PONI: Potential Functions for ObjectGoal Navigation with Interaction-free Learning Supplementary Materials

Santhosh Kumar Ramakrishnan[1,2], **Devendra Singh Chaplot**[1], **Ziad Al-Halah**[2], **Jitendra Malik**[1,3], **Kristen Grauman**[1,2]
[1]Meta AI [2]UT Austin [3]UC Berkeley

This document provides additional information about our experimental settings and supporting qualitative visualizations. Below is a summary of the sections in the supplementary file:

- (§S1) Limitations
- (§S2) Additional experimental details
- (§S3) Non-interactive baseline implementation details
- (§S4) Masking strategy for semantic maps
- (§S5) Action costs for long-term goal sampling
- (§S6) Influence of object PF over time
- (§S7) Examples of semantic maps
- (§S8) Examples of potential functions
- (§S9) Visualizing ObjectNav episodes

Additionally, we provide a supplementary video that visualizes complete ObjectNav trajectories and provides an intuition of how **PONI** works. These are animated versions of the ObjectNav episodes visualized in Fig. 5 from the main paper, and Fig. S6 in the supplementary.

S1. Limitations

In Sec. 5 from the main paper, we discussed the benefits of our proposed PONI method both in terms of achieving state-of-the-art results on ObjectNav, as well as computational benefits during training. However, we would like to acknowledge some limitations of our approach.

One of our main limitations is our reliance on the semantic map as the only source for deciding when an object is found (i.e., to execute STOP). As discussed in the ablation study from Sec. 5, our performance is sensitive to the image segmentation quality. The success rate goes down by 14.9% in Gibson and 45.4% on MP3D relative to the performance with ground-truth segmentation (see Tab. S1). Note that our ratio of SPL to success remains relatively stable (only 6% reduction on Gibson and 19% on MP3D), indicating that our search efficiency is not affected significantly by segmentation errors. Unlike end-to-end RL methods which may learn to be robust to the sensory noise, we do not have an inbuilt mechanism to handle failures in segmentation. This limitation of interaction-free learning can potentially be addressed by using the latest advances in segmentation. Additionally, segmentation errors in simulation can be caused by reconstruction artifacts in the 3D scenes. Experimenting on higher quality scenes, or testing in the real world may address this limitation.

	Gibson (val)			MP3D (val)		
Method	Succ.	SPL	SPL / Succ.	Succ.	SPL	SPL / Succ.
PONI + GT-s	86.5	51.5	0.596	58.2	27.5	0.47
PONI	73.6	41.0	0.557	31.8	12.1	0.38
Relative drop	-14.9%	-20.4%	-6.5%	-45.4%	-56.0%	-19.1%

Table S1. Impact of segmentation errors on **PONI**'s ObjectNav performance. The first row shows performance with ground-truth segmentation. The last row shows the relative drop in the performance when we remove ground-truth segmentation.

We also rely on access to human-annotated semantic information in 3D scenes. While this is standard practice for most approaches in ObjectNav [6, 8, 11, 12], alternative strategies exist for learning ObjectNav without access to any ground-truth semantic annotations in 3D scenes [5]. Such self-supervised approaches have the potential to be more scalable than our supervised approach. However, to the best of our knowledge, there are no purely self-supervised methods that achieve state-of-the-art results for ObjectNav.

S2. Additional experimental details

We provide additional information about the experiments to supplement the main paper. The Gibson Ob-

jectNav dataset from [6] consists of 6 object categories: 'chair', 'couch', 'potted plant', 'bed', 'toilet', and 'tv'. The train split episodes are generated on-the-fly during training from 25 train scenes in Gibson tiny. The val split consists of 1,000 episodes from 5 val scenes in Gibson tiny. The MP3D ObjectNav dataset from the Habitat challenge consists of 21 object categories: 'chair', 'table', 'picture', 'cabinet', 'cushion', 'sofa', 'bed', 'chest of drawers', 'plant', 'sink', 'toilet', 'stool', 'towel', 'tv monitor', 'shower', 'bathtub', 'counter', 'fireplace', 'gym equipment', 'seating', and 'clothes'. The train / val splits consist of 263,2422 / 2,195 episodes from 61 / 11 MP3D scenes. We share these datasets publicly on our project website: https://vision.cs.utexas.edu/projects/poni/.

S3. Non-interactive baseline details

We provide more details about the non-interactive baselines from Sec.4.1 in the main paper.

BC: We train a recurrent policy using behavior cloning. The policy consists of a ResNet-50 backbone for encoding RGB-D observations, and MLP layers to encode the agent's pose and goal object category. The outputs of these models are concatenated and fed to a 2-layer LSTM with 512-D hidden states to aggregate observations over time. The LSTM hidden states are used by a linear layer to predict a probability distribution over the set of agent actions. This is a standard policy architecture for recent navigation methods [8, 10]. The idea in behavior cloning is to supervise the policy to classify the ground-truth action sampled by an expert at each step. We use the greedy shortest-path sampler from Habitat [9] to sample expert actions to the goal object. The model is trained using the cross-entropy loss.

Predict-θ: We modify the potential function network from Sec. 3.3 in the main paper to predict the direction to the nearest object from each category. We discretize the directions from 0° to 360° into 8 classes. The model uses the partial semantic map as input and predicts a $N \times 8$ array of direction probabilities for the N object categories. The model is trained on the semantic maps dataset from Sec. 3.6 in the main paper with the cross-entropy loss per object category. During ObjectNav, we sample the most-likely direction to the goal object category, and navigate to the closest frontier along this direction.

Predict-xy: We modify the potential function network from Sec. 3.3 to predict the (x, y) map location of the nearest object from each category. The model uses the partial semantic map as input and regresses the normalized position values from 0 to 1 (same action space as [6]). The model is trained on the semantic maps dataset from Sec. 3.6 in the main paper with the mean-squared error loss per object category. During ObjectNav, we sample the predicted (x, y) location as the long-term navigation goal.

Method	MP3D (val)		
	Succ. ↑	SPL ↑	DTS ↓
PONI (square)	31.8	**12.1**	**5.1**
PONI (view-cone)	**31.9**	**12.1**	**5.1**

Table S2. We measure the impact of masking strategy for generating training samples during PF training. In 'square', we unmask a $3\text{m} \times 3\text{m}$ square region centered around each shortest-path location on the semantic map. In 'view-cone', we unmask a viewing cone in-front of the agent with 3m radius and 90° field-of-view. Both strategies perform comparably on the MP3D (val) split.

Predict-$\mathcal{A}$: We modify the potential function network from Sec. 3.3 to predict the low-level navigation action for reaching the nearest object from each category. The model uses the partial semantic map as input and classifies, per object category, the action for reaching the nearest object along the shortest-path. The model is trained on the semantic maps dataset from Sec. 3.6 in the main paper with the cross-entropy loss per object category. During ObjectNav, we sample the most-likely prediction action to reach the goal object.

S4. Masking strategy for semantic maps

We described our strategy to sample exploration masks in Sec. 3.6 in the main paper, where we sampled random shortest paths and revealed a $3\text{m} \times 3\text{m}$ square patch around each location on the shortest-path. We now experiment with an alternative strategy where we reveal a viewing cone in-front of the agent, where we aim to mimic the agent's visibility in 3D space. The results are shown in Tab. S2. We find that it performs comparably with the 'square' strategy, which we use as the default option for all of our experiments.

S5. Action costs for long-term goal sampling

As described in Sec. 3.3 in the main paper, we sample long-term goals by selecting the maxima of the overall potential (see Eqn. 2). An alternative is to take into account the cost of navigating from the agent's location to each map location as well (i.e., an action cost). For example, when there are two locations with similarly valued potentials, the agent could choose to navigate to the nearer one. We incorporate this into PONI by adding a distance potential function U^d that is 1.0 at the agent's location and linearly decreases as we move away:

$$U_t = \alpha U_t^a + \beta U_t^o + \gamma U_t^d, \quad \text{where} \quad \alpha + \beta + \gamma = 1. \quad (1)$$

The constants α, β, γ are determined through a grid-search over MP3D (val). We compare the best results from

Method	MP3D (val)		
	Succ. ↑	SPL ↑	DTS ↓
`PONI`	**31.8**	**12.1**	**5.1**
`PONI + act-cost`	30.3	11.6	5.3

Table S3. We measure the impact of using action costs to sample long-term goals for PONI.

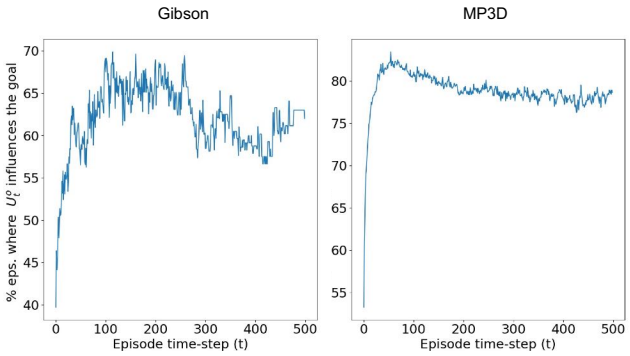

Figure S1. **Influence of object PF on action selection**: The plots show the percentage of of episodes where the selected goal location was influenced by the object PF (y-axis) at a given time-step of an episode (x-axis). The contribution of the object PF is higher during later stages of the episode.

this grid-search (**`+ act-cost`**) with our current method in Tab. S3. PONI does not benefit from adding the action costs. Based on our qualitative analysis, we find that the PONI agent typically continues to explore a single frontier sufficiently before moving away to other frontiers. Therefore, prioritizing the best frontier at all times (regardless of how far away it is) works well in practice.

S6. Influence of object PF over time

In Fig. 5 from main and Fig. S6, we qualitatively demonstrated that the agent explores using the area PF in early stages of the episode, and then uses the object PF to find objects. We now quantitatively demonstrate this. In Fig. S1, we plot the influence of the object PF on the goal location selection at a given time t on Gibson (val) and MP3D (val), i.e, the percentage of episodes where the selected goal location differs from the maxima of the area PF at t (by atleast 1m euclidean distance). The contribution of the object PF is higher in the later stages of the episode, after sufficient information has been gathered. This is intuitive: we cannot anticipate unseen objects without sufficient context on the map.

S7. Examples of semantic maps

We show examples from the semantic map datasets we used for training the potential function network in Figs. S2 and S3. The Gibson semantic maps contain up to 15 object categories of which 6 categories are goal categories (same as [6]). The MP3D semantic maps contain up to 21 object categories of which all are goal categories (same as the Habitat challenge [2]). These maps are computed by performing an orthographic projection of the 3D point-cloud annotations (following [3]). In addition to the pipeline from [3], we perform additional pre-processing to obtain per-floor maps. Specifically, we segment the 3D semantic point-cloud from Gibson [1] and MP3D [4] annotations into different floors. We do this by loading each scene into Habitat [9], identifying the navigable points and clustering them along the Y-coordinate to automatically discover the number of floors and their extents using DBScan [7]. We then perform orthographic projection independently for each floor of the scene.

S8. Examples of potential functions

In Fig. 3 from the main paper, we showed an example of potential functions from MP3D. We now show more examples of such potential functions for Gibson and MP3D in Figs. S4 and S5.

S9. Visualizing ObjectNav episodes

In Figure 5 from the main paper, we qualitatively visualized an episode showing how the potential functions are used to perform ObjectNav. We provide two additional examples in Fig. S6.

References

[1] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5664–5673, 2019. 3

[2] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020. 3

[3] Vincent Cartillier, Zhile Ren, Neha Jain, Stefan Lee, Irfan Essa, and Dhruv Batra. Semantic mapnet: Building allocentric semantic maps and representations from egocentric views. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 964–972, 2021. 3

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *Fifth International*

Figure S2. Examples of semantic maps from Gibson. The maps contain objects from up to 15 object categories (legend on the last row).

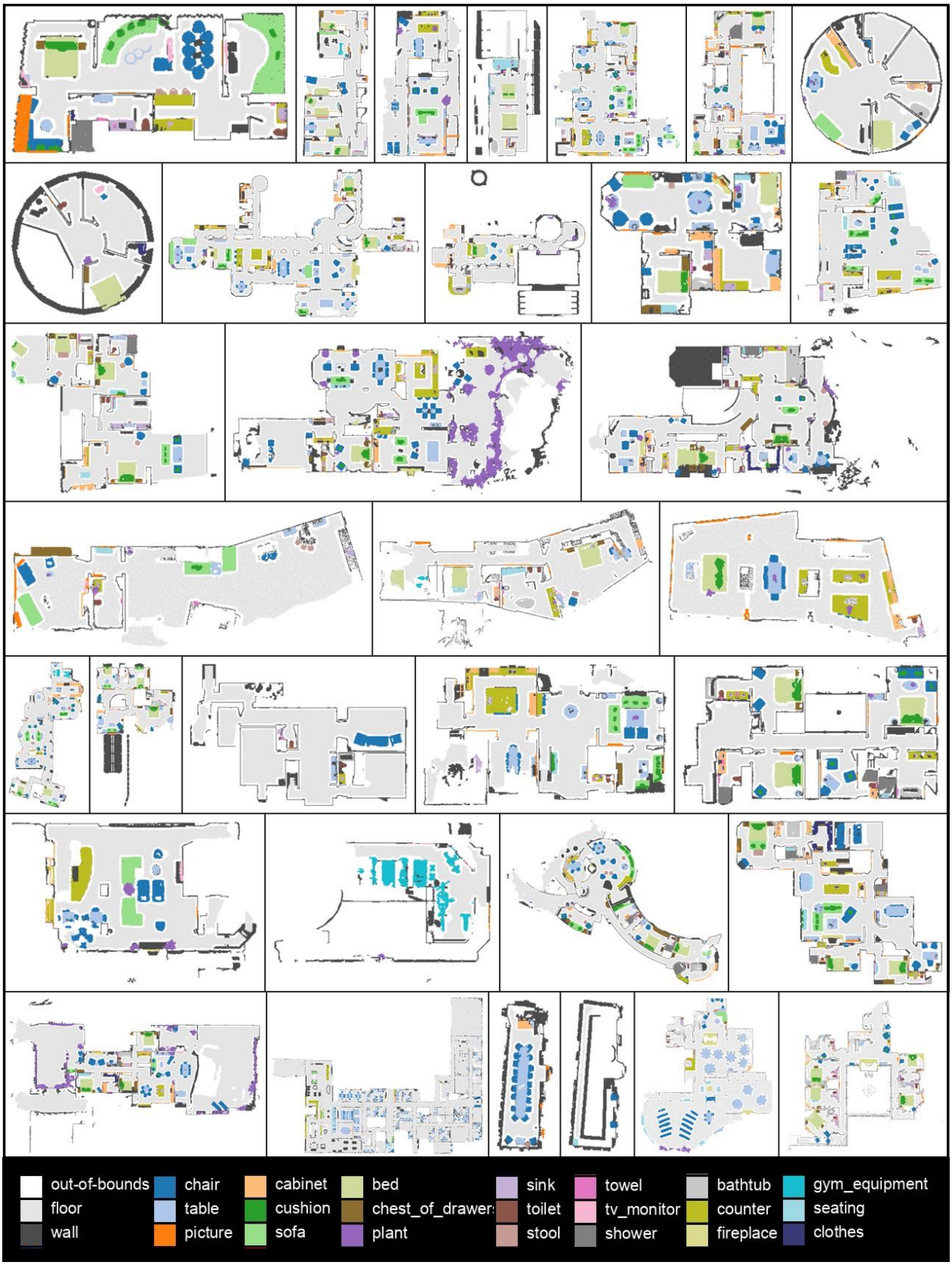

Figure S3. Examples of semantic maps from MP3D. The maps contain objects from up to 21 object categories (legend on the last row).

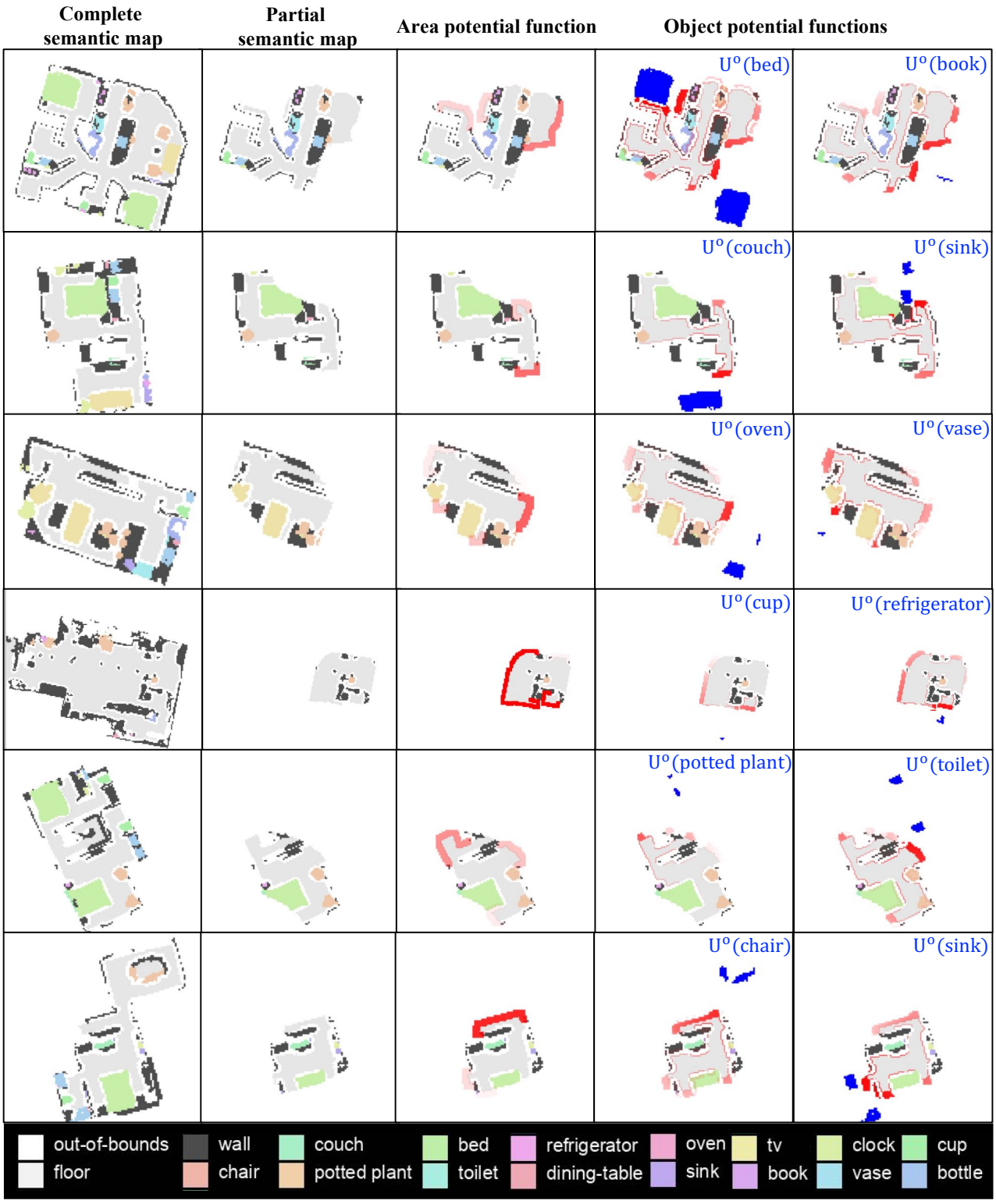

Figure S4. **Examples of potential functions from Gibson.** On each row, we show the complete semantic map, partial semantic map, the area potential function, and object potential functions for two unseen objects (from left to right). The potential functions are computed at the map frontiers using the analytical procedure described in Sec. 3.3 from the main paper. Both the potential functions range from 0.0 to 1.0, which the intensity of red indicating the strength of the potential function (1.0 is highest intensity). For the object potential function, we state the object category on the top-right corner of the map, and also highlight the spatial locations on the map in bright blue.

Figure S5. **Examples of potential functions from MP3D.** On each row, we show the complete semantic map, partial semantic map, the area potential function, and object potential functions for two unseen objects (from left to right). The potential functions are computed at the map frontiers using the analytical procedure described in Sec. 3.3 from the main paper. Both the potential functions range from 0.0 to 1.0, which the intensity of red indicating the strength of the potential function (1.0 is highest intensity). For the object potential function, we state the object category on the top-right corner of the map, and also highlight the spatial locations on the map in bright blue.

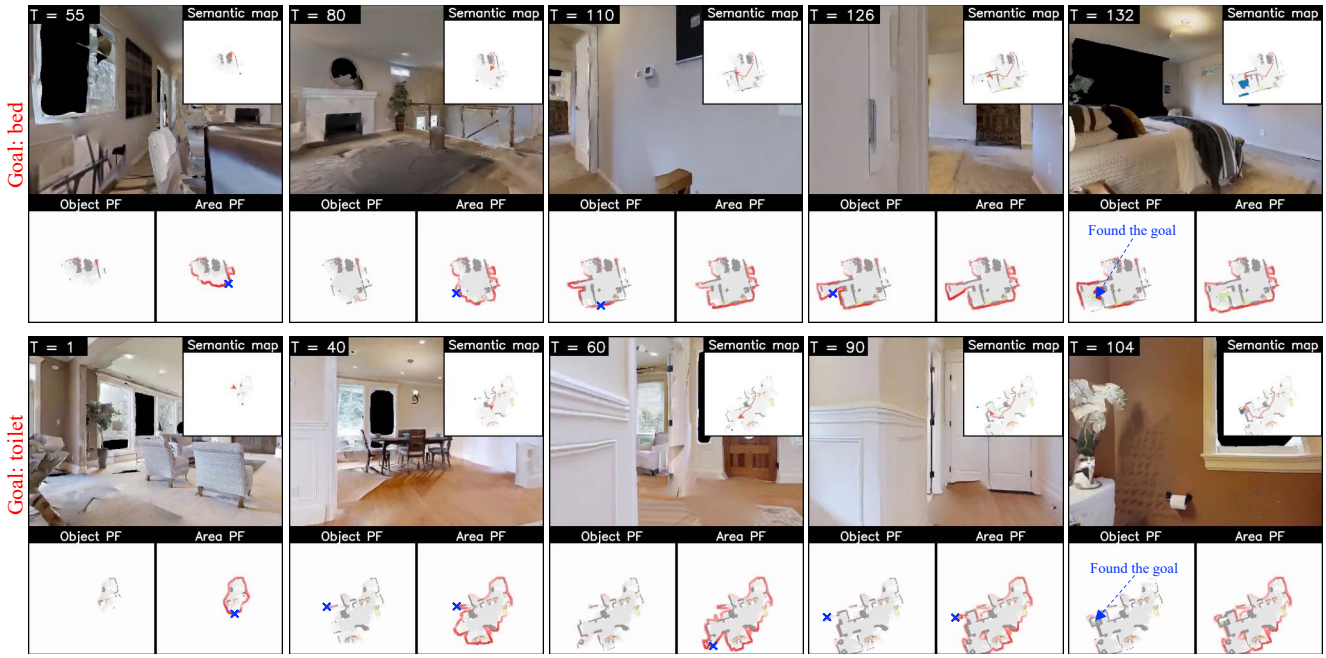

Figure S6. **Qualitative examples of navigation using potential functions**. On each row, we visualize parts of an ObjectNav episode on Gibson (val) as the agent searches for and finds the goal object. For each step, we show the egocentric RGB view, the predicted semantic map, object and area potential functions (PFs). We indicate the maximum location that the agent navigates to using a blue cross on the PF map(s) responsible for the maximum. **Row 1:** The agent searches for a bed in an unexplored scene. In the first several steps of the episode (T=1 until 110), the agent is guided by the area PF which is high near frontiers leading to unexplored areas, allowing it to explore efficiently and gather information. The object PF plays a limited role here. After having explored sufficient parts of the environment, the model predicts high object PF at two frontiers (one of which corresponds to the bedroom entrance), while the area PF remains high at multiple frontiers unrelated to the object location. Guided by the signal from the object PF, the agent starts entering the bedroom at T=126, and eventually finds the goal at T=132. **Row 2:** The agent searches for a toilet in an unexplored scene. In the initial steps of the episode (T=1 to T=60), the agent is primarily guided by the area PF to explore the scene and gather information. At T=90, the object PF activates near the toilet room entrance. Note that while the absolute value of the object PF is not very high, it is sufficient to bias the overall PF towards the goal (and away from other frontiers). This is critical since the area PF has high values along multiple frontiers, while the object PF focuses on frontiers that could lead to the object. The agent follows this signal to eventually reach the goal at T=104. These examples highlight the value of the two potential functions and how they are combined to perform ObjectNav.

Conference on 3D Vision (3DV), 2017. Matterport3D license available at http://kaldir.vc.in.tum.de/matterport/MP_TOS.pdf. 3

[5] Matthew Chang, Arjun Gupta, and Saurabh Gupta. Semantic visual navigation by watching youtube videos. *arXiv preprint arXiv:2006.10034*, 2020. 1

[6] 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, 2020. 1, 2, 3

[7] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996. 3

[8] Oleksandr Maksymets, Vincent Cartillier, Aaron Gokaslan, Erik Wijmans, Wojciech Galuba, Stefan Lee, and Dhruv Batra. Thda: Treasure hunt data augmentation for semantic navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15374–15383, 2021. 1, 2

[9] 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*, pages 9339–9347, 2019. 2, 3

[10] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *International Conference on Learning Representations*, 2019. 2

[11] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. *arXiv preprint arXiv:1810.06543*, 2018. 1

[12] Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary tasks and exploration enable objectgoal navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16117–16126, 2021. 1