Supplementary Material for "Multi-Robot Active Mapping via Neural Bipartite Graph Matching"

This document provides the additional supplemental material that cannot be included in the main paper due to its page limit:

- Implementation details.
- Training with a single scene.
- Generalization to more unseen robots.

A. Implementation details

We train the global planner with 12 parallel threads. We use 12 mini-batches and do 4 epochs in each PPO update. We adopt the Adam optimizer with a learning rate of 0.0005/0.000025 (actor/critic), a discount factor of 0.99, an entropy coefficient of 0.0001 and value loss coefficient of 3.0. The network structure for the multiplex graph neural network is detailed below:

- f_{init}: a 5-layer MLP (3-32-64-128-256-32).
- $f_{query}/f_{key}/f_{value}$: a linear projection (32-32).
- f_{node}: a 2-layer MLP (64-64-32).
- f_{edge}: a 2-layer MLP (96-32-1).

Each fully connected layer in the above networks is followed by a Batch Normalization layer and a ReLU layer.

In the iGibson environment, each robot has a physical body and can be visually observed by other robots and hence becomes obstacles (occupied) in the occupancy map. Therefore, unlike the previous works [2, 1, 3] that do not update the obstacle once it is constructed, in our framework, we update the explored region (free and explored region) when it has been scanned more than once to alleviate the issues raised by the multi-robot scenario.

B. Training with a single scene

In the main paper, we demonstrate that our algorithm is trained only on 9 scenes in the Gibson dataset, and is able to generalize well to various indoor scenes even in the unseen Matterport3D dataset. This mainly benefits from the proposed multiplex graph neural network, which is the solely learnable module in our algorithm, and only relies on the simple robot, frontier, and their geodesic distance information extracted from the occupancy map for goal position estimation. Such a design makes our algorithm relatively robust to the geometry, appearance, and layout variations of the indoor scene distributions.

NeuralCoMapping (Ours)	Cov. (%)	Time (#steps)
single training scene	97.1	691.3
nine training scenes	96.3	661.7

Table 1. Experiments of training with a single scene or nine scenes for our algorithm. The results are reported on the Matterport3D dataset.

To further explore the potential of training with fewer scenes for our algorithm, we experiment with an extreme case, where our algorithm is trained with only a single scene. The results are shown in Table 1. Surprisingly, we observe that in this extreme case, our algorithm still performs comparably with the original one trained with nine scenes. It demonstrates the strong learning ability and robustness of our algorithm.

C. Generalization to more unseen robots

We further evaluate the generalization ability of our algorithm on more unseen robots. We train our algorithm with 3 robots, same as the multi-robot scenario in the main paper, and evaluate it with 2, 4, 5, 7 and 9 robots. We also test its upper bound performance by training and evaluating on the same number (2, 4, 5, 7, 9) of robots. We observe that when we run more robots for scene reconstruction, the performance of time efficiency tends to saturate, hence the advantage of using more robots cannot be fully exposed. To tackle this issue, we evaluate the generalization ability on the large scenes ($\geq 50m^2$) in the Matterport3D dataset. The results are shown in Table 2. Our algorithm is able to achieve very similar performance compared to its upper bound even when it generalizes to the 9-robot scenario. It validates the exceptional generalization ability of our algorithm again.

In this work, we focus on active mapping of indoor scenes from the Gibson and Matterport3D datasets, where usually less than 10 robots are sufficient. We further evaluate 100 robots in a scene of $633.6m^2$, and observe consistently better results of our algorithm (450 steps) than the best competitor CoScan (525 steps).

	Train with 3 robots		Upper bound	
Test number of robots	Cov. (%)	Time (#steps)	Cov. (%)	Time (#steps)
2 robots	97.1	1293.8	96.6	1276.5
4 robots	98.4	798.7	98.4	776.3
5 robots	98.1	728.7	98.2	693.5
7 robots	96.9	694.0	98.4	662.3
9 robots	98.6	589.7	98.5	580.8

Table 2. Generalization to more robots on the Matterport3D dataset. Our algorithm is trained with 3 robots, and evaluated with 2, 4, 5, 7 and 9 robots separately. The upper bound performance is computed by training and evaluated on the same number of robots.

References

[1] 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)*, 2020.

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

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