Move to See Better: Self-Improving Embodied Object Detection

Zhaoyuan Fang†
zhaoyuaf@andrew.cmu.edu
Ayush Jain†
ayushj2@andrew.cmu.edu
Gabriel Sarch†
gsarch@andrew.cmu.edu
Adam W. Harley
aharley@cs.cmu.edu
Katerina Fragkiadaki
katef@cs.cmu.edu

Carnegie Mellon University
5000 Forbes Ave,
Pittsburgh, PA 15213

Abstract

Passive methods for object detection and segmentation treat images of the same scene as individual samples, and do not exploit object permanence across multiple views. Generalization to novel or difficult viewpoints thus requires additional training with lots of annotations. In contrast, humans often recognize objects by simply moving around, to get more informative viewpoints. In this paper, we propose a method for improving object detection in testing environments, assuming nothing but an embodied agent with a pre-trained 2D object detector. Our agent collects multi-view data, generates 2D and 3D pseudo-labels, and fine-tunes its detector in a self-supervised manner. Experiments on both indoor and outdoor datasets show that (1) our method obtains high quality 2D and 3D pseudo-labels from multi-view RGB-D data; (2) fine-tuning with these pseudo-labels improves the 2D detector significantly in the test environment; (3) training a 3D detector with our pseudo-labels outperforms a prior self-supervised method by a large margin; (4) given weak supervision, our method can generate better pseudo-labels for novel objects.

1 Introduction

For tasks that require high-level reasoning, intelligent systems must be able to recognize objects despite partial occlusions or uncommon poses. Humans and other mammals actively move their eyes, head, and body to obtain less occluded and more familiar viewpoints of the objects of interest [17, 43]. They then use familiar viewpoints to inform viewpoints they are less confident about. For example, to recognize an occluded object (such as the TV in Figure 1) from an unfamiliar viewpoint, an intelligent agent can increase its accuracy in this task simply by moving to a less occluded and more familiar viewpoint, and then mapping these confident beliefs of the object back to the unfamiliar views.

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

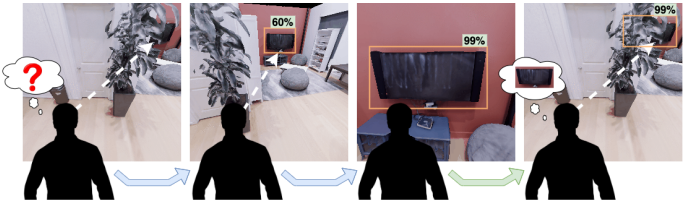

Figure 1: **Improving object recognition by moving.** An agent is viewing an object from an occluded, unfamiliar viewpoint. By moving to less occluded, more familiar viewpoints of the object (blue arrow), the agent can use the familiar viewpoints to self-supervise the previously unfamiliar viewpoints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

Significant improvements have been made in the accuracy and reliability of 2D and 3D visual recognition systems [29]. However, recent works [8, 33] show that current detectors are less likely to recognize an object correctly under occlusions and uncommon viewpoints. Advances in active visual learning [11, 12, 51] 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 either ground truth 3D segmentation of the environment or 2D human annotations of the images (in addition to a pre-trained detector).

In this work, we demonstrate that by obtaining a diverse set of views of an object and propagating confident detections across viewpoints, we can increase detector performance without requiring any additional human annotations for rigid objects. We propose a novel method for an embodied agent to improve its 2D and 3D object detection in test environments that is robust to realistic actuation noise. Our pipeline has three phases: *(1) Data collection*: randomly move in the environment to collect observations and occasional high-confidence detections, then plan paths to collect diverse posed RGB-D images of the detected objects; *(2) 3D segmentation*: segment the detected objects in 3D using aggregated RGB-D images, and then re-project that segmentation to form 2D pseudo-labels in all views; *(3) Detector improvement*: fine-tune the pre-trained detector on the pseudo-labels.

We show that fine-tuning with pseudo-labels generated by our method significantly improves the pre-trained detector in challenging indoor and outdoor datasets. Adding weak supervision further increases performance. We extend our self-supervised method to 3D detection where our model outperforms a state-of-the-art self-supervised 3D detection method by a large margin, while achieving performance comparable to a fully supervised model with the same architecture. We will be making our code and data publicly available.

2 Related Work

2D Object Recognition Deep networks achieve good performance on 2D object detection [38] and segmentation [22]. Data augmentation techniques [16, 55] have also shown promising results to enhance training with scarce annotations. However, recent works [8, 33] show that detectors are unlikely to correctly recognize an object from uncommon viewpoints. Augmentations fail to capture viewpoint invariances which are essential for 2D recognition systems. Banani et al. [7] proposed a fully supervised shape and descriptor network that takes as input two labelled views of an object and generates object mask for a novel view. Xiao et al. [50] trained a model to perform few-shot object detection and viewpoint estimation together, but they do not improve detection performance for viewpoints that are unfamiliar

to the detector. Our method bridges this gap by improving the detector in *new environments and viewpoints without additional supervision*.

3D Object Recognition Some methods use 3D voxel grids [37, 54]. PointNet [34, 35, 36] and SPGN [48] directly operates on unordered pointclouds for learning deep point set features applicable to object detection and segmentation. Later works [27, 47] integrate convolution into pointclouds. While those methods require 3D supervision, Armeni et al. [4] uses a pre-trained detector to build a 3D scene graph semi-automatically, and embeds its knowledge into the 3D scene representation akin to a SLAM method. Our method instead embeds its knowledge into detectors, which allows it to work even under settings where the scenes change over time. LDLS [46] performs 3D segmentation in a semi-supervised way, diffusing pre-trained detectors' predictions from RGB-D images onto the scene pointcloud. With our pseudo-labels we can train a 3D object detector [36] that outperforms LDLS and achieves performance comparable to the same detector trained on ground truth labels.

Active Visual Learning The problem of active vision [2, 5, 41] presents an agent with a large unlabelled set of images and asks the agent to select a subset for labelling, which will provide the maximal amount of information about the full dataset [39]. Psychology research suggests active vision as a natural method used by humans to attend to relevant visual features [6, 17, 43, 53]. This has been applied to object detection [23, 24, 45, 52], instance segmentation [32] and feature learning [1]. Chaplot *et al*. [12] explored a closely related setting, where a policy is trained to efficiently acquire data where detections are not multi-view consistent. In this work, we propose a self-supervised technique complementary to both directions, where we select "easy" viewpoints according to the confidence of a pre-trained detector, and propagate information from these viewpoints to more challenging ones. Additionally, this selection of "easy" viewpoints can be accomplished even with a simple navigation policy and random viewpoint sampling, and thus does not require additional learning of a navigation policy to improve detection performance.

Embodiment Embodied agents can move and interact with their environment through a physical apparatus and 3D simulators help model embodiment in a virtual setting. Many of the environments are photo-realistic reconstructions of indoor [8, 10, 42, 49] and outdoor [15, 19] scenes. These simulated environments have been used to study tasks such as visual navigation and exploration [11, 18, 20], visual question answering [14], tracking [21], and object recognition [12, 51]. In our work, we use a simulated embodied agent to discover objects and collect diverse posed data for fine-tuning a detector.

Pseudo-Label Generation A general paradigm in Pseudo-Label generation is to train a teacher network on few labelled examples and then fire the teacher network on a large unlabelled dataset to generate "pseudo" labels which can then be used to train a "student" network. Prior works typically generate pseudo labels lying in the same dimension space as the labelled examples consumed by the teacher network. Caine et al. [9] trained a teacher network on 3D labelled data to generate more 3D pseudo labels. Similarly, Chen et al. [13] generated 2D pseudo labels using a teacher network trained on 2D labelled data. In our setup, we use pre-trained MaskRCNN as teacher network to generate pseudo labels which are then used to finetune a 2D "student" MaskRCNN and 3D "student" Frustum Pointnet. Often careful engineering is required to select augmentations like in Li et al. [26] to generate reasonable quality pseudo labels. In contrast, our pseudo labels naturally targets to improve the detector on uncertain viewpoints and significantly improves detection performance with simple fine-tuning and standard augmentations.

3 Move to See Better

We propose a method for an embodied agent to improve its 2D and 3D object detection in unseen environments, assuming only a pre-trained 2D object detector, a depth sensor, and self-provided egomotion information. Most previous methods that attempt to improve detection of embodied agents [11, 12, 51] require either 2D or 3D human annotations after they have been collected by the embodied agent. Some of those methods train the movement of the agent to select specific viewpoints for later labelling [12, 51]. However, acquiring the annotations for the collected images still remains extremely expensive.

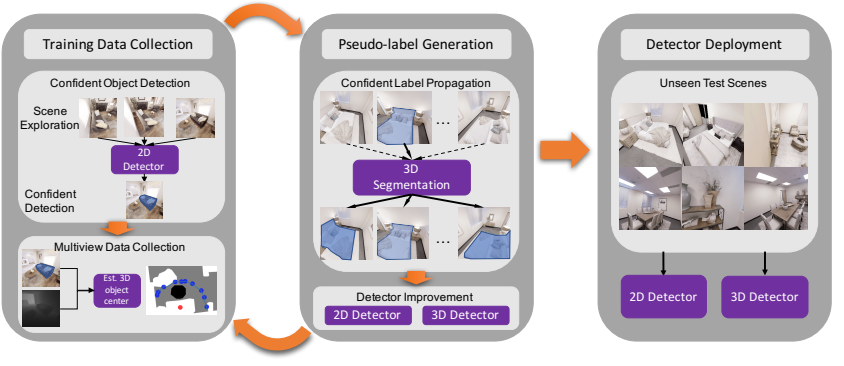

Figure 2: **Seeing by Moving (SbM).** Confident detections of a pre-trained 2D object detector guide self-supervised multi-view data collection and pseudo-label generation. Our 3D segmentation module segment the detected objects in 3D using aggregated RGB-D images. The 2D and 3D detectors fine-tuned on the pseudo-labels perform better on unseen test scenes.

We introduce a "Seeing by Moving" (SbM) framework, which removes the bottleneck of expensive human annotations, by driving the annotation process with the agent itself. An overview of our framework is shown in Figure 2. In the data collection stage, 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 serve as a cue to help us select good views of objects in the agent's environment. Although the confidence of a detector is not always well-calibrated with its accuracy, we are able to maintain high precision by setting a strict confidence threshold (see supp. for more details). In the pseudo-label generation stage, we propagate the high-confidence detections from "easy" views to "hard" views. We then fine-tune the 2D object detector using generated pseudo-labels. We demonstrate large improvements in both indoor and outdoor benchmarks. Additionally, we train a 3D detector from scratch using the pseudo-labels. Experimental results are discussed in Section 4.

3.1 Embodied Data Collection

The aim of our data collection policy is to capture a diverse set of viewpoints of the rigid objects present in the environment. Note that the datasets used for training deep object detection models most often capture objects from unoccluded and canonical viewpoints. In our data collection policy, we seek to obtain these unoccluded canonical viewpoints *as well as* other viewpoints where a pre-trained detector would be less certain. In our experiments,

it is possible to simply obtain a diverse set of viewpoints directly from the simulator, but we present a more general method here that could work in real-world scenarios as well.

We opted for a simple policy where the agent chooses an initial viewpoint randomly and navigates deterministically. While other methods of viewpoint selection and navigation could be used, we thought this data collection policy would be best for showcasing our proposed method. Most importantly, we wanted to highlight the fact that our method can be used without optimal target or viewpoint selection, and simply obtaining a diverse set of views of a scene is enough to improve detector performance with our method. Thus, an agent simply navigating to various locations around a room could apply our method to improve its detector. We believe active data collection could improve our method further, although we leave that for future work.

Navigation Policy We consider an embodied agent equipped with a pre-trained object detector, a depth sensor, and approximate egomotion information. Data collection proceeds in object-centric *episodes*. Episodes have two stages: localizing a random object and collecting N views (here 25). To localize a random object, the agent naively explores the scene with a random policy and runs the detector on every frame. When the detector returns a sufficiently confident detection (determined by a threshold), we proceed to collect additional views of that object. To collect views, the agent needs to navigate to positions at various viewing angles and distances from the object. We begin by estimating the 3D centroid of the object, using the predicted 2D object mask, the depth map, and the camera intrinsics. We then unproject (see section: 3.2) the depth map to construct a 3D occupancy map of the region, and use this map to sample a valid navigation location near the agent and within a distance from the object centroid, similar to Gupta *et al*. [20]. Given the occupancy map and goal location, we use a fast marching planner to reach the goal [40]. Once we reach the goal location, we use the object centroid and estimated pose to orient the viewing angle of the sensors so that the object is in view, and capture the sensor readings (i.e., the RGB-D image and the pose). We repeat this navigation and view-capture process until N views have been obtained. At the end of an episode, we navigate away from the target object, and restart the random localization process. We collect 30 such episodes per environment.

3.2 Multi-View Object Segmentation

After collecting N observations of an object from diverse viewpoints, our goal is to segment the object from its background. We first aggregate a colorized pointcloud of the region, by unprojecting each frame using its depth observation and pose, then segment the object from its background in 3D. Figure 3 shows an overview of this process.

View Aggregation For the i-th view, a 2D pixel coordinate (u,v) with depth z is unprojected and transformed to its coordinate $(X,Y,Z)^T$ in the reference frame:

$$(X,Y,Z,1) = \mathbf{G}_i^{-1}\left(z\frac{u-c_x}{f_x}, z\frac{v-c_y}{f_y}, z, 1\right)^T \tag{1}$$

where (f_x,f_y) and (c_x,c_y) are the focal lengths and center of the camera model and $\mathbf{G}_i \in SE(3)$ is the camera pose for view i relative to the reference view. This module first unprojects each RGB image $I_i \in \mathbb{R}^{H\times W\times 3}$ into a colored pointcloud in the reference frame $P_i \in \mathbb{R}^{M_i\times 6}$ with M_i being the number of pixels with an associated depth value. We concatenate the spatial location with the color, forming an aggregated colorized scene point cloud. The aggregated pointcloud P can be partitioned into three sets: the foreground set P_{fg}, the

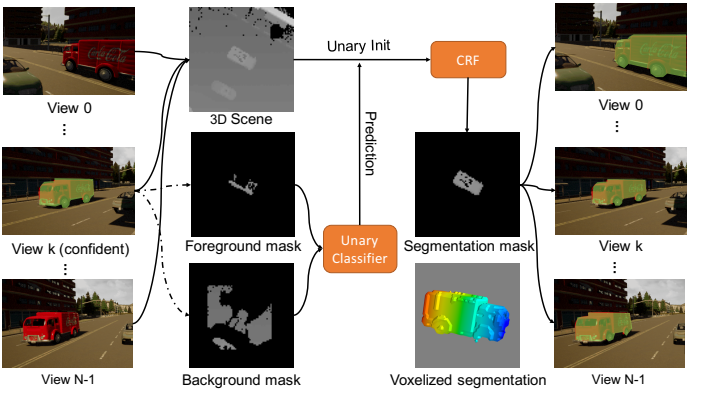

Figure 3: **3D Segmentation.** Images from N views and a segmentation mask from confident view k are unprojected into 3D using depth and pose. We sample foreground and background to train a unary classifier, whose outputs are used to initialize the unary potentials of the CRF model. The final 3D segmentation is then reprojected to all views to obtain pseudo-labels.

background set P_{bg}, and the unknown set P_{unk}: $P = \bigcup_{i=0}^{N-1} P_i = P_{fg} \cup P_{bg} \cup P_{unk}$. The per-view foreground/background masks of the detector provide P_{fg} and P_{bg}; the rest of the points form P_{unk}. To account for mis-classified points near the boundary, we find it helpful to perform a morphological erosion the predicted object mask to form the foreground mask. Similarly, we apply a morphological dilation on the inverse of the object mask to form the 2D background mask.

3D Segmentation To label points in P_{unk}, we apply a simple yet effective two-stage segmentation method. In the first stage, based on the information available in P_{fg} and P_{bg}, we train a unary model to classify whether a point belongs to the foreground or background. We then ask the unary classifier to predict the log probability of *all* points, including the unlabelled ones in P_{unk}. In the second stage, we apply a fully connected conditional random field (CRF) [23] to refine the segmentation. Each node in the CRF model is a point in P and its unary potential is initialized with the log probabilities from the unary classifier. To inject spatial information into the CRF model, we add a pairwise potential between every point pair $(P^{(a)}, P^{(b)})$ in P: $\psi_p\left(P^{(a)}, P^{(b)}\right) = \mu\left(X^{(a)}, X^{(b)}\right) k(P^{(a)}, P^{(b)})$, where $X^{(a)}$ and $X^{(b)}$ are the first three dimensions of $P^{(a)}$ and $P^{(b)}$ that corresponds to their spatial locations, μ is the label compatibility function from the Potts Model, and k is a contrast-sensitive potential given by the combination of an appearance kernel and a smoothness kernel (both parameterized by a Gaussian kernel). In our experiments, we use a support vector machine (SVM) as the unary classifier for its efficiency. We additionally experimented with a "deep" method for this task, but found this simple two-stage "shallow" method to be superior in both accuracy and runtime (see supp.). The output of this process is a high-quality 3D pointcloud segmentation P_{seg}, distinguishing the object from its local background.

3.3 Supervising with Self-Generated Labels

After generating 3D pointcloud segmentations of the found objects, we distill this knowledge into the weights of a neural network. To do this, we treat the estimated 3D segmentations as

pseudo-labels, and supervise standard deep architectures to mimic the labels.

3D Detection Training We compute 3D boxes to encapsulate the 3D segmentations, to match the training format of modern 3D object detectors. We then train a standard object detector on this self-supervised labelled data from scratch. Our experiments show that this simple self-supervision scheme outperforms a state-of-the-art self-supervised 3D detection method by a large margin and achieves performance comparable to a supervised detector with the same architecture.

2D Detection Training We produce 2D pseudo-labels by re-projecting P_{seg} to all views. For a point $(X,Y,Z)^T$ in P_{seg}, we can get its 2D pixel coordinate in the i-th frame with: $(u,v)^T = \mathbf{G}_i \left(f_x \frac{X}{Z} + c_x, f_y \frac{Y}{Z} + c_y\right)$. The reprojected points in P_{seg} are sparse in 2D, so we fit a concave hull to convert them into a connected binary mask. Our experiments show that these pseudo-labels provide a significant boost in performance to a pre-trained detector.

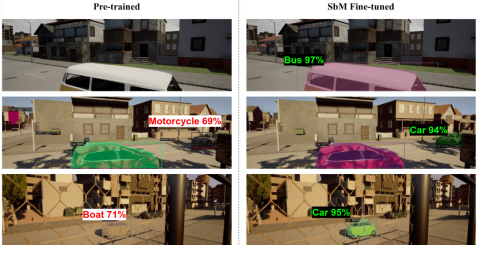

Figure 4: **Visualizations of 2D detector performance on the CARLA test set.** We show qualitative examples of the detections of pre-trained detector (left) and SbM fine-tuned detector (right). The improvements are shown in larger fonts for better visibility.

mAP@IoU	Method	Train	Test
0.5	Pre-trained	68.05	68.23
	SbM Labels	86.88	81.81
	SbM fine-tuned	-	80.15
	GT fine-tuned	-	93.76
0.3	Pre-trained	73.09	75.55
	SbM Labels	92.93	92.49
	SbM fine-tuned	-	88.84
	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.

4 Experiments

4.1 Datasets

Environments We test our method in an indoor and outdoor environment. We use the CARLA simulator [15] as the outdoor environment, which renders realistic urban driving scenes. We use the Habitat simulator [30] with the Replica dataset [42] as the indoor environment, which contains high quality and realistic reconstructions of indoor spaces. 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 10 for training, 4 for validation, and 4 for testing. In our self-supervised data collection, we capture 25 views in each episode, resulting in 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 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 randomly spawn two vehicles in the scene for each episode. Since CARLA has the same semantic label for all vehicles, we consider detection of all

vehicle classes in the COCO dataset [28] during evaluation. For Replica, we keep the default layout of objects in each scene. We consider a subset of object categories based on the following standards: (1) the category is shared between COCO and Replica, and 2) enough instances (more than 10) occur in the dataset. This includes chair, couch, plant, tv and bed.

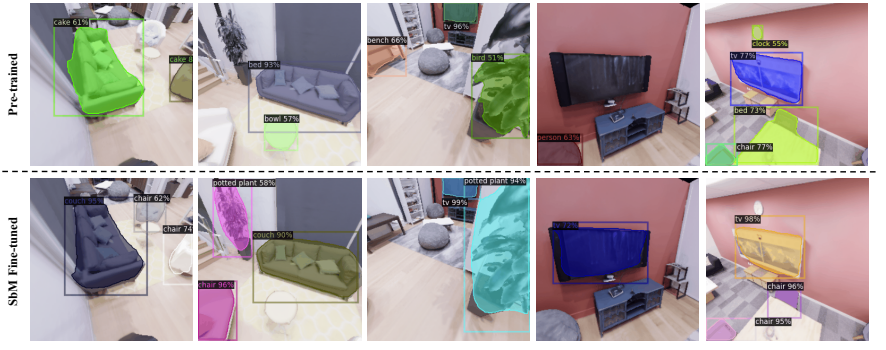

Figure 5: **Visualizations of 2D detection on Replica test set.** We show qualitative results from the pre-trained 2D detector (top) and the SbM fine-tuned 2D detector (ours, bottom).

4.2 2D Object Detection

We analyze our method for improving 2D object detection by asking: (1) do our pseudo-labels improve performance over the detector on which they are based? (2) does fine-tuning the object detector on the pseudo-labels improve detection performance on unobserved scenes? Experiments in CARLA and Replica show that the answer to both is "yes".

CARLA The performance of our method, the pre-trained detector, and the detector fine-tuned on ground truth data is shown in Table 3.3. We report mAP at IoU of 0.5 and 0.3 using the PascalVOC setup [11]. 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. Pseudo-labels generated by SbM have much better performance than the pre-trained detector outputs. This shows that moving and propogating information across viewpoints improves the detector, when compared to treating multi-view images as individual observations. We also fine-tune the detector with the SbM pseudo-labels generated from the training set. At test time, the SbM fine-tuned model is deployed in unseen towns where only a single RGB image is given as input. Results show that the fine-tuned detector outperforms the pre-trained detector by a large margin. We emphasize that this is accomplished with no additional human labels required. Figure 3.3 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 pseudo-labels on the training set is shown in Table 2. Our pseudo-labels are more accurate than the pre-trained detector on most classes, indicating that moving around helps generate better labels. 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 a large margin. In Figure 5, 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 can help training a better detec-

mAP@IoU	Method	Bed	Chair	Couch	Table	Plant	TV	Avg
0.5	Pre-trained	**7.50**	11.08	17.20	**7.09**	20.44	46.79	18.35
	SbM (ours)	6.08	**21.41**	**39.67**	4.12	**27.15**	**58.78**	**26.20**
	SbM-ws (ours)	7.34	40.53	58.33	38.33	64.68	58.23	44.57
0.3	Pre-trained	8.12	13.18	17.97	7.41	48.28	46.79	23.62
	SbM (ours)	**10.04**	**31.76**	**45.63**	**8.30**	**66.99**	**66.04**	**38.12**
	SbM-ws (ours)	39.03	58.93	82.37	59.74	82.85	75.87	66.47

Table 2: **2D object detection performance of the pre-trained detector vs self-supervised SbM vs weakly supervised SbM on the Replica *training set*.** Self-Supervised SbM consistently outperforms the pre-trained detector across most categories. Weak supervision (a single-view ground truth annotation) increases performance of SbM on all categories.

mAP@IoU	Method	Bed	Chair	Couch	Table	Plant	TV	Avg
0.5	Pre-trained	**15.18**	21.51	**23.54**	2.37	11.74	43.71	19.67
	SbM Fine-tuned (ours)	5.57	**36.19**	18.86	**8.50**	**37.34**	**57.85**	**27.38**
	GT Fine-tuned	27.20	53.56	48.65	26.99	35.04	58.28	41.62
0.3	Pre-trained	**27.71**	22.95	**25.83**	2.80	19.79	43.71	23.79
	SbM Fine-tuned (ours)	10.55	**45.60**	21.17	**8.82**	**40.80**	**57.85**	**30.79**
	GT Fine-tuned	38.25	60.15	52.84	28.65	42.59	58.28	46.79

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

tor without requiring additional ground truth. In addition, we show in supplementary that this improvement over the baseline is maintained even under actuation noise modeled by a real robot. We further experiment to test whether we can generate higher-quality labels if provided very weak supervision instead of no supervision. For this, we only provide ground truth annotation for one view out of the 25 available views for each object instance. We denote this setup as SbM-ws. We report the label quality on the training set in Table 2. We observe that with a single labelled view, the pseudo-label quality is better than both the pre-trained detector and self-supervised SbM by a large margin. This suggests that our method can generate much better pseudo-labels with an improved pre-trained detector. In supplementary, we show applicability of this method to generate high quality labels for novel objects that are not part of COCO.

4.3 3D Object Detection

Can we train a 3D object detector self-supervised without requiring any 3d annotations? To answer this question, we compare the two versions of frustum PointNet: one trained on SbM's self-supervised 3D and 2D labels (Figure 4.2), and the other trained on ground truth 3D and 2D labels. We also compare our method with the semi-supervised LDLS [46] method. The experiments are conducted in CARLA.

Table 4.2 shows the test set performance of LDLS [46], frustum PointNet trained on SbM segmentations, and frustum PointNet trained on ground truth. Our self-supervised frustum

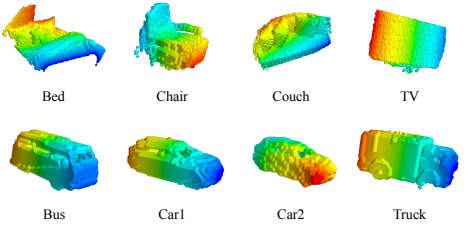

Figure 6: **Visualizations of 3D object segmentation.** We show colorized visualisations of the voxelized 3D segmentations of our method, on Replica (top) and CARLA (bottom).

Method	mAP
LDLS [46]	44.03
SbM Self-Sup. F-PointNet (ours)	**83.87**
Supervised F-PointNet	85.06

Table 4: **Fine-tuning with SbM labels outperforms self-supervised LDLS.** 3D object detection performance of LDLS [46], frustum PointNet trained on SbM segmentations, and GT-trained frustum PointNet on the CARLA test set at IoU@0.25.

PointNet model outperforms LDLS significantly. Our model also achieves close performance to the fully supervised model. We show qualitative examples of the 3D detections from LDLS and SbM fine-tuned frustum PointNet in Supplementary. This demonstrates that the 3D segmentation labels produced by SbM are high quality and could be successfully used to train 3D detection models without ground truth 3D annotations.

5 Conclusion

While visual recognition systems trained on large internet data have shown great advancements, they still require a lot of additional human annotations to work well on novel domains, unusual poses, or heavy occlusion conditions. Motivated by how humans learn, we utilize an active agent that can move in the environment, discover objects, and generate its own pseudo-labels for self-supervision. In both indoor and outdoor settings, we show that our method significantly improves the performance of a pre-trained 2D detector in test environments for rigid objects. Moreover, we show that our method can be used to train a 3D detector without any human-provided 3D annotations. Our experiments with simple exploration policies and realistic actuation noise show promising results for real-world conditions.

We believe that active visual learning remains an important problem for future work. We note several limitations of our method: (1) our method assumes that the pre-trained detector makes correct high-confidence predictions for at least some of the available views and the experiments contain limited number of object classes; it may be helpful to use contextual or common-sense cues to ensure accurate, highly confident defections occur in some views for long-tail classes [44]; (2) for simplicity, we used a random exploration policy in our method; recent works on active exploration [11, 12] can be used as the high level exploration policy to make the data collection more efficient; (3) our method isn't specifically designed to handle more challenging object categories, such as objects with complex articulated parts; (4) finally, applying this method on a real robot is a direct avenue for future research.

6 Acknowledgements

This work has been funded by Sony AI, DARPA Machine Common Sense, a NSF CAREER award, the Air Force Office of Scientific Research under award number FA9550-20-1-0423, and the National Science Foundation Graduate Research Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force. The authors declare no competing interests. The authors would also like to thank Hsiao-Yu Fish Tung for helpful discussions.

References

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

[2] John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. Active vision. *IJCV*, 1(4): 333–356, 1988. doi: 10.1007/BF00133571. URL https://doi.org/10.1007/BF00133571.

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

[4] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *ICCV*, 2019.

[5] Dana H. Ballard. Animate vision. *Artif. Intell.*, 48(1):57–86, February 1991. ISSN 0004-3702. doi: 10.1016/0004-3702(91)90080-4. URL https://doi.org/10.1016/0004-3702(91)90080-4.

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

[7] Mohamed El Banani, Jason J Corso, and David F Fouhey. Novel object viewpoint estimation through reconstruction alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3113–3122, 2020.

[8] 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.

[9] Benjamin Caine, Rebecca Roelofs, Vijay Vasudevan, Jiquan Ngiam, Yuning Chai, Zhifeng Chen, and Jonathon Shlens. Pseudo-labeling for scalable 3d object detection. *arXiv preprint arXiv:2103.02093*, 2021.

[10] 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.

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

[12] DS Chaplot, H Jiang, S Gupta, and A Gupta. Semantic curiosity for active visual learning. In *ECCV*, 2020.

[13] Nicholas FY Chen. Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under ego-motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 644–653, 2018.

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

[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.

[16] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *ICCV*, 2017.

[17] 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.

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

[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.

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

[21] 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.

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

[23] 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. doi: 10.1109/TPAMI.2018.2840991.

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

[25] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*, 2011. ISBN 9781618395993.

[26] Duo Li, Sanli Tang, Zhanzhan Cheng, Shiliang Pu, Yi Niu, Wenming Tan, Fei Wu, and Xiaokang Yang. Rethinking pseudo-labeled sample mining for semi-supervised object detection. 2020.

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

[28] 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.

[29] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jiechen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *IJCV*, 2020.

[30] 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.

[31] R. Padilla, S. L. Netto, and E. A. B. da Silva. A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 237–242, 2020.

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

[33] Senthil Purushwalkam and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. *arXiv preprint arXiv:2007.13916*, 2020.

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

[35] 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.

[36] 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.

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

[38] 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.

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

[40] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[41] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. URL http://axon.cs.byu.edu/~martinez/classes/778/Papers/settles.activelearning.pdf.

[42] 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.

[43] 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.

[44] Antonio Torralba. Contextual priming for object detection. *International journal of computer vision*, 53(2):169–191, 2003.

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

[46] 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. doi: 10.1109/LRA.2019.2922582.

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

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

[49] 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.

[50] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *European Conference on Computer Vision*, pages 192–210. Springer, 2020.

[51] J Yang, Z Ren, M Xu, X Chen, DJ Crandall, D Parikh, and D Batra. Embodied amodal recognition: Learning to move to perceive objects. In *ICCV*, 2019.

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

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

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

[55] Barret Zoph, Ekin D. Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V. Le. Learning data augmentation strategies for object detection. In *ECCV*, 2020.