LEARNING EFFICIENT MULTI-AGENT COOPERATIVE VISUAL EXPLORATION

Chao Yu[1♯]*, **Xinyi Yang**[1*], **Jiaxuan Gao**[2*], **Huazhong Yang**[1], **Yu Wang**[1], **Yi Wu**[23♮]
[1] Department of Electronic Engineering, Tsinghua University
[2] Institute for Interdisciplinary Information Sciences, Tsinghua University
[3] Shanghai Qi Zhi Institute
♯zoeyuchao@gmail.com, ♮jxwuyi@gmail.com

ABSTRACT

We consider the task of visual indoor exploration with multiple agents, where the agents need to cooperatively explore the entire indoor region using as few steps as possible. Classical planning-based methods often suffer from particularly expensive computation at each inference step and a limited expressiveness of cooperation strategy. By contrast, reinforcement learning (RL) has become a trending paradigm for tackling this challenge due to its modeling capability of arbitrarily complex strategies and minimal inference overhead. We extend the state-of-the-art single-agent RL solution, *Active Neural SLAM* (ANS), to the multi-agent setting by introducing a novel RL-based global-goal planner, *Spatial Coordination Planner* (SCP), which leverages spatial information from each individual agent in an end-to-end manner and effectively guides the agents to navigate towards different spatial goals with high exploration efficiency. SCP consists of a transformer-based relation encoder to capture intra-agent interactions and a spatial action decoder to produce accurate goals. In addition, we also implement a few multi-agent enhancements to process local information from each agent for an aligned spatial representation and more precise planning. Our final solution, *Multi-Agent Active Neural SLAM* (MAANS), combines all these techniques and substantially outperforms 4 different planning-based methods and various RL baselines in the photo-realistic physical testbed, Habitat.

1 INTRODUCTION

Navigation is an important problem for building intelligent robot systems, which has been extensively studied in enormous domains, including rescue (Kleiner et al., 2006), autonomous driving (Bresson et al., 2017), drone (von Stumberg et al., 2017) and mobile robots (Rubio et al., 2019). In this paper, we consider a multi-agent exploration problem, where multiple homogeneous robots simultaneously explore an unknown spatial region via visual and sensory signals in a cooperative fashion. The existence of multiple agents enables complex cooperation strategies to effectively distribute the workload among different agents, which could lead to remarkably higher exploration efficiency than the single-agent counterparts. However, planning over multiple agents becomes substantially harder as well thanks to an exponentially large action space.

Planning-based solutions have been widely adopted for robotic navigation problems for both single-agent and multi-agent scenarios (Chaplot et al., 2020a; Burgard et al., 2005; Umari & Mukhopadhyay, 2017). Planning-based methods require little training and can be directly applied to different scenarios. However, these methods often suffer from limited expressiveness capability on coordination strategies, require non-trivial hyper-parameter tuning for each test scenario, and are particularly time-consuming due to repeated re-planning at each decision step. By contrast, reinforcement learning (RL) has been promising solution for a wide range of decision-making problems (Lillicrap et al., 2015; Mnih et al., 2013), including various visual navigation tasks (Chen et al., 2019a; Chaplot et al., 2020a; Savinov et al., 2019). An RL-based agent is often parameterized as a deep neural network and directly produces actions based on raw sensory signals. Once a policy is well trained by an RL algorithm, the robot can capture arbitrarily complex strategies and produce real-time decisions with efficient inference computation (i.e., a single forward-pass of neural network). However, training effective RL policies can be particularly challenging. As a result, most existing RL methods for robot exploration

*Equal contribution. Website: https://sites.google.com/view/maans problems focus on the single-agent setting while most multi-agent RL methods are only evaluated on simple scenarios like maze or grid world (Wakilpoor et al., 2020).

We develop *Multi-Agent Active Neural SLAM* (MAANS), the first RL-based solution for cooperative multi-agent exploration that substantially outperforms classical planning-based methods in a photo-realistic physical simulator, Habitat (Savva et al., 2019). MAANS extends the single-agent Active Neural SLAM method (Chaplot et al., 2020a) to the multi-agent setting and consists of 4 components, a neural SLAM module, a planning-based local planner, a local policy for control, and the most critical one, an RL-based global planner, which performs multi-agent planning and generates a navigation target for each agent.

We propose a novel multi-agent global planner, *Spatial Coordination Planner* (SCP), which is trained by the multi-agent PPO algorithm (Yu et al., 2021a) and integrates all the local spatial information for effective planning in an end-to-end manner. SCP leverages a transformer-based relation encoder to effectively model the complex intra-agent interactions, and a spatial action decoder to generate accurate and effective goal assignments towards accelerated exploration. In addition to the SCP module, we also implement a map refiner to align the spatial representation of each agent's local map, and a map merger, which enables the local planner to perform more precise sub-goal generation over a manually combined approximate 2D map. Experiment results show that MAANS achieves substantially higher exploration efficiency than a collection of planning-based competitors as well as various RL baselines on a set of carefully selected Habitat maps with sufficient difficulty-level.

Our contributions are summarized as follows:

- We introduce a multi-agent cooperative navigation framework, *Multi-Agent Active Neural SLAM* (MAANS).
- We develop a novel RL-based multi-agent planner, *Spatial Coordination Planner* (SCP), which applies a transformer-based relation encoder to effectively fuse each agent's local information and adopts a spatial action decoder to generate accurate global goals.
- We implement a map refiner, which unifies the map representation in MAANS, and a map merger, which improves local trajectory planning.
- MAANS produces substantially higher exploration efficiency than 4 planning-based methods in a photo-realistic physical environment, Habitat, with a 16.4% ACS improvement on 2-agent training maps, 8.2% on 2-agent unseen maps and 12.1% on 3-agent maps than the best of planning-based baselines. Ablation studies also show that MAANS outperforms various RL-based methods.

2 Related Work

Visual Navigation. Navigation with visual sensory is a critical task for building mobile robots. Classical methods typically apply a SLAM algorithm (Fuentes-Pacheco et al., 2015; Thrun, 2002) to reconstruct a spatial map from visual signals, and then derive a trajectory towards the target via path planning (Kavraki et al., 1996; LaValle et al., 2001; Sethian, 1996). For effective exploration, frontier-based sampling (Holz et al., 2010; Yamauchi, 1997; Umari & Mukhopadhyay, 2017; Dornhege & Kleiner, 2013) is the most widely used heuristic, which adaptively selects navigation goals located on the boundary between the explored and the unexplored region. Recently, with the advances in deep RL, many RL-based solutions have been developed (Mikolov et al., 2010; Mirowski et al., 2017; Zhu et al., 2017; Fang et al., 2019). Many works also leverage the spatial inductive bias to effective reconstruct the 2D map via an end-to-end spatial memory layer (Chen et al., 2019b; Mousavian et al., 2019; Henriques & Vedaldi, 2018; Parisotto & Salakhutdinov, 2018; Zhang et al., 2017) or an explicitly constructed graph structure to reflect the semantic topological structure of the environment (Bhatti et al., 2016; Wu et al., 2019; Yang et al., 2018; Chaplot et al., 2020c; Savinov et al., 2018; 2019). The Active Neural SLAM (ANS) method (Chaplot et al., 2020a) is the state-of-the-art framework for single-agent visual exploration, which takes advantage of both planning-based and RL-based techniques via a modular design (details in Sec. 3.2). There are also follow-up enhancements based on the ANS framework, such as improving map reconstruction with occupancy anticipation (Ramakrishnan et al., 2020) and incorporating semantic signals into the reconstructed map for semantic exploration (Chaplot et al., 2020b). Our MAANS can be viewed as a multi-agent extension of ANS with a few multi-agent-specific components.

Multi-agent Exploration. Cooperation among multiple robots is another important challenge in robotics. Many works have extended classical planning-based navigation methods from the single-agent setting to the multi-agent setting by sharing the reconstructed map (Dornhege & Kleiner, 2013; Holz et al., 2010; Umari & Mukhopadhyay, 2017; Yamauchi, 1997) or by applying some explicit cooperation heuristics (Wurm et al., 2008; Cohen, 1996; Burgard et al., 2005; Cohen, 1996). These simple techniques require little training but may have very limited expressiveness power to represent complex multi-agent cooperative strategies. Meanwhile, similar multi-agent extensions have been also developed for RL-based methods, such as introducing communication channels between individual agents (Wang et al., 2021; Zhu et al., 2021), utilizing intrinsic reward (Wang* et al., 2020), or cooperative training with curriculum learning (Long et al., 2020; Yang et al., 2020; Wang et al., 2020). However, jointly optimizing multiple policies makes multi-agent RL training remarkably more challenging than its single-agent counterpart. Hence, these end-to-end RL methods either focus on much simplified domains, like grid world or particle world, or still produce poor exploration efficiency compared with classical planning-based solutions. Our MAANS framework adopts a modular design and is the first RL-based solution that significantly outperforms classical planning-based baselines in a photo-realistic physical environment. Finally, we remark that MAANS utilizes a centralized global planner SCP, which assumes perfect communication between agents. There are also works on multi-agent cooperation with limited or constrained communication (Sukhbaatar et al., 2016; Peng et al., 2017; Jiang & Lu, 2018), which are parallel to our focus.

3 Preliminary

3.1 Task Setup

We use the Habitat environment (Savva et al., 2019), which provides photo-realistic visual signals and physics dynamics. At each step, each agent observes an first-person view RGB image in the shape of 128×128 as well as its pose sensory signals. The available actions include moving forward and rotation. We remark that Habitat also introduces random noise to pose signals and actions to simulate the real-world physics.

We consider a multi-agent cooperative exploration task, where a team of robots needs to explore a given house to maximize the accumulative amount of explored region as fast as possible within a limited horizon. In this task, we assume centralized decision setting, i.e., each agent can be fully aware of all the information from its teammates. Note that in this cooperative exploration setting, the birthplace of agents matters a lot, i.e., if all the agents are uniformly spread over the room, even random exploration could result in satisfactory room coverage rate. Hence, we focus on a more challenging setting where all the agents are always spawn together, i.e., within a randomly positioned circle of 2 meters in diameter per episode.

3.2 Single-agent Active Neural SLAM

The ANS framework (Chaplot et al., 2020a) consists of 4 parts: a neural SLAM module, a RL-based global planner, a planning-based local planner and a local policy. The neural SLAM module, which is trained by supervised learning, takes an RGB image, the pose sensory signals, and its past outputs as inputs, and outputs an updated 2D reconstructed map and a current pose estimation. Note that in ANS, the output 2D map only covers a neighboring region of the agent location and always keeps the agent at the egocentric position. For clarification, we call this raw output map from the SLAM module a *agent-centric local map*. The global planner in ANS takes in an augmented agent-centric *local map*, which includes channels indicating explored regions, unexplored regions and obstacles and the history trajectory, as its input, and outputs two real numbers from two Gaussian distributions denoting the coordinate of the long-term goal. This global planner is parameterized as a CNN policy and trained by the PPO algorithm (Schulman et al., 2017). Note that the global planner is actually trained over a much simpler 2D state-space with a shorter episode length compared to the original task since the agent needs to take a few environment steps to reach each global goal. The local planner performs classical planning, i.e., Fast Marching Method (FMM) (Sethian, 1996), over the agent-centric local map towards a given long-term goal, and outputs a trajectory of short-term sub-goals. Finally, the local policy produces actions given an RGB image and a sub-goal and is trained by imitation learning.

4 Methodology

We illustrate the overall framework of *Multi-Agent Active Neural SLAM* (MAANS) in Fig. 1. Each agent first passes its pose sensory signals and RGB image to the neural SLAM module to obtain

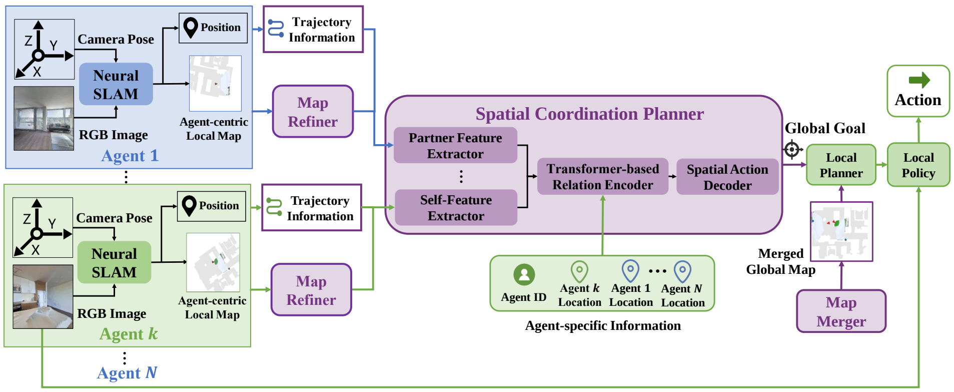


Figure 1: Overview of *Multi-Agent Active Neural SLAM* (MAANS).

the agent-centric local map and the pose estimation. Each local map is normalized by the map refiner and combined with additional agent-specific information as a input global map to the *Spatial Coordination Planner* (SCP). For each agent with ID k, SCP takes in the ID information, applies a transformer-based relation encoder over the extracted features of *all* the input maps, and generates a global goal via the spatial action decoder for agent k. The local planner performs trajectory planning on the merged global map towards the global goal. Finally, an action is generated by the local policy. Note that the neural SLAM module and the local policy do not involve multi-agent interactions, so we directly reuse these two modules from ANS (Chaplot et al., 2020a).

4.1 Spatial Coordination Planner

The SCP module has 3 parts, i.e., CNN-based feature extractors, a transformer-based relation encoder and a spatial action decoder. A illustration of SCP is shown in Fig. 2.

4.1.1 CNN-based Feature Extractor

We apply a separate 5-layer CNN tower to process each agent's input map and extract a 8×8 feature map with 32 channels. Note that we simply use N CNN-towers in the setting of N agents without parameter sharing. Then, a channel-wise concatenation is performed over all the feature maps to produce the input to the relation encoder. In order to make sure the network is fully aware of the current decision agent k, we always use the *first* CNN-tower to process the input map for that agent.

4.1.2 Transformer-based Relation Encoder

We apply a simplified transformer block as a multi-agent relation encoder over the 8×8 spatial feature map to better capture the intra-agent interactions. The relation encoder consists of two parts, including position encoding and spatial self-attention.

Agent-Specific Position Encoding: We adopt an agent-specific variant of relative position encoding (Shaw et al., 2018). When computing attention from grid (x_1, y_1) to grid (x_2, y_2) from the 8×8 feature map, conventional position encoding computes the embedding of the relative distances between the two attending grids, i.e., $[\text{embed}(x_1 - x_2), \text{embed}(y_1 - y_2)]$. In our task, the agent positions, i.e., *the grids containing agents*, are more important than those grids without agents. Hence, for each grid position (x, y) and each agent i at grid (x_i, y_i), we compute $[\text{embed}(x - x_i), \text{embed}(y - y_i)]$, which yields a total of N embeddings at each spatial position as additional channels to the feature map. Finally, to ensure the network is fully aware of the decision agent k, we similarly make the embedding for agent k as the leading channels and include an embedding of ID k at each grid.

Spatial Self-Attention: Inspired by the vision transformer model (Dosovitskiy et al., 2020), we convert the 8×8 feature map with position encoding into a sequence of 64 tokens and apply multi-head self-attention (Vaswani et al., 2017). The output sequence is then reshaped back to an 8×8 feature map with 16 channels.

4.1.3 Spatial Action Decoder

We utilize a hierarchical action space to better capture the spatial structure in our exploration problem. We design 3 action heads, a discrete region head denoting which grid to choose from the 8×8 feature map, and two continuous point heads denoting the x and y coordinate *within the grid of choice by the region head*. In particular, we first apply a CNN layer with 1 kernel size to project the $8 \times 8 \times 16$ feature map to a 8×8 logit map, and apply a spatial softmax operator over these logits to derive

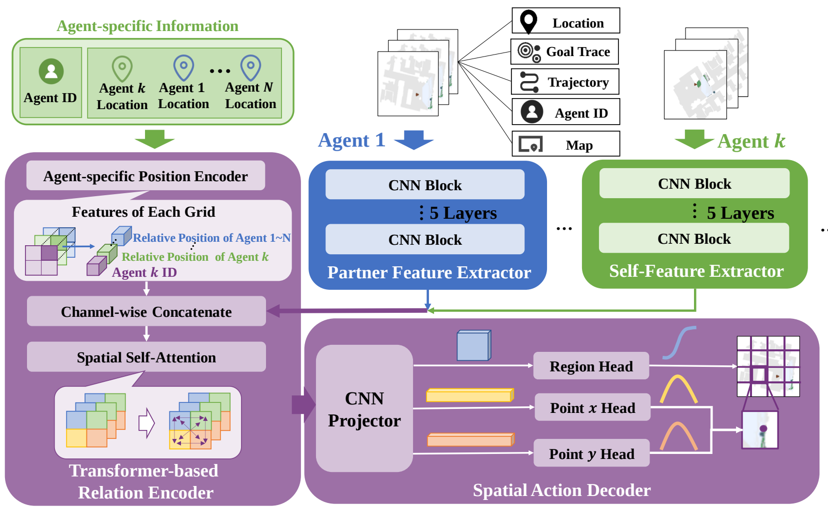


Figure 2: Workflow of *Spatial Coordination Planner* (SCP), including N CNN-based feature extractors, a relation encoder and a spatial action decoder.

the categorical distribution over the 64 grid. For each of point head, we parameterize the action distribution as a Gaussion distribution and directly project the entire feature map to the Gaussian mean and standard deviation. A sigmoid function is applied to ensure the Gaussian output is bounded within 0 and 1. When producing an action, we sample a discrete grid position from the region head and a relative coordinate from the point head, which suggests the global goal is located at a specific position within the selected grid region.

4.2 Map Refiner for Aligned 2D Maps

We develop a map refiner to ensure all the maps from the neural SLAM module are within the same coordinate system. The workflow is shown as the blue and green part in Fig. 3. The map refiner first composes all the past *agent-centric local maps* to recover the agent-centric *global* map. Then, we transform the coordinate system based on the pose estimates to normalize the global maps from all the agents w.r.t. the same coordinate system. Note that when an agent explores the border of the house, the agent-centric local map often covers a large portion of invisible region. As a result, the normalized global map will accordingly contain a large unexplorable boundary surrounding the actual explorable house region. To ensure the feature extractor in SCP concentrates only on the viable part and also induce a more focused spatial action space, we crop the unexplorable boundary of the normalized map and enlarge the house region as our final refined map.

4.3 Map Merger for Improved Local Planning

The local planner from ANS plans sub-goals on the agent-centric local map, while in our setting, we can also leverage the information from other agents to plan over a more accurate map. The diagram of map merger is shown in Fig. 3. After obtaining N enlarged global maps via the map refiner, the map merger simply integrates all these maps by applying a max-pooling operator for each pixel location. That is, for each pixel in the merged global map, the probability of it being an obstacle is the maximum value at that pixel over all the individual enlarged global maps. We remark that the artificial merged global map is only utilized in the local planner, but not in the global planner SCP. We empirically observe that having a coarse merged map produces better short-term local goal while such an artificial map is not sufficient for accurate global planning (more details can be found in Sec. 5.5.).

4.4 RL Training for SCP

We adopt the multi-agent PPO framework (Yu et al., 2021a) with parameter sharing. Some of the most important training factors are presented below while full details can be found at `https://sites.google.com/view/maans`.

4.4.1 Reward

The team-based reward includes 4 terms, i.e., (1) a coverage reward, which is the increment of the explored region, (2) a success reward, which will be given when 90% and 95% coverage is achieved,

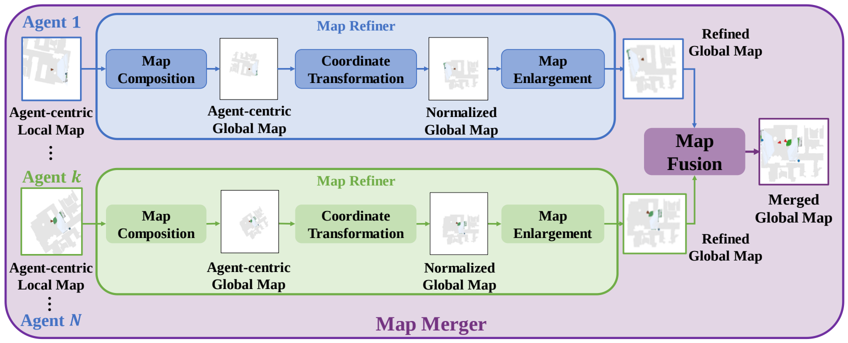


Figure 3: Computation workflow of *map refiner* (blue and green) and *map merger* (light purple).

(3) the overlapping penalty to promote non-overlapping global goals, which is the increment of overlapped explored area between agent k and other agents, and (4) a time penalty to encourage fast exploration. The final reward is a linear combination of these 4 terms. Detailed definition can be found in Appendix A.1.3.

4.4.2 INPUT REPRESENTATION

Each agent input map for SCP is a 240×240 map with 7 channels, including (1) an obstacle channel, where each pixel value denotes the probability of being an obstacle, (2) an explored region channel, which is a probability map for each pixel being explored, (3) a one-hot position channel, which denotes the position of that agent, (4) a trajectory channel, which covers all the historical position of that agent with an exponentially decaying weight, (5) a one-hot global goal channel denoting the position of previous global goal, (6) a goal history channel with all the past global goals, and (7) an agent ID channel. We remark that SCP takes N such refined input maps as its input state.

4.4.3 MULTI-TASK TRAINING AND DISTILLATION

We empirically notice that in our multi-agent exploration task, the training progress varies a lot on different maps. Therefore, when SCP is directly trained over a uniform distribution of all training maps, SCP may easily overfit to some particularly easy maps and stop learning on those harder ones. In order to obtain a single policy that can work on multiple maps and eventually generalize to unseen maps, we propose a simple training-and-distillation solution, i.e., *MAANS-TD*: at the first stage, we train a specialized policy using RL for every training map while at the distillation stage, we train a general policy over all the maps by running imitation learning over the expert policies. We adopt a KL-divergence loss for the region head and a MSE loss for point heads as our imitation objective.

5 EXPERIMENT RESULTS

We consider the 3D houses from the Gibson dataset (Xia et al., 2018) and eliminate those houses with disconnected viable regions or multiple floors. Based on the size of available area, we adopt 7 middle maps and 2 large maps for training, and 3 middle maps for evaluation. We assume the two-agent scenario ($N = 2$) in all the following experiments unless otherwise mentioned (i.e., Sec. 5.3.3). Every RL training is performed with 10^4 training episodes over 3 random seeds. Each evaluation score is expressed in the format of "mean (standard deviation)", which is averaged over a total of 300 testing episodes, i.e., 100 episodes per random seed.

5.1 EVALUATION METRIC

We proposed ***Accumulative Coverage Score*** (ACS) as our performance metric, which captures the overall exploration progress throughout an episode. In particular, let C_t denote the coverage rate, i.e., the ratio of explored region to the total explorable area, at timestep t for an episode. The ACS number at timestep k, A_k, is computed by $A_k = \sum_{t=0}^{k} C_t$. A higher ACS number implies faster exploration. We take the ACS number at timestep 200 in all our experiments.

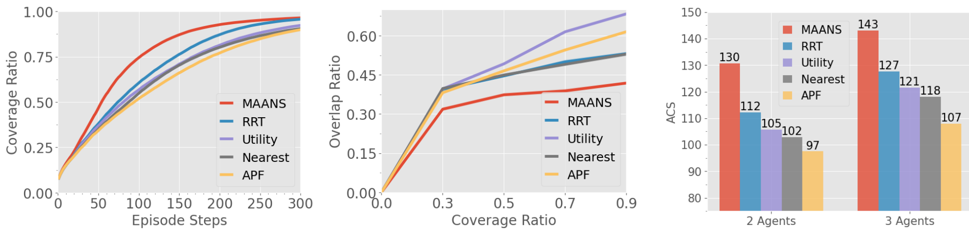


Figure 4: The left two figures show within-episode exploration efficiency on training maps by different trained policies. We measure the coverage rate w.r.t. episode step (left, higher the better) and overlap ratio w.r.t. coverage ratio (right, lower the better). As exploration proceeds, agents by MAANS cover explorable space much faster with a significantly lower overlap ratio. The rightmost figure shows the ACS performance of 2 agents and 3 agents.

5.2 Behavior Statistics

In addition to ACS, we also consider 3 additional behavior statistics measurement to capture different characteristics of a particular exploration strategy. We remark that these metrics is only for analysis purpose, and we primarily focus on ACS as our performance criteria.

- *Coverage Ratio*: the *final* ratio of explored area to total area when an episode terminates. Higher coverage implies more exhaustive exploration.
- *Steps*: the timesteps that the agents use to reach a 90% coverage. Fewer steps implies more efficient exploration.
- *Overlap Ratio*: the ratio of the overlapped region explored by *multiple* agents to the current explored area when 90% coverage is reached. Lower overlap ratio implies better credit assignment.

Table 1: Average training performance.

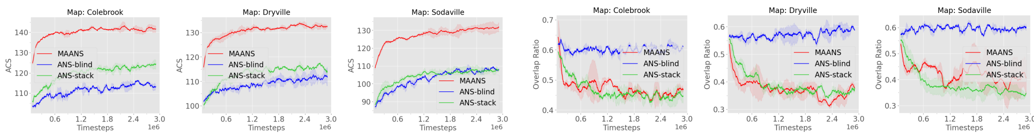
Methods	ACS	Behavior Statistics		
		Coverage Ratio	Steps	Overlap Ratio
Nearest	102.79(1.55)	0.91(0.01)	246.79(3.90)	0.53(0.02)
APF	97.49(1.57)	0.90(0.01)	251.41(3.15)	0.61(0.01)
Utility	105.62(0.89)	0.92(0.01)	236.15(3.61)	0.68(0.01)
RRT	112.21(1.39)	**0.96**(0.00)	199.59(3.27)	0.53(0.02)
MAANS	**130.59**(1.53)	**0.96**(0.00)	**165.67**(4.64)	**0.42**(0.02)

Table 2: Average evaluation performance on unseen maps.

Methods	ACS	Behavior Statistics		
		Coverage Ratio	Steps	Overlap Ratio
Nearest	122.39(1.05)	0.94(0.00)	198.90(3.90)	**0.55**(0.02)
APF	120.07(1.15)	0.93(0.00)	202.09(3.10)	0.61(0.01)
Utility	128.34(0.91)	0.95(0.00)	173.40(2.66)	0.68(0.01)
RRT	127.43(0.98)	**0.96**(0.00)	168.24(2.16)	0.59(0.02)
MAANS-TD	**137.60**(0.67)	**0.96**(0.00)	**165.21**(1.00)	0.57(0.01)

Table 3: *3-agent-setting* training performance ($N = 3$).

Methods	ACS	Behavior Statistics		
		Coverage Ratio	Steps	Overlap Ratio
Nearest	118.05(0.63)	0.91(0.00)	188.58(2.02)	0.46(0.01)
APF	107.88(1.39)	0.87(0.01)	207.20(2.41)	0.45(0.01)
Utility	121.62(0.85)	0.94(0.00)	180.82(2.25)	0.58(0.00)
RRT	127.64(1.31)	0.95(0.01)	155.13(3.26)	0.44(0.01)
MAANS	**143.09**(0.71)	**0.96**(0.00)	**132.95**(1.86)	**0.35**(0.02)



(a) ACS (b) Overlap Ratio

Figure 5: Comparison between MAANS and other ANS variants on 3 representative training maps.

5.3 Comparison with Planning-based Methods

We consider 4 planning-based competitors, including a utility-maximizing method (*Utility*) (Juliá et al., 2012), a search-based nearest-frontier method (*Nearest*) (Yamauchi, 1997), the artificial potential field method (*APF*) (Yu et al., 2021b), which applies resistance forces among agents as a cooperation mechanism, and a rapid-exploring-random-tree-based method (*RRT*) (Umari & Mukhopadhyay, 2017). Since these planning-based methods also require a SLAM module and a controller, we simply substitute SCP in our framework with a planning algorithm while the neural SLAM module, the local planner and the local policy are all preserved.

5.3.1 Training Performance

Tab. 1 summarizes the average performance of MAANS and planning-based baselines over the training maps (7 middle maps and 2 large maps). We notice that RRT consistently produces the best result among all the planning-based methods under every evaluation metric. However, it is still outperformed by MAANS with a clear margin. MAANS produces the same final coverage ratio as RRT while achieves a 15% higher ACS, more than 30 fewer steps for 90% coverage (horizon is 300), and about 20% lower overlap ratio. These metrics suggest that MAANS is able to explore more space faster and with better multi-agent coordination.

To better measure the overall exploration process, Fig. 4 visualizes how the coverage ratio as well as the overlap ratio varies throughout each testing episode. In particular, we plot the average coverage rate at each episode step, and the overlap ratio w.r.t. different coverage ratio. Regarding coverage rate, we can observe that MAANS explores substantially more area than other methods particularly in the early stage of exploration, leading to a higher ACS value consequently. For overlap ratio, MAANS maintains a significantly lower curve particularly in the later period where more than 30% of the house is explored. This suggests that MAANS is able to well control the agents to avoid repetitive exploration.

5.3.2 Generalization Performance

Tab. 2 shows the evaluation performance of our distillation policy (*MAANS-TD*) and planning-based methods on 3 unseen test maps. MAANS-TD achieves the best final coverage ratio, the fewest steps for 90% coverage and a comparable overlap ratio. More importantly, MAANS-TD yields the highest ACS score with a clear margin (i.e., 9+ ACS number) compared to the best planning-based competitor, which suggests that our training-and-imitation solution leads to a generalizable policy.

5.3.3 Performance with More Agents

Here we apply MAANS and other baselines to the setting of $N = 3$ agents. Tab. 3 summarizes the average training performance w.r.t. different evaluation metrics on 7 middle maps and 2 large maps. With $N = 3$ agents, MAANS still produces the best final coverage ratio, the fewest steps towards 90% coverage, the least overlap ratio, as well as the highest ACS number with a clear margin (15+ in ACS value) over other baselines. The rightmost figure of Fig. 4 shows the comparison of ACS performance between 2 agents and 3 agents. This surprising advantage of MAANS over planning-based methods shows the potential to further improve collaborative exploration efficiency with more agents.

5.4 Comparison with Other ANS Variants

We consider 2 additional ANS variants other than MAANS. For simplicity, we report the training performances on 3 selected maps, *Colebrook*, *Dryville* and *Sodaville*.

ANS-blind: We train N ANS agents to explore blindly, i.e., without any communication, in the environment.

ANS-stack: We directly stack all the agent-centric local maps from the neural SLAM module as the input representation to the global planner, and retrain the ANS global planner under our multi-agent task setting.

We measure the *ACS* and the *Overlap Ratio* metric over these 3 maps and demonstrate the training curves in Fig. 5. Regarding the ACS, both ANS variants perform consistently worse than MAANS on each of the map. Regarding the overlap ratio, the blind variant fails to cooperate completely while the stack variant produces comparable overlap ratio to MAANS despite its low exploration efficiency. We hypothesis that this is due to the fact that ANS-stack performs global and local planning completely on the agent-centric *local* map. The local map is a narrow sub-region over the entire house, which naturally leads to a much conservative exploration strategy and accordingly helps produces a low overlap ratio.

5.5 Ablation Study on SCP

We consider 3 SCP variants and report the training performances on map *Quantico*.

SCP w.o. RE: We consider the SCP without the relation encoder. The output feature maps from the CNN-based feature extractors are channel-wise concatenated and directly fed into the spatial action decoder.

SCP w.o. AE: We remove the region head from the spatial action decoder, so that the global goal is directly generated over the entire refined global map via two Gaussian action distributions. We remark that such an action space design follows the original ANS paper (Chaplot et al., 2020a).

SCP-merge: We consider another SCP variant that applies a single CNN feature extractor over the manually merged global map from the map merger, instead of forcing the network to learn to fuse each agent's information.

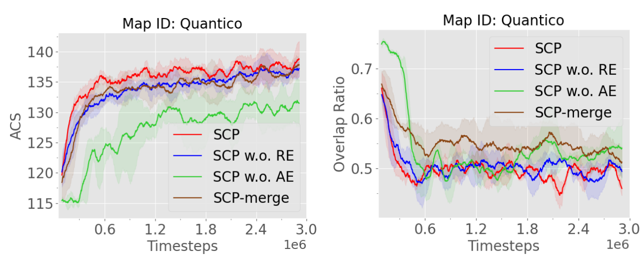


Figure 6: Ablation studies on SCP components.

We measure the *ACS* and the *Overlap* metric and visualize the training curves in Fig. 6. In summary, the full MAANS module produces both the highest ACS and the lowest overlap ratio. Among all the SCP variants, *SCP w.o. AE* produces the lowest ACS. This suggests that a simple Gaussian representation of actions may not be able to fully capture the distribution of good long-term goals, which can be highly multi-modal in the early exploration stage. In addition, *SCP-merge* produces a much higher overlap ratio than all the other methods. We hypothesis that this is due to the fact that many agent-specific information are lost in the manually merged maps while SCP can learn to utilize these features implicitly.

6 Conclusion

We propose the first multi-agent cooperative navigation framework, *Multi-Agent Active Neural SLAM* (MAANS) that outperforms planning-based competitors in a photo-realist physical environment. MAANS consists of a novel RL-based multi-agent planner, Spatial Coordination Planner, and a collection of multi-agent-specific enhancements. Experiments in the Habitat testbed shows that MAANS achieves better training and generalization performances than all the baseline methods. We also show that our method can be applied to a system with various number of agents. We hope that our method can be a starting point for more efficient multi-agent exploration systems in even more challenging settings, such as a large number of agents or constrained communication, in the future research.

REFERENCES

Shehroze Bhatti, Alban Desmaison, Ondrej Miksik, Nantas Nardelli, N Siddharth, and Philip HS Torr. Playing doom with slam-augmented deep reinforcement learning. *arXiv preprint arXiv:1612.00380*, 2016.

Guillaume Bresson, Zayed Alsayed, Li Yu, and Sébastien Glaser. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220, 2017.

Wolfram Burgard, Mark Moors, Cyrill Stachniss, and Frank E Schneider. Coordinated multi-robot exploration. *IEEE Transactions on robotics*, 21(3):376–386, 2005.

Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *International Conference on Learning Representations*. ICLR, 2020a.

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

Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological slam for visual navigation. In *CVPR*, 2020c.

Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning exploration policies for navigation. In *International Conference on Learning Representations*. ICLR, 2019a.

Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning exploration policies for navigation. In *International Conference on Learning Representations*. ICLR, 2019b.

William W Cohen. Adaptive mapping and navigation by teams of simple robots. *Robotics and autonomous systems*, 18(4):411–434, 1996.

Christian Dornhege and Alexander Kleiner. A frontier-void-based approach for autonomous exploration in 3d. *Advanced Robotics*, 27(6):459–468, 2013.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Kuan Fang, Alexander Toshev, Li Fei-Fei, and Silvio Savarese. Scene memory transformer for embodied agents in long-horizon tasks. In *Computer Vision and Pattern Recognition*. CVPR, 2019.

Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial intelligence review*, 43(1):55–81, 2015.

Joao F Henriques and Andrea Vedaldi. Mapnet: An allocentric spatial memory for mapping environments. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8476–8484, 2018.

Dirk Holz, Nicola Basilico, Francesco Amigoni, and Sven Behnke. Evaluating the efficiency of frontier-based exploration strategies. In *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, pp. 1–8. VDE, 2010.

Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation. *Advances in Neural Information Processing Systems*, 31:7254–7264, 2018.

Miguel Juliá, Arturo Gil, and Oscar Reinoso. A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Autonomous Robots*, 33(4):427–444, 2012.

Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 1996.

Alexander Kleiner, Johann Prediger, and Bernhard Nebel. Rfid technology-based exploration and slam for search and rescue. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4054–4059. IEEE, 2006.

Steven M LaValle, James J Kuffner, BR Donald, et al. Rapidly-exploring random trees: Progress and prospects. *Algorithmic and computational robotics: new directions*, 5:293–308, 2001.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Qian Long, Zihan Zhou, Abhinav Gupta, Fei Fang, Yi Wu, and Xiaolong Wang. Evolutionary population curriculum for scaling multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2020.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. In *ICLR*, 2017.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Arsalan Mousavian, Alexander Toshev, Marek Fišer, Jana Košecká, Ayzaan Wahid, and James Davidson. Visual representations for semantic target driven navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8846–8852. IEEE, 2019.

Emilio Parisotto and Ruslan Salakhutdinov. Neural map: Structured memory for deep reinforcement learning. In *International Conference on Learning Representations*. ICLR, 2018.

Peng Peng, Quan Yuan, Ying Wen, Yaodong Yang, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games. *CoRR*, abs/1703.10069, 2017. URL `http://arxiv.org/abs/1703.10069`.

Santhosh K Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. Occupancy anticipation for efficient exploration and navigation. In *European Conference on Computer Vision*, pp. 400–418. Springer, 2020.

Francisco Rubio, Francisco Valero, and Carlos Llopis-Albert. A review of mobile robots: Concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems*, 16(2):1729881419839596, 2019.

Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. In *International Conference on Learning Representations*. ICLR, 2018.

Nikolay Savinov, Anton Raichuk, Raphaël Marinier, Damien Vincent, Marc Pollefeys, Timothy Lillicrap, and Sylvain Gelly. Episodic curiosity through reachability. In *International Conference on Learning Representations*. ICLR, 2019.

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

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *NAACL*, 2018.

Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29:2244–2252, 2016.

Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

Hassan Umari and Shayok Mukhopadhyay. Autonomous robotic exploration based on multiple rapidly-exploring randomized trees. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1396–1402, 2017. doi: 10.1109/IROS.2017.8202319.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Lukas von Stumberg, Vladyslav Usenko, Jakob Engel, Jörg Stückler, and Daniel Cremers. From monocular slam to autonomous drone exploration. In *2017 European Conference on Mobile Robots (ECMR)*, pp. 1–8. IEEE, 2017.

Ceyer Wakilpoor, Patrick J Martin, Carrie Rebhuhn, and Amanda Vu. Heterogeneous multi-agent reinforcement learning for unknown environment mapping. *arXiv preprint arXiv:2010.02663*, 2020.

Haiyang Wang, Wenguan Wang, Xizhou Zhu, Jifeng Dai, and Liwei Wang. Collaborative visual navigation. *arXiv preprint arXiv:2107.01151*, 2021.

Tonghan Wang*, Jianhao Wang*, Yi Wu, and Chongjie Zhang. Influence-based multi-agent exploration. In *International Conference on Learning Representations*, 2020.

Weixun Wang, Tianpei Yang, Yong Liu, Jianye Hao, Xiaotian Hao, Yujing Hu, Yingfeng Chen, Changjie Fan, and Yang Gao. From few to more: Large-scale dynamic multiagent curriculum learning. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7293–7300. AAAI Press, 2020.

Yi Wu, Yuxin Wu, Aviv Tamar, Stuart Russell, Georgia Gkioxari, and Yuandong Tian. Bayesian relational memory for semantic visual navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2769–2779, 2019.

Kai M Wurm, Cyrill Stachniss, and Wolfram Burgard. Coordinated multi-robot exploration using a segmentation of the environment. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1160–1165. IEEE, 2008.

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

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

Jiachen Yang, Alireza Nakhaei, David Isele, Kikuo Fujimura, and Hongyuan Zha. Cm3: Cooperative multi-goal multi-stage multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2020.

Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. *arXiv preprint arXiv:1810.06543*, 2018.

Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of mappo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021a.

Jincheng Yu, Jianming Tong, Yuanfan Xu, Zhilin Xu, Haolin Dong, Tianxiang Yang, and Yu Wang. Smmr-explore: Submap-based multi-robot exploration system with multi-robot multi-target potential field exploration method. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021b.

Jingwei Zhang, Lei Tai, Ming Liu, Joschka Boedecker, and Wolfram Burgard. Neural slam: Learning to explore with external memory. *arXiv preprint arXiv:1706.09520*, 2017.

Fengda Zhu, Siyi Hu, Yi Zhang, Haodong Hong, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Main: A multi-agent indoor navigation benchmark for cooperative learning. 2021.

Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3357—3364. IEEE, 2017.