscariol, Albano Lanzutti, and Renato Vidoni. 2015. Path planning and trajectory planning algorithms: A general overview. *Motion and operation planning of robotic systems* (2015), 3–27.

Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. 2017. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 3389–3396.

Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. 2017. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2616–2625.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. 2018. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-second AAAI conference on artificial intelligence*.

Ruizhen Hu, Juzhan Xu, Bin Chen, Minglun Gong, Hao Zhang, and Hui Huang. 2020. TAP-Net: transport-and-pack using reinforcement learning. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–15.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. 2016. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397* (2016).

Deepali Jain, Atil Iscen, and Ken Caluwaerts. 2019. Hierarchical reinforcement learning for quadruped locomotion. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 7551–7557.

Bilal Kartal, Pablo Hernandez-Leal, and Matthew E Taylor. 2019. Terminal prediction as an auxiliary task for deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 15. 38–44.

Sven Koenig and Maxim Likhachev. 2002. D^* lite. *Aaai/iaai* 15 (2002).

Michael Krainin, Brian Curless, and Dieter Fox. 2011. Autonomous generation of complete 3D object models using next best view manipulation planning. In *2011 IEEE International Conference on Robotics and Automation*. IEEE, 5031–5037.

Simon Kriegel, Christian Rink, Tim Bodenmüller, Alexander Narr, Michael Suppa, and Gerd Hirzinger. 2012. Next-best-scan planning for autonomous 3d modeling. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2850–2856.

Chengshu Li, Fei Xia, Roberto Martin-Martin, and Silvio Savarese. 2020. Hrl4in: Hierarchical reinforcement learning for interactive navigation with mobile manipulators. In *Conference on Robot Learning*. PMLR, 603–616.

Ligang Liu, Xi Xia, Han Sun, Qi Shen, Juzhan Xu, Bin Chen, Hui Huang, and Kai Xu. 2018. Object-aware guidance for autonomous scene reconstruction. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–12.

Jan Matas, Stephen James, and Andrew J Davison. 2018. Sim-to-real reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*. PMLR, 734–743.

Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. 2016. Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673* (2016).

Ofir Nachum, Michael Ahn, Hugo Ponte, Shixiang Gu, and Vikash Kumar. 2019. Multi-agent manipulation via locomotion using hierarchical sim2real. *arXiv preprint arXiv:1908.05224* (2019).

Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*. IEEE, 127–136.

Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. 2013. Real-time 3D reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)* 32, 6 (2013), 1–11.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. 2017. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*. PMLR, 2778–2787.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018a. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. 2018b. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 3803–3810.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. 2019. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9339–9347.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* 362, 6419 (2018), 1140–1144.

Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. 2005. Information Gain-based Exploration Using Rao-Blackwellized Particle Filters.. In *Robotics: Science and systems*, Vol. 2. 65–72.

Richard S Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112, 1-2 (1999), 181–211.

Hassan Umari and Shayok Mukhopadhyay. 2017. Autonomous robotic exploration based on multiple rapidly-exploring randomized trees. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 1396–1402.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. 2017. Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*. PMLR, 3540–3549.

Hanqing Wang, Wei Liang, and Lap-Fai Yu. 2020. Scene mover: Automatic move planning for scene arrangement by deep reinforcement learning. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–15.

Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. 2016. ElasticFusion: Real-time dense SLAM and light source estimation. *The International Journal of Robotics Research* 35, 14 (2016), 1697–1716.

Shihao Wu, Wei Sun, Pinxin Long, Hui Huang, Daniel Cohen-Or, Minglun Gong, Oliver Deussen, and Baoquan Chen. 2014. Quality-driven poisson-guided autoscanning. *ACM Transactions on Graphics* 33, 6 (2014).

Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. 2018. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9068–9079.

Kai Xu, Hui Huang, Yifei Shi, Hao Li, Pinxin Long, Jianong Caichen, Wei Sun, and Baoquan Chen. 2015. Autoscanning for coupled scene reconstruction and proactive object analysis. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–14.

Kai Xu, Yifei Shi, Lintao Zheng, Junyu Zhang, Min Liu, Hui Huang, Hao Su, Daniel Cohen-Or, and Baoquan Chen. 2016. 3D attention-driven depth acquisition for object identification. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–14.

Kai Xu, Lintao Zheng, Zihao Yan, Guohang Yan, Eugene Zhang, Matthias Niessner, Oliver Deussen, Daniel Cohen-Or, and Hui Huang. 2017. Autonomous reconstruction of unknown indoor scenes guided by time-varying tensor fields. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–15.

Brian Yamauchi. 1997. 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, 146–151.

Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. 2021. Auxiliary Tasks and Exploration Enable ObjectGoal Navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 16117–16126.

Lintao Zheng, Chenyang Zhu, Jiazhao Zhang, Hang Zhao, Hui Huang, Matthias Niessner, and Kai Xu. 2019. Active scene understanding via online semantic reconstruction. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 103–114.