Semantic MapNet: Building Allocentric Semantic Maps and Representations from Egocentric Views

Vincent Cartillier,[1] **Zhile Ren,**[1] **Neha Jain,**[1] **Stefan Lee,**[2] **Irfan Essa,**[1,4] **Dhruv Batra,**[1,3]
[1] Georgia Institute of Technology
[2] Oregon State University
[3] Facebook AI Research
[4] Google Research
{vcartillier3, zren80, nehaj, irfan, dbatra}@gatech.edu, leestef@oregonstate.edu

Abstract

We study the task of *semantic mapping* – specifically, an embodied agent (a robot or an egocentric AI assistant) is given a tour of a new environment and asked to build an allocentric top-down semantic map (*'what is where?'*) from egocentric observations of an RGB-D camera with known pose (via localization sensors). Towards this goal, we present Semantic MapNet (SMNet), which consists of: (1) an Egocentric Visual Encoder that encodes each egocentric RGB-D frame, (2) a Feature Projector that projects egocentric features to appropriate locations on a floor-plan, (3) a Spatial Memory Tensor of size floor-plan length $\times$ width $\times$ feature-dims that learns to accumulate projected egocentric features, and (4) a Map Decoder that uses the memory tensor to produce semantic top-down maps. SMNet combines the strengths of (known) projective camera geometry and neural representation learning. On the task of semantic mapping in the Matterport3D dataset, SMNet significantly outperforms competitive baselines by $4.01 - 16.81\%$ (absolute) on mean-IoU and $3.81 - 19.69\%$ (absolute) on Boundary-F1 metrics. Moreover, we show how to use the neural episodic memories and spatio-semantic allocentric representations built by SMNet for subsequent tasks in the same space – navigating to objects seen during the tour (*'Find chair'*) or answering questions about the space (*'How many chairs did you see in the house?'*). Project page: https://vincentcartillier.github.io/smnet.html.

Introduction

Imagine yourself receiving a tour of a new environment. Maybe you visit a friend's house and they show you around (*'This is our living room, and down here is the study'*). Or maybe you accompany a real-estate agent as they show you a new office space (*'These are the cubicles, and down here is the conference room'*). Or someone gives you a tour of a mall or a commercial complex. In all these situations, humans have the ability to form episodic memories and spatio-semantic representations of these spaces (O'keefe and Nadel 1978). We can recall which spaces we visited (living room, kitchen, bedroom, *etc.*), what objects were present (chairs, tables, whiteboards, *etc.*), and what their relative arrangements were (the kitchen was combined with the open plan living room, the bedroom was down the hallway, *etc.*). We can also leverage these representations to

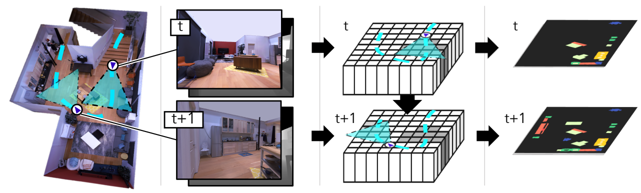

Figure 1: Semantic Mapping: (a) While moving through a 3D space (with known pose), our agent converts egocentric RGB-D observations (b) to representations in an allocentric spatial memory (c), which is used to predict top-down semantic segmentations (d) showing *'what objects are where'* from a birds-eye view.

perform new tasks in these spaces (*e.g.* navigate to the restroom via a path shorter than the one demonstrated on the tour). Of course, human memory is limited in time and in the level of metric detail it stores (Epstein et al. 2017). Our long-term goal is to develop super-human AI agents that can build rich, accurate, and reusable spatio-semantic representations from egocentric observations. This capability is an essential building block for autonomous navigation, mobile manipulation, and egocentric personal AI assistants.

In this paper, we study the specific task of creating an allocentric top-down semantic map of an indoor space, illustrated in Fig. 1. An embodied agent (a virtual robot or an egocentric AI assistant) is equipped with an RGB-D camera with known pose (extracted via localization sensors such as GPS and IMU). The agent is provided a tour of a new environment, represented as a trajectory of camera poses (shown in Fig. 1(a)). The task then is to produce an allocentric top-down semantic map (shown in Fig. 1(d)) from the sequence of egocentric observations with known pose (shown in Fig. 1(b)). Our experiments focus on top-down *semantic segmentation*, *i.e.* each pixel in the top-down map is assigned to a single class label (of the *tallest* object at that location on the floor, *i.e.* the one visible from the top-down view). Our produced semantic maps are *metric* – each pixel corresponds to a 2cm $\times$ 2cm grid on the floor – as opposed to topological maps (Fraundorfer, Engels, and Nister 2007; Nagarajan et al. 2020) that lack spatial information (scale, position) and do not support our downstream tasks of interest. Importantly, while the semantic top-down map is our

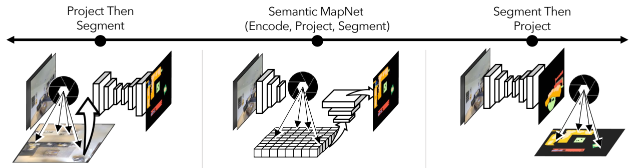

Figure 2: A spectrum of approaches to top-down semantic segmentation: (Right) perform egocentric semantic segmentation and project *labels*; (Left) Construct an overhead imagery (project *pixels*) and perform semantic segmentation; (Middle) SMNet: encode pixels, project *features*, decode labels.

primary 'product', our goal is to build neural episodic memories and spatio-semantic *representations* of 3D spaces in the process. Representations that enable the agent to easily learn and accomplish subsequent tasks in the same space – navigating to objects seen during the tour (*'Go to a chair'*) or answering questions about the space (*'How many chairs did you see in the house?'*).

What should we project? Approaches to top-down or overhead semantic segmentation can be arranged on spectrum illustrated in Fig. 2. At one end (on the right), are approaches (Sengupta et al. 2012; Sünderhauf et al. 2016; Maturana et al. 2018a) that first perform egocentric semantic segmentation and then use the known camera pose and the depth of each pixel to project *labels* to an allocentric map. In our experiments, we find that this results in 'label splatter' – any mistakes in the egocentric semantic segmentation made at the depth boundaries of objects get splattered on the map around the object. This problem can be slightly assuaged via image processing heuristics (median filtering). However, the fundamental issue persists even after those 'bells and whistles'.Quantitatively, this results in high precision but low recall of the object segmentation. At the other end of the spectrum (on left) are approaches that operate on a single overhead image (projecting *pixels* if needed) and perform semantic segmentation on this image (Singh et al. 2018; Máttyus et al. 2015). While this may be appropriate for aerial or geospatial imagery, converting multiple high-res egocentric images into a single bird's eye view is wasteful and throws out significant visual information. Qualitatively, we find that this results in coarse segmentations; object sizes are under-estimated, small objects missed entirely. Quantitatively, we see low precision and low recall.

We pursue an approach called Semantic MapNet (SMNet) that lies in the middle of this spectrum. Specifically, as shown in Fig. 2 (middle), SMNet extracts visual *features* in the egocentric reference frame, but predicts semantic segmentation *labels* in the allocentric reference frame. This is accomplished by projecting egocentric features to appropriate locations in an allocentric spatial memory, and using this memory to decode a top-down semantic segmentation. This design addresses both deficiencies in prior work – (a) the spatial-memory-to-map decoder in SMNet is based on transposed convolutions and learns to smooth out any 'feature splatter'; and (b) the egocentric feature extractor in SMNet operates directly on high-res egocentric images and is able to recognize and segment small objects that may not be visible from a bird's eye view.

We conduct experiments on the photo-realistic scans of building-scale environments (homes, offices, churches) in the Matterport3D dataset (Chang et al. 2017) using the Habitat simulation platform (Savva et al. 2019) (giving us access to agent state, navigation trajectories, RGB-D renderings, *etc*.). We choose the Matterport3D dataset because it provides semantic annotations in 3D, the spaces are large enough to allow multi-room traversal by the agent, and the use of a 3D simulator allows us to render RGB-D from any viewpoint, create top-down semantic annotations, and study embodied AI applications in the same environments. Quantitatively, on the task of semantic mapping, SMNet significantly outperforms the aforementioned baselines by $4.01 - 16.81\%$ (absolute) on mean-IoU and $3.81 - 19.69\%$ (absolute) on Boundary-F1 metrics.

SMNet combines the strengths of (known) projective camera geometry with neural representation learning, and address our key desideratum – learning rich, reusable spatio-semantic representations. We demonstrate via extension experiments how representations built by SMNet from a single tour of an environment can be reused for ObjectNav and Embodied Question Answering (Das et al. 2018).

Related Work

Spatial Episodic Memories for Embodied Agents. Building and dynamically updating a spatial memory is a powerful inductive bias that has been studied in many embodied settings. Most SLAM systems perform localization by registration to sets of localized keypoint features (Mur-Artal and Tardós 2017). Many recent works in embodied AI have developed agents for navigation (Anderson et al. 2019; Beeching et al. 2020; Gupta et al. 2017; Georgakis, Li, and Kosecka 2019; Blukis et al. 2018) and localization (Henriques and Vedaldi 2018; Parisotto and Salakhutdinov 2017; Zhang et al. 2017) that build 2.5D spatial memories containing deep features from egocentric observation. Like our approach, these all involve some variation of egocentric feature extraction, pin-hole camera projection, and map update mechanisms. However, these works focus on spatial memories as part of an end-to-end agent for a downstream task and do not evaluate the quality of the generated maps in terms of environment semantics directly, nor study how segmentation quality affects downstream tasks.

Semantic Mapping from Egocentric Observations. Predicting top-down semantic segmentation from egocentric observations has been studied in the context of robotics as the semantic SLAM (or semantic mapping) problem (Rosinol et al. 2019; Maturana et al. 2018b; Grinvald et al. 2019; McCormac et al. 2017). We compare with a recent representative algorithm in this family as our baseline (Grinvald et al. 2019). Further work has examined the use of semantic labels as an intermediate step in an end-to-end model (Gordon et al. 2018; Chaplot et al. 2020a) or to derive supervision to reward agent trajectories (Chaplot et al. 2020c). These works have not evaluated the quality of the semantic map and instead focused on downstream tasks. All these works follow the Segment-then-Project paradigm – invoking a segmenta-

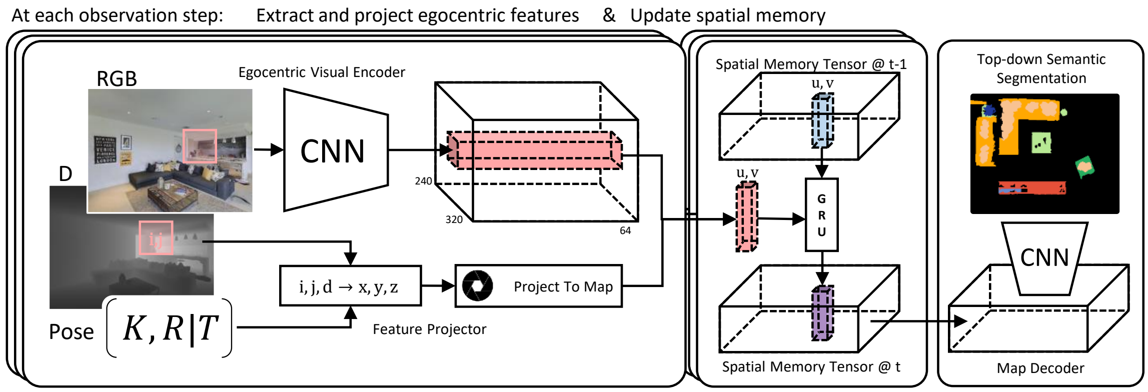

Figure 3: At each step in a trajectory, SMNet updates an allocentric map based on egocentric observations. Egocentric RGB-D observations are represented using a CNN encoder and the feature vectors are projected to the spatial memory (left). Memory cells are updated by a GRU to incorporate this new information (middle). The spatial memory can then be decoded by to perform top-down semantic segmentation (right).

tion network on the 2D observations and then projecting labels into an allocentric map. In contrast, our findings suggest it is more effective to project intermediate features and allow an allocentric decoder to produce the final segmentation.

Closely related to our approach is a line of work focusing on volumetric recurrent memory architectures for 3D semantic segmentation of small objects (Cheng, Wang, and Fragkiadaki 2018; Tung, Cheng, and Fragkiadaki 2019). Like Semantic MapNet, these approaches project intermediate features into a spatial memory and then decode segmentations from that structure. However, these works focus on relatively small objects due to the large memory constraints of 3D volumetric memory. For example, a 25m $\times$ 20m footprint indoor environment with standard ceiling height (2.75m) would require storing 171.875 million feature vectors at 2cm^3 resolution – or a total of 176 gigabytes if features are 256 dimensional as in our experiments. Semantic MapNet is designed to work on the large environments (average footprint of 24.5m $\times$ 23.4m) of the Matterport3D dataset, and achieves state of the art results.

Recently, Pan et al. examined cross-view semantic segmentation – i.e. the task of predicting a local top-down semantic map from a single first-person observation (Pan et al. 2020). Unlike SMNet, their proposed approach does not include projective geometry – instead learning a small network to transform first-person views to top-down feature maps – nor does it accumulate observations over a trajectory. In contrast, we addresses the problem of building a global top-down semantic map based on a trajectory.

Simulation Platforms and Embodied Vision Tasks. The creation of large 3D datasets (Chang et al. 2017; Armeni et al. 2016) and simulators (Savva et al. 2019; Anderson et al. 2018; Kolve et al. 2017; Xia et al. 2018) has spurred development in Embodied AI. Recent work examines interactive agents navigating in the environment to answer questions (Das et al. 2018; Wijmans et al. 2019; Gordon et al. 2018), reach desired locations (Wijmans et al. 2020), and infer the shape of occluded objects (Yang et al. 2019). We study a fundamental building block for these tasks – building top-down semantic maps of indoor environments.

Semantic MapNet (SMNet)

We now describe our proposed approach for semantic mapping, called Semantic MapNet (SMNet), in detail. As shown in Fig. 3, SMNet consists of the following modules:

- An **Egocentric Visual Encoder** that converts each egocentric RGB-D frame into a $\mathbb{R}^{w\times h\times d}$ feature tensor, representing the content of each image region.
- A **Feature Projector** that uses the known camera pose and the depth of each pixel to project these egocentric features at appropriate locations on a floor-plan,
- A **Spatial Memory Tensor** of size floor-plan length $\times$ width $\times$ feature-dims that accumulates these projected egocentric features. Repeated observations of the same spatial locations are incorporated through a learned recurrent model operating at each location.
- A **Map Decoder** that uses the accumulated memory tensor to produce top-down semantic segmentations.

Problem Setup and Notation. Let I denote an RGB-D image. We assume a known camera – specifically, let K be the camera intrinsic matrix, and $[R \mid \mathbf{t}]$ denote the camera extrinsic matrix (rotation and translation needed to convert world coordinates to camera coordinates). Thus, an agent's trajectory through an environment is represented as a sequence of egocentric RGB-D observations $I^{(1)}, \ldots, I^{(T)}$ at known poses $[R \mid \mathbf{t}]^{(1)}, \ldots, [R \mid \mathbf{t}]^{(T)}$. Strictly speaking, our approach does not require knowing camera pose in world coordinates at all times – all we need are successive pose transformations $[R \mid \mathbf{t}]^{(t\to t+1)}$, a problem known in robotics and computer vision as egomotion estimation. The entire approach could be defined in terms of the camera coordinates at time $t = 1$. However, for sake of clarity of the exposition, we describe our approach with global pose.

Let S denote the top-down semantic segmentation. Each pixel in S represents a 2cm$\times$2cm cell in the environment and is labeled with the class of the tallest object in that cell (*i.e.* the object visible from above). At each time t, let $M^{(t)}$

denote the memory tensor incorporating all the information observed in the trajectory so far, and let $\hat{S}^{(t)}$ denote the segmentation predicted using $M^{(t)}$. Note that test-time evaluation is done using $\hat{S}^{(T)}$, but during training our agent predicts and receives supervision for intermediate predictions along the trajectory $\hat{S}^{(1)}, \ldots, \hat{S}^{(T)}$.

There are a number of coordinate systems in this discussion which we define now for clarity – pixel positions in the egocentric RGB-D image I are indexed with i, j, and the depth at this pixel is denoted with $d_{i,j}$ (or d when its clear from context which pixel is being talked about). A 3D point in world coordinates is denoted with x, y, z. For notational simplicity, we follow the standard convention in computer graphics – negative Y-axis aligned gravity in the world coordinate system. Finally, cells in the memory tensor are indexed with u, v. Next, we describe each module in detail.

Egocentric Visual Encoder. Each egocentric frame $I^{(t)}$ gives a local glimpse of the environment – providing information about objects and their locations in the current view. To represent these, we encode each RGB-D frame using RedNet (Jiang et al. 2018), a recently proposed architecture for semantic segmentation of indoor scenes. In principle, one may choose any standard image encoder network for semantic segmentation such as Mask-RCNN (He et al. 2017). We chose RedNet simply because the network structure has proven to be effective for parsing indoor environments and pre-trained models (learned on SUN-RGBD dataset (Song, Lichtenberg, and Xiao 2015)) are publicly available. We initialize with these pre-trained weights and fine-tune RedNet on our dataset. We conducted several experiments by extracting egocentric features at different stages in the RedNet network (encoder, last layer, scores, softmax, one-hot encoded labels). We found that encoding each RGB-D frame with the last layer RedNet features yields to the best performances. The output of this encoder for image $I^{(t)}$ is an egocentric feature map $F^{(t)} \in \mathbb{R}^{240\times320\times64}$ with each of the 240×320 cells storing a 64-d feature. We upscale this tensor to the resolution of the depth image (480×640) with bilinear interpolation, resulting in each pixel i, j having an associated feature $F^{(t)}_{i,j} \in \mathbb{R}^{64}$ and depth value $d_{i,j}$.

Feature Projector. To project an egocentric feature $F^{(t)}_{i,j}$ to the spatial memory, we must (a) shoot a ray from the camera center through the image pixel (i, j) out to a depth $d_{i,j}$ to get a 3D point in the camera coordinate system, (b) convert from camera to world coordinates to get the corresponding (x, y, z), and then (c) project it to cell indices u, v in the memory tensor. With known camera pose and intrinsics, these transformations for the standard pinhole camera can be written compactly as:

$$\underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix} = d_{i,j} R^{-1} K^{-1} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} - \mathbf{t},}_{\text{(Inverse) Pinhole Camera Projection}} \quad \text{and} \quad \underbrace{\begin{bmatrix} u \\ v \\ 0 \\ 1 \end{bmatrix} = P_v \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}_{\text{Orthographic Projection}} \tag{1}$$

where P_v is a known orthographic projection matrix converting 3D world coordinates to 2D memory cell indices. When several points are projected to the same index in $M^{(t)}$, we retain the one with the maximum height in the world coordinates. This results in a set of projected features $F^{(t)}_{u,v}$.

Spatial Memory Tensor M is a 3D tensor of size $U \times V \times 256$. Each grid cell (u, v) stores a 256-d feature vector and corresponds to a 2cm×2cm area on the floor-plan, which is the same spatial resolution as the segmentation S. The memory must be updated at each time step to incorporate new observations. Specifically, given the current memory $M^{(t-1)}$ and a new observation $F^{(t)}_{u,v}$ for cell u, v, we compute $M^{(t)}_{u,v} = GRU(F^{(t)}_{u,v}, M^{(t-1)}_{u,v})$ where $M^{(t-1)}_{u,v}$ is the hidden state and $F^{(t)}_{u,v}$ the input for the GRU. This GRU can learn to accumulate incoming observations. Notice that the GRU parameters are shared spatially, *i.e.* for all (u, v). Importantly, this independent updating of modified memory cells (as opposed to something like a ConvGRU) ensures that observations only affect local regions of the memory – keeping previously observed areas stable.

Map Decoder. The memory tensor $M^{(t)}$ is used to decode a top-down semantic segmentation map. We use a simple architecture consisting of five convolutional layers with batch norm and ReLU activations. As the memory M and segmentation S are the same spatial resolution, no learned upsampling or downsampling is involved in this decoding.

Together, these modules form SMNet and implement the basic principle of 'encode pixels, project features to a spatial memory, decode labels'. Notice that all modules and thus the entire architecture is end-to-end differentiable.

Matterport Semantic-Map Dataset

For our experimental evaluation, we need 3D environments for an agent to traverse that have dense semantic segmentations. While our extension experiments involve agent-driven navigation, our core task of semantic mapping is defined w.r.t. a fixed trajectory provided to the agent as input. The more challenging task of simultaneous semantic mapping and goal-driven navigation is left for future work.

Given this fixed-trajectory setting, our task has the input (but not output) specification of video segmentation – both taking a sequence of input images along a trajectory. We choose the Matterport3D scans (Chang et al. 2017) with the Habitat simulator (Savva et al. 2019) over video segmentation datasets for a number of reasons – Matterport3D provides semantic annotations in 3D (as opposed to 2D annotations in video datasets), the spaces are large enough to allow multi-room traversal by the agent (as opposed to (Dai et al. 2017; Nathan Silberman and Fergus 2012)), and the use of a 3D simulator (as opposed to (Geiger, Lenz, and Urtasun 2012; Cordts et al. 2016)) allows us render RGB-D from any viewpoint, create top-down semantic annotations, and study embodied AI applications in the same environments.

Matterport3D Environments. Matterport3D dataset (Chang et al. 2017) contains reconstructed 3D meshes of 90 indoor environments (homes, offices, churches). These meshes are densely annotated with 40 object categories. Many of these are rare or would appear

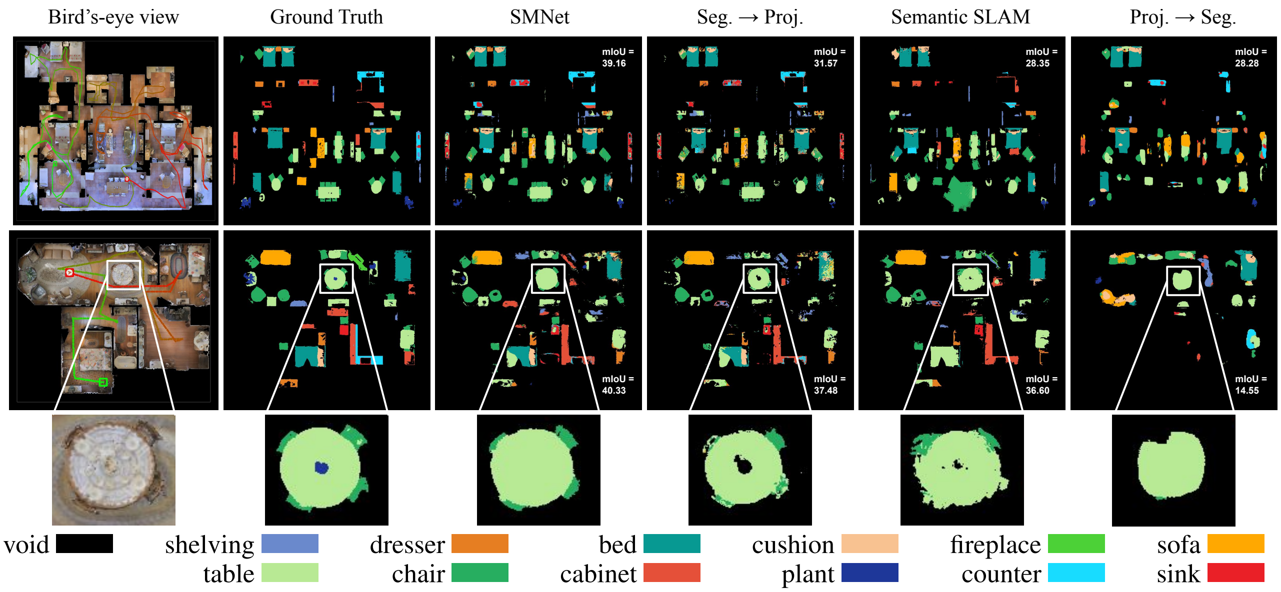

Figure 4: Example semantic segmentation predictions. SMNet makes cleaner and more accurate predictions than the baseline approaches.

as thin lines in a top-down view (*e.g.* walls and curtains); we focus on the 12 most common object categories: `chair`, `table`, `cushion`, `cabinet`, `shelving`, `sink`, `dresser`, `plant`, `bed`, `sofa`, `counter`, `fireplace` (sorted in descending order by number of object instances). We treat all other classes and background pixels as `void` class. We divide multi-story environments in Matterport3D into separate floors by manually refining floor dividers present in the meta-data. This is not always possible given a single dividing plane (*e.g.* split level homes), resulting in inaccurate top-down maps – we discard such environments. Utilizing the same data split as (Wijmans et al. 2019), we keep 85 unique floors in our dataset: 61 for training, 7 for validation, and 17 for testing. See supplement (Cartillier et al. 2020) for these splits.

Ground-Truth Top-down Semantic Segmentations. To supervise our model, we need access to ground-truth top-down semantic maps from these environments. These are created by applying an orthographic projection for the 3D mesh annotations in a similar manner to Eqn. 1 (right). In this process we only project vertices labeled with one of the 12 kept object categories. The resulting ground-truth top-down semantic maps are free from occlusions caused by non-target objects. There will be cases where from the egocentric view the agent won't be able to visualize the object entirely either because it is occluded or the object is too high (wall cabinet). In table 1 we report numbers on the Seg. GT $\rightarrow$ Proj. experiment where the agent projects egocentric ground-truth semantic labels. This will account for such occlusion and set an upper-bound to our experiments.

Modal Maps and Viewing Frustum. We perform *modal* top-down semantic segmentation (as opposed to *amodal*). Specifically, the agent receives supervision on map cells it has *actually observed*; it is not evaluated on hallucinating unseen regions. We do this by projecting the viewing frustum (*i.e.* region the agent can currently see) to the floor-plan at each navigation step. We can then keep track of which regions have been observed during a trajectory.

Navigation Paths. We assume that an agent's path through the environment is provided by some external policy – *e.g.* a goal-oriented path or a general exploration policy – and that we are constructing the map and memory opportunistically from this experience. To simulate this behavior, we manually record a navigation path through each floor using the Habitat simulator (Savva et al. 2019). The action space is move forward 10cm, and rotate left or right 9° . To encourage trajectories with high environment coverage, our human navigation interface included the top-down RGB map with agent position drawn. On average, agents move 2500 steps in each environment. Note that this is an order of magnitude longer than most navigation trajectories in contemporary works (Savva et al. 2019; Wijmans et al. 2020; Kadian et al. 2019; Gordon et al. 2018; Chaplot et al. 2020b)

Training Samples. To train our model, we consider 20-step navigation segments from these trajectories. Starting from a random location on the trajectory, we step the agent forward 20 steps along it, capturing the corresponding viewpoints to mask the top-down semantic map. We generate 50 examples for each environment leading to 3050/350 train/val training samples. This both greatly increases the number of training instances and increases training speed by requiring a smaller semantic memory tensor. At evaluation/testing time, the agent builds the map from the entire trajectory.

Semantic Mapping Experiments

Baselines. As depicted in Fig. 2, there exists a spectrum of methodologies for our task based on what is being projected from egocentric observations to the top-down map – pixels, features, or labels. Our approach stakes a middle-ground on

	Matteport3D (test)					Replica				
	Acc	mRecall	mPrecision	mIoU	mBF1	Acc	mRecall	mPrecision	mIoU	mBF1
Seg. GT → Proj.	89.49 ± 0.09	73.73 ± 0.06	74.58 ± 0.10	59.73 ± 0.09	54.05 ± 0.11	96.83 ± 0.07	83.84 ± 0.05	94.05 ± 0.06	79.76 ± 0.07	86.89 ± 0.04
Proj. → Seg.	83.18 ± 0.07	27.32 ± 0.08	35.30 ± 0.13	19.96 ± 0.07	17.33 ± 0.08	81.25 ± 0.09	26.64 ± 0.12	41.50 ± 0.12	20.06 ± 0.09	19.08 ± 0.12
Seg. → Proj.	88.06 ± 0.07	40.53 ± 0.09	**58.92 ± 0.11**	32.76 ± 0.07	33.21 ± 0.08	88.61 ± 0.09	48.11 ± 0.09	**65.20 ± 0.11**	40.77 ± 0.09	45.86 ± 0.12
Semantic SLAM	85.17 ± 0.08	37.51 ± 0.09	51.54 ± 0.15	28.11 ± 0.08	31.05 ± 0.12	88.30 ± 0.09	45.80 ± 0.08	62.41 ± 0.12	37.99 ± 0.09	**46.71 ± 0.10**
SMNet	**88.14 ± 0.09**	**47.49 ± 0.11**	58.27 ± 0.11	**36.77 ± 0.09**	**37.02 ± 0.09**	**89.26 ± 0.10**	**53.37 ± 0.12**	64.81 ± 0.09	**43.12 ± 0.10**	45.18 ± 0.14

Table 1: Results on top-down semantic segmentation on the Matterport3D and Replica datasets. Models have not been trained on Replica and those results are purely transfer experiments. SMNet outperforms the baselines on mIoU and mBF1 for Matterport3D and mIoU in Replica.

this spectrum – projecting egocentric features. We compare with approaches at either end and existing work:

- **Project → Segment.** As agents traverse the scene, the observed RGB pixels are projected to the top-down map using our mapper architecture – resulting in a top-down RGB image of the environment. We train a model similar to RedNet (Jiang et al. 2018) to decode the semantic map directly from this top-down RGB image.
- **Segment → Project.** At the other extreme, agents perform semantic segmentation on each egocentric frame and then project the resulting labels using our mapper architecture to create the top-down segmentation. The produced top-down semantic maps are post-processed using a median filter (3×3) to reduce the label splatter noise caused by egocentric prediction errors around object boundaries. In addition, we found experimentally that down-sampling the egocentric resolution from (480×640) to (120×160) helps reducing the impact of egocentric errors on the top-down maps and leads to best performances for this baseline. Any missed pixels in the observed area of the top-down semantic map caused by this down-sampling is filled using median filtering. We finetune a RedNet (Jiang et al. 2018) model for this task. We also present an oracle baseline that projects ground-truth segmentations – **Segment GT → Project.** This experiment sets an upper-bound of performances (perfect predictions being not possible due to occlusions)
- **Semantic SLAM.** We use VoxBlox++, an off-the-shelf implementation of semantic SLAM (Grinvald et al. 2019) where we replaced the object detection module with our pre-trained RedNet model plus a hand-crafted instance segmentation applied on-top of the semantic predictions (connected components). The algorithm takes as input RGB-D frames and simultaneously estimates agent's pose and constructs a point cloud of semantically labelled points. We project the point cloud to a top-down segmentation using the same mapping functions we described in Sec. . For fairness, we provide the ground truth pose to (Grinvald et al. 2019) at each time step.

In all experiments, detections 50cm over the agent's camera position are discarded. This prevents detecting ceilings.

Implementation Details We pretrain two RedNet (Jiang et al. 2018) models for semantic segmentation in our setting – one from egocentric RGB-D (Segment→Project) and another from top-down RGB alone (Project→Segment). SMNet is initialized with the encoder from the egocentric RedNet. We use a single-layer GRU to update the spatial memory. SMNet is trained end-to-end under cross-entropy loss using SGD with learning rate $1e{-4}$, momentum 0.9, weight decay $4e{-4}$, and batch size 8 across 8 Titan XPs. Training took 2-3 days. Back propagation is applied after 20 steps.

Evaluation Metrics. We report the entire range of evaluation metrics for semantic segmentation: (a) the overall pixel-wise labeling accuracy (**Acc**), (b) the average of pixel recall or precision scores for each class (**mRecall/mPrecision**), (c) the average of the intersection-over-union score of all object categories (**mIoU**), and (d) the average of the boundary F1 score of all object categories (**mBF1**). mBF1 is contour-based metric defined in (Csurka et al. 2013). mIoU and mBF1 serve as our primary metrics.

Results. Table 1(left) shows a summary of the results with bootstrapped standard error (see supplement (Cartillier et al. 2020) for category-level breakdowns). Fig. 4 shows qualitative results. Project→Segment achieves low performance (mBF1 17.33, mIoU 19.96) compared to the approaches that operate on egocentric images prior to projection (either via segmentation or feature extraction). This suggests details lost in the top-down view are important for disambiguating objects – *e.g.* the chairs at the table in Fig. 4 (bottom) are difficult to see in the top-down RGB and are completely lost by this approach. Segment→Project performs significantly better (mBF1 33.21, mIoU 32.76), but faces a problem with errors in the egocentric predictions resulting in noise in the top-down map. Semantic SLAM (VoxBlobx++) performs worse than the Segment→Project baseline (mBF1 31.05, mIoU 28.11). VoxBlox++ follows a segment-then-project paradigm, making it prone to the same errors as the Segment → Project baseline. In addition, the data association module of VoxBlox++ will sometimes group objects of different categories (*e.g.* the two bottom chairs are grouped with the table in Fig. 4). As our approach reasons over a spatial memory tensor, it can reason about multiple observations of the same point – achieving mBF1 37.02, and mIoU 36.77. The Segment GT →Project experiment sets an upper-bound of mBF1 54.05, and mIoU 59.73. We also evaluated SMNet on the replica dataset (Straub et al. 2019). Table 1 (right) shows a summary of the results with bootstrapped standard error. Similarly, SMNet performs best on the mIoU metric at 43.12. These results demonstrate that an approach which interleaves projective geometry and learning can provide more robust allocentric semantic representations.

Re-using Maps for Downstream Tasks

The map and spatio-semantic allocentric representation our method constructs while exploring an environment provide

Figure 5: Object Navigation: Visualization of paths found by A* using SMNet maps. Green and red squares indicate agent's starting and stopping locations. The grey color represents the floor pixels. The left example shows a case of success with high SPL $= 0.8276$ and the example on the right shows a case of success with low SPL$= 0.4282$.

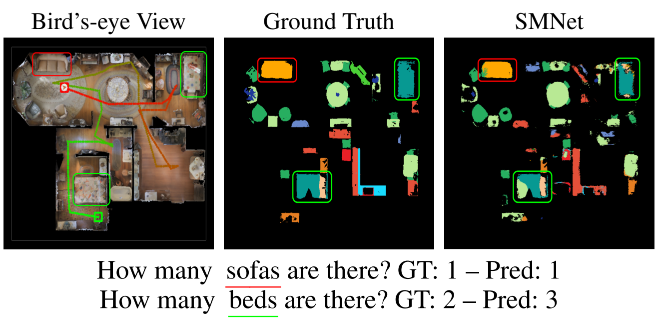

Figure 6: Visualizations of MemoryQA

a rich description of the space. In this section, we explore proof-of-concepts for various downstream embodied AI tasks based on these representations.

Object Navigation. A natural extension is navigating to specific objects, or ObjectNav for short. In ObjectNav, an agent is randomly initialized in a scene and tasked to navigate to an instance of a given object class as quickly as possible (Savva et al. 2019). In the standard setting, the environment is novel; however, we consider a pre-exploration setting where the agent first traverses the environment to construct a top-down semantic map. In parallel we compute a top-down map of heights and use it to compute a free space map of the environment. We opt for an open loop planing strategy by running A* search (with a Euclidean heuristic) on the free space map combined with the semantic map to find a path from the start location to the nearest target object instance and then run the trajectory in the Habitat simulator (Savva et al. 2019). We evaluated this strategy on the validation set of the ObjectNav Habitat challenge (hab 2020), agents are able to achieve a success rate of 9.658%, with SPL of 5.714%, soft SPL of 8.702% and average distance to the target of $7.31576m$. Note that 26% of the episodes in this set are targeting object categories falling outside of our list of object classes, we consider those as failure. The evaluation metrics limited to episodes targeting objects in our list of classes are: success rate of 13.070%, with SPL of 7.733%, soft SPL of 11.777% and average distance to the target of $6.70981m$. These results are in the same order of magnitude as the state-of-the-art methods submitted to the Habitat Challenge (hab 2020), suggesting that the memory tensor contains useful spatial and semantic information in this pre-exploration setting. Experimentally we found that inaccuracies in the free space map computation and objects misclassification in the top-down semantic map are the two major sources of error. While the latter is harder to cope with, the former can be limited by extended SMNet to predict free space. Fig. 5 shows qualitative results of two successful examples – start locations are shown as green squares with trajectory transitioning to red until terminating. Using the predicted semantic maps provides interpretability – when the navigation fails, we can know why. On the example on the right in Fig. 5 the chair at the top has been mislabeled as sofa, thus leading the agent to the second closest chair slightly on the left.

Question Answering. We also consider an embodied question answering (Das et al. 2018) task where agents are asked questions about the environment. Again considering a pre-exploration setting, the agent first navigates the environment on a fixed trajectory to generate the spatial memory tensor. We consider counting questions (*e.g. 'How many beds are there?'*) and design a decoder directly from the spatial memory. The decoder outputs the number of instances detected per object category for a given memory input. We train this decoder using 5m x 5m memory samples. We design this task as a classification problem with 21 classes corresponding to values ranging from 0 to 19 and 20+. When testing on larger environments, we apply this decoder using a sliding window over the full memory – accumulating counts. We compare our approach to a 'prior' baseline that answers with the most frequent answer in the training set. Our approach outperforms this baseline across the board: 27.78% *vs*. 20.83% on accuracy, 13.19% *vs*. 9.18% class-balanced accuracy and 5.35 *vs*. 6.98 on RMSE.

Conclusion

Taken holistically, our results show SMNet is able to outperform competitive baselines in constructing semantic maps, and spatio-semantic representations built show promise on downstream tasks. Note that the specific sub-task of counting instances highlights a limitation in our current problem setup – using semantic segmentation does not preserve instance information. The generalization to producing top-down instance segmentation maps is an interesting avenue for future work.

References

2020. Habitat Challenge 2020 @ Embodied AI Workshop. CVPR 2020. https://aihabitat.org/challenge/2020/.

Anderson, P.; Shrivastava, A.; Parikh, D.; Batra, D.; and Lee, S. 2019. Chasing Ghosts: Instruction Following as Bayesian State Tracking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 369–379.

Anderson, P.; Wu, Q.; Teney, D.; Bruce, J.; Johnson, M.; Sünderhauf, N.; Reid, I.; Gould, S.; and van den Hengel, A. 2018. Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H.; Brilakis, I.; Fischer, M.; and Savarese, S. 2016. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1534–1543.

Beeching, E.; Dibangoye, J.; Simonin, O.; and Wolf, C. 2020. EgoMap: Projective mapping and structured egocentric memory for Deep RL. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*.

Blukis, V.; Misra, D.; Knepper, R. A.; and Artzi, Y. 2018. Mapping Navigation Instructions to Continuous Control Actions with Position-Visitation Prediction. In *Conference on Robot Learning*, 505–518.

Cartillier, V.; Ren, Z.; Jain, N.; Lee, S.; Essa, I.; and Batra, D. 2020. Semantic MapNet: Building Allocentric SemanticMaps and Representations from Egocentric Views. *arXiv preprint arXiv:2010.01191* .

Chang, A.; Dai, A.; Funkhouser, T.; Halber, M.; Niessner, M.; Savva, M.; Song, S.; Zeng, A.; and Zhang, Y. 2017. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)* MatterPort3D dataset license available at: http://kaldir.vc.in.tum.de/matterport/MP_TOS.pdf.

Chaplot, D. S.; Gandhi, D.; Gupta, A.; and Salakhutdinov, R. 2020a. Object Goal Navigation using Goal-Oriented Semantic Exploration. *arXiv preprint arXiv:2007.00643* .

Chaplot, D. S.; Gandhi, D.; Gupta, S.; Gupta, A.; and Salakhutdinov, R. 2020b. Learning To Explore Using Active Neural SLAM. In *International Conference on Learning Representations (ICLR)*.

Chaplot, D. S.; Jiang, H.; Gupta, S.; and Gupta, A. 2020c. Semantic Curiosity for Active Visual Learning. In *ECCV*.

Cheng, R.; Wang, Z.; and Fragkiadaki, K. 2018. Geometry-aware recurrent neural networks for active visual recognition. In *Advances in Neural Information Processing Systems*, 5081–5091.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Csurka, G.; Larlus, D.; Perronnin, F.; and Meylan, F. 2013. What is a good evaluation measure for semantic segmentation?. In *BMVC*, volume 27, 2013.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Das, A.; Datta, S.; Gkioxari, G.; Lee, S.; Parikh, D.; and Batra, D. 2018. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2054–2063.

Epstein, R. A.; Patai, E. Z.; Julian, J. B.; and Spiers, H. J. 2017. The cognitive map in humans: spatial navigation and beyond. *Nature Neuroscience* 20(11): 1504–1513. doi:10.1038/nn.4656. URL https://doi.org/10.1038/nn.4656.

Fraundorfer, F.; Engels, C.; and Nister, D. 2007. Topological mapping, localization and navigation using image collections. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Georgakis, G.; Li, Y.; and Kosecka, J. 2019. Simultaneous Mapping and Target Driven Navigation. *arXiv preprint arXiv:1911.07980* .

Gordon, D.; Kembhavi, A.; Rastegari, M.; Redmon, J.; Fox, D.; and Farhadi, A. 2018. IQA: Visual question answering in interactive environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4089–4098.

Grinvald, M.; Furrer, F.; Novkovic, T.; Chung, J. J.; Cadena, C.; Siegwart, R.; and Nieto, J. 2019. Volumetric instance-aware semantic mapping and 3D object discovery. *IEEE Robotics and Automation Letters* 4(3): 3037–3044.

Gupta, S.; Davidson, J.; Levine, S.; Sukthankar, R.; and Malik, J. 2017. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2616–2625.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask R-CNN. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2961–2969.

Henriques, J. F.; and Vedaldi, A. 2018. Mapnet: An allocentric spatial memory for mapping environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8476–8484.

Jiang, J.; Zheng, L.; Luo, F.; and Zhang, Z. 2018. Rednet: Residual encoder-decoder network for indoor rgb-d semantic segmentation. *arXiv preprint arXiv:1806.01054* .

Kadian, A.; Truong, J.; Gokaslan, A.; Clegg, A.; Wijmans, E.; Lee, S.; Savva, M.; Chernova, S.; and Batra, D. 2019. Are We Making Real Progress in Simulated Environments? Measuring the Sim2Real Gap in Embodied Visual Navigation. *arXiv preprint arXiv:1912.06321* .

Kolve, E.; Mottaghi, R.; Han, W.; VanderBilt, E.; Weihs, L.; Herrasti, A.; Gordon, D.; Zhu, Y.; Gupta, A.; and Farhadi, A. 2017. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474* .

Maturana, D.; Chou, P.-W.; Uenoyama, M.; and Scherer, S. 2018a. Real-Time Semantic Mapping for Autonomous Off-Road Navigation. In Hutter, M.; and Siegwart, R., eds., *Field and Service Robotics*, 335–350. Springer International Publishing. ISBN 978-3-319-67361-5.

Maturana, D.; Chou, P.-W.; Uenoyama, M.; and Scherer, S. 2018b. Real-time semantic mapping for autonomous off-road navigation. In *Field and Service Robotics*, 335–350. Springer.

McCormac, J.; Handa, A.; Davison, A.; and Leutenegger, S. 2017. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and automation (ICRA)*, 4628–4635. IEEE.

Mur-Artal, R.; and Tardós, J. D. 2017. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics* 33(5): 1255–1262.

Máttyus, G.; Wang, S.; Fidler, S.; and Urtasun, R. 2015. Enhancing Road Maps by Parsing Aerial Images Around the World. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*.

Nagarajan, T.; Li, Y.; Feichtenhofer, C.; and Grauman, K. 2020. EGO-TOPO: Environment Affordances from Egocentric Video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nathan Silberman, Derek Hoiem, P. K.; and Fergus, R. 2012. Indoor Segmentation and Support Inference from RGBD Images. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

O'keefe, J.; and Nadel, L. 1978. *The hippocampus as a cognitive map*. Oxford: Clarendon Press.

Pan, B.; Sun, J.; Leung, H. Y. T.; Andonian, A.; and Zhou, B. 2020. Cross-view semantic segmentation for sensing surroundings. *IEEE Robotics and Automation Letters* 5(3): 4867–4873.

Parisotto, E.; and Salakhutdinov, R. 2017. Neural map: Structured memory for deep reinforcement learning. *arXiv preprint arXiv:1702.08360* .

Rosinol, A.; Abate, M.; Chang, Y.; and Carlone, L. 2019. Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping. *arXiv preprint arXiv:1910.02490* .

Savva, M.; Kadian, A.; Maksymets, O.; Zhao, Y.; Wijmans, E.; Jain, B.; Straub, J.; Liu, J.; Koltun, V.; Malik, J.; Parikh, D.; and Batra, D. 2019. Habitat: A Platform for Embodied AI Research. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*.

Sengupta, S.; Sturgess, P.; Ladický, L.; and Torr, P. H. S. 2012. Automatic dense visual semantic mapping from street-level imagery. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Singh, S.; Batra, A.; Pang, G.; Torresani, L.; Basu, S.; Paluri, M.; and Jawahar, C. V. 2018. Self-supervised Feature Learning for Semantic Segmentation of Overhead Imagery. In *Proceedings of the British Machine Vision Conference (BMVC)*.

Song, S.; Lichtenberg, S. P.; and Xiao, J. 2015. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 567–576.

Straub, J.; Whelan, T.; Ma, L.; Chen, Y.; Wijmans, E.; Green, S.; Engel, J. J.; Mur-Artal, R.; Ren, C.; Verma, S.; et al. 2019. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797* .

Sünderhauf, N.; Dayoub, F.; McMahon, S.; Talbot, B.; Schulz, R.; Corke, P.; Wyeth, G.; Upcroft, B.; and Milford, M. 2016. Place categorization and semantic mapping on a mobile robot. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Tung, H.-Y. F.; Cheng, R.; and Fragkiadaki, K. 2019. Learning spatial common sense with geometry-aware recurrent networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2595–2603.

Wijmans, E.; Datta, S.; Maksymets, O.; Das, A.; Gkioxari, G.; Lee, S.; Essa, I.; Parikh, D.; and Batra, D. 2019. Embodied question answering in photorealistic environments with point cloud perception. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6659–6668.

Wijmans, E.; Kadian, A.; Morcos, A.; Lee, S.; Essa, I.; Parikh, D.; Savva, M.; and Batra, D. 2020. Decentralized Distributed PPO: Solving PointGoal Navigation. *International Conference on Lefvarning Representations (ICLR)* .

Xia, F.; R. Zamir, A.; He, Z.-Y.; Sax, A.; Malik, J.; and Savarese, S. 2018. Gibson env: real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Yang, J.; Ren, Z.; Xu, M.; Chen, X.; Crandall, D. J.; Parikh, D.; and Batra, D. 2019. Embodied Amodal Recognition: Learning to Move to Perceive Objects. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2040–2050.

Zhang, J.; Tai, L.; Boedecker, J.; Burgard, W.; and Liu, M. 2017. Neural SLAM: Learning to explore with external memory. *arXiv preprint arXiv:1706.09520* .