Hierarchical Object-to-Zone Graph for Object Navigation
— Supplementary Materials —

Sixian Zhang[1,2], Xinhang Song[1,2], Yubing Bai[1,2], Weijie Li[1,2], Yakui Chu[4], Shuqiang Jiang[1,2,3]
[1]Key Lab of Intelligent Information Processing Laboratory of the Chinese Academy of Sciences (CAS),
Institute of Computing Technology, Beijing [2]University of Chinese Academy of Sciences, Beijing
[3]Institute of Intelligent Computing Technology, Suzhou, CAS [4]Huawei Application Innovate Laboratory, Beijing
{sixian.zhang, xinhang.song, yubing.bai, weijie.li}@vipl.ict.ac.cn
chuyakui@huawei.com; sqjiang@ict.ac.cn

1. Video Demo

A video demo that visualizes the construction of HOZ graph, navigation with HOZ graph and more case studies can be found at the following url:

https://drive.google.com/file/d/1UtTcFRhFZLkqgalKom6_9GpQmsJfXAZC/view?usp=sharing

2. Navigation Target

The target objects of different scenes in AI2THOR [4] are shown in Table 1. Our training and testing share the consistent target objects categories, though the testing environments are new and unseen.

Considering that each environment in AI2THOR usually contains one room, the agent navigation may be limited to short trajectories. Thus, for longer trajectories object navigation, we also conduct experiments on a more complex simulator RoboTHOR [2], which has 2.4 times larger area and 5.5 times longer trajectory length than AI2THOR. The environment in RoboTHOR usually contains a variety of rooms. To highlight the differences between AI2THOR and RoboTHOR, we define each environment in AI2THOR as *room* and that in RoboTHOR as *apartment*. In RoboTHOR, 12 objects categories are selected as target objects for training and testing, involving *Book, Bowl, Chair, Plate, Television, Floor Lamp, Garbage Can, Alarm Clock, Desk Lamp, Laptop, Pot, CellPhone*. The experimental results are shown in Section 4.1.

3. More Ablation Studies

3.1. Clustering information

In our method, we sample a set of features (f, l) according to the observations in the environments, where f is a bag-of-objects vector representing objects categories detected in view, and l represents the sample location. Then we implement feature clustering on f, and each obtained cluster serves as a zone node in room-wise HOZ. That is to say, our zone node is only based on visual information. In order to further explore the impact of clustering, we introduce the additional location information and cluster on both (f, l). Table 2 demonstrates the navigation performance with these two clustering methods. The results show that clustering on both visual and location information drops 2.40/2.12% and 2.16/1.05% in SR and SAE and slightly improves in SPL, suggesting that the additional location information narrows the range of our proposed *zone*. In other words, our HOZ (clustering on visual information) treats all regions where agent can observe similar objects with a specified direction as a zone, while clustering with both visual and location information restrains the zone region merely around these objects. Thus, location is more like a constraint rather than helpful information, limiting the visual

Table 1. **Object categories for navigation.** The target objects categories of different room types in AI2THOR [4].

Scenes	Objects
Kitchen	Fridge, Light Switch, Pot, Coffee Machine, Sink, Pan, Chair, Plate, Bowl, Toaster, Stove Burner, Kettle, Microwave, Garbage Can
Living Room	FloorLamp, Chair, Plate, Light Switch, Garbage Can, Laptop, Remote Control, Book, Television, Desk Lamp
Bedroom	Book, Light Switch, Bowl, Desk Lamp, Laptop, Chair, Alarm Clock, Garbage Can,
Bathroom	Light Switch, Garbage Can, Sink

Table 2. **Comparisons with different information used for clustering (%).** The zone clustering is based on different information, including visual information f (Visual) and location information l (Location).

Visual	Location	ALL			$L \geq 5$		
		SR	SPL	SAE	SR	SPL	SAE
$\surd$		**70.62**$_{\pm 1.70}$	40.02$_{\pm 1.25}$	**27.97**$_{\pm 2.01}$	**62.75**$_{\pm 1.73}$	**39.24**$_{\pm 0.56}$	**30.14**$_{\pm 1.34}$
$\surd$	$\surd$	68.22$_{\pm 1.54}$	**40.48**$_{\pm 1.07}$	25.81$_{\pm 1.78}$	60.63$_{\pm 1.46}$	37.92$_{\pm 0.48}$	29.09$_{\pm 1.01}$

Table 3. **Comparisons with different detection modules (%).** We compare the impact of utilizing a pre-trained detection model (Detection Pre) or the ground truth of object detection (Detection GT).

Module	ALL			$L \geq 5$		
	SR	SPL	SAE	SR	SPL	SAE
Detection Pre	65.12$_{\pm 1.03}$	37.86$_{\pm 0.93}$	24.36$_{\pm 0.91}$	53.42$_{\pm 1.43}$	35.37$_{\pm 0.71}$	25.32$_{\pm 1.04}$
Detection GT	**66.78**$_{\pm 0.73}$	**55.91**$_{\pm 0.46}$	**26.73**$_{\pm 0.26}$	**55.02**$_{\pm 0.68}$	**48.73**$_{\pm 0.31}$	**30.23**$_{\pm 0.33}$

generalization of the proposed HOZ graph. When the target object is not in view, agent needs to search more zones until discovering the target. It is obviously inefficient so that we obtain zone nodes for HOZ only based on visual information.

3.2. Object detection module

Table 3 shows the impact of different detection modules on navigation performance, where *Detection Pre* indicates that the detection module is pre-trained with labeled egocentric images sampled in simulator, and *Detection GT* indicates that the detection module is ground truth provided by simulator. The ablation with ground truth detection improves performance by 1.66/1.60 , 2.37/4.91 and **18.05/13.36** in SR, SAE and SPL (ALL/$L \geq 5$, %) respectively. The results demonstrate that accurately recognizing more objects can help agent navigate successfully in shorter trajectories. It is easy to understand because agent can take the most likely action at each step to obtain the high SPL. However, since the navigation task includes multiple decision steps, its success rate does not rely on taking the perfect action at each step. As long as most actions are reasonable, the agent can still achieve success. So the approximate results on SR and SAE indicate that our HOZ graph still makes sense in guiding unseen object navigation.

3.3. The ablations of graph settings

Since our HOZ graph adds more parameters to the model, we perform additional ablations of zone nodes and edges, as indicated in Table 4. To assess if the gain in network performance is due to the increased number of parameters or the information contained in the HOZ graph's nodes and edges, We respectively set the edges and nodes of the HOZ graph to random. The experimental results show that the control experiments with random settings perform worse than the original value, demonstrating the efficacy of zone information (nodes) and spatial priors (edges).

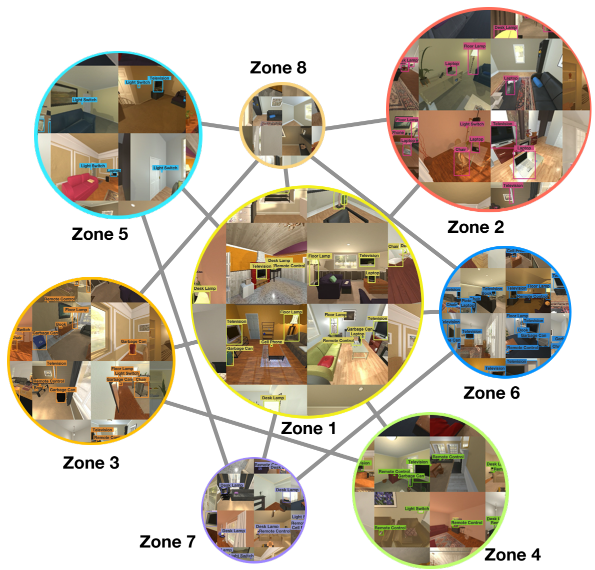

Figure 1. **Zones nodes of Hierarchical Object-to-Zone Graph.** 8 different colors represent different zones. To highlight the objects contained in these zones, we mark them with bounding boxes.

4. More comparisons with the related works

4.1. Experiments on RoboTHOR

For longer trajectories object navigation, we also conduct experiments on RoboTHOR [2] simulator. RoboTHOR consists of 89 apartments, 75 for training and validation, while the testing data have not yet been made public. Therefore, we choose 60 apartments for training, 5 for validation and 10 for testing. Since the regions in RoboTHOR are simply separated with several clapboard, we treat each apartment as a whole rather than subdividing it into scattered scenes. Therefore, different from the construction of scene-wise HOZ graph in AI2THOR, we build apartment-wise HOZ graph in RoboTHOR and establish a unified HOZ graph combing all apartments.

Table 5 illustrates that our method still outperforms the state-of-the-art with a large margin by 2.66/2.30 in SR,

Table 4. **More ablations of graph settings (%).** The parameters of nodes or edges are randomly set (R) or kept (K).

Nodes	Edges	ALL			$L \geq 5$		
		SR	SPL	SAE	SR	SPL	SAE
R	R	$67.81_{\pm 0.62}$	$38.92_{\pm 0.22}$	$24.13_{\pm 0.35}$	$57.84_{\pm 0.81}$	$38.22_{\pm 0.44}$	$24.02_{\pm 0.52}$
	K	$68.52_{\pm 1.05}$	$39.83_{\pm 0.52}$	$26.52_{\pm 0.62}$	$58.61_{\pm 0.82}$	$38.73_{\pm 0.62}$	$28.73_{\pm 0.53}$
K	R	$69.33_{\pm 0.32}$	$39.71_{\pm 0.32}$	$26.63_{\pm 0.13}$	$59.93_{\pm 0.53}$	$39.14_{\pm 0.45}$	$29.01_{\pm 0.312}$
	K	$\mathbf{70.47}_{\pm 0.35}$	$\mathbf{40.66}_{\pm 0.47}$	$\mathbf{27.85}_{\pm 0.44}$	$\mathbf{62.17}_{\pm 0.26}$	$\mathbf{40.14}_{\pm 0.46}$	$\mathbf{30.33}_{\pm 0.25}$

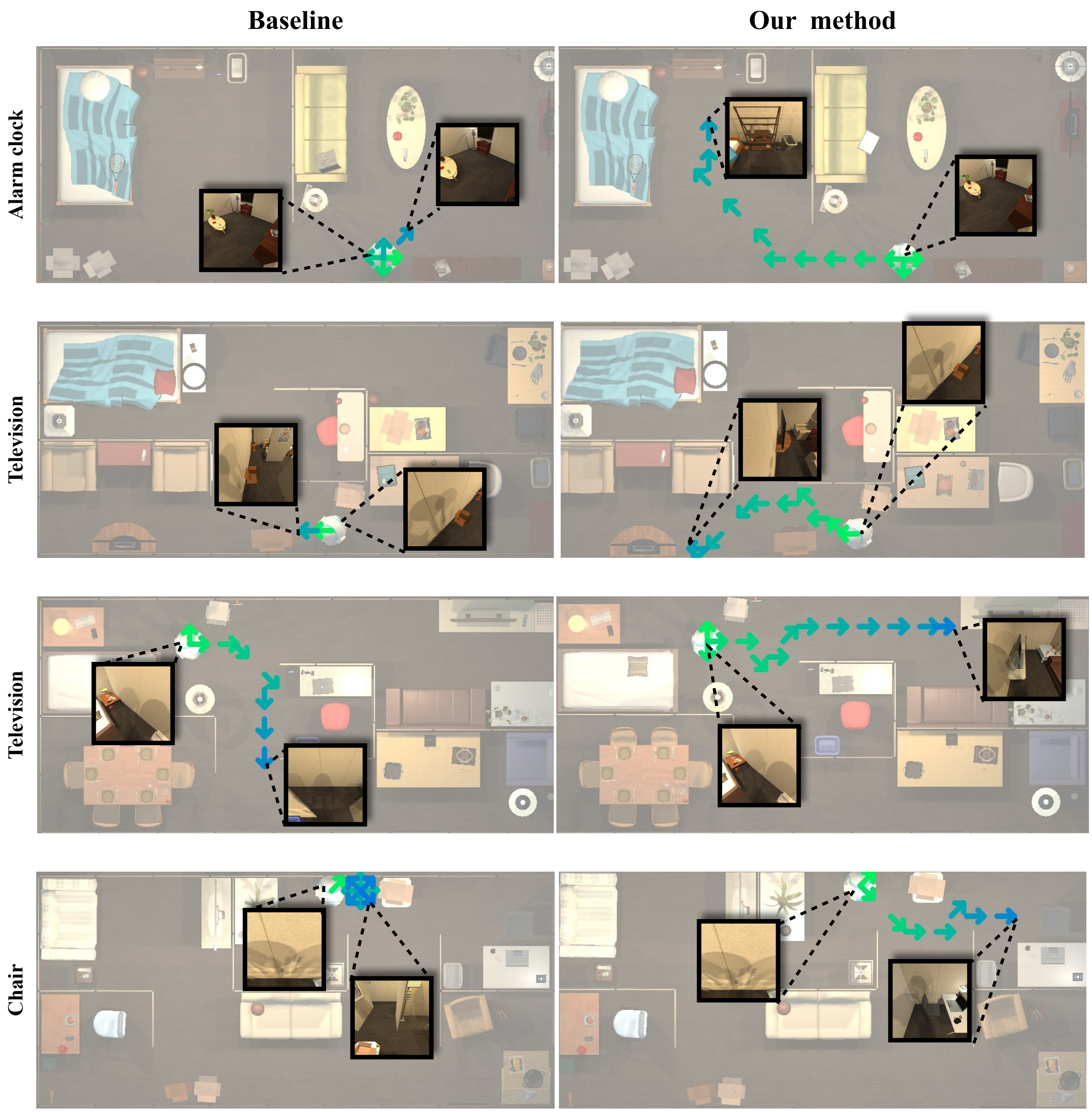

Figure 2. **Visualization of trajectory in RoboTHOR.** Black arrows represent rotations. The trajectory of the agent is illustrated with green and blue arrows, where green is the beginning and blue is the end.

Table 5. **Comparisons with the related works in RoboTHOR [2] (%).** We repeat the evaluations similar to AI2-Thor on RoboTHOR.

Method	ALL SR	ALL SPL	ALL SAE	$L \geq 5$ SR	$L \geq 5$ SPL	$L \geq 5$ SAE
Non-adaptive method						
Random	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
A3C (baseline)	$26.41_{\pm 0.52}$	$16.61_{\pm 0.34}$	$13.15_{\pm 0.43}$	$17.42_{\pm 0.21}$	$12.23_{\pm 0.66}$	$10.94_{\pm 0.35}$
SP [7]	$28.04_{\pm 0.33}$	$17.63_{\pm 0.26}$	$14.23_{\pm 0.25}$	$21.66_{\pm 0.32}$	$15.14_{\pm 0.46}$	$13.27_{\pm 0.34}$
ORG [3]	$29.61_{\pm 0.71}$	$19.23_{\pm 0.94}$	$14.72_{\pm 0.64}$	$22.53_{\pm 0.55}$	$15.73_{\pm 0.86}$	$13.82_{\pm 0.44}$
Ours (HOZ)	$\mathbf{32.27}_{\pm 1.14}$	$\mathbf{20.48}_{\pm 0.63}$	$\mathbf{17.18}_{\pm 0.42}$	$\mathbf{24.83}_{\pm 0.72}$	$\mathbf{16.89}_{\pm 0.50}$	$\mathbf{15.62}_{\pm 0.55}$
Self-supervised method						
SAVN [5]	$28.42_{\pm 0.41}$	$17.82_{\pm 0.33}$	$13.91_{\pm 0.24}$	$22.13_{\pm 0.32}$	$15.34_{\pm 0.45}$	$13.01_{\pm 0.24}$
ORG-TPN [3]	$30.01_{\pm 1.22}$	$20.51_{\pm 0.74}$	$14.52_{\pm 0.93}$	$22.25_{\pm 0.63}$	$16.64_{\pm 0.35}$	$13.83_{\pm 0.45}$
Ours (HOZ-TPN)	$\mathbf{33.28}_{\pm 1.62}$	$\mathbf{22.13}_{\pm 0.91}$	$\mathbf{16.66}_{\pm 0.62}$	$\mathbf{24.98}_{\pm 1.32}$	$\mathbf{18.05}_{\pm 0.64}$	$\mathbf{15.57}_{\pm 0.76}$

Table 6. **Comparisons with the related works in AI2THOR (%).** These results are the supplement for Table 3 in the main text.

Method	All Suc.	All SPL	All SAE	$L \geq 5$ Suc.	$L \geq 5$ SPL	$L \geq 5$ SAE
Non-adaptive method						
Random	$3.56_{\pm 2.74}$	$1.73_{\pm 1.52}$	$0.41_{\pm 0.52}$	$0.27_{\pm 0.22}$	$0.07_{\pm 0.06}$	$0.06_{\pm 0.05}$
A3C (baseline)	$57.35_{\pm 1.92}$	$33.78_{\pm 1.33}$	$19.02_{\pm 1.36}$	$45.77_{\pm 2.17}$	$30.65_{\pm 1.01}$	$20.04_{\pm 1.87}$
SP [7]	$62.16_{\pm 0.70}$	$37.01_{\pm 0.68}$	$23.39_{\pm 0.69}$	$50.86_{\pm 0.34}$	$34.17_{\pm 0.85}$	$24.35_{\pm 0.74}$
ORG [3]	$66.38_{\pm 0.95}$	$38.42_{\pm 0.22}$	$25.36_{\pm 0.43}$	$55.55_{\pm 1.89}$	$36.26_{\pm 0.39}$	$27.53_{\pm 0.48}$
Ours (HOZ)	$\mathbf{70.62}_{\pm 1.70}$	$\mathbf{40.02}_{\pm 1.25}$	$\mathbf{27.97}_{\pm 2.01}$	$\mathbf{62.75}_{\pm 1.73}$	$\mathbf{39.24}_{\pm 0.56}$	$\mathbf{30.14}_{\pm 1.34}$
Self-supervised method						
SAVN [5]	$63.32_{\pm 1.17}$	$37.62_{\pm 0.86}$	$21.97_{\pm 0.21}$	$52.38_{\pm 0.73}$	$35.31_{\pm 0.79}$	$24.64_{\pm 0.52}$
ORG-TPN [3]	$67.31_{\pm 1.14}$	$\mathbf{39.53}_{\pm 1.01}$	$23.07_{\pm 0.24}$	$57.41_{\pm 0.71}$	$38.27_{\pm 0.63}$	$26.37_{\pm 0.57}$
Ours (HOZ-TPN)	$\mathbf{73.15}_{\pm 1.01}$	$39.22_{\pm 1.27}$	$\mathbf{29.49}_{\pm 0.11}$	$\mathbf{64.58}_{\pm 0.74}$	$\mathbf{39.80}_{\pm 0.57}$	$\mathbf{30.92}_{\pm 0.40}$

1.25/1.16 in SPL and 2.46/1.80 in SAE metric (ALL/$L \geq 5$, %). Besides, compared with self-supervised methods, our method equipped with the equal self-supervised adaptive module also gains significant improvement of 3.27/2.73 in SR, 1.62/1.41 in SPL and 2.14/1.74 in SAE metric (ALL/$L \geq 5$, %).

In addition, we supplement the experimental results of variance for Table 3 in the main text. The complete experimental results are shown in Table 6.

4.2. Comparisons with semantic map

In addition, Chaplot et al. [1] attempt to construct the episodic semantic map and use it to explore the unseen environment. Different from our method that only relies on RGB input, the semantic map is constructed based on a variety of inputs, including RGB-D input, segmentation mask and GPS coordinate. We evaluate the HOZ graph and the semantic map in Gibson [6], where all methods utilize the RGB-D input, segmentation mask and GPS coordinate. As indicated in Table 7, since the SLAM-based method processes multiple inputs more completely, the performance of the baseline with the HOZ graph is slightly inferior than SemExp. However, incorporating the HOZ graph for SemExp improves the SR, SPL and SAE by 1.18, 0.34, 0.41 (ALL, %) respectively, indicating that the HOZ graph and SLAM-based method learn complementary information. The experimental results demonstrate that the HOZ graph is also effective when combined with SLAM-based methods.

Table 7. **Comparisons with the semantic map in Gibson (%).** The baseline is the A3C model with a simple visual embedding layer to encode various inputs. Since the path lengths of all episodes are larger than 5, the subset of $L \geq 5$ is excluded.

Method	SR	SPL	SAE
Baseline + HOZ	$43.47_{\pm 0.51}$	$12.88_{\pm 0.36}$	$11.67_{\pm 0.51}$
SemExp [1]	$44.01_{\pm 0.47}$	$14.34_{\pm 0.42}$	$12.32_{\pm 0.43}$
SemExp + HOZ	$\mathbf{45.19}_{\pm 0.35}$	$\mathbf{14.68}_{\pm 0.38}$	$\mathbf{12.73}_{\pm 0.45}$

5. Qualitative Results

5.1. The HOZ graph visualization

Figure 1 illustrates the visualization of our HOZ graph. We visualize the zones nodes in a scene-wise HOZ graph (e.g., living room), which is the fusion of 20 room-wise HOZ graphs. There are 8 zones marked with different colors and each zone consists of similar objects distribution. Even though there are overlapped objects among zones, each zone has semantically representative objects. For instance, in Figure 1, $zone_2$, $zone_3$, $zone_6$ focus on laptop, garbage can and television, respectively.

5.2. Navigation trajectory

Figure 2 qualitatively compares our method with the baseline in RoboTHOR. Benefiting from the sub-goals guidance and online-updating of proposed HOZ graph, agent can still adopt reasonable actions even in the long trajectory unseen navigation task, while the baseline model often falls into confusion and struggles with spinning around.

References

[1] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Russ R. Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 4.2, 7

[2] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, Luca Weihs, Mark Yatskar, and Ali Farhadi. Robothor: An open simulation-to-real embodied AI platform. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 3161–3171, 2020. 2, 4.1, 5

[3] Heming Du, Xin Yu, and Liang Zheng. Learning object relation graph and tentative policy for visual navigation. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VII*, pages 19–34, 2020. 5, 6

[4] Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: an interactive 3d environment for visual AI. *CoRR*, abs/1712.05474, 2017. 2, 1

[5] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6750–6759, 2019. 5, 6

[6] Fei Xia, Amir Roshan Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 9068–9079, 2018. 4.2

[7] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. 5, 6