Seeing by Moving: Towards Self-Supervised Amodal Object Detection

Zhaoyuan Fang†
Carnegie Mellon University
zhaoyuaf@andrew.cmu.edu

Ayush Jain†
Carnegie Mellon University
ayushj2@andrew.cmu.edu

Gabriel Sarch†
Carnegie Mellon University
gsarch@andrew.cmu.edu

Adam W. Harley
Carnegie Mellon University
aharley@cmu.edu

Katerina Fragkiadaki
Carnegie Mellon University
katef@cs.cmu.edu

[1]

Abstract

Humans learn to better understand the world by moving around their environment to get more informative viewpoints of the scene. Most methods for 2D visual recognition tasks such as object detection and segmentation treat images of the same scene as individual samples and do not exploit object permanence in multiple views. Generalization to novel scenes and views thus requires additional training with lots of human annotations. In this paper, we propose a self-supervised framework to improve an object detector in unseen scenarios by moving an agent around in a 3D environment and aggregating multi-view RGB-D information. We unproject confident 2D object detections from the pre-trained detector and perform unsupervised 3D segmentation on the point cloud. The segmented 3D objects are then re-projected to all other views to obtain pseudo-labels for fine-tuning. Experiments on both indoor and outdoor datasets show that (1) our framework performs high quality 3D segmentation from raw RGB-D data and a pre-trained 2D detector; (2) fine-tuning with self supervision improves the 2D detector significantly where an unseen RGB image is given as input at test time; (3) training a 3D detector with self supervision outperforms a comparable self-supervised method by a large margin.

1. Introduction

For tasks that require high-level reasoning, intelligent systems must be able to recognize objects despite partial occlusions or uncommon poses. The ability to perceive both the visible and the occluded regions of the environment, known as amodal perception, is especially important for understanding fundamental relationships in the scene such as depth ordering and object permanence [39]. Humans and other mammals actively move their eyes, head, and body in order to obtain less occluded and more familiar viewpoints of the objects of interest [16, 51]. Thus, in a self-supervised manner, animals are able to inform viewpoints they are less confident about by moving to highly certain viewpoints. However, amodal perception remains challenging for state-of-the-art deep learning methods which typically build complex models to hallucinate 3D instances from 2D labels [17, 29, 65]. Do we need to hallucinate what's behind a mug, if we can simply lean over and see around it?"

Imagine a scenario where one is attempting to determine the extent or identity of an occluded object from an unfamiliar viewpoint, such as in Figure 1. One can increase their certainty by simply moving to a less-occluded and more-

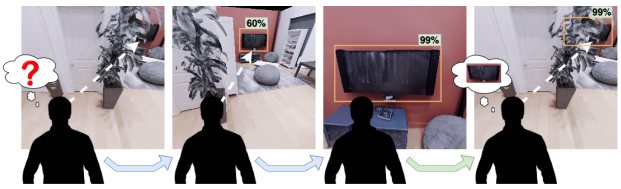

Figure 1. **Improving object recognition by moving.** An agent is viewing an object from an occluded, unfamiliar viewpoint. By moving to less occluded, more familiar viewpoints of the object (blue arrow), the agent can use the familiar viewpoints to self-supervise the previously unfamiliar viewpoints (green arrow). Subsequently, perception of objects in the previously unfamiliar views improves.

[1] † The authors contributed equally. The order is sorted alphabetically by last names.

familiar viewpoint. Then, by mapping these confident beliefs of the object back to the unfamiliar views, perception of the object from the previously unfamiliar views will improve over time and experience. In a similar manner, intelligent machine vision recognition systems can exploit simple movements and self-supervised learning to improve scene understanding, and consequently, the performance on 2D and 3D recognition tasks.

Since the rise of deep neural networks, significant improvements in accuracy and reliability of 2D [20, 32, 34, 44, 46, 4, 12, 23, 35, 62] and 3D [30, 42, 43, 41, 55, 64] visual recognition tasks have been made. Methods trained on large static image datasets perform well in the domains they are trained on, but require large amounts of human annotation for training, and have difficulty generalizing well to novel contexts and viewpoints [7]. Recent advances in active visual learning[10, 11, 60] have focused on efficient data collection techniques, so that the detector adapts to new scenes and views after fine-tuning on the collected data. However, these approaches require ground truth 3D segmentation of the environment or 2D human annotations of the images to train, making such methods expensive. This dependence on human annotation also makes it difficult to scale to a continual learning setting where the agent can extract knowledge from the unstructured world and adapt to new information [38, 13, 8].

In this work, we propose a fully self-supervised method for obtaining labels to train a network to perform amodal 2D and 3D object detection and segmentation. We have an embodied agent that moves around in its environment until it finds a high confidence detection using a pre-trained object detector. It then moves around the detected object and collects diverse posed RGB-D images. We then use the high confidence object detections to segment them in 3D space and propagate the segmentations to 2D to use as labels in all other views. Modern depth sensors, such as Lidar and stereo cameras, and pose estimation methods, such as simultaneous localization and mapping (SLAM), allow robots to represent the 3D world as a point cloud with very detailed precision [10, 49]. Thus, our label generation method is applicable to a real-world setting, and would allow the robot to obtain labels for adapting a detector to a target environment with little to no human supervision. While 2D human annotations are expensive to obtain, 3D segmentations is even more difficult to obtain in a real-world setting. Our method provides a robust way for obtaining 3D detection and segmentation labels unsupervised.

Our key contribution is the generation of high-quality 2D and 3D labels for training an object detection and segmentation network without any human annotation needed in the entire process. On indoor and outdoor datasets, we show that fine-tuning Mask-RCNN with self-supervised labels generated by our method significantly improves the Mean Average Precision (mAP) over the pre-trained detector. When we apply our method with weak supervision, we are able to further increase performance. Additionally, we show that our self-supervised 3D detection method outperforms state-of-the-art self-supervised 3D detection methods while achieving performance comparable to fully supervised methods. We finally demonstrate that our method can iteratively improve with self-supervision, which fits our model into a continual visual learning framework, whereby the system would be able to perpetually learn to reason about the world without human intervention. We will make our code and data publicly available.

2. Related Work

2D Object Recognition 2D object recognition has been one of the most important tasks in the field of Computer Vision. With the surge of deep learning, researchers collected large scale datasets to ground the development of methods [28, 33, 63]. Deep networks now achieve extraordinary performance on visual recognition tasks, including object detection [20, 32, 34, 44, 46] and semantic segmentation [4, 12, 23, 35, 62]. However, a recent study [7] showed that state-of-the-art detectors are less likely to recognize an object under unique viewpoints correctly by testing them on a new manually generated dataset of uncommon views. In this work, we aim to improve a pre-trained Mask-RCNN [57] in new environments and viewpoints in a fully self-supervised way.

3D Object Recognition 3D object recognition has also been explored in various forms. Some methods quantize pointclouds into 3D voxel grids [45, 64] to get structured data, but often voxels become expensive as the resolution increases. The PointNet series [42, 43, 41] designed an architecture to directly operate on unordered pointclouds for learning deep point set features for object detection. SPGN [55] extends the direct consumption of pointcloud to instance segmentation. Later works [30, 54] integrate direct convolution into pointclouds. All of these methods require 3D annotations, which are even more expensive than 2D annotations. Wang et al. [53] proposes a semi-supervised method, named LDLS, which performs 3D segmentation by diffusing the pre-trained detectors prediction from single view RGB-D images through building graph connections between 2D pixels and 3D points without requiring 3D ground truth information. In this work, we show that from our pseudo-labels, we can train a 3D object detector [41] which outperform LDLS and achieves comparable performance to fully supervised methods.

Active Visual Learning The problem of active vision [2, 5, 48] asks an agent to actively choose images to be labelled

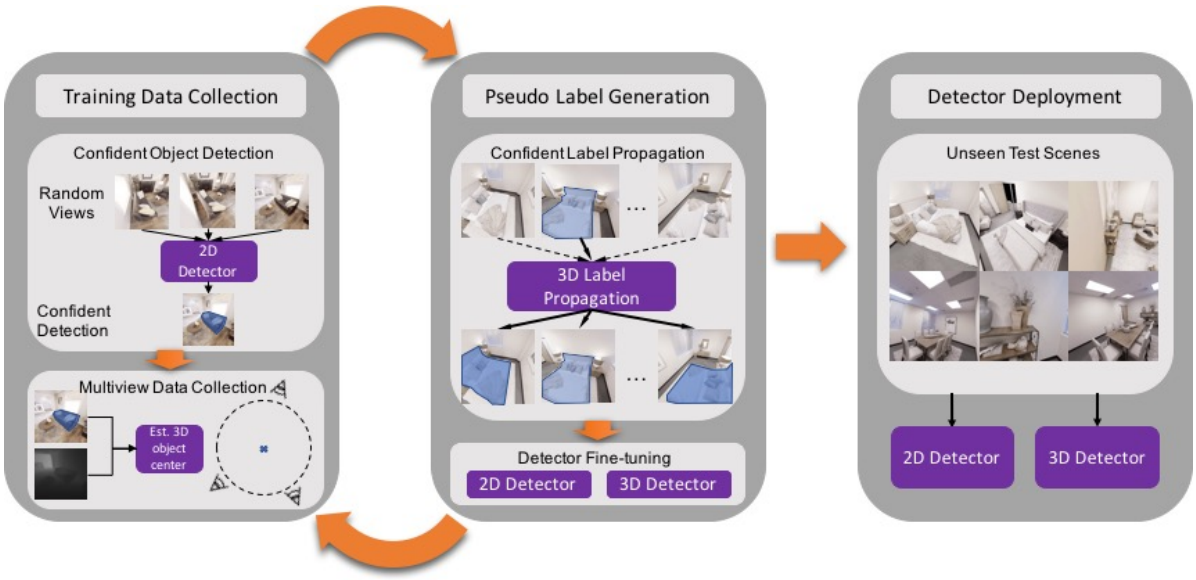

Figure 2. **Seeing by Moving (SbM).** We use confident detections of a pre-trained 2D object detector to guide self-supervised multi-view data collection and pseudo-label generation. Our 3D segmentation module uses confident detections in some views to label the object in other views, generating pseudo-labels automatically. The 2D detector fine-tuned on the pseudo-labels performs better on unseen test scenes.

from a large collection of unlabelled images in a smart way [47]. Psychology research also confirms active vision as a natural method used by humans to attend to relevant visual features [6, 16, 51, 61]. This has been applied to several fields including training object detectors [24, 25, 52, 59], instance segmentation [40], feature learning [1], and medical image analysis [27]. Another related setting is explored by Chaplot *et al*. [11] where instead of selecting images to label from pre-collected data, a policy is trained for efficiently acquiring unlabeled data. In this work, we propose a self-supervised technique complementary to both directions, where we use images with high confidence detections of a pre-trained COCO Mask-RCNN to label a set of collected multi-view images.

Embodiment Embodied agents can move and interact with their environment through a physical apparatus. 3D simulators have been an important part of modelling embodiment in a virtual setting. Many of the environments are photo-realistic reconstructions of indoor [50, 3, 58, 9] and outdoor [15, 19] scenes, and provide 3D ground truth labels for objects. These simulated environments have been used to study tasks such as visual navigation and exploration [10, 21, 18], visual question answering [14], tracking [22], and object recognition [11, 60]. In our work, we use a simulated embodied agent to discover objects and fixate their sensors on them to obtain object-centric data for fine-tuning a detector.

3. Seeing by moving

We aim to improve a (2D or 3D) object detector in novel scenes and viewpoints in a self-supervised way without using additional object annotations. Most previous methods that attempt to improve a detector [10, 11, 60] require either ground truth 3D object segmentation or human annotations of 2D images after they have been collected by the embodied agent. Some of those methods train the movement of the embodied agent without supervision to attend to specific inputs [11, 60]. However, acquiring the annotations for the collected images still remains extremely expensive. Thus, fine-tuning the detector would be prohibitive if human annotations were not available for some environments.

We propose the Seeing by Moving (SbM) model to remove the bottleneck of expensive annotations by making the labeling of images self-supervised. We take advantage of the classifier head in a pre-trained object detector, which has high confidence when the object is viewed unoccluded in a common pose. The confidence values of the pre-trained detector serves as a cue to help us select good views of objects in unseen scenarios. By unprojecting the high confidence 2D detections to 3D, segmenting, and re-projecting the 3D segmentations to all views, we are effectively propagating the high confidence detections from one view to less confident views. Note that this label propagation can also be used on with any embodied agent that collects a series of images of the same scene. We then fine-tune the 2D / 3D object detector using generated pseudo-labels and show

large improvements on both indoor and outdoor datasets. At test time, the detector is able to work on a single view. The overview of our framework is shown in Figure 2.

3.1. Self-supervised Data Collection

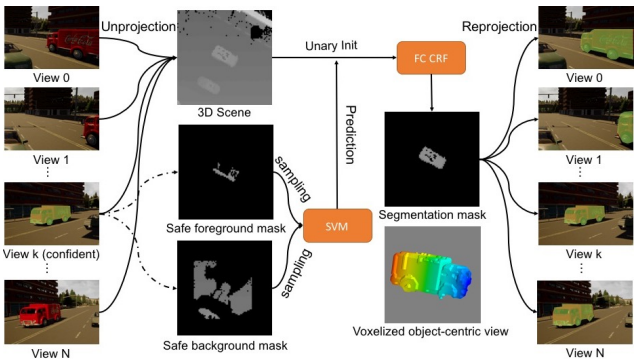

Figure 3. **Label Propagation.** All the observations are unprojected into 3D space, as well as the segmentation mask from confident view k. The foreground / background points are sampled to train an SVM, which is used to initialize the unary potentials of the fully connected CRF model. The final 3D segmentation is reprojected to all views to obtain pseudo-labels.

We have two stages in our data collection pipeline. The first stage uses an embodied agent with a depth sensor and a pre-trained object detector to obtain the 3D locations of objects in the scene. The second stage makes the agent rotate around the estimated object centroids to get multiview RGB-D observations. Note that any multiview RGB-D data collection method (for example [11]) would be acceptable for our label generation to work. We focus on a hand-designed fixation policy to ensure we obtain a diverse set of views of the objects, and to show that our method works with easily programmable movements.

In the first stage of our data collection, we randomly spawn the agent in the environment and rotate the agent about its axis until there is at least one highly confident detection of an object in view. For that view, we take all high confident detection masks, and estimate the 3D centroid of each object using depth and pose information. The 3D mask centroid is used as the approximation of the actual centroid of the object. Repeated centroid detections of the same object from multiple views are suppressed based on the heuristic that for detections of the same object, the 3D centroids are close together. Note that this is an important step for removing the bias of the pre-trained 2D detector and for capturing a uniform distribution of views for all objects.

In the second stage, we iterate through each centroid, moving the agent inside an annulus around the object. This is accomplished by first uniformly sampling navigable and non-obstructed points at various radii from the object centroid. We then have the agent move along a trajectory connecting these locations, and fixate the camera on the object such that the centroid projection is centered in the frame. Using this pipeline, we obtain a set of multi-view images in which there is at least one view with a high confidence detection, which we then propagate to other views.

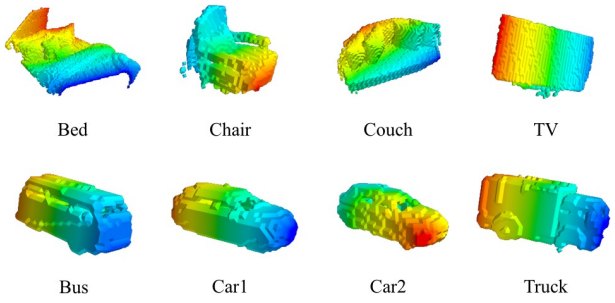

Figure 4. **Visualizations of 3D object segmentation on CARLA and Replica datasets.** We show qualitative examples of the voxelized 3D segmentations of our method, on both Replica (top) and CARLA (bottom).

3.2. Self-supervised Label Generation

Given N observations from different views for each episode obtained in our data collection, we propagate high confidence detections in some views to all frames in the episode. Taking advantage of object permanence, our self-supervised label generator segments objects in 3D space and reprojects the 3D segmentation to produce 2D pseudo-masks. The core of the label generator is a fully connected conditional random field (CRF) with Gaussian edge potentials [26], whose unary potential is computed using an SVM classifier. The diagram of our label propagation is shown in Figure 3.

From the RGB observations and depth measurements, we first unproject the RGB images from all views into a common 3D reference frame using camera pose and depth. For each high confidence detection, we dilate its 2D object mask and take the inverse to obtain a safe 2D background mask. Then we also erode the 2D object mask to obtain a safe 2D object mask, to make sure pixels covered by masks are accurately labelled. This is important because the pre-trained detector, even with high confidence, has unavoidable flaws at object boundaries [53]. We unproject all object masks and background masks to the 3D reference frame. Hence, we obtain a pointcloud $S = \bigcup_{i=1}^{N} S^{(i)}$, where $S^{(i)} = \{(x, y, z)_j\}_{j=1}^{K_i}$ denotes the set of points unprojected from view i, with K_i being the number of points.

We now describe the segmentation of objects on pointclouds. For clarity, we focus on one high confidence detection in view k. Note that we can partition the point-

cloud $S^{(k)}$ into three sets: foreground, background, and unknown border. For all other views $S^{(k')}$ where $k' \neq k$ the points are unlabelled. To segment the unknown border and the other unlabelled points, we first train an SVM on the foreground and background sets from $S^{(k)}$, where the features are the normalized color and 3D coordinate. Note that we can also use a pre-trained deep network to obtain the feature representation, but we noticed that the simple 6 dimensional color and position feature works just as well and is much faster. We conjecture that this is due to the sparse nature of points in the 3D space, making segmentation easier. However, the SVM predictions are oftentimes inaccurate, hence we use a fully connected CRF to obtain the final refined 3D segmentation. In the CRF model, each node is a point $S_j^{(i)}$ in the pointcloud and we set its unary potential as the predicted class probabilities from SVM. The pairwise potential between two points $S_j^{(i)}$ and $S_{j'}^{(i')}$ can be written as:

$$\psi_p\left(S_j^{(i)}, S_{j'}^{(i')}\right) = \mu\left(S_j^{(i)}, S_{j'}^{(i')}\right) k(\mathbf{f}_j^{(i)}, \mathbf{f}_{j'}^{(i')}) \quad (1)$$

where μ is the label compatibility function given by the Potts Model, and k is a contrast-sensitive potential given by the combination of an appearance kernel and a smoothness kernel, defined similarly as in [26]. The Gibbs energy is therefore the summation of the unary and pairwise potentials.

Finally, we obtain 3D pointcloud segmentation for each object and use them as pseudo-labels to train a 3D object detector on RGB-D data. Visualizations of the voxelized pointcloud segmentation are shown in Figure 4. For visualizations of 2D pseudo-labels generated from 3D segmentations, please see supplementary materials. To obtain pseudo-labels of 2D segmentation, we re-project the 3D segmentation to all views. If multiple confident input masks of the same object (from different views) are given, we perform label densification at reprojection time, so that a pixel is labelled by the most confident prediction. Note that as a benefit of the segmentation in 3D space, we can obtain both modal and amodal masks in 2D, by reprojecting either only points from this view or all points that belong to the object. From those we can train modal / amodal 2D object detectors.

4. Experiments

4.1. Datasets

Environments We run our experiments in both indoor and outdoor environments. For the indoor environment, we use the Habitat simulator [36] with the Replica dataset [50], which contains high quality and realistic reconstructions of indoor spaces. For the outdoor environment, we use the CARLA simulator [15], which renders urban driving scenes. We have an agent move around in the 3D simulator while collecting multi-view RGB-D images of objects in the scenes. Though the simulated environments are noise-free in pose and depth measurements, previous work has shown that accurate estimates of pose and depth can be obtained from RGB image under noisy odometry [10]. Since pose and depth estimation is tangential to our work, we assume ground truth pose and depth. The Replica dataset consists of 18 distinct indoor scenes, such as offices, hotels, and apartments. We split the scenes into disjoint sets such that there are 15 for training, 1 for validation, and 2 for testing. In our self-supervised data collection, we capture 25 views in each episode. We have 17k images for training, 1k for validation, and 2.3k for testing. The CARLA driving scenes consist of five distinct towns. We again split them dis-jointly into 3 towns for training, 1 for validation, and 1 for testing. In our self-supervised data collection, we capture 25 views in each episode. We have 5.3k images for training, 1.8k for validation, and 1.8k for testing.

Objects For CARLA, we spawn two vehicles each episode randomly. Since CARLA has the same semantic label for all vehicles, we consider detection of all vehicle classes of the COCO dataset when we evaluate [33]. For Replica, we keep the default layout of objects in each scene. We consider a subset of the objects categories in each simulator based on the following standards: (1) the object category is shared between COCO and Replica, and 2) enough instances occurred in the dataset. These include chair, couch, plant, tv, fridge, bed, table and toilet for Replica. Though our method do not necessitate choosing shared objects between COCO and Replica, its essential for grounding our experimental results in comparison to fully supervised methods that require ground truth annotations for training.

4.2. Implementation Details

2D object detection We use the Mask-RCNN model [57] with FPN [31] using ResNet-50 as the backbone. The detector is pre-trained on the COCO dataset. It is fine-tuned on the 2D pseudo-labels of the training set and its mAP with at IoU threshold of 50% is computed on a validation set every 5000 iterations. For comparison, we also fine-tuned the network on the same datasets but with ground truth annotations. For both datasets and both settings, we use a learning rate of 0.001 and a batch size of 2.

3D object detection We train the frustum PointNet model [41] with PointNet [42] backbone on Carla in a self-supervised way. The original frustum PointNet model uses ground truth 2D bounding boxes and camera pose to define a 3D frustum search space and then performs 3D segmentation on it using PointNet based architecture and ground

truth 3D boxes for supervision. In our experiment, we use the self-supervised 2D and 3D bounding boxes generated from SbM's 2D and 3D bounding boxes. To gauge our self-supervised frustum PointNet's performance, we also train the same network using ground truth 2D and 3D bounding boxes. For both settings, we train it using a learning rate of 0.001 and a batch size of 32 until convergence using the train-validation curves. Similar to previous experiments, we test both the models on a new unseen town and compare their performance.

4.3. 2D Object Detection

We analyze our SbM framework for 2D object detection by asking the following questions: (1) does the label propagation procedure by moving around produce labels of higher quality for the images? (2) does fine-tuning the object detector on pseudo-labels improve the detector's performance on unseen scenes? Our experiments on both Carla and Replica show that the answer to both is "yes".

mAP@IoU	Method	Train	Test
0.5	Pre-trained	68.05	68.23
	SbM (ours)	**75.94**	**78.59**
	GT fine-tuned	-	93.76
0.3	Pre-trained	73.09	75.55
	SbM (ours)	**85.62**	**86.33**
	GT fine-tuned	-	94.71

Table 1. **2D object detection performance comparison on CARLA test set** Fine-tuning 2D detector on self-supervised SbM labels increases pre-trained models performance taking its performance closer to supervised fine-tuning.

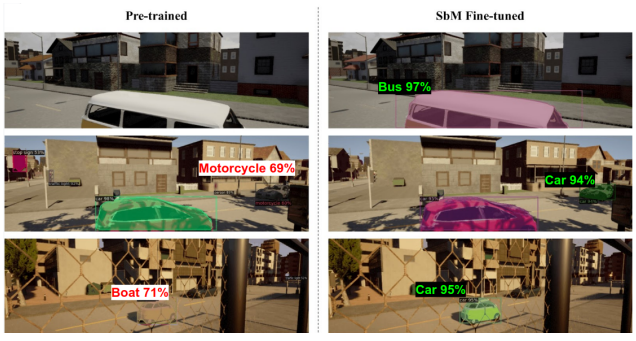

Figure 5. **Visualizations of 2D detector performance on CARLA test set.** We show paired qualitative examples of the detections of pre-trained 2D detector (left) and SbM fine-tuned 2D detector (right) on the CARLA test set. The improvements are shown in larger fonts for better visibility. The pre-trained detector misses objects and classifies the object as the wrong category, while the fine-tuned model produces accurate class predictions, bounding boxes, and semantic masks.

CARLA The performance comparison of our method, the pre-trained detector, and the detector fine-tuned on ground truth data is shown in Table 1. Classless mAP is reported because all vehicles in CARLA have the same semantic label. We report mAP at IoU of 0.5 and 0.3. At training time, we investigate the setting where the embodied agent is free to move around, obtain observations, and use SbM to generate predictions for all views. We observe that pseudo-labels generated by SbM have better performance than the pre-trained detector outputs. This shows that moving and performing segmentation in the 3D space helps the detector see better, when compared to treating multi-view images as individual observations. At test time, the detector is fine-tuned with the pseudo-labels and is deployed in unseen environments where only a single RGB image is given as input. Results show that the detector fine-tuned with SbM outperforms the pre-trained detector by a large margin. This indicates that we can improve the detector's performance in unseen environments with no additional human labels. By moving a detector around to gather and label data using information it acquired previously, the detector is able to improve itself. Figure 5 shows qualitative comparisons of the detections of the pre-trained detector and the detector fine-tuned by SbM pseudo-labels.

Replica The performance of our SbM label propagation on the training set is shown in Table 2. We observe that SbM-generated labels are more accurate than the pre-trained detector on all classes, indicating that moving around helps generate better labels. Note that the performance on "table" category is very low for both the pre-trained detector and SbM. This is because the dining table class in COCO is visually very different from the table class in Replica (see supplementary). The performance comparison of the pre-trained, SbM fine-tuned, and ground truth fine-tuned detectors on the test set is shown in Table 3. The SbM fine-tuned detector overall outperforms the pre-trained detector by large margins while improving performance on most categories. In Figure 6, we also present qualitative comparisons of the detections of the pre-trained detector and the detector fine-tuned by SbM pseudo-labels. This confirms that fine-tuning on pseudo-labels generated by moving around makes the detector robust to viewpoint change and output accurate predictions. Surprisingly, the performance of couch decreased after fine-tuning even though the pseudo-label mAP of couch is higher than pre-trained MaskRCNN on the training set. From visualizations (see supplementary), we found that the pre-trained detector often recognizes couch as bed with high confidence, which are propagated to other views by SbM, thus corrupting the pseudo-labels. This reveals a limitation of our method, which can possibly be mitigated by using context aware detectors [37, 56] or by providing weak supervision.

mAP@IoU	Method	Bed	Chair	Couch	Table	Plant	Fridge	Toilet	TV	Avg
0.5	Pre-trained	3.13	12.48	22.81	0.14	13.18	8.37	1.45	30.43	11.58
	SbM (ours)	**7.03**	**22.65**	**34.15**	0.14	**16.26**	**26.49**	**3.72**	**31.39**	**17.73**
	SbM-ws (ours)	29.82	46.91	51.54	29.65	52.15	32.85	42.18	52.10	42.15
0.3	pre-trained	12.79	14.06	23.22	**0.16**	35.57	8.37	2.38	30.86	15.93
	SbM (ours)	**19.39**	**33.03**	**36.74**	0.14	**38.20**	**30.97**	**7.18**	**38.57**	**25.53**
	SbM-ws (ours)	52.21	67.46	75.76	46.69	68.58	56.23	62.53	69.51	61.99

Table 2. **2D object detection performance of pre-trained Mask-RCNN vs self-supervised SbM vs weakly supervised SbM on Replica training set.** Self-Supervised SbM consistently outperform pre-trained MaskRCNN across most categories. Providing weak supervision (a single view ground truth annotation) to SbM increases its performance significantly on all categories.

mAP@IoU	Method	Bed	Chair	Couch	Table	Plant	Fridge	Toilet	TV	Avg
0.5	Pre-trained	-	11.87	**30.44**	0.76	33.07	**0.50**	-	41.60	19.71
	SbM Fine-tuned (ours)	-	**25.81**	26.33	**5.59**	**44.47**	0.00	-	**43.82**	**24.34**
	GT Fine-tuned	-	57.86	53.47	5.31	89.92	46.46	-	87.20	56.70
0.3	Pre-trained	-	15.61	**40.59**	0.76	36.86	**0.50**	-	41.60	22.65
	SbM Fine-tuned (ours)	-	**20.48**	36.99	**5.59**	**64.55**	0.00	-	**57.99**	**32.60**
	GT Fine-tuned	-	62.22	57.09	5.31	93.48	50.28	-	88.38	59.46

Table 3. **2D object detection performance of pre-trained, SbM fine-tuned (ours), and ground truth fine-tuned Mask-RCNN on Replica test set.** Training on SbM generated pseudo-labels improve the detector performance on the test set by a large margin.

Figure 6. **Visualizations of 2D detector performance on Replica test set.** We show paired qualitative examples of the predictions of the pre-trained 2D detector (top) and the SbM fine-tuned 2D detector (ours) (bottom) on Replica test set. The pre-trained detector misses objects and classifies the object as the wrong category, while the fine-tuned model produces accurate class predictions, bounding boxes, and semantic masks.

Weakly-supervised label propagation As previously noted, moving around does not help much in cases where the pre-trained detector performs poorly on all views. Can we generate higher quality labels if provided weak supervision? We show that our SbM framework can also perform pseudo-label generation for the set of multi-view images when we have some ground-truth 2D annotations, enabling us to generate high quality labels for non-COCO instances, as well as for categories where the pre-trained detector fails. In our experiments, we only provide ground truth annotation on 1 view for each object out of the 25 views, making the label propagation procedure weakly supervised in 2D.

mAP@IoU	Method	Bed	Chair	Couch	Table	Plant	Fridge	Toilet	TV	Avg
0.5	SbM	7.03	22.65	34.15	**0.14**	**16.26**	**26.49**	**3.72**	31.39	17.73
	SbM Fine-tuned	**9.12**	**25.10**	**42.20**	0.01	11.75	21.84	0.27	**40.47**	**18.85**
0.3	SbM	19.39	33.03	36.74	**0.14**	38.20	**30.97**	**7.18**	38.57	25.53
	SbM Fine-tuned	**36.08**	**38.83**	**54.56**	0.01	**38.74**	26.03	0.47	**57.50**	**31.53**

Table 4. **Fine-tuning with pseudo labels increases quality of generated labels.** Reported is the performance of pseudo labels obtained from using pre-trained Mask-RCNN versus using our fine-tuned Mask-RCNN in the segmentation pipeline. This demonstrates how our method can be used in continual learning setup.

We report the label quality in Table 2, where the weakly supervised SbM is denoted as SbM-ws. We observe that with one ground truth 2D annotation per object, the pseudo-label quality is better than both pre-trained MaskRCNN and self-supervised SbM by a large margin. This suggests that our SbM framework generates better labels with improved pre-trained detector.

Continual improvement From previous experiments, we see that (1) the pre-trained detector can be improved in a self-supervised manner on collected data; (2) better detection (weak supervision) on some views improve the pseudo-label quality. Therefore, we can naturally formulate the problem in a continual learning setting: we can deploy the fine-tuned detector again in the training environments and repeat the first and second stage of our framework to improve the detector further. For clearer comparison, we deploy the fine-tuned detector on the collected training images again and perform confident label propagation. The results are presented in Table 4. Comparing the pseudo-labels generated from pre-trained detector detections and the ones from fine-tuned detector detections, we see that the quality of the generated labels is further improved in this continual setting. While the overall mAP and the mAP for most categories improved upon deploying the fine-tuned detector for generating labels, interestingly, performance on some categories dropped slightly. We believe that adding more relationship constraints within environment categories is essential for scaling to full continual learning setup as proposed by [13, 8].

4.4. 3D Object Detection

Can we train a 3D object detector self-supervised without any ground truth data? To answer this question, we compare the two versions of frustum PointNet: one trained on SbM's self-supervised 3D and 2D labels (Figure 4), and the other trained on ground truth 3D and 2D labels. We also compare our method with semi-supervised LDLS [53] method. The experiments are conducted in CARLA.

The test set performance comparison of LDLS [53], frustum PointNet trained on SbM segmentations, and frustum PointNet trained on ground truth is shown in Table 5. Our self-supervised frustum PointNet model outperforms LDLS significantly. Our model also achieves reasonably good performance as compared to the fully supervised frustum PointNet model. We show qualitative examples of the 3D detections from LDLS and SbM fine-tuned frustum PointNet in Figure 7. This demonstrates that the 3D segmentation labels produced by SbM are high quality and could be successfully used to train state of the art 3D detection models without ground truth 3D annotations.

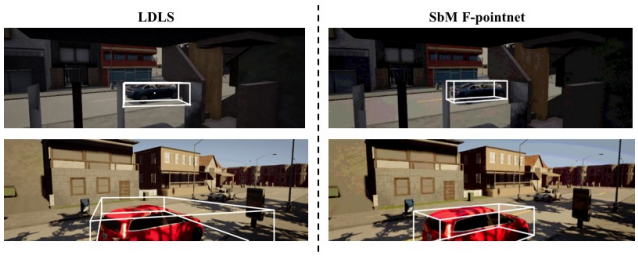

Figure 7. **Visualizations of 3D detector performance on CARLA test set.** We show paired qualitative examples of the detections of LDLS [53] (left) and SbM-trained frustum PointNet (ours) (right) on the CARLA test set. While LDLS gets a rough box estimation, the SbM-trained frustum PointNet is able to obtain bounding boxes that are much tighter and better-oriented.

Method	mAP@IoU=0.25
LDLS [53]	44.03
SbM Self-Sup. F-PointNet (ours)	**64.39**
Supervised F-PointNet	85.06

Table 5. **Fine-tuning with SbM labels outperform self-supervised LDLS.** 3D object detection performance of LDLS [53], frustum PointNet trained on SbM segmentations, and GT-trained frustum PointNet on the CARLA test set.

5. Conclusion

In this work, we introduce a fully self-supervised method for obtaining labels for fine-tuning a pre-trained detector.

In multiple domains, we demonstrate that our method significantly improves performance of a pre-trained detector (just by moving around), and generalizes to a test domain. We also show our method can be extended to train a 3D detector without human annotation. For future work, we believe there is a lot more to explore in the area of self-supervised or weakly-supervised improvement of detectors. Our method currently assumes perfect odometry and localization, whereas those oftentimes need to be estimated in practice. Another limitation of our method is that it depends on the assumption that the pre-trained detector makes correct predictions at least for some views. Loosening these constraints so that novel object discovery can be performed are direct avenues for future work.

References

[1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *ICCV*, 2015. 3

[2] John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. Active vision. *IJCV*, 1(4):333–356, 1988. 2

[3] Phil Ammirato, Patrick Poirson, Eunbyung Park, Jana Košecká, and Alexander C Berg. A dataset for developing and benchmarking active vision. In *ICRA*, 2017. 3

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017. 2

[5] Dana H. Ballard. Animate vision. *Artif. Intell.*, 48(1):57–86, Feb. 1991. 2

[6] Sven Bambach, David J. Crandall, Linda B. Smith, and Chen Yu. Toddler-inspired visual object learning. In *NeurIPS*, 2018. 3

[7] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *NeurIPS*, 2019. 2

[8] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. Toward an architecture for never-ending language learning. In *Aaai*, volume 5. Atlanta, 2010. 2, 8

[9] 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. In *3DV*, 2017. 3

[10] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *ICLR*, 2020. 2, 3, 5

[11] Devendra Singh Chaplot, Helen Jiang, Saurabh Gupta, and Abhinav Gupta. Semantic curiosity for active visual learning. In *ECCV*, 2020. 2, 3, 4, 12

[12] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 2

[13] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. Neil: Extracting visual knowledge from web data. In *ICCV*, 2013. 2, 8

[14] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *CVPR Workshops*, 2018. 3

[15] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 3, 5

[16] Shimon Edelman and Heinrich H Bülthoff. Orientation dependence in the recognition of familiar and novel views of three-dimensional objects. *Vision research*, 32(12):2385–2400, 1992. 1, 3

[17] Kiana Ehsani, Roozbeh Mottaghi, and Ali Farhadi. Segan: Segmenting and generating the invisible. In *CVPR*, 2018. 1

[18] Kuan Fang, Alexander Toshev, Li Fei-Fei, and Silvio Savarese. Scene memory transformer for embodied agents in long-horizon tasks. In *CVPR*, 2019. 3

[19] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 3

[20] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 2

[21] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *CVPR*, 2017. 3

[22] Adam W. Harley, Shrinidhi Kowshika Lakshmikanth, Paul Schydlo, and Katerina Fragkiadaki. Tracking emerges by looking around static scenes, with neural 3d mapping. In *ECCV*, Lecture Notes in Computer Science, 2020. 3

[23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2

[24] Dinesh Jayaraman and Kristen Grauman. End-to-end policy learning for active visual categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1601–1614, 2019. 3

[25] Edward Johns, S. Leutenegger, and A. Davison. Pairwise decomposition of image sequences for active multi-view recognition. In *CVPR*, 2016. 3

[26] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*, 2011. 4, 5

[27] Weicheng Kuo, Christian Häne, E. Yuh, P. Mukherjee, and Jitendra Malik. Cost-sensitive active learning for intracranial hemorrhage detection. In *MICCAI*, 2018. 3

[28] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018. 2

[29] Ke Li and Jitendra Malik. Amodal instance segmentation. In *ECCV*, 2016. 1

[30] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on χ-transformed points. In *NeurIPS*, 2018. 2

[31] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 5

[32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 2

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *ECCV*, 2014. 2, 5

[34] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 2

[35] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[36] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *ICCV*, 2019. 5

[37] Kenneth Marino, Ruslan Salakhutdinov, and Abhinav Gupta. The more you know: Using knowledge graphs for image classification. In *CVPR*, 2017. 6

[38] Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, et al. Never-ending learning. *Communications of the ACM*, 61(5):103–115, 2018. 2

[39] Bence Nanay. The importance of amodal completion in everyday perception. *i-Perception*, 9(4):2041669518788887, 2018. 1

[40] Deepak Pathak, Yide Shentu, Dian Chen, Pulkit Agrawal, Trevor Darrell, Sergey Levine, and Jitendra Malik. Learning instance segmentation by interaction. In *CVPR Workshop*, 2018. 3

[41] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *CVPR*, 2018. 2, 5

[42] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CVPR*, 2017. 2, 5

[43] Charles R. Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 2

[44] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2

[45] Mengye Ren, Andrei Pokrovsky, B. Yang, and R. Urtasun. Sbnet: Sparse blocks network for fast inference. *CVPR*, 2018. 2

[46] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, Cambridge, MA, USA, 2015. MIT Press. 2

[47] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018. 3

[48] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. 2

[49] Nikolai Smolyanskiy, Alexey Kamenev, and Stan Birchfield. On the importance of stereo for accurate depth estimation: An efficient semi-supervised deep neural network approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1007–1015, 2018. 2

[50] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 3, 5

[51] Michael J. Tarr. Rotating objects to recognize them: A case study on the role of viewpoint dependency in the recognition of three-dimensional objects. *Psychonomic Bulletin & Review*, 2(1):55–82, 1995. 1, 3

[52] Sudheendra Vijayanarasimhan and Kristen Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. In *CVPR*, 2011. 3

[53] Brian H. Wang, Wei-Lun Chao, Yan Wang, Bharath Hariharan, Kilian Q. Weinberger, and Mark Campbell. Ldls: 3-d object segmentation through label diffusion from 2-d images. *IEEE Robotics and Automation Letters*, 4(3):2902–2909, July 2019. 2, 4, 8

[54] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In *CVPR*, 2018. 2

[55] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *CVPR*, 2018. 2

[56] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *CVPR*, 2018. 6

[57] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 2, 5

[58] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *CVPR*, pages 9068–9079, 2018. 3

[59] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Visual curiosity: Learning to ask questions to learn visual recognition. In *CoRL*, 2018. 3

[60] Jianwei Yang, Zhile Ren, Mingze Xu, Xinlei Chen, David J. Crandall, Devi Parikh, and Dhruv Batra. Embodied amodal recognition: Learning to move to perceive objects. In *ICCV*, 2019. 2, 3

[61] Scott Cheng-Hsin Yang, Mate Lengyel, and Daniel M Wolpert. Active sensing in the categorization of visual patterns. *Elife*, 5:e12215, 2016. 3

[62] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2

[63] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 2

[64] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018. 2

[65] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *CVPR*, 2017. 1

A. Apendix Overview

In Section B, we provide implementation details for our SbM method and data collection. In Section C, we provide additional visualizations of our output. In Section D, we discuss the limitations of self-supervised SbM in detail. In Section F, we discuss the weakly supervised SbM method that can address some of the limitations of fully supervised method and also work on novel categories. Finally in Section E, we investigate the effects of adding multi-view consistency constraints on the quality of SbM generated labels.

B. Method Details

B.1. Point Cloud

2D-to-3D unprojection This module converts the input RGB image $I \in \mathbb{R}^{w \times h \times 3}$ and depth map $D \in \mathbb{R}^{w \times h \times 1}$ into a 3D point cloud $\mathbf{P} \in \mathbb{R}^{x \times y \times z}$. For each image, using known intrinsic parameters of the camera $\mathbf{K} \in \mathbb{R}^{4 \times 4}$, we calculate the 3D coordinate of each pixel in the image relative to the camera. To obtain an aggregated point cloud over all viewpoints, for each of the K views in the multi-view sequence, we rotate and translate the point cloud to the reference frame, defined as the coordinate frame of the first view, using the ground truth transformation matrix $[\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{4 \times 4}$. We then aggregate the transformed 3D points across all viewpoints to obtain a dense multi-view point cloud.

3D-to-2D projection Given a point cloud in a camera coordinate frame, this module computes the visible 2D projection of the point cloud onto a target viewpoint. Before projection, we rotate and translate the point cloud to be in the coordinate frame of the target camera using ground truth transformation matrix $[\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{4 \times 4}$. For each point in the 3D point cloud indexed by (i, j, k), we compute the 2D pixel location (x, y) which the point projects onto, from the current camera viewpoint:

$$[x, y] = [f \cdot i/k, f \cdot j/k], \tag{2}$$

where f is the focal length of the camera. If multiple points fall along the same casted ray, we take only the point which is closest to the camera as the projection.

B.2. Data collection

Replica We discover objects using a pretrained COCO MaskRCNN detector. A detected object with a confidence threshold of 0.9 is required for the object to be used for data collection. The centroid of the object is obtained by taking the mean (x, y, z) coordinate of the 3D point cloud obtained by unprojecting the object mask to 3D world-centric coordinates using the depth map, camera intrinsics, and agent pose. We use a radius range of 1-2 meters from the estimated object centroid for sampling points for the agent to obtain observations.

Carla For Carla, we spawn two vehicles so that one is randomly selected from a naturally-occurring spawning location and pose (given by the Carla simulator), and the second is the closest naturally-occurring spawning location at least 3 meters from the first car. We spawn the agent near the vehicles and randomly sample locations for the agent to obtain observations at a radius range of 3-15 meters from the centroid of the first car.

C. SbM pseudo-label visualizations

We show visualizations of the 2D pseudo-masks reprojected from the 3D segmentation for a variety of classes in the Replica dataset in Figure 8. We can see that the borders of the objects are nicely segmented.

D. Limitations of Self-Supervised SbM

Novel and unmatched categories One evident limitation of the method is that if objects in the new environment are not among the classes the detector is trained on, our SbM label propagation would not be applicable. For example, Replica contains objects like bean bag and cushion which are not among the COCO classes, so they are never detected by the pre-trained detector. A similar case is when the categories exist in both environments, but they are semantically and visually very different. This is indicated in the paper by the poor performance of SbM on the "table" class is due to the unmatched definitions of the "table" category in COCO and Replica. In Figure 9, we show that the tables in COCO and Replica differ significantly in semantic meaning and appearances.

Dependency of SbM on pre-trained detector As briefly discussed in the paper, another limitation of the completely self-supervised version of SbM is that in order to propagate high-quality labels, the pre-trained detector must detect objects correctly with high confidence. Due to novel environments and viewpoints, the pre-trained detector sometimes detects wrong objects with high confidence, as shown in Figure 10.

E. Multi-view Consistency

Inspired by previous work [11], we also examined whether adding additional constraints on the detection consistency of the semantic category predictions across views would increase the quality of our labels on the Replica dataset. We kept our pipeline the same except that we removed images from our training set which did not

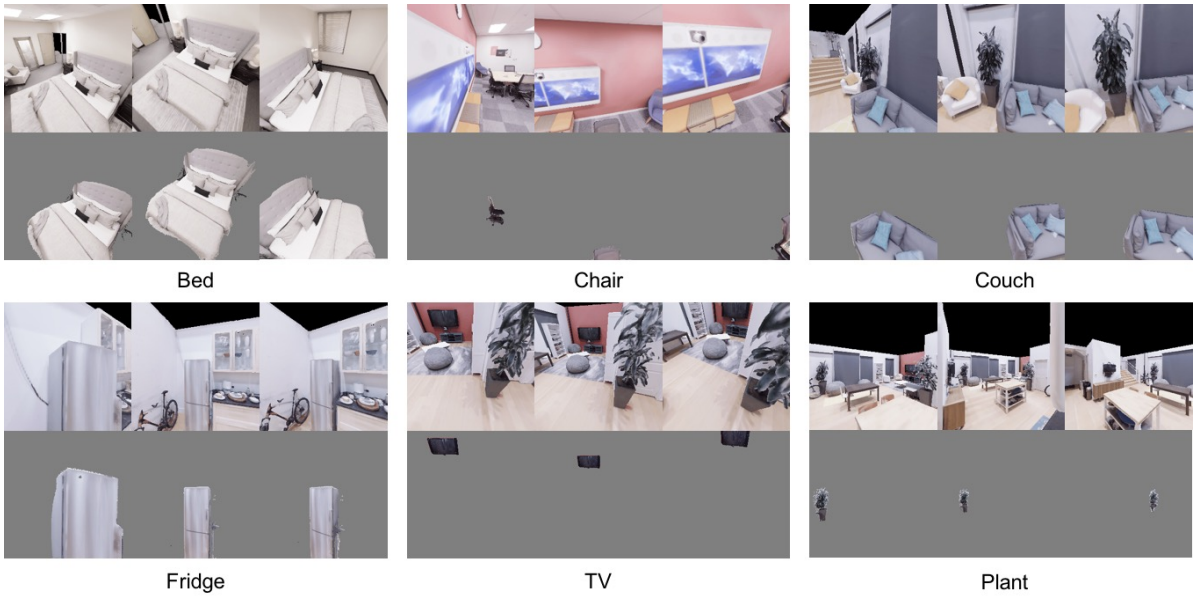

Figure 8. **Example 2D pseudo-labels reprojected from 3D segmentation.** We show examples of the reprojections of 3D segmentation (which are used as 2D pseudo-labels) on a variety of objects in the Replica dataset.

mAP@IoU	Method	Bed	Chair	Couch	Table	Plant	Fridge	Toilet	TV	Avg
0.5	Pre-trained	-	11.87	30.44	0.76	33.07	**0.50**	-	41.60	19.71
	SbM w/o consistency	-	**25.81**	26.33	**5.59**	44.47	0.00	-	**43.82**	**24.34**
	SbM w/ consistency	-	4.78	**40.10**	1.08	**69.68**	0.00	-	19.20	22.47
0.3	Pre-trained	-	15.61	40.59	0.76	36.86	**0.50**	-	41.60	22.65
	SbM w/o consistency	-	**20.48**	36.99	**5.59**	64.55	0.00	-	**57.99**	**32.60**
	SbM w/ consistency	-	7.21	**46.91**	1.08	**69.68**	0.00	-	27.22	25.35

Table 6. **2D object detection performance of MaskRCNN pre-trained, SbM fine-tuned, and SbM fine-tuned with a multi-view semantic consistency constraint on Replica test set.** We implemented an additional constraint to exclude views for label generation if the semantic predictions of the same object instance across multiple viewpoints was not the same. Fine-tuning MaskRCNN with the consistency constraint demonstrates worse mAP on average on the Replica dataset compared to fine-tuning without the constraint. The pretrained COCO MaskRCNN is included for reference.

mAP@IoU	Method Name	Cushions	Nightstand	Shelf	Beanbag	Avg
0.5	SbM-ws	66.92	49.36	43.72	75.74	58.93
0.3	SbM-ws	75.68	65.16	65.82	87.90	73.64

Table 7. **SbM-ws can generate high quality labels for novel categories** Reported is the performance of pseudo labels obtained from SbM-ws on the train set. The high mAP values demonstrate that SbM-ws can be effectively used to generate labels for categories not present in COCO.

meet the following criteria for each detected object in the image:

1. There were atleast three other views in the mutli-view sequence with high confidence detections of the object instance from the pretrained COCO MaskRCNN.

2. All detections of the object instance predicted the same class (i.e. unique classes predicted for the object instance must not exceed one)

This enforced semantic consistency across viewpoints

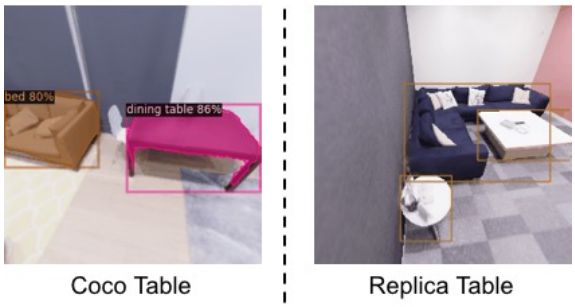

Figure 9. **Semantically and visually different tables in COCO and Replica.** We show a table (predicted as "dining table" by the pre-trained MaskRCNN) on the left, and two actual tables in Replica dataset on the right.

Figure 10. **Incorrect detections by pre-trained detector with high confidence.** We show three examples where the pre-trained detector incorrectly classify the object with high confidence.

such that the Mask-RCNN semantic predictions of an object instance across multiple viewpoints was required to be both confident and reliable to keep the object instance for label generation. We hypothesized this consistency constraint would improve the quality of our generated labels, and subsequently the performance when we fine-tuned the detector with the obtained labels. However, the mAP of the detector fine-tuned on labels with the consistency constraint did not improve as much overall from the pretrained detector as compared to the fine-tuned detector trained on labels without the consistency constraint, as shown in Table 6. We attribute this to a large class imbalance in object categories that were more likely to be tagged as consistent in the dataset, such as plant and couch. We did not investigate this further, and used fine-tuning without this constraint for our main results.

F. Weakly-supervised SbM

In section 4.3 of the main paper, we described weakly supervised SbM (SbM-ws) which assumes that each object of interest has a ground truth label for one view out of the 25 recorded views. We showed that assuming one ground truth label increases the mAP performance significantly for categories contained in the COCO dataset used to train MaskRCNN. We also examined this weak supervision in the context of novel object categories not contained in the COCO dataset. Since embodied agents typically encounter a lot of new objects while exploring, it would thus be useful if those new objects could be learned with a smaller number of human annotations. In this section, we show that SbM-ws can be used to generate labels for novel objects too, which are not contained in the category set from COCO. We also show that using these labels, we can train MaskRCNN to recognize those novel objects. We use the exact same experimental settings as in Section 4.3 for data collection and label generation, except we assume that in each multi-view episode, we have a single segmentation label for a novel object. We use this ground truth label as the weak supervision for label generation.

Figure 11. **Visualizations of the detection results of the detector trained with SbM-ws pseudo-labels** From visualizations of detection results on the test set, we can see that the detector supervised by SbM-ws pseudo-labels generates robust predictions.

mAP@IoU	Method Name	Cushion
0.5	SbM-ws	83.74
	SbM Trained	**93.62**
	GT Trained	87.24
0.3	SbM-ws	91.58
	SbM Trained	**94.23**
	GT Trained	87.24

Table 8. **SbM-ws labels can be used to train MaskRCNN on novel categories** We compare the performance of MaskRCNN trained on labels produced by SbM-ws (2nd row) vs the MaskRCNN trained on ground truth labels (3rd row) vs the performance of SbM-ws labels. The results show that MaskRCNN trained on SbM-ws labels outperforms the MaskRCNN trained on the ground truth labels.

Experiment Details For this experiment we consider four objects: Cushion, Nightstand, Shelf and Beanbag. Notice that these categories are not present in COCO and hence considered categories novel to the pre-trained detector. Using SbM-ws, we generate pseudo labels for these categories on the train set. We only included the cushion pseudo labels obtained from weakly-supervised label generation for training MaskRCNN because it was the only category occurring in more than 30% of the training set. We use a learning rate of 0.001 and a batch size of 2.

Experiment Results To gauge the quality of SbM-ws labels for novel objects, we computed the Mean Average Precision (mAP) of the propagated labels against the ground truth labels. The results are shown in Table 7. Our results show that SbM-ws can generate highly accurate labels for other views under weak supervision.

We further investigate if we can use SbM-ws labels for training the MaskRCNN on novel categories. We also trained Mask-RCNN on the ground truth views that the weak supervision method assumed in Section 4.3 of the main paper. The results of the experiment are shown in Table 8 and qualitative examples are shown in Figure 11. We see that the detector trained on SbM pseudo-labels outperforms the detector trained on the limited ground truth data.