Labeling Indoor Scenes with Fusion of Out-of-the-Box Perception Models

Yimeng Li[1*], Navid Rajabi[1*], Sulabh Shrestha[1], Reza Alimoor[2], and Jana Košecká[1]

[1]George Mason University [2]Drake University

{yli44, nrajabi, sshres2, kosecka}@gmu.edu, md.reza@drake.edu

Abstract

The image annotation stage is a critical and often the most time-consuming part required for training and evaluating object detection and semantic segmentation models. Deployment of the existing models in novel environments often requires detecting novel semantic classes not present in the training data. Furthermore, indoor scenes contain significant viewpoint variations, which need to be handled properly by trained perception models. We propose to leverage the recent advancements in state-of-the-art models for bottom-up segmentation (SAM), object detection (Detic), and semantic segmentation (MaskFormer), all trained on large-scale datasets. We aim to develop a cost-effective labeling approach to obtain pseudo-labels for semantic segmentation and object instance detection in indoor environments, with the ultimate goal of facilitating the training of lightweight models for various downstream tasks. We also propose a multi-view labeling fusion stage, which considers the setting where multiple views of the scenes are available and can be used to identify and rectify single-view inconsistencies. We demonstrate the effectiveness of the proposed approach on the Active Vision dataset [1] and the ADE20K dataset [27]. We evaluate the quality of our labeling process by comparing it with human annotations. Also, we demonstrate the effectiveness of the obtained labels in downstream tasks such as object goal navigation and part discovery. In the context of object goal navigation, we depict enhanced performance using this fusion approach compared to a zero-shot baseline that utilizes large monolithic vision-language pre-trained models.

1. Introduction

In recent years, there has been an increased interest in the design and evaluation of embodied agents (e.g., mobile robots) that operate in indoor environments such as restaurants, households, and hospitals. These embodied agents often need to perform tasks such as object goal navigation
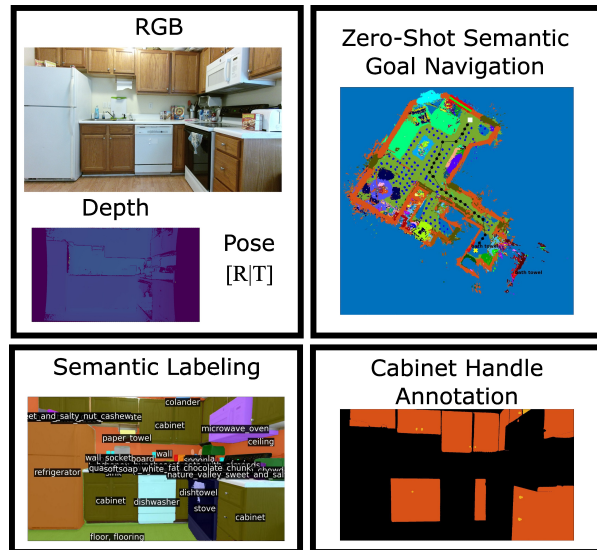
---
*Equal contribution

Figure 1. **This diagram gives an overview of our contributions.** Starting from an RGB-D dataset, we propose a labeling approach for semantic segmentation annotations. On top of the semantic segmentation results, we additionally proposed two downstream tasks for robot navigation. We build top-down-view semantic maps and use them for zero-shot semantic-goal navigation. We proposed an object part segmentation task for the 'cabinet handle' related to the robot mobile manipulation task.

or fetch and delivery tasks, both of which require navigating to objects specific by natural language. Towards this end, various indoor scene datasets of real-world interiors have been recently used [1, 21, 23]. The perception in these tasks relies on the ability of the agent to detect objects of interest. Obtaining dense semantic annotation of indoor scenes is challenging and time-consuming; The tasks of instance segmentation and semantic segmentation require pixel-level accuracy. For example, it has been reported that labeling HM3D dataset [23] took over 14,200 hours of human effort for annotation and verification by 20+ annotators. Since the indoor environments have a more diverse set of objects compared to the autonomous driving datasets, choosing the object category from over 1K object categories adds to the

complexity of the task.

In this paper, we explore the effectiveness of fusing the predictions from existing state-of-the-art models for bottom-up class-agnostic image segmentation [11], object detection [28] and semantic segmentation [3] to obtain instance and semantic segmentation. While the class agnostic image segmentation model SAM [11] produces accurate boundaries, it requires either a good bounding box hypothesis for the classes or a dense point initialization of the input image to generate segmentations. Hence, we exploit the accurate boundaries of SAM, jointly with the object detection model [28] and the semantic segmentation model [3] to get labels and masks for foreground and background classes respectively. We demonstrate the effectiveness of this labeling approach by evaluating the resulting labels using human-annotated labels and through the performance of agents that utilize our approach for various downstream tasks. Our contributions are as follows:

- We design a labeling approach that fuses predictions from the state-of-the-art semantic segmentation and object detection models to obtain semantic labels for the class-agnostic image segmentation.

- We further refine the single-view labeling by enforcing multi-view semantic consistency on Active Vision Dataset [1] (densely sampled multi-view dataset of indoor scenes) and evaluate the effectiveness of our approach by directly comparing it to the labels obtained with human annotations.[1]

- We demonstrate the effectiveness of our approach in evaluating the performance of the object goal navigation task and part discovery on the AVD dataset, which utilizes the fused segmentation results.

2. Related Work

Label Propagation. The idea of propagating labels between different views of the environment has been used extensively as a means of obtaining additional training examples for either contrastive learning or fine-tuning existing models semantic segmentation models. Label propagation has been used for scene completion in indoor scenes [19,26] and in videos [2, 19]. labeling pixels covering a range of categories from small objects to large furniture and backgrounds. Authors in [26] model the 3D environment implicitly as a neural radiance field. Label propagation has also been used for gathering more training data either by directly using the available motion information [6] or by training models to predict motion [29]. However, the approaches that utilize label propagation in one form or another either assume that at least a few human-annotated labeled frames are available, keep the number of labels fixed

and relatively small, and strive to train or fine-tune a single model.

Domain Adaptation. Additional labels have been obtained in the past by training a model in a source domain of a particular environment (e.g., indoor) and then adapting the trained model to another domain of the same environment [9]. While this does increase the performance of the model for the known classes, the classes novel to the model are missed entirely. Most, if not all, unsupervised or self-supervised domain adaptation approaches to target the autonomous driving domain, which has smaller view-point variations and less challenging occlusions [22, 24, 25] and does not utilize association between multiple views from the environment.

Open-Vocabulary Object Detection. The powerful large multi-modal vision and language models [18] have been used effectively for semantic segmentation [13] or as open vocabulary object detectors in zero-shot setting in multiple recent works on object goal naviation [4, 7, 10, 17]. These approaches mainly leverage (1) open-vocabulary (as opposed to the traditional fixed-category methods), (2) joint embedding space (for vision and language representations), and (3) large-scale pre-training on 400 million images of models like CLIP. For instance, CLIP-Nav [4] has applied this technique to a variation of Vision-and-Language Navigation (VLN) task, while CLIP on Wheels (CoW) [7] approach introduces a model for zero-shot semantic goal navigation. It combines classical frontier exploration with the generation of CLIP heatmaps, achieved by calculating the dot product between egocentric visual embeddings and target object text embeddings. Upon detecting the target object during exploration, the approach projects its pixels onto a map, facilitating straightforward robot navigation toward the goal.

While zero-shot navigation-based approaches have been aiming to eliminate the need for domain adaptation and enable open-vocabulary navigation, there is still significant room for improvement of observed poor zero-shot performances. We think a fundamental issue is the reliance on dot-product scores when comparing the holistic vision and language-embedded representations in joint high-dimensional embedding space. Due to the nature of the representations these models learn, it is difficult to scrutinize the details of concepts/categories covered during pre-training and the frequency of those samples. We hypothesize that fusing the predictions from the state-of-the-art perception models and building a richer semantic map, including long-tail object instance predictions, could be more effective in downstream tasks as a trade-off between the fixed-category and zero-shot open-vocabulary approaches.

---

[1]The resulting annotated dataset will be made publicly available.

3. Preliminaries

We briefly review three main components of our labeling approach.

Segment Anything Model (SAM) [11] SAM is a Vision Transformer (ViT) based model trained extensively on a very large dataset of 1 billion masks from their SA-1B dataset. SAM accommodates diverse input prompts, including points, bounding boxes, and dense point grids, and returns boundaries corresponding to the prompts. Such prompts provide a coarse estimation of what to segment in an image, which enables SAM to undertake a wide range of segmentation tasks without the need for additional training. A notable strength of SAM is its innate grasp of what constitutes an object, which enables zero-shot generalization to unfamiliar objects that are commonplace in indoor environments.

Object Detector (Detic) [28] The Detic model is an object detector that can efficiently identify 21,000 object classes, including many previously considered challenging. Leveraging the idea of training object detectors in a Weakly-Supervised manner, Detic is trained on the ImageNet-21K dataset and sidesteps the traditional reliance on assigning labels to bounding boxes. Using text embeddings from CLIP [18] and a classifier based on concept recognition, Detic recognizes objects beyond predefined categories, enhancing its adaptability. The model accommodates various categories from various datasets such as COCO [15], OpenImages [20], and LVIS [8]. Because of its use of CLIP text embeddings for class specification, the model can also be adapted to incorporate a custom vocabulary.

MaskFormer [3] The MaskFormer model presents a novel approach to tackling semantic and instance segmentation tasks using a unified mask classification framework. MaskFormer employs a Transformer decoder to compute a set of pairs comprising class predictions and mask embeddings, with the mask embeddings used to compute binary mask predictions. The model excels in handling datasets with diverse categories. It achieves a new state-of-the-art performance on the ADE20K dataset, outperforming per-pixel models with similar backbones.

4. Labeling Approach

The proposed labeling approach is tailored for indoor scene datasets featuring high-resolution images of cluttered environments, including diverse object types. The single-view labeling stage leverages the open-vocabulary object detection Detic [28], the state-of-the-art semantic segmentation model MaskFormer [3], and the foundational segment anything model SAM [11] to annotate all the images with pixel-wise labels. When presented with depth images and corresponding camera poses, the integration of the multi-view verification stage can further enhance the results

through the merging of single-view stage results. The stages are explained in detail below.

4.1. Single-view Labeling Stage

The initial stage of our labeling pipeline involves labeling single views from the environment independent of the other views. This stage encompasses three distinct branches that can be processed separately: (i) Generate masks for the entire image utilizing the semantic segmentation model (MaskFormer) and SAM; (ii) Generate masks for foreground classes utilizing object detection model (Detic) and SAM; (iii) If manual bounding boxes are available, generate corresponding masks utilizing SAM.

Figure 3. An example of labeling using Semantic Segmentation (MaskFormer) and SAM. MaskFormer produces good predictions for background classes but not so well for foreground classes: the coffee machine is misclassified as a 'stove' (black bounding box), and the cooking pot (cyan bounding box) is missed entirely.

Semantic Segmentation with SAM We prompt SAM with dense point grids in the input image. This operation yields a collection of non-overlapping class agnostic segments. We run MaskFormer (trained on ADE20K [3]) to get the semantic segmentation of the image. We then assign labels to each segment generated by SAM by voting. Specifically, each segment gets votes per class based on the number of pixels that lie inside the segment and belong to the class based on MaskFormer results. The segment is assigned the class label with the highest vote. This step is illustrated in Fig. 2 (F). This step produces good results for background classes but misses a lot of foreground classes, as shown in Fig. 3. Hence, we opt to use an object detector for foreground classes.

Object Instance Detection and Segmentation with SAM Semantic segmentation models usually perform well on background classes but may lack the same accuracy for foreground classes. Furthermore, they may be trained with a limited vocabulary. For example, MaskFormer is trained on ADE20K, which lacks many common indoor objects
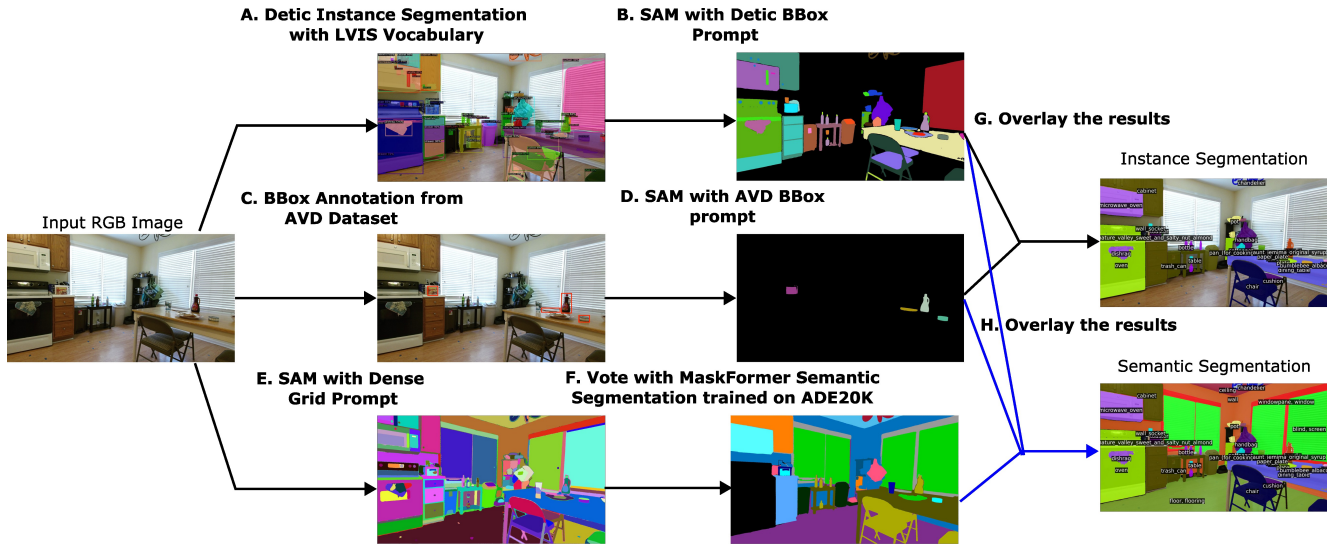
Figure 2. **This diagram gives an overview of our labeling approach at the single-view labeling stage.** Given an input image, we generate masks for foreground classes utilizing Detic and SAM at steps (A) and (B). If any classes with manual bounding boxes are available, we generate masks utilizing SAM at steps (C) and (D). We generate masks for the entire image utilizing MaskFormer and SAM at steps (E) and (F). We overlay the results of foreground class masks on top of the entire image's mask and achieve the semantic segmentation and instance segmentation annotations.

Figure 4. An example of labeling using Object Detector (Detic) and SAM.

like 'a cooking pot' and 'knife'. To address these shortcomings, we employ Detic with LVIS vocabulary, which includes 1203 classes and covers many known indoor object classes. However, the jointly predicted object bounding boxes and masks from Detic often do not have the desired level of accuracy, especially when compared with SAM results, as shown in Fig.4. This is due to the fact that Detic is partially trained on ImageNet, a dataset without mask annotations. We utilize SAM to generate high-quality masks by using the bounding boxes as input prompts. In practice, we prompt SAM with bounding boxes and a point corresponding to the centroid of the masks from Detic, leading to high-quality object instance segmentation shown in Fig. 4 and Fig.2 (B).

We select the background classes *floor*, *ceiling*, *door*, *blind*, and *wall* from the semantic segmentation model and discard other classes. The final result is obtained by superimposing foreground class masks, obtained from the use of object detection and SAM, on top of semantic segmentation

results, achieving a comprehensive and refined final representation.

4.2. Multiview Verification Stage

The single-view annotation may produce incorrect predictions due to model prediction errors, especially in challenging scenarios such as partial occlusion of objects or irregular viewing angles. With the assumption that models perform well in most scenarios and that objects are less occluded, at least in some viewpoints, results from such views can rectify these prediction errors. Hence, we fuse masks from multiple views using a per-class voting approach.

Consider an image I_k and its corresponding single-frame annotation A_k that requires label verification. We designate two keyframes, namely I_m and I_n, with their respective single-view annotations A_m and A_n, respectively, which serve as reference frames for I_k. In practice, the selection of I_m and I_n is guided by their proximity to the viewpoint of I_k, thereby ensuring contextual relevance. It's worth noting that all frames apart from I_k can be utilized as reference images in this process, provided they overlap with frame I_k.

We begin by projecting the annotations A_m and A_n into a 3D spatial context, utilizing the respective image poses $[R_m|T_m]$ and $[R_n|T_n]$ and their depth maps. The projections yield the corresponding 3D point clouds P_m and P_n, respectively. Each point within P_m and P_n has an associated semantic label from A_m and A_n. Using a 4-way connected component in A_k, we get the set of regions Reg_k in I_k. Subsequently, we traverse each region in Reg_k within

A_k, treating each region as a superpixel, similar to the approach outlined in [19]. We then project P_m and P_n onto the I_k frame using its known camera pose $[R_k|T_k]$. To ensure the correctness of the projected points, we filter out erroneous projections by cross-checking the projected point's depth value with the depth of the original pixel.

For each region in Reg_k, we calculate the votes for each class c_j using projected pixels from the two reference point clouds P_m and P_n. First, we calculate the number of pixels that were projected from P_m to this region that has the class c_j. We normalize this value using the total number of pixels in the region to get the normalized score $f_m^{c_j}$ for class c_j. We perform similar steps to get score $f_n^{c_j}$ using the other reference point cloud P_n. The score $f_k^{c_j}$ is set to 1 because the region is obtained using the class labels in A_k itself. The final vote for class c_j in the region is obtained by taking the average of the scores $f_m^{c_j}$, $f_n^{c_j}$ and $f_k^{c_j}$. The region is assigned the class which has the highest vote.

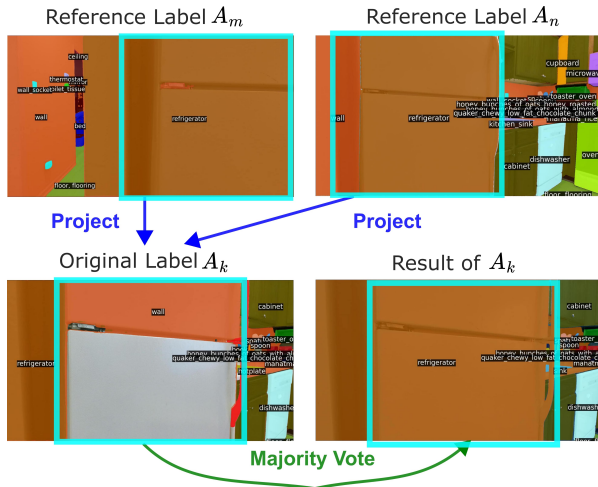Fig. 5 illustrates the multiview verification process.

Figure 5. An example of Multiview Verification. The refrigerator (cyan bounding box) in view A_k is originally labeled as wall and missing mask for some of its parts. The error is resolved by fusing labels from views A_m and A_n with the correct annotation for the class.

5. Downstream Tasks

We next evaluate the performance of the obtained instance segmentation and semantic segmentation on two downstream tasks: part discovery and object goal navigation.

5.1. Object Part Discovery

Part discovery is important for mobile manipulation tasks such as opening doors, cabinets, bottles, or adjusting thermostats. Since the shape of hinges and handles can differ significantly across different houses, we describe a simple approach for detecting these parts in a particular environment. We demonstrate this approach to the detection of cabinet handles. We use SAM to generate an over-segmentation of the image, followed by cabinet detection. SAM's segmentation typically separates the cabinet into three distinct regions: the handle, the drawer, and the cabinet's main body. Focusing on the segments within the predicted 'cabinet' bounding boxes from Detic, We first extract the ResNet-50 feature map of the entire image and extract the feature vector for each cabinet segment using RoIAlign. This is followed by k-means clustering with these features and manually identifying the cluster containing points corresponding to cabinet handles. We project data points within this cluster back to the images, thereby identifying segments of cabinet handles. In practice, we apply k-means clustering to each scene individually. We use 3 to 5 clusters considering the cabinet's appearance in the environment.

Fig. 6 illustrates the object part discovery approach.

5.2. Object Goal Navigation

Next, we evaluate the effectiveness of obtained semantic labels for the semantic object goal navigation task. More specifically, we focus on long-tail object instances and compare them to the state-of-the-art navigation approaches that use CLIP joint embedding space of large multimodal vision-language models [18] to localize objects in images and maps like VLMaps [4, 7, 10].

We first build a top-down semantic map for each scene of AVD using the semantic segmentation labels. The semantic map is a metric map of size $m \times m$ where each cell's value is in the range $[0, N]$. Each value from 1 to N corresponds to one of the N object categories, and 0 refers to the undetected class. Each cell is a $5cm \times 5cm$ region in the real world. Given an RGB-D view, we build the semantic map by projecting semantic segmentation images to a 3D point cloud using the available depth maps, and the robot poses, discretizing the point cloud into a voxel grid and taking the top-down view of the voxel grid. The semantic map depends on the majority category of the points located at the top grid of each cell. We summarize the semantic categories that exist in the scene by going through each semantic category and localizing the corresponding cells on the map.

For each AVD scene, we collect a semantic goal to evaluate object navigation. Then, we randomly specify a starting position of the robot in the scene and then pick a target object to navigate to. We consider the navigation to the target successful when the agent reaches the nearest navigable pose of the target object. (see Section 6.3 for experiment details)
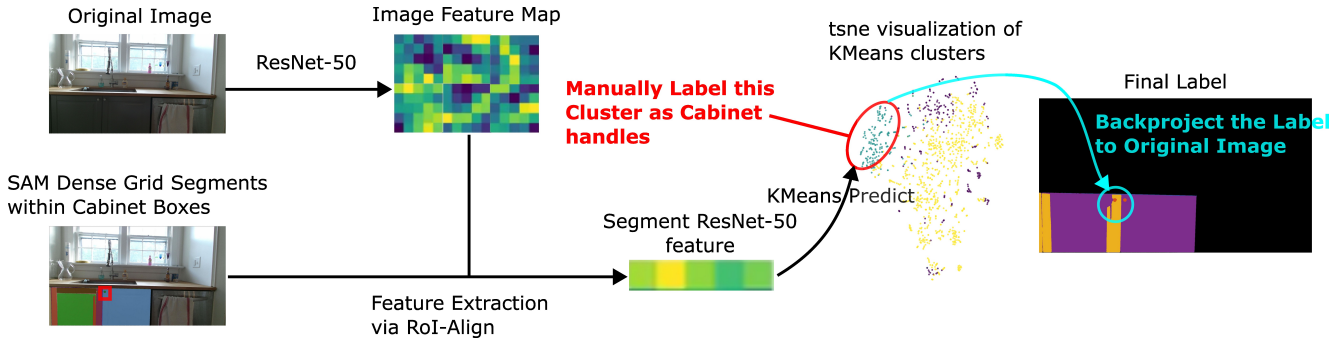
Figure 6. **This diagram gives an overview of the labeling cabinet handle.** We choose SAM segments within the detected 'cabinet handle' bounding box. Then, we extract ResNet50 features for these segments. We cluster these feature points through KMeans and manually label the cluster containing 'cabinet handle' points. We backproject the label to the original image to get the 'cabinet handle' annotations.

6. Experiments

Datasets. We use the Active Vision Dataset (AVD) [1] and the ADE20K [27] dataset for our experiments. We demonstrate and evaluate our labeling approach on AVD and ADE20K, while we only use AVD for downstream tasks.

AVD is comprised of 20 distinct environments that cover a variety of realistic indoor scenes, ranging from kitchens and offices to living rooms and bedrooms. We use the AVD dataset as a basis for our labeling effort because it captures real-world indoor scenarios characterized by intricate clutter and the presence of open-set objects, posing significant challenges to existing perception models. AVD also contains realistic depth images captured by Kinect V2, and the poses associated with the observations are available, making it an ideal dataset for downstream robotics applications. Furthermore, AVD has high-resolution RGB images, making annotation of fine-grained objects more feasible. The semantic categories of AVD are consistent with ADE20K [27] and LVIS [8] datasets. To evaluate the labeling approach, we manually labeled the first ten images of each scene with the instance and semantic segmentation. In total, we labeled 200 images for 20 scenes. We name this annotated subset of the dataset AVD-GT.

ADE20K has room-type labels for each image. Since the proposed approaches are tailored for indoor scenes, owing to the classes selected, we only select images labeled as *bathroom*, *bedroom*, *kitchen*, *living room*, *office*, *dining room*, *hotel room*, *dorm room*, *home office* or *waiting room*. We used 440 indoor scene images from the ADE20K for the evaluation. We name this subset of the dataset ADE20K-indoor.

Implementation Details. For the single-view annotation process described in Section 4.1, we set the parameters of the SAM model, namely the mask IoU threshold and the mask prediction stability, to 0.86 and 0.92, respectively. Considering AVD's high-resolution RGB images and the

abundance of small objects within each frame, we prompt SAM with a densely spaced point grid comprising 64×64 in the input image. The complete labeling is carried out on the entire AVD dataset and takes approximately 120 hours, utilizing two A100 GPUs.

6.1. Labeling

We compare the labeling results with human-labeled ground truths. Originally, the AVD dataset has approximately 20,000 RGB-D images and a total of approximately 50,000 bounding boxes. We provide semantic segmentation annotations for all images covering more than 300 categories. Distinguishing itself from the HM3DSem dataset [23], where 3D mesh models undergo time-consuming semantic labeling and where semantic segmentation images are rendered at each viewpoint, our pipeline delivers high-quality single-view annotations with lesser computational requirements.

Metrics. We use *mIoU* as the metric for evaluating the segmentation results, which is the average of the intersection-over-union (IoU) of all the categories. We also compute *mIoU-small*, which is the mIoU of only the small object categories. For ADE20K-indoor, we choose *bottle*, *plant*, *lamp*, *glass*, *flower*, and *vase* as the small object categories. For AVD, we add *knife*, *bowl*, *plate*, and *wall socket* into the small object category list. We compare the following approaches:

MaskFormer: A semantic segmentation model trained on ADE20K. We use the publicly available code and the trained weights of this model [3].

MaskFormer+SAM: Prompt SAM [11] with a dense grid to get an over-segmentation of the input image. The assignment of labels to each segment is determined by voting as described in Sec. 4.1.

MaskFormer+SAM+Detic (Ours): On the outcome of the 'Mask+SAM' approach, we overlay the results of Detic detections.

MaskFormer+SAM+Detic+MV (Ours): On the outcome

Table 1. mIoU(%) of semantic segmentation methods (higher is better)

Dataset	ADE20K-indoor		AVD-GT	
Method	mIoU (%)	mIoU-small (%)	mIoU (%)	mIoU-small (%)
MaskFormer [3]	55.7	62.3	58.6	60.2
MaskFormer+SAM [11]	56.8	63.0	64.9	70.7
MaskFormer+SAM+Detic (Ours)	**58.3**	**63.6**	64.2	92.8
MaskFormer+SAM+Detic+MV (Ours)	–	–	**65.4**	**97.7**

of the 'Mask+SAM+Detic' approach, we further apply multiview verification if depth images and camera poses are available.

Note that ADE20K-indoor does not provide depth images and camera poses. Hence, we did not evaluate the MaskFormer+SAM+Detic+MV approach with it. Besides, as the ADE20K semantic category set is a subset of LVIS categories, we create new annotations for MaskFormer and MaskFormer+SAM approaches when evaluating them with AVD. The new annotations are created by finding the synonyms of the LVIS class in the ADE20K category set. If no synonyms are found, e.g. *knife*, we label it as class *void*. We show these results in Table 1. The SAM-based approach consistently improves the model's performance by having better boundary segmentation of the segments. Mask-Former+SAM outperforms MaskFormer+SAM+Detic on the AVD dataset. This is because the labels of AVD-GT cover fewer classes with the ADE20K vocabulary.

6.2. Object Part Discovery

We create the 'cabinet handle' dataset with 1000+ images and 3000+ bounding boxes and associated masks. Fig. 7 shows the successful annotation examples. We suppose the feature clustering approach applied for the 'cabinet handle' can be easily utilized for bottle part segmentation.

cabinet_handle annotation examples

Figure 7. This figure gives examples of cabinet handles marked as red circles discovered through our clustering process.

6.3. Object Goal Navigation

We compare the recently released VLMaps [10] approach on the zero-shot navigation task and compare the performance with the semantic map-based approach generated with our labeling framework, named GT-Maps. The robot observations are RGB, depth, and semantic segmentation images. We also assume the robot can traverse the entire scene. The action space contains moving to a neighboring location on the navigation grid and STOP action. We adapt **VLMaps** approach to object goal navigation as follows:

- Traverse the entire scene, running LSeg [13] and get the pixel embeddings of each frame I_t.

- Project each frame's pixel embeddings onto the semantic map of size $m \times m$ using the frame's camera pose $(R_t|T_t)$. Take the average of the embeddings projected on the same cell.

- Localize the goal object on the semantic map by computing the inner product between the object's text embedding and the semantic map. The cell with the maximum inner product is the location of the object.

- Run A* algorithm to find the path from the robot's current pose to the selected location.

Our (**GT-Maps**) approach is using the semantic map built with auto-labeled semantic segmentations. For each scene, we select the object goal randomly out of all the object categories detected in the scene. We localize the target object in the map and plan the path the same way as the VLMaps approach from several randomly sampled start poses. For the **VLMaps**, the target location is obtained by feeding the sampled object target name as input to the LSeg language encoder, query the LSeg heatmap activations and pick the maximum activation of the semantic heatmap as the location of the semantic target. To evaluate the navigation episodes, we manually inspect the STOP location of the agent and its distance to the ground-truth target, both on the top-down semantic map and egocentric RGB views. If the agent stops close to the target, the success rate (SR) would be 1; otherwise, 0. In order to have a more reliable experiment, we repeat the above experiment with three different random seeds (indicated as R1, R2, and R3 on Table 2), which means each run covers 20 episodes, and the numbers under each column are the average SR for all 20 episodes for that run. Finally, the last column (Avg-SR) indicates the average and standard deviation of 3 runs for each approach/map.

As depicted in Table 2, we demonstrated that our generated semantic map using our auto-annotation approach

Instruction: Go to *'Coca cola glass bottle'*

GT-Maps (Semantic Map) **VLMaps (Heatmap)**

Last View in the Navigation Episode

Instruction: Go to *'Remote Control'*

GT-Maps (Semantic Map) **VLMaps (Heatmap)**

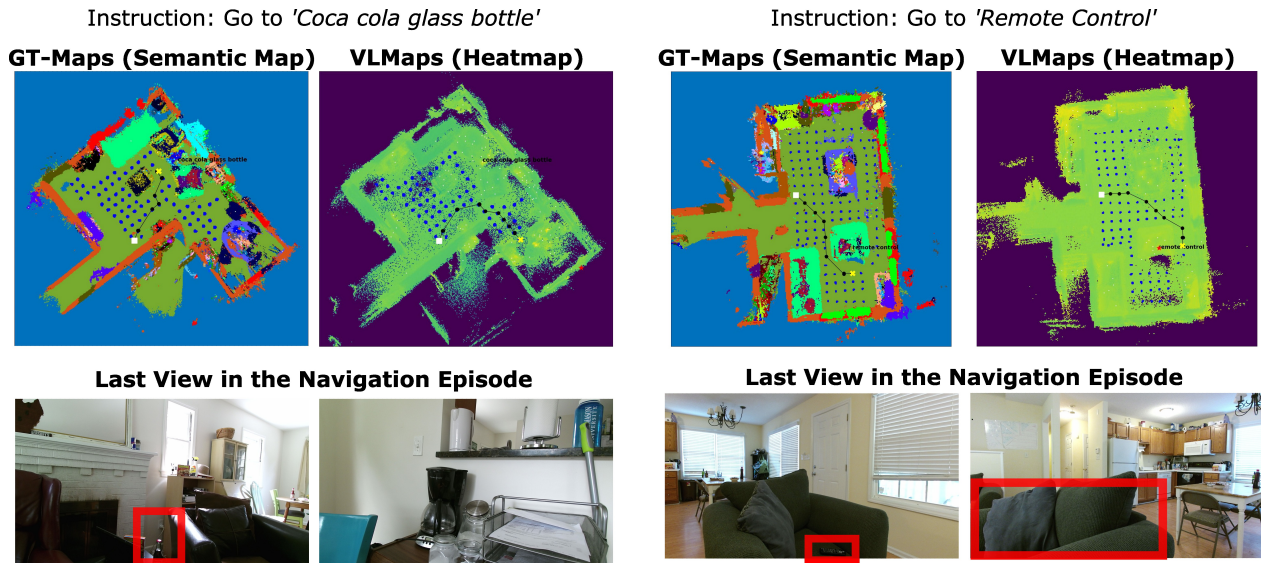**Last View in the Navigation Episode**

Figure 8. Qualitative results of two different episodes for navigating to small targets. The left example shows a case in which our GT-Maps successfully navigated to *coca coal glass bottle*, while VLMaps failed. The right example demonstrates a case where both approaches successfully navigate to *remote control*. For each example, the top row shows our semantic map on the left (GT-Map), in which blue points are the navigable points in the scene, the white point is the starting pose, and the yellow star is the selected target. While the VLMaps approach on the right demonstrates the queried LSeg-based map for the target object, the highest activation (argmax) is indicated as a red star. The bottom row shows the last view where the agent stops.

Table 2. Object Goal Navigation Results

Baselines	R1	R2	R3	Avg-SR (%)
VLMaps [10]	84.2	73.6	78.9	78.9±5.2
GT-Maps (Ours)	**94.7**	**78.9**	**94.7**	**89.4±9.1**

could outperform the VLMaps state-of-the-art baseline in zero-shot semantic-goal navigation by $\sim 10.5\%$ in terms of the average success rate. The results of this experiment endorse our hypothesis of the effectiveness of augmenting the semantic maps with the fusion of dense predictions using large-scale pre-trained models through labeling, compared to using large vision-language models in a zero-shot setting. Two qualitative results can be seen in Fig. 8.

7. Conclusion

This paper proposes a labeling approach for image semantic segmentation using pre-trained vision models. We design two downstream tasks based on the segmentation results for object part segmentation and zero-shot robot semantic-goal navigation. We use the labeling approach to annotate an RGB-D dataset, AVD. Our experiments demonstrate that: (i) SAM effectively discriminates object boundaries, especially for small objects and (ii) The built top-down-view semantic map with our labeling approach for semantic segmentation is competitive in zero-shot semantic-goal navigation compared to VLMaps. In terms of fu-

ture works, our pseudo-labels can be utilized for curating benchmarks for various vision-and-language tasks in indoor scenes, such as grounding (particularly for fine-grained concepts) [14], spatial relationships understanding [16], referring expression comprehension [5], descriptive image captioning, instruction following [12], and beyond.

8. Acknowledgements

References

[1] Phil Ammirato, Patrick Poirson, Eunbyung Park, Jana Kosecka, and Alexander C. Berg. A dataset for developing and benchmarking active vision. *ICRA*, pages 1378–1385, 2017. 1, 2, 6

[2] Vijay Badrinarayanan, Fabio Galasso, and Roberto Cipolla. Label propagation in video sequences. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3265–3272, 2010. 2

[3] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 2, 3, 6, 7

[4] Vishnu Sashank Dorbala, Gunnar Sigurdsson, Robinson Piramuthu, Jesse Thomason, and Gaurav S Sukhatme. Clip-

nav: Using clip for zero-shot vision-and-language navigation. *arXiv preprint arXiv:2211.16649*, 2022. 2, 5

[5] Zi-Yi Dou, Aishwarya Kamath, Zhe Gan, Pengchuan Zhang, Jianfeng Wang, Linjie Li, Zicheng Liu, Ce Liu, Yann LeCun, Nanyun Peng, et al. Coarse-to-fine vision-language pre-training with fusion in the backbone. *Advances in neural information processing systems*, 35:32942–32956, 2022. 8

[6] Zhaoyuan Fang, Ayush Jain, Gabriel Sarch, Adam W. Harley, and Katerina Fragkiadaki. Move to see better: Self-improving embodied object detection. In *BMVC*, 2021. 2

[7] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Clip on wheels: Zero-shot object navigation as object localization and exploration. *arXiv preprint arXiv:2203.10421*, 3(4):7, 2022. 2, 5

[8] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. Lvis: A dataset for large vocabulary instance segmentation. *CVPR*, pages 5351–5359, 2019. 3, 6

[9] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1989–1998. PMLR, 10–15 Jul 2018. 2

[10] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615. IEEE, 2023. 2, 5, 7, 8

[11] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. *ArXiv*, abs/2304.02643, 2023. 2, 3, 6, 7

[12] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 104–120. Springer, 2020. 8

[13] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. 2, 7

[14] Liunian Harold Li*, Pengchuan Zhang*, Haotian Zhang*, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *CVPR*, 2022. 8

[15] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 3

[16] Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023. 8

[17] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. *Advances in Neural Information Processing Systems*, 35:32340–32352, 2022. 2

[18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3, 5

[19] Md. Alimoor Reza, Akshay U. Naik, Kai Chen, and David J. Crandall. Automatic annotation for semantic segmentation in indoor scenes. pages 4970–4976, 2019. 2, 5

[20] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. *ICCV*, pages 8429–8438, 2019. 3

[21] 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. 1

[22] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. Dacs: Domain adaptation via cross-domain mixed sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021. 2

[23] Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, Alexander William Clegg, and Devendra Singh Chaplot. Habitat-matterport 3d semantics dataset. *ArXiv*, abs/2210.05633, 2022. 1, 6

[24] Jinyu Yang, Weizhi An, Chaochao Yan, Peilin Zhao, and Junzhou Huang. Context-aware domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021. 2

[25] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[26] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. *ICCV*, pages 15818–15827, 2021. 2

[27] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. *CVPR*, pages 5122–5130, 2017. 1, 6

[28] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Phillip Krahenbuhl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022. 2, 3

[29] Yi Zhu, Karan Sapra, Fitsum A. Reda, Kevin J. Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8848–8857, 2019. 2