Interactive Exploration for Mapping

Cafer Mertcan Akçay
Mechanical and Process Engineering
ETH Zürich Switzerland
cakcay@student.ethz.ch

İrem Kaftan
Information Technology and Electrical Engineering
ETH Zürich Switzerland
ikaftan@student.ethz.ch

Abstract: The ability to interact with objects in an environment is becoming an increasingly important skill for autonomous agents to perform complex robotic tasks. It is also important to have a notion of individual objects, their shape, and pose in the map for successful interactions. We hypothesize that interactions can help with learning these concepts. In this work, we propose a reinforcement learning (RL) framework to encourage an agent to navigate in an unknown environment and interact with objects to perform more complete object-level mapping. We also implement a bridge between the RL and the mapping framework to exchange the sensor readings and the map metric. Furthermore, we train different policies for the agent with and without interaction capabilities and assess the completeness of the resulting object-level maps for each policy by analyzing the number of observed voxels of objects and their qualitative reconstructions in the map. We show that the agent with interaction capabilities observes more voxels of objects and creates more complete object-level maps. Hence, we validate our hypothesis that interactions help with performing more complete object-level mapping. The code for the RL and the mapping framework is publicly available.

Keywords: Reinforcement learning, Object-level mapping

1 Introduction

Humans have the ability to enhance their visual perception through interactions with the environment, such as pushing an object aside to resolve occlusions. Enabling this behavior in robots can help with creating more accurate object-level maps which are necessary for successful completion of many robotic tasks. Moreover, interactions can potentially help with creating such object-level maps, which are beyond pure geometric reconstruction. There has been advancements in training exploration policies using deep RL methods [1, 2] and creating volumetric maps of environments [3] using submapping approaches [4, 5, 6, 7, 8]. However, training interactive exploration policies for performing more accurate object-level mapping has been a rather under-explored topic.

In this work, we propose an RL framework to incentivize the agent to navigate in an unknown environment and interact with objects to create a more complete object-level map using the panoptic mapping framework introduced in [3]. We implement an RL framework using Stable-Baselines3 [9] in which we use the egocentric RGB camera images and ground truth (GT) affordance masks as the observations. We reward the agent for each new interacted object and for each new visited position to encourage both exploration and interaction. Our RL framework is inspired by [1]. With this framework, we train RL policies using proximal policy optimization (PPO) [10] for the agent in the AI2-iTHOR simulation environment [11] with and without interaction capabilities. This simulator contains realistic indoor scenes and supports interactions with objects. We evaluate these policies by calculating the object coverage and position coverage and analyzing the number of observed voxels of objects and their qualitative reconstructions in the map. Moreover, we implement a ROS bridge between the RL and the mapping framework [3]. This bridge allows us to exchange the sensor readings and the map metric which is used in the evaluation. We show that our reward signal succeeds in making the agent explore the environment and interact with the objects and demonstrate that the

agent with interaction capabilities creates more complete object-level maps. Our contributions are as follows:

- We introduce an RL framework to perform more complete object-level mapping in indoor scenes using [3] by leveraging interactions.
- We implement a ROS bridge between the RL and the mapping framework [3] to exchange the sensor readings and the map metric.
- We show that interactions help with creating more complete object-level maps by comparing the trained policies for the agent with and without interaction capabilities.

2 Related Work

2.1 Reinforcement learning for exploration

With the recent advancements in the performance of deep RL approaches, numerous methods [12, 13, 14, 15] have been proposed for training exploration policies for navigating in and mapping static simulation environments [16, 17]. These methods utilize different reward metrics, such as novelty [18, 19], curiosity [20, 19], and coverage [12, 13, 15, 14, 19]. Our work differs in that the focus is on dynamic simulation environments where we train policies for both exploring the environment and interacting with the objects.

Many recent works also leverage dynamic simulation environments [11, 21] in order to learn affordance maps, where affordance is a potential for action [22], and develop agents that can perform interaction tasks [1, 2]. In [1], a deep RL approach is proposed in which the agent learns the affordance landscape of a new environment. The agent is rewarded for maximizing successful interactions with objects while simultaneously training an affordance segmentation model. Although [1] forms the inspiration of our work, it differs in that no explicit map of the environment is built. Moreover, there has not been a work on training interactive exploration policies for improving the completeness of object-level maps. Our work fills this gap since our agent learns to navigate in the environment and interact with the objects to perform more complete object-level mapping.

2.2 Object-level mapping

Most dense semantic mapping approaches [23, 24, 6, 7, 25, 26] obtain segmentation masks using convolutional neural networks (CNNs) and fuse them into a global map to estimate the label. However, these approaches assume that the environment is static. There are also object-based mapping approaches [6, 7, 8] which focus on reconstructing multiple moving objects. Although these methods are designed for short-term dynamics, there is no self-induced motion which is a central point as looked at in this work.

In [3], panoptic multi-TSDFs is proposed as a novel method for semantic volumetric mapping in dynamic environments which focuses on long-term change detection. It differentiates between object instances, background classes, and free space based on [27] and treats objects as the minimal units of change. Moreover, it represents the scene as a collection of submaps where each submap contains a locally consistent panoptic entity, such as an object. This allows for capturing long-term changes and maintaining semantic consistency. Although it does not focus on short-term dynamics, we use this framework for object-level mapping since it works in dynamic environments.

3 Method

In this work, we implement an RL framework to encourage the agent to navigate in an environment and interact with objects. This section provides a detailed overview of the RL framework and the choice of our reward signal, as well as the ROS bridge we implement to exchange the sensor readings and the map metric.

3.1 Training pipeline

We implement our RL framework using Stable-Baselines3 [9]. For this purpose, we use AI2-iTHOR [11] which is a Unity-based simulation environment that contains realistic indoor scenes with interactable objects. We wrap this environment in a gym environment [28] to make it compatible with Stable-Baselines3 [9]. The action space consists of navigation (move ahead, turn left, turn right, look up, look down) and interaction (take, put) actions for the agent shown in Figure 1a. The observation space consists of RGB images (3, 80, 80) and ground truth (GT) affordance masks (2, 80, 80) where we have one channel for each interaction action. Affordance masks are binary images where each pixel indicates whether that action can be performed at that location. An example RGB image and its corresponding affordance mask for the action 'put' are shown in Figure 1b and 1c.

(a) Agent in iTHOR [11]. (b) RGB image. (c) Affordance mask for 'put'.

Figure 1: (a) The agent in AI2-iTHOR [11]: it can move its base forward, turn clockwise or counterclockwise, look up or down, and take or put objects. (b) An example RGB image and (c) its corresponding affordance mask for the action 'put'.

Since we want our agent to create more complete object-level maps by leveraging interactions, we can give reward for each new interacted object and for the increase in the number of observed voxels. However, the ROS bridge limits the number of different scenes that we can use in training due to memory constraints which makes it harder for the agent to learn generalized policies. Moreover, obtaining one part of the reward signal through the bridge slows down the training process. Due to these reasons, we decided to use a proxy reward signal that encourages the agent to explore the environment and interact with the objects. We give reward for each new interacted object (R_{int}) and for each new visited position (R_{nav}) as given in Eq. 1. We discretize the floor of the scene and consider each grid cell as a new position where the size of a grid cell is 0.25 meters.

$$R_{int} = \begin{cases} 1, & \text{each new interacted object} \\ 0, & \text{otherwise} \end{cases} \qquad R_{nav} = \begin{cases} 1, & \text{each new visited position} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

R_{int} and R_{nav} are summed after a relative weighting through parameters α and β to obtain the total reward which is given in Eq. 2. We experiment with different combinations of α and β to investigate the influence of these parameters on the interactive and explorative behaviors of the agent.

$$R_{tot} = \alpha R_{int} + \beta R_{nav} \tag{2}$$

We then implement a custom policy network whose pipeline is given in Figure 2. The inputs are RGB images and GT affordance masks obtained from AI2-iTHOR [11]. They are stacked together (5, 80, 80) and fed to a 3 layer CNN (Conv + ReLU x 3) with channel dimensions (5, 32, 64), kernel sizes (8, 4, 3), and strides (4, 2, 1) followed by a fully connected layer to reduce the dimension to 512. The output is fed to two fully connected layers with dimensions (512, 512) each to generate the next action distribution and value. We train our network using PPO [10] for 1 million frames with a learning rate of 0.0001 and rollouts of 256 time steps.

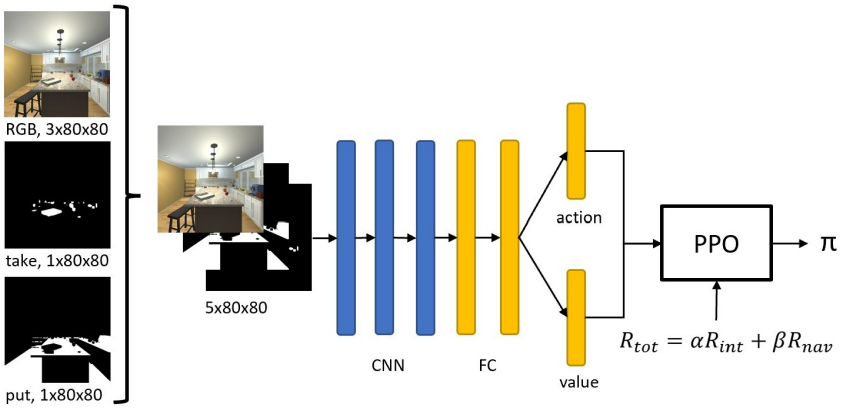


Figure 2: Training pipeline based on [1].

3.2 Mapping framework

We use panoptic multi-TSDFs [3] as our mapping framework which is explained in Section 2. Given color and depth image, segmentation mask, and pose, it creates an object-level map of the environment. Moreover, representing the scene as a collection of submaps allows for capturing changes. We add the functionality to publish the number of observed voxels of a selected object to this mapping framework. We then implement a ROS bridge between the simulator and the mapping framework which we use in the evaluation. The simulator publishes the egocentric RGB-D camera images, GT segmentation masks, and the pose of the agent which are used for creating an object-level map of the environment using [3]. The mapping framework subscribes to these and publishes the number of observed voxels of the objects chosen from the scene.

4 Experiments

In this section, we first provide details about our training procedure as well as the different experimental settings that we consider. We then explain the evaluation metrics and present the results of the ablation study that we perform for investigating the influence of the parameters α and β on the behaviors of the agent. Finally, we report the quantitative and qualitative evaluation results and demonstrate that the agent with interaction capabilities creates more complete object-level maps.

4.1 Training details

We train RL policies in a vectorized setting with 12 environments where a random scene among 30 training scenes is selected after each episode. The positions of the objects in the scene as well as the initial position of the agent are randomized at the beginning of each episode. We perform validation in 8 episodes from 5 different scenes. All policies are trained for 1 million frames with a learning rate of 0.0001 and rollouts of 256 time steps. Training lasts approximately 12 hours with an NVIDIA Quadro T2000 Mobile GPU.

We consider two experimental settings: interaction plus navigation and navigation. In the first setting, we encourage the agent to explore the environment and interact with objects by giving reward for each new interacted object and for each new visited position. This means that $\alpha > 0$ and $\beta > 0$ in Eq. 2. In the second setting, we encourage the agent to explore the environment by only giving reward for each new visited position. This means that $\alpha = 0$ and $\beta > 0$ in Eq. 2.

4.2 Evaluation metrics

For quantitative evaluations, we use three metrics: object coverage, position coverage, and the number of observed voxels of objects. The evaluation pipeline is shown in Figure 3. We obtain the first

two metrics from the simulator and the last metric from the mapper using the ROS bridge. Object coverage (OC) is the fraction of objects that the agent interacts with out of all interactable objects and position coverage (PC) is the fraction of positions that the agent visits out of all reachable positions as given in Eq. 3. These metrics are used to investigate the influence of the parameters α and β on the behaviors of the agent. We have 5 evaluation scenes and for each evaluation scene, we generate 10 randomized episodes of 1024 time steps each after which OC and PC roughly converge.

$$\text{OC} = \frac{\text{\# of interacted objects}}{\text{\# of all interactable objects}} \quad \text{PC} = \frac{\text{\# of visited positions}}{\text{\# of all reachable positions}} \tag{3}$$

We use the ROS bridge to send the egocentric RGB-D camera images, GT segmentation masks, and the pose of the agent to the mapping framework [3] and receive the number of observed voxels of an object. This metric is used to determine whether the agent with interaction capabilities observes more voxels of objects and creates more complete object-level maps. Since we use GT segmentation masks, all the voxels that we map for an object actually belongs to that object. Therefore, a higher number of observed voxels corresponds to a more complete object reconstruction. We first run the trained policies in each evaluation scene to see which objects the agent interacts with. For each evaluation scene, we then randomly select some of these objects and generate a randomized episode of 400 time steps after which the metric roughly converges. We perform evaluation for the selected objects in these episodes to obtain their number of observed voxels. In addition to the quantitative evaluation metrics, we also qualitatively evaluate the reconstructions of objects in the resulting maps in terms of completeness.

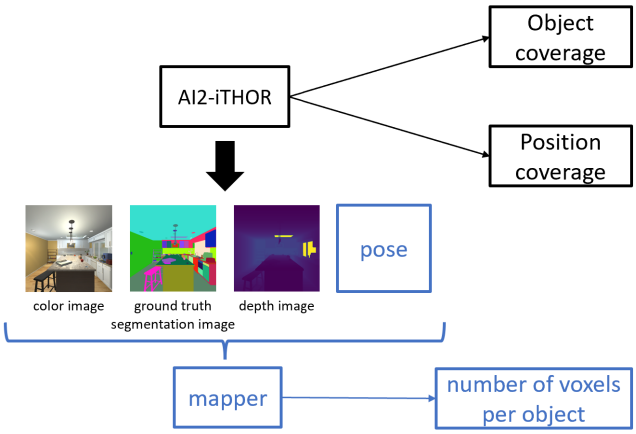


Figure 3: Evaluation pipeline.

4.3 Results

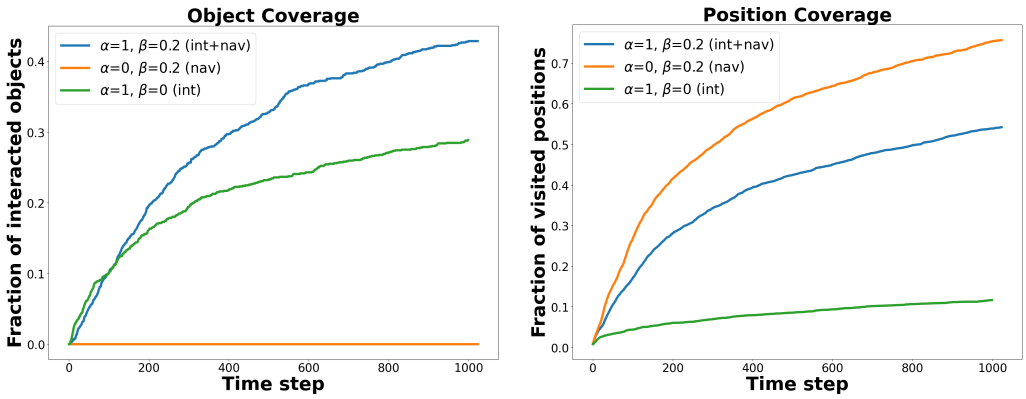
Ablation study. We experiment with different α and β values to investigate the influence of these parameters and validate that our reward design makes the agent perform the desired interaction and navigation behaviors. In addition to the two experimental settings explained in Section 4.1, we also consider a setting (interaction) where we encourage the agent to interact with objects by only giving reward for each new interacted object. This means that $\alpha > 0$ and $\beta = 0$ in Eq. 2. The α and β values that we use for training these three policies are given in Table 1. We observed that navigation dominates interaction when we assign high values to β. Hence, we decided to give less weight to navigation in the first setting to maximize interactive behaviors while navigating in the scene.

The results for object coverage and position coverage for these settings are shown in Fig 4a and 4b. For the generated 50 episodes (see Section 4.2), the average number of reachable positions is 130 and the average number of interactable objects is 25. The agent which is rewarded for both navigation and interaction interacts with 45% of all possible objects and visits 55% of all possible positions.

Setting	α	β
Interaction + navigation	1.0	0.2
Navigation	0.0	0.2
Interaction	1.0	0.0

Table 1: The α and β values that we use for training three different policies.

Moreover, the agent which is rewarded only for navigation visits 75% of possible positions. This means that there is a trade-off between interaction and exploration. On the other hand, the agent which is rewarded only for interaction does not outperform the interaction plus navigation case. It interacts with 30% of all possible objects and visits 10% of all possible positions. Since it is not incentivized to explore the environment, it only visits a few new locations and sees only a few new objects to interact with. This is the reason why the object coverage is lower than the interaction plus navigation case.



(a) Object coverage.

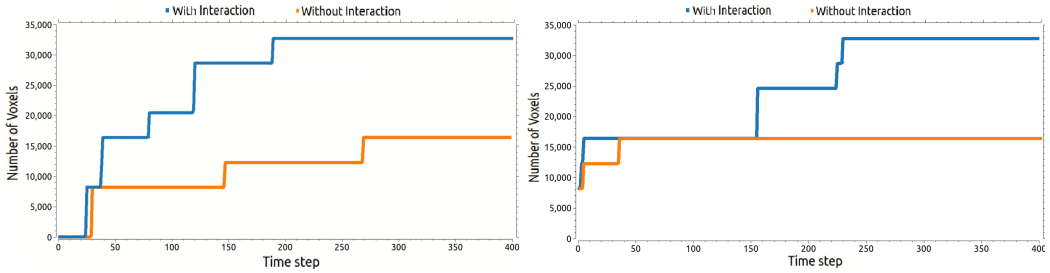
(b) Position coverage.

Figure 4: (a) Object coverage and (b) position coverage for three different policies: interaction + navigation (blue), navigation (orange), and interaction (green).

These results show that our reward signal succeeds in making the agent explore the environment and interact with the objects. This is a requirement to validate our hypothesis that interactions help with creating more complete object-level maps. Hence, we use $\alpha = 1$ and $\beta = 0.2$ to train the agent with interaction capabilities and $\alpha = 0$ and $\beta = 0.2$ to train the agent without interaction capabilities.

Object reconstruction results. We determine the number of observed voxels of individual objects for the agent with and without interaction capabilities to see whether interactions help with creating more complete object-level maps. We report the results for two selected objects (bowl and pan) in Figure 5a and 5b. It should be noted that all the voxels that we map for an object actually belongs to that object since we use GT segmentation masks. Therefore, a higher number of observed voxels of an object corresponds to a more complete object reconstruction.

The results show that the agent with interaction capabilities observes more voxels of objects. Since the agent with interaction capabilities can pick up and put down objects, it can see more parts of the objects compared to the agent without interaction capabilities. The jumps in the curves given in Figure 5a and 5b either corresponds to a viewpoint change or an interaction with the object. The curves for the case with interaction have more jumps because the agent not only sees the objects from different viewpoints while navigating in the environment but also interacts with them and sees different parts. We also visualize the reconstructed submaps in the mapper for qualitative evaluation to assess whether the increase in the number of voxels actually reflects more complete object-level reconstructions. The reconstructed submaps are shown along with the real scenes in Figure 6 where each color represents a different submap, i.e. object.



(a) Number of observed voxels of bowl.

(b) Number of observed voxels of pan.

Figure 5: The number of observed voxels of (a) bowl and (b) pan for the agent with (blue) and without (orange) interaction capabilities.

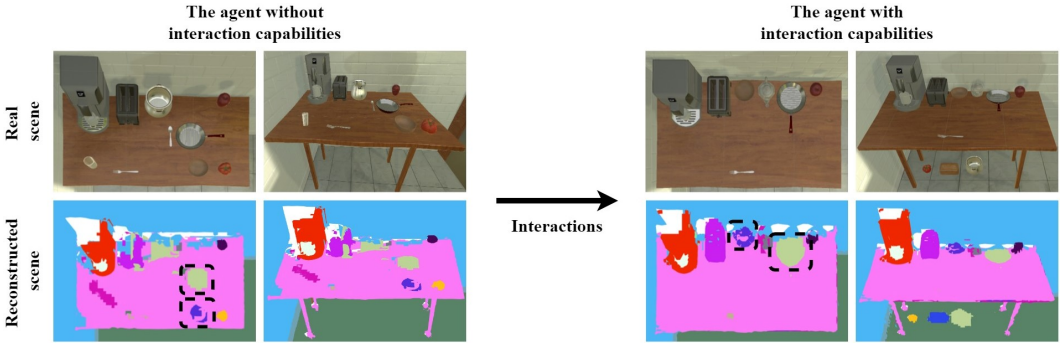


Figure 6: The real scene (top) and the reconstructed submaps (bottom) for the agent with (right) and without (left) interaction capabilities. The bowl (dark blue) and the pan (light green) appear more complete in the case with interaction. On the other hand, there are more incomplete and bad reconstructions in the case without interaction.

The results show that the objects are reconstructed better in the case with interaction since the agent observes more voxels of the objects. The selected objects (bowl and pan) also appear more complete which is consistent with the results in Figure 5. Moreover, there are fewer bad reconstructions since the agent can observe the areas where the objects are located more closely. Hence, we can say that the agent can create more complete object-level maps by leveraging interactions.

We then look into the case where there are small objects hidden behind big objects. We expect that interactions would particularly be useful in this case since it is not possible to see the hidden objects without picking up the big objects. Therefore, we create a custom scene in Unity similar to the ones in AI2-iTHOR [11] where there is a shelf containing a row of big objects which occlude the row of small objects. The reconstructed submaps are shown along with the real scenes in Figure 7.

The results show that the shelf appears more complete in the case with interaction since the agent can observe the back of it by picking up the big objects. The small objects are also reconstructed in the case with interaction since the agent can resolve occlusions. Therefore, we can say that the agent can observe small objects hidden behind big objects by leveraging interactions. This means that interactions help with creating a more complete map of the environment.

5 Discussion

5.1 Limitations and future work

One limitation is that we do not use a direct reward signal from the mapper that encourages observing more voxels. Although we tried to reward the agent for each new interacted object (R_{int}) and for the increase in number of observed voxels which we used in the evaluation, the ROS bridge leads to

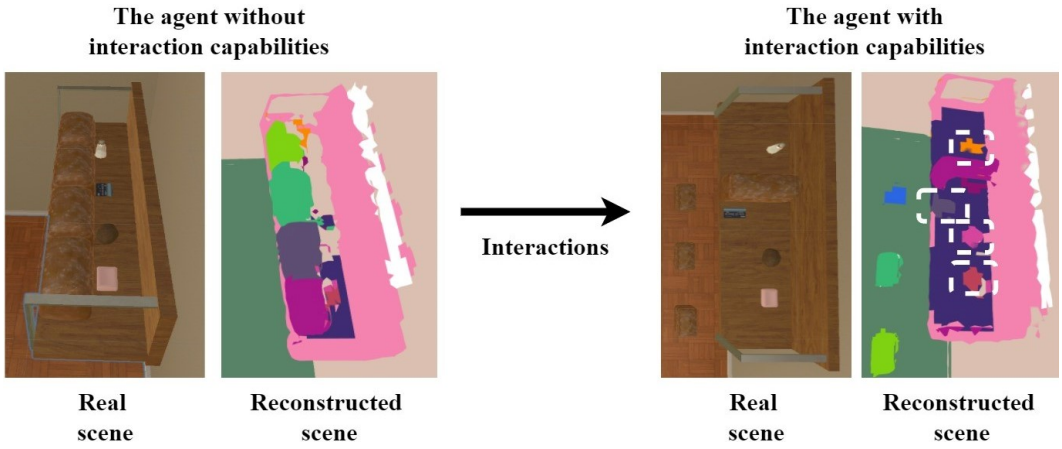


Figure 7: The real scenes and the reconstructed submaps for the agent with (right) and without (left) interaction capabilities. The saltcellar (orange), credit card (gray), potato (pink), and soap (red) are observed in the case with interaction since the agent picks up the breads in front of them. The shelf (dark purple) also appears more complete in the case with interaction.

memory and time constraints as explained in 3.1. Hence, we used the the proxy reward signal given in Eq. 2. Another limitation is that the simulator does not provide us a GT map that we can use for evaluating the accuracy of the reconstructed map as a whole. Therefore, we looked only at object reconstructions through the number of observed voxels. Since we use GT segmentation masks, we assumed that observing higher number of voxels means more complete object-level reconstruction.

There are also limitations caused by the mapping framework [3]. Firstly, the objects should be reconstructed at the right position due to using GT segmentation masks, but this is limited by the ability of the mapping framework [3] to update changes. Since panoptic multi-TSDFs [3] is designed for capturing long-term changes, it is not the most optimal mapping framework for capturing short-term changes. Secondly, a trail of objects is created when the agent picks up an object and navigates in the environment while holding it. We overcame this problem by not mapping the objects that are closer than 0.5 meters to the agent. However, this means that the objects disappear in the map when the agent picks them up and reappear when the agent puts them down.

There are possible directions to make our work more suitable for real-life operation. Firstly, the mapping framework [3] currently uses the GT segmentation masks from the simulator [11]. However, these masks are not given in reality and their usage highly influences the mapping quality as they govern the boundaries of objects. For these reasons, we can feed the output of a segmentation network, such as Detectron2 [29], to the mapping framework instead of GT segmentation masks from the simulator. Secondly, we use the GT affordance masks provided by the simulator in training which are not available in reality. To overcome this limitation, we can learn the affordance landscape of the environment through interactions using a U-Net [30] based network similar to [1]. Finally, we can utilize a different object-level mapper that focuses on short-term dynamics on the scene level.

5.2 Conclusion

In this work, we introduced an RL framework to incentivize an agent to navigate in an environment and interact with objects to perform more complete object-level mapping. We also created a ROS bridge between the RL and the mapping framework [3] to exchange the sensor readings and the number of observed voxels. Our results show that the agent with interaction capabilities creates more complete object-level maps compared to the agent without interaction capabilities. This means that interactions indeed help with performing more complete object-level mapping.

References

[1] T. Nagarajan and K. Grauman. Learning affordance landscapes for interaction exploration in 3d environments. In *NeurIPS*, 2020.

[2] W. Qi, R. T. Mullapudi, S. Gupta, and D. Ramanan. Learning to move with affordance maps. *CoRR*, abs/2001.02364, 2020. URL http://arxiv.org/abs/2001.02364.

[3] L. Schmid, J. Delmerico, J. Schönberger, J. Nieto, M. Pollefeys, R. Siegwart, and C. Cadena. Panoptic multi-tsdfs: a flexible representation for online multi-resolution volumetric mapping and long-term dynamic scene consistency. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[4] V. Reijgwart, A. Millane, H. Oleynikova, R. Siegwart, C. Cadena, and J. I. Nieto. Voxgraph: Globally consistent, volumetric mapping using signed distance function submaps. *CoRR*, abs/2004.13154, 2020. URL https://arxiv.org/abs/2004.13154.

[5] L. M. Schmid, V. J. F. Reijgwart, L. Ott, J. I. Nieto, R. Siegwart, and C. Cadena. A unified approach for autonomous volumetric exploration of large scale environments under severe odometry drift. *CoRR*, abs/2010.09859, 2020. URL https://arxiv.org/abs/2010.09859.

[6] M. Strecke and J. Stückler. Em-fusion: Dynamic object-level SLAM with probabilistic data association. *CoRR*, abs/1904.11781, 2019. URL http://arxiv.org/abs/1904.11781.

[7] M. Rünz and L. Agapito. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. *CoRR*, abs/1706.06629, 2017. URL http://arxiv.org/abs/1706.06629.

[8] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. J. Davison, and S. Leutenegger. Mid-fusion: Octree-based object-level multi-instance dynamic SLAM. *CoRR*, abs/1812.07976, 2018. URL http://arxiv.org/abs/1812.07976.

[9] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22 (268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

[11] E. Kolve, R. Mottaghi, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi. AI2-THOR: an interactive 3d environment for visual AI. *CoRR*, abs/1712.05474, 2017. URL http://arxiv.org/abs/1712.05474.

[12] T. Chen, S. Gupta, and A. Gupta. Learning exploration policies for navigation. *CoRR*, abs/1903.01959, 2019. URL http://arxiv.org/abs/1903.01959.

[13] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov. Learning to explore using active neural SLAM. *CoRR*, abs/2004.05155, 2020. URL https://arxiv.org/abs/2004.05155.

[14] N. Savinov, A. Dosovitskiy, and V. Koltun. Semi-parametric topological memory for navigation. *CoRR*, abs/1803.00653, 2018. URL http://arxiv.org/abs/1803.00653.

[15] K. Fang, A. Toshev, L. Fei-Fei, and S. Savarese. Scene memory transformer for embodied agents in long-horizon tasks. *CoRR*, abs/1903.03878, 2019. URL http://arxiv.org/abs/1903.03878.

[16] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra. Habitat: A platform for embodied AI research. *CoRR*, abs/1904.01201, 2019. URL http://arxiv.org/abs/1904.01201.

[17] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese. Gibson env: Real-world perception for embodied agents. *CoRR*, abs/1808.10654, 2018. URL http://arxiv.org/abs/1808.10654.

[18] N. Savinov, A. Raichuk, R. Marinier, D. Vincent, M. Pollefeys, T. P. Lillicrap, and S. Gelly. Episodic curiosity through reachability. *CoRR*, abs/1810.02274, 2018. URL http://arxiv.org/abs/1810.02274.

[19] S. K. Ramakrishnan, D. Jayaraman, and K. Grauman. An exploration of embodied visual exploration. *CoRR*, abs/2001.02192, 2020. URL http://arxiv.org/abs/2001.02192.

[20] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. *CoRR*, abs/1705.05363, 2017. URL http://arxiv.org/abs/1705.05363.

[21] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox. ALFRED: A benchmark for interpreting grounded instructions for everyday tasks. *CoRR*, abs/1912.01734, 2019. URL http://arxiv.org/abs/1912.01734.

[22] J. J. Gibson. *The Ecological Approach to Visual Perception*. Psychology Press, Nov. 2014. doi:10.4324/9781315740218. URL https://doi.org/10.4324/9781315740218.

[23] J. McCormac, A. Handa, A. J. Davison, and S. Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. *CoRR*, abs/1609.05130, 2016. URL http://arxiv.org/abs/1609.05130.

[24] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. *CoRR*, abs/1903.01177, 2019. URL http://arxiv.org/abs/1903.01177.

[25] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. I. Nieto. Volumetric instance-aware semantic mapping and 3d object discovery. *CoRR*, abs/1903.00268, 2019. URL http://arxiv.org/abs/1903.00268.

[26] A. Rosinol, M. Abate, Y. Chang, and L. Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. *CoRR*, abs/1910.02490, 2019. URL http://arxiv.org/abs/1910.02490.

[27] A. Kirillov, K. He, R. B. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. *CoRR*, abs/1801.00868, 2018. URL http://arxiv.org/abs/1801.00868.

[28] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym, 2016.

[29] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[30] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.