SupeRGB-D: Zero-shot Instance Segmentation in Cluttered Indoor Environments

Evin Pınar Örnek[1], Aravindhan K Krishnan[2], Shreekant Gayaka[2],
Cheng-Hao Kuo[2], Arnie Sen[2], Nassir Navab[1], Federico Tombari[1,3]

Abstract—Object instance segmentation is a key challenge for indoor robots navigating cluttered environments with many small objects. Limitations in 3D sensing capabilities often make it difficult to detect every possible object. While deep learning approaches may be effective for this problem, manually annotating 3D data for supervised learning is time-consuming. In this work, we explore zero-shot instance segmentation (ZSIS) from RGB-D data to identify unseen objects in a semantic category-agnostic manner. We introduce a zero-shot split for Tabletop Objects Dataset (TOD-Z) to enable this study and present a method that uses annotated objects to learn the "objectness" of pixels and generalize to unseen object categories in cluttered indoor environments. Our method, SupeRGB-D, groups pixels into small patches based on geometric cues and learns to merge the patches in a deep agglomerative clustering fashion. SupeRGB-D outperforms existing baselines on unseen objects while achieving similar performance on seen objects. We further show competitive results on the real dataset OCID. With its lightweight design (0.4 MB memory requirement), our method is extremely suitable for mobile and robotic applications. Additional DINO features can increase the performance with a higher memory requirement. The dataset split and code is available at https://github.com/evinpinar/supergb-d.

Index Terms—RGB-D Perception; Deep Learning for Visual Perception; Object Detection, Segmentation and Categorization; Zero-shot Instance Segmentation

I. INTRODUCTION

INSTANCE segmentation is a long-standing and important problem with its applications from robotic perception to augmented reality [1], [2]. Especially for an indoor autonomous agent, detecting and identifying objects encountered in an environment is critical for visual navigation. However, the existing 3D sensing capabilities prevent detecting every possible object. Deep learning-based approaches are promising, but annotating 3D data is cumbersome and time-consuming. The problem is further exacerbated due to the variety of objects observed in a typical home environment. In this work, we address this issue by focusing on unseen object discovery in zero-shot from only a limited amount of annotated data.
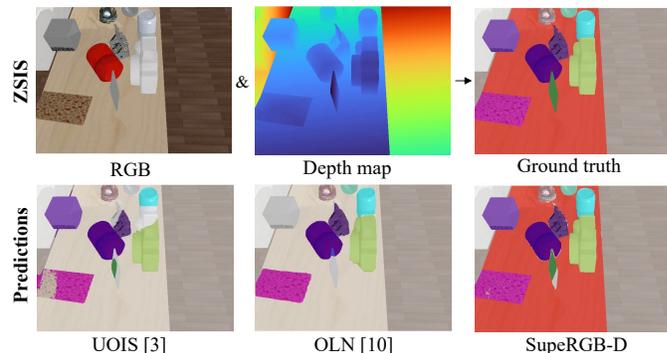
Fig. 1: We study zero-shot instance segmentation from RGB and depth modalities, where a training split includes seen categories (e.g. plate, mug), and the test split includes both seen and unseen categories (e.g. camera, printer). We propose a dataset split and a method for ZSIS from RGB-D in cluttered environments. In this paradigm, compared to UOIS [9] and OLN [10], the proposed method SupeRGB-D segments unseen objects with accurate boundaries (camera, helmet), and can even identify things in the background (e.g. table).

Leveraging different sensor modalities such as RGB images and depth maps and carrying the task from 2D to 3D comes into further rescue when detecting object instances. Using only 2D cues (RGB) cannot capture the complexity of the 3D world, especially in high-textured areas [3]. Some methods use only depth, but they tend to result poorly when sensors fail (e.g. non-Lambertian surfaces, sharp edges) [4], [5], [6]. Few approaches use RGB-D, but they depend on large-scale annotated datasets or hand-crafted features [7]. Relying heavily on annotations is not scalable for an indoor agent, given the variations in the object categories encountered in different indoor environments [8]. Prior work aims to solve this task in sim2real [9], yet, employment of these methods could be difficult due to memory contraints.

Towards this, we tackle the zero-shot instance segmentation (ZSIS) problem, where a training split can include objects from seen categories, whereas a test split with seen and unseen categories. We rely on RGB and depth modalities to solve this problem. As there are no available RGB-D datasets to study the problem under this multi-modal setup, we propose a novel dataset split, TOD-Z, where we divide the dataset into seen and unseen categories to enable a structured study into ZSIS. Through this dataset, we establish a benchmark and propose a suitable approach for the task. Our method, SupeRGB-D,

first detects local regions from RGB images and depth maps by simultaneously applying super-pixel over-segmentation on the input modalities. Then, it builds instance masks in a bottom-up manner by learning to group these extracted small patches through visual and geometric cues in an agglomerative clustering fashion. We show that SupeRGB-D outperforms compared methods on unseen objects as illustrated in Fig. 1, obtains strong boundaries due to super-pixels acquired edges, and is extremely suitable for mobile applications with its lightweight design.

Our contributions can be summarised as follows:

- We study zero-shot instance segmentation from RGB-D in an open-world scenario. Notably, it is one of the first works on generalization towards unseen categories in zero-shot paradigm as opposed to cross-data sim2real.
- To this end, we provide an in-depth analysis of the generalization performance under different setups and offer a zero-shot split over the original TOD, namely, the TOD-Z split. With this split, we establish a novel benchmark to enable studying ZSIS on RGB-D.
- We propose a novel method, SupeRGB-D, that is explicitly tailored towards generalization for unseen objects under limited data. SupeRGB-D first over-segments the input into local patches, then learns to merge them to reflect objectness and infer instance mask.

II. RELATED WORK

Instance segmentation is one of the fundamental tasks in computer vision [2]. Prior to the deep learning (DL) era, learning-free algorithms aimed to group the pixels according to a similarity in a bottom-up manner. These works used hand-crafted features, e.g., brightness, color, texture, and defined heuristics to learn the objectness. Some of these study segmentation in terms of graph partitioning and seek a global optimal solution ([11], [12], [13]). Recently, with the availability of large-scale datasets and advances in DL, the performance of instance segmentation improved [14]. The majority of methods first detect the object and then segment the detected region in a top-down manner. A nominal work is based on identifying probable object regions through Region Proposal Network (RPN) and then using predictors to classify and segment objects, i.e., Mask R-CNN [1]. Albeit these methods perform remarkably, learning generalizable object recognition is data-hungry.

Low-shot segmentation has gained importance to address the generalizability of instance segmentation to enable its application in real-world scenarios. Zero-shot segmentation aims to densely segment input images along with the object labels for zero-shot objects, i.e., the objects that were never seen during training, through the usage of semantic cues such as WordNet embeddings [15], [16], [17]. Few-shot simplifies the assumption by providing one or a few samples during training [18], [19]. ZSIS aims to detect different instances of unseen categories. Zheng et al. studied ZSIS [20] through learning a semantic alignment on a Fast R-CNN instance predictor. Following, Kim et al. proposed Object Localization Network (OLN) [10] which turns off the object classifier
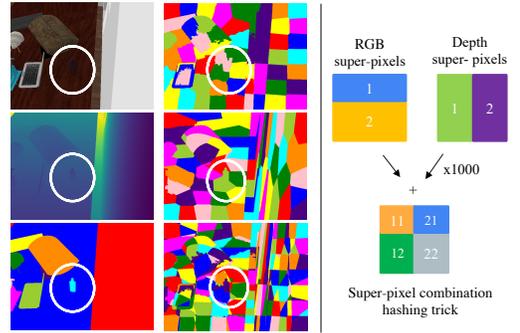

Fig. 2: **RGB vs. depth modalities and super-pixels combination.** RGB and depth maps have complementary information. As depicted on the left, the bottle object was segmented out only by using both super-pixels. On the right, the super-pixels merging algorithm is illustrated.

of Mask R-CNN and learns a common objectness score. In an orthogonal direction, Wang et al. offered a ground-truth mask generation strategy to be used for detector supervision [21]. We also study class-agnostic segmentation and aim to generalize beyond observed shapes. In contrast, we focus on cluttered indoor spaces for robotic applications and work on multi-modal RGB-D cues.

Segmentation on RGB-D modality became another possible direction of study after the availability of consumer level range sensors [22] and the relevant datasets [23]. The segmentation can be applied on either on single-view RGB-D images [24], or on full 3D point clouds which are reconstructed through multi-view RGB-D fusion algorithms [5]. Initial works relied on hand-crafted features and heuristics such as normal, curvature and planarity [24], [25], [3], [26]. Richtsfeld et al. generated patches with planar surface estimation and learned merging with SVM[27]. Recent works use deep learning methods [5], [28], [7]. *Low-shot segmentation in 3D* is a rather novel area of research [29], [30], [31]. Some works looked into part discovery [32] or zero-shot semantic segmentation and classification for unseen 3D shapes [30]. Meanwhile, in terms of robotics applications, previous methods looked into feature extraction recent trends perform *unseen object instance segmentation* in a simulation to real data transfer setup [33], [34], [35], [36], [9], [37]. Xie et al. proposed UOIS-Net to learn objectness from RGB-D input, and provided with a large simulation dataset TOD. Back et al. predicts amodal instance segments [38], while Durner et al. proposes a stereo dataset and a suitable method [39]. As opposed to prior works, instead of cross-dataset sim2real paradigm, we study ZSIS.

III. METHOD

In the following, we formalize the ZSIS problem and explain the proposed method.

Problem formulation. We denote the set of seen classes as $\mathcal{S}$, a disjoint set of novel classes as $\mathcal{U}$ and the union of them as $\mathcal{Y} = \mathcal{S} \cup \mathcal{U}$. Let $\mathcal{T} = \{(x^1, x^2, y) | x \in \mathcal{X}_s, y \in \mathcal{Y}_s\}$, where $\mathcal{T}$ defines the training set, x^1 is an input image $I \in R^{3 \times H \times W}$ in the image space $\mathcal{X}_s$, x^2 is its input depth map $I \in R^{H \times W}$,
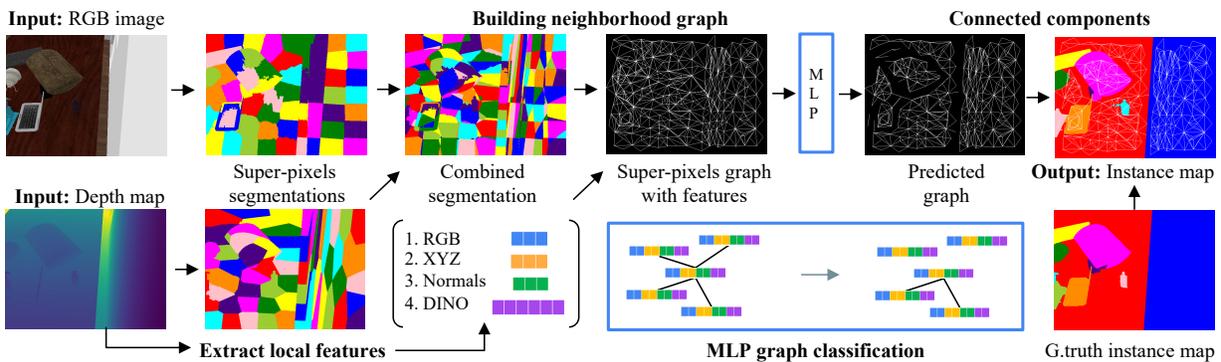
Fig. 3: **Method illustration.** Our method SupeRGB-D first obtains local patches from input by applying super-pixels over-segmentation on RGB and depth maps separately and combining them to attain intersected areas, and the regions that lie in between. It then learns to merge the local patches to reflect the connectedness of objects by relying on extracted explicit (XYZ, normal and RGB) and implicit (DINO [40]) features for patches and feeding each patch pair to a multi-layer perceptron. The final instance map is inferred by applying connected components on the predicted graph.

y is its class agnostic instance label mask with the same size. Training images include objects from seen object classes $\mathcal{S}$, whereas test images contain both seen and unseen classes from $\mathcal{U}$. Hence, the goal of ZSIS is to predict pixel-wise dense instance predictions among both seen and unseen classes. Note that, instead of semantic segmentation, we predict the class-agnostic instance segmentation map which enables open-world object discovery in real-world environments.

Method overview. We design a lightweight bottom-up strategy to learn instance segmentation from a small amount of data, without depending on object detection. We design a bottom-up strategy to infer instance segmentation which does not depend on object recognition and detection. Our algorithm first segments the input into small regions and then learns to merge these regions to infer a full instance segmentation map. Formally, we regard the segmentation task as a graph partitioning problem, where we initially generate an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ on a super-pixels level to represent an image. A vertex $v_i \in \mathcal{V}$ represents one super-pixel region, and an edge $e = (v_i, v_j) \in \mathcal{E}$ connects two vertices v_i and v_j. A segmentation solution S is a partition of $\mathcal{V}$ into multiple connected components. To find the partitioning, as opposed to prior work ([13], [11]) that uses hand-crafted heuristics to find a threshold, our method learns the partitioning implicitly through a neural network.

A. Extracting local regions from RGB-D

We aim towards a lightweight method for practical mobile and robot applications. Hence, we use the super-pixel over-segmentation instead of pixel-wise dense input to build our graph. Super-pixel methods can capture and summarize the most informative local regions from an image in an unsupervised manner. Furthermore, they provide strong boundaries reliably without requiring additional semantic input. As for the super-pixels algorithm, we select SLIC [41] because it is a simple yet robust approach, does not require any pretraining, has efficient runtime, and is accepted in the community for different downstream tasks [42], [8].

Combining superpixels from RGB and depth maps. As illustrated in Fig. 2, we apply SLIC separately on two input modalities, RGB and depth maps, resulting in two patch maps. Applying super-pixels on RGB and on depth maps has different indications. From RGB, it segments the regions according to color and brightness information. From the depth map, it understands the objectness of instances and the topology of surfaces through 3D cues. We aim to combine two maps to retrieve the intersecting areas between the patches and the regions lying in between. We use the following hashing trick: suppose each mask has N patches, where each patch is identified with a unique mask id, i.e. P_i^{RGB} and P_i^{depth}, where $i \in R^N$. For that, we multiply RGB superpixel ids with 1000 (shifting them three digits left) and add depth ids:

$$P_i^{RGB} * 1000 + P_i^{depth}. \tag{1}$$

Explicit local features. For each super-pixel region, we extract the local features that describe the properties of that local patch. As we aim to use the cues both from RGB and depth maps, we use the following features: (1) averaged R,G,B values within the patch, (2) X,Y,Z coordinates of the patch centroid, (3) surface normals of the centroid. For X,Y coordinates, we use image coordinates, whereas for Z, we use normalized depth value on that pixel d_i. We calculate the surface normal n_i of the centroid pixel via vertical and horizontal gradients of the depth map as:

$$n_i^g = [-\nabla_x(d_i), -\nabla_y(d_i), 1]^\top. \tag{2}$$

Implicit ViT features. Explicit features already bring strong cues for patch merging, however, are limited in terms of global cues. To make the model globally aware, we rely on self-supervised Vision Transformer features from DINO[40]. Notably, our goal is open-world generalizability, and we cannot rely on neural networks pretrained on large-scale datasets that already observe all categories (ImageNet ResNet [43]). Self-supervised ViT, on the other hand, is trained under a self-distillation setup and complies with the zero-shot setup where no category annotations are needed. We extract attention maps from the ViT, average the features that lie within the regions

and incorporate them as additional feature inputs in the super-pixel neighborhood graph.

To sum up, a patch feature can be constructed by concatenating the explicit and implicit features in a vector:

$$F_{sp} = [R, G, B, X, Y, Z, n_x, n_y, n_z, \\ DINO_1, ..., DINO_M]$$

(3)

where M denotes the number of ViT attention heads. This is a superset of features and we ablate the effect of each component in Experiments Section.

B. Learning to merge

In order to infer instance segments, our method learns to merge super-pixel regions through an agglomerative clustering fashion. After building a super-pixel neighborhood graph where neighboring patches are connected through an edge, we use a neural network to learn the merging of the patches by classifying the edge as a cut or no-cut criteria. For supervision, we generate a ground truth pair graph from the instance mask and use this mid-representation.

Model training. We first pre-process the data to extract the super-pixels along with the explicit and implicit features. For learning to merge, we use shared Multilayer Perceptron (MLP) layers, and as an input, we feed neighboring super-pixel patch pairs features P_i, P_j. The network is tasked to classify whether this patch pair is connected. We simply use binary cross-entropy loss to train the neural network:

$$L_{BCE} = -(y \log(p) + (1 - y) \log(1 - p)).$$

During training, we randomly swap half of the pairs to enable permutation invariance P_j, P_i. Furthermore, we realized that the positive/negative ratio of the data is naturally biased, as the boundary pixels on images occupy fewer pixels than the objects themselves. The training data indeed have 80% positive samples, and this prevents the network to learn the negatives appropriately. Hence, we modify the sampling ratio to enable the learning of negative edges. We provide more details on this in the Experiments.

Inferring the instance segmentation map. The model predicts binary edge labels representing which patch pairs are connected to each other. To infer the instance segmentation map, we need to identify disjoint connected sets from the patch neighborhood graph. As shown in Fig. 3, we use connected-components algorithm for this purpose.

IV. TABLETOP OBJECTS DATASET FOR ZERO-SHOT (TOD-Z)

Zero-shot *semantic* segmentation problem is well-studied with RGB image datasets [15], [42], whereas there are only a limited number of works on zero-shot *instance* segmentation [21]. To systematically study ZSIS on RGB-D data, we believe that one should initially understand the effect of different modalities where sensor noise is minimized. Hence, to benchmark the RGB-D ZSIS framework, we select the synthetic Tabletop Object Dataset (TOD) [9] and propose a zero-shot split upon this, namely, TOD-Z. TOD is composed
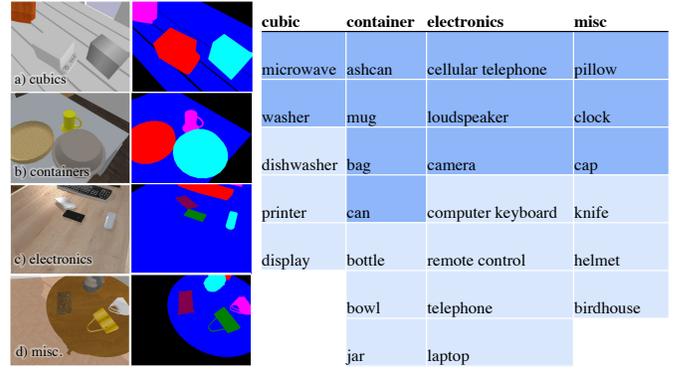
cubic	container	electronics	misc
microwave	ashcan	cellular telephone	pillow
washer	mug	loudspeaker	clock
dishwasher	bag	camera	cap
printer	can	computer keyboard	knife
display	bottle	remote control	helmet
	bowl	telephone	birdhouse
	jar	laptop	

Fig. 4: **Proposed TOD-Z Split**. We group the 25 object classes of TOD into 4 categories, and create a seen/unseen split by selecting equal numbers of classes from each category. The seen classes are highlighted with dark blue, whereas the unseen ones with light blue.

of ShapeNet [44] objects cluttered on tabletop environments. There are 25 different categories of common household objects, such as *bottle, helmet, microwave, pillow* with 40K scenes, containing 5 to 25 random objects randomly thrown on tables acquired from SUNCG environments [45].

Data pre-processing. The raw TOD data does not include the semantic segmentation masks. However, we need them to be able to identify the object classes and study the proposed task. We hence extract the Synset id of each CAD object from TOD metadata and extract semantic masks for each image. Furthermore, this data is created by setting up a scene and rendering multiple images for each scene from different viewpoints. Each scene contains one empty SUNCG home background image, as well as another image with an empty table placed in SUNCG home. Furthermore, some images contain only a single object occupying a very small fraction of an image (a couple of pixels). Therefore, we filter out these redundant and information-poor images.

Selecting seen and unseen object categories. For dividing the classes into seen (train) and unseen (test) classes, we follow a procedure similar to [30], [46]. The 25 object classes of TOD have a high variance in terms of both semantics and 3D structures. We initially group the classes in terms of semantics, by applying K-means clustering at the word2vec vectors as well as WordNet hierarchical distances. We further visually analyze images by extracting ResNet features for the objects appearing in images and checking similarities. Through this analysis, we group the object classes into four shape categories: containers (e.g., bowl, bottle, ashcan), cubics (e.g., washer, microwave), electronics (e.g., telephone, cellular), and miscellaneous (e.g., knife, birdhouse, pillow). We randomly select half of the classes from each group and assign them as seen, and the remaining as unseen. The final training split contains 12 classes, whereas the open test set contains 25 classes (12 seen & 13 unseen), with 7118 train images and 881 test samples with equal object distributions. The TOD-Z classes and sample images can be seen in Fig. 4. We provide an in-depth analysis on of different classes and the number of samples in the Experiments.

TABLE I: **Comparing SupeRGB-D (our method) to baseline methods over harmonic mean, seen and unseen splits** in terms of object overlap precision, recall, and f-score, as well as boundaries. Higher values are better for all metrics.

Set	Method	Overlap			Boundary		
		P	R	F	P	R	F
HM	UOIS	81.27	77.08	72.92	65.97	**67.38**	64.96
	OLN	72.00	**83.12**	70.72	64.99	65.61	63.95
	SupeRGB-D	**89.67**	74.44	**76.48**	**69.17**	67.99	**66.75**
Seen	UOIS	86.54	80.70	**79.23**	**70.39**	**72.79**	**70.20**
	OLN	74.12	**85.81**	73.05	66.08	67.43	65.41
	SupeRGB-D	**90.35**	73.69	76.31	66.37	64.16	63.71
Unseen	UOIS	76.60	73.78	67.53	62.08	62.73	60.45
	OLN	69.99	**80.60**	68.53	63.94	63.88	62.55
	SupeRGB-D	**89.01**	75.20	**76.66**	**69.90**	**66.75**	**66.52**

TABLE II: **Computational requirements** in terms of number of parameters in the neural network, model size in megabyte and runtime in seconds. SupeRGB-D has a lower number of parameters, fewer memory requirements, and a faster runtime than the compared baselines.

Method	# Param.	Memory (MB)	Runtime [s]
UOIS	45.407.488	173.216	7.42
OLN	43.744.526	167.935	4.50
SupeRGB-D	**103.682**	**0.400**	1.44

V. EXPERIMENTS

Metrics. We evaluate the **instance segmentation performance** with precision, recall and F-score metrics following the literature [47], [9]. These metrics are calculated by first measuring the F-score between every ground truth and predicted instance mask pair, which are then matched with the Hungarian algorithm by the highest overlap F-score. The metrics between the matched objects are then calculated by:

$$P = \frac{\sum_i |s_i \cap g(s_i)|}{\sum_i |s_i|}, R = \frac{\sum_i |s_i \cap g(s_i)|}{\sum_j |g_j|}, F = \frac{2 * P * R}{P + R}$$

where s_i denotes the set of pixels belonging to predicted object i, $g(s_i)$ is the set of pixels of the matched ground truth object of s_i, and g_j is the set of pixels for ground truth object j. We denote these metrics as Overlap P/R/F. Furthermore, we report the same numbers for the boundary pixels, where the boundaries are extracted by the edge detector and dilated. These metrics are calculated by considering the boundary pixels on the matched objects and they express how well the methods perform on fine-level details. Higher is the better for all of the metrics.

To measure the **zero-shot performance**, following the previous works [15], [30], we report the instance segmentation metrics P/R/F averaged for seen classes, unseen classes and harmonic mean (HM) of seen&unseen, which is computed as $HM = \frac{2*S*U}{S+U}$. The HM measures how well a model balances seen and unseen classes.

Datasets. We use TOD to examine the generalization gap between different classes as this synthetic dataset enables a structured study by precluding the effect of sensor noise. We propose TOD-Z split, where we demonstrate the usefulness of our method and compare it with others. To show SupeRGB-D performance on real images, we use the Object Clutter Indoor
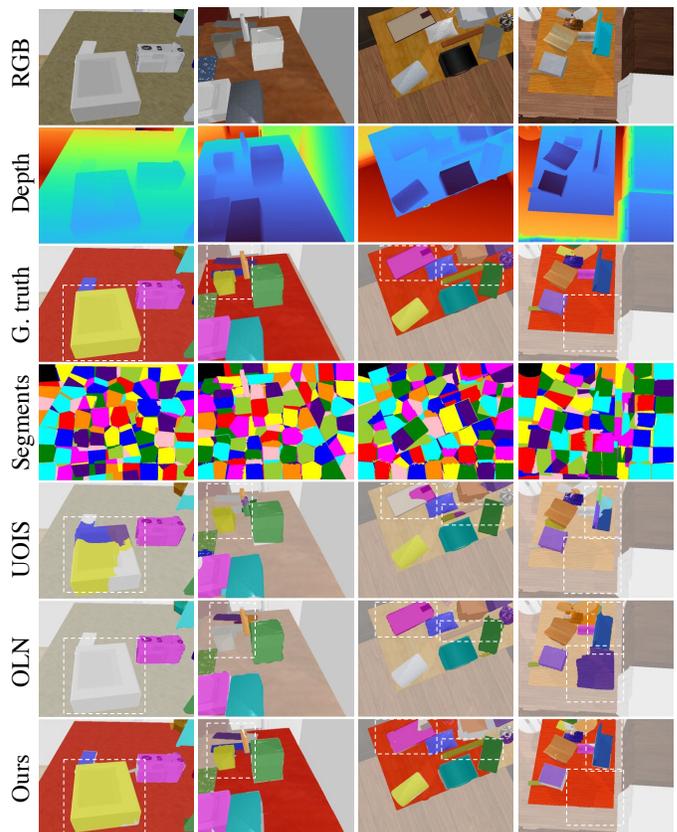

Fig. 5: **Qualitative results.** Samples in cluttered environments with unseen object categories varying from the dishwasher, bottle, keyboard, display, and laptop (left to right). SupeRGB-D can segment these instances, whereas the compared methods fail to detect the objects as they tend to overfit into seen categories. Further, we can identify distinct objects in cluttered scenes captured from different perspectives such as floor parallel view (left two) or top view (right two), and even the table and objects beyond the background (right bottom).

Dataset (OCID) [48], which is captured in real-world cluttered environments through a structured light depth sensor (ASUS-Pro) mounted on a robot.

Implementation details. We implement the merging network with a 3-layer MLP of latent feature sizes $(256, 1024, 256)$, each followed by ReLU, and dropout layers. We use Adam optimizer with a step size learning rate scheduler on an initial learning rate $1e^{-3}$ and a reduction rate of 0.5. We train our network for 10 epochs and select the checkpoint with the highest validation F-score. For over-segmentation, we use the SLIC algorithm, where we find the ideal patch size of 128 after ablation studies.

Baseline methods. We select two strong methods that aim toward detecting and segmenting unseen objects. The first one is the aforementioned UOIS-Net method, which was tailored for RGB-D segmentation within sim2real [9]. Their method consists of two steps, first using the depth map to detect the object instances, followed by improving the object boundaries through RGB image. We train it with the proposed TOD-Z split to measure the ZSIS performance. Our second baseline

TABLE III: **Data sampling studies.** Increasing the negative ratio helps the model learn objects better, increasing all metrics. The best positive/negative ratio is $25/75$.

Set	p/n ratio	Overlap		
		P	R	F
HM	Random	55.89	66.32	51.25
	50/50	65.15	69.07	59.55
	25/75	86.91	**78.71**	**78.43**
	10/90	**93.37**	66.76	73.64
Seen	Random	55.29	67.23	50.94
	50/50	64.49	69.83	59.24
	25/75	87.66	**79.16**	**79.11**
	10/90	**94.24**	66.23	75.53
Unseen	Random	56.45	65.43	51.56
	50/50	65.82	68.33	59.86
	25/75	86.18	**78.26**	**77.76**
	10/90	**92.53**	67.29	73.74

is Object Localization Network (OLN) [10], which was built upon Mask R-CNN [1], but aims to detect in an open-world setup by replacing the RPN object classification head with a generic object understanding head. After modifying this network for taking RGB-D input, we train it with TOD-Z. We follow the public code and training details provided by the authors.

A. Comparison to baselines

We **quantitatively** compare our method without DINO features against the baselines in terms of instance segmentation performance in Tab. I and in terms of computational requirements in Tab. II. Compared to baselines, SupeRGB-D results with the best overlap F and P on harmonic mean and unseen splits and the best boundary scores on all splits. On seen objects, SupeRGB-D has the highest P, whereas UOIS has the highest F, showing that this method is strong on seen objects and is better than OLN probably because their model design considers 3D cues explicitly. On unseen objects, SupeRGB-D achieves nearly 10% improvement on F and boundary scores over the compared baselines. This shows that the baseline methods overfit into seen object categories and are rather limited in generalization over unseen types of shapes. In terms of **model size**, SupeRGB-D has fewer neural network parameters and has lower memory requirements than the compared methods. We furthermore compare in terms of runtime efficiency in seconds, where we measure the runtime of our method, including the data preprocessing (superpixels extraction) and postprocessing (segmentation inference via connected components) on an i7-8700 3.20GHz 12-core CPU. SupeRGB-D has the fastest runtime with 1.44 seconds, followed by 4.50 OLN, which is three times longer than SupeRGB-D, and 7.42 UOIS, with the longest probably due to data preprocessing. The lightweight design of SupeRGB-D enables running it on any robot or a mobile device with low-cost hardware. Our method's performance can be improved by additional ViT features with an increased memory requirement as we show in ablations.

We further provide a **qualitative** comparison to baselines in Fig. 5. We provide samples with unseen objects, dishwasher, bottle, keyboard, display, and laptop, captured from different viewpoints (bottom and top). It can be observed that compared methods perform poorly on these objects, either cannot detect

TABLE IV: **Features ablation.** We evaluate SupeRGB-D by ablating different feature inputs.

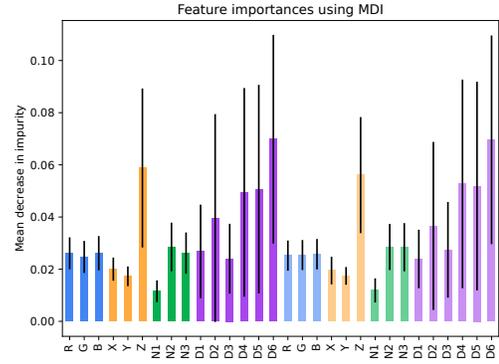| Input				Overlap			Boundary		
RGB	D	N	DINO	P	R	F	P	R	F
✓				76.51	75.81	69.51	60.42	68.95	61.91
	✓			53.49	62.95	48.26	35.20	37.24	34.24
	✓*			61.62	60.26	52.62	36.79	36.50	34.81
			✓	**94.46**	55.67	65.47	58.04	46.56	50.26
	✓*		✓	88.26	63.37	67.64	59.22	51.68	53.25
✓	✓			87.62	75.33	76.33	68.09	65.42	65.08
✓	✓	✓		**89.67**	74.44	76.48	69.17	67.99	66.75
✓	✓	✓	✓	86.91	**78.71**	**78.43**	72.05	71.23	**69.69**

Fig. 6: **Feature importances for an input patch pair.** Depth values (Z) and DINO6 (ViT 6th attention head) features have the highest importance on a Random Forest Classifier.

them or hallucinate a couple of different objects on the overlapping pixels. Comparably, SupeRGB-D can segment and identify the objects as it relies on local cues and does not overfit into categories. Interestingly, it can even detect the objects beyond the categories, that lie in the background such as the table.

B. Ablation studies

Positive/negative sampling. SupeRGB-D requires balancing the positive and negative edge ratios to encounter the data imbalance, i.e., boundary vs. in-object pixels. We compare the default sampling, where the input reflects the dataset ratio of 80% positives and modified ratios of $50/50$, $25/75$, and $10/90$ p/n in Tab. III. There is a clear correlation between the negative ratio and performance. First, having a $50/50$ ratio keeps P similar, but it increases R significantly by around 10%. Further, increasing the negative ratio to $25/75$ boosts both P and R. More increase, however, results in a drop in R and F, probably due to over-weighting the edge pixels.

Effect of features. SupeRGB-D relies on features from different modalities, namely, RGB, XYZ, normals, and DINO features. We compare the effect of each in Tab. IV on the harmonic mean. We observe that (1) using only RGB features enables class-agnostic instance segmentation, whereas (2) using only depth does not give strong enough cues for the task. (3) Combining RGB and depth features improve upon the RGB on all overlap and boundary metrics. (4) The addition of normals does not significantly affect overlap scores, but it helps with boundaries. (5) The full model with DINO features boosts the performance further.

TABLE V: **Quantitatives on OCID.** Sim2real (S2R) and ZSIS setup comparison on YCB10.

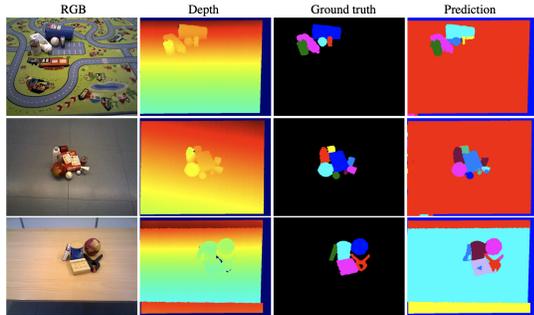| Method		Overlap			Boundary		
S2R	ZSIS	P	R	F	P	R	F
UOIS ✓		51.73	34.26	30.86	25.05	27.83	23.22
OLN ✓		47.79	15.90	17.46	34.81	14.94	17.66
Ours ✓		59.02	47.09	38.44	36.12	32.50	30.42
OLN	✓	58.31	46.48	38.21	46.44	50.82	43.12
Ours	✓	55.46	60.22	51.23	41.52	55.92	45.65

Fig. 7: Results on real-world RGB-D dataset OCID YCB10 split, samples from floor and tabletop environments.

Feature importance. To understand the effect of features out of a black box model, we fit a Random Forest classifier and analyze the importance of each input feature on the output in Fig. 6. For this study, we randomly select 512 patches and fit a classifier. The feature importance is calculated as the decrease in node impurity weighted by the probability of reaching that node [49]. Higher values show higher importance in the model decision. The Z and DINO 6 features have the highest effect on the classification outcome.

Superpixel size. We compare superpixel sizes of 32, 64, and 128 in Tab. VI. Increasing the patch size from 32 to 64 has significant gains on all overlap and boundary metrics (e.g. 12% on overlap, 13% on boundary F). This is expected, as a smaller number of patches don't cover objects well and result in missing objects. Doubling 64 to 128 only helps in boundary F, as finer patches cover boundaries better.

C. Results on real-world images

As of today, there are no zero-shot splits available on real-world RGB-D datasets. Yet, to test SuperGB-D capability under real sensor depth maps, we provide results on the OCID dataset in Fig. 7 and Tab. V. This dataset consists of various types of objects in different tabletop and floor environments of different materials. For complementary reasons, we first test on sim2real setup with the TOD-Z trained models. Then for the ZSIS problem, we train SupeRGB-D and OLN methods with ARID objects and test on YCB10 objects. SupeRGB-D performs well on high-texture images, as well as noisy depth maps, and segments things in the background.

D. Generalization studies on TOD

Before proposing TOD-Z split, we conducted a set of experiments to study the generalization gap between different seen and unseen classes on the original TOD. We split TOD into seen train, seen test, and closed unseen test, and run

TABLE VI: **Number of super-pixels.** We compare the effect of patch sizes in terms of the number of super-pixels in the over-segmentation mask and provide results on the harmonic mean. Increasing the number of super-pixels results in better instance segmentation.

# patches	Overlap			Boundary		
	P	R	F	P	R	F
32	71.70	74.09	66.03	59.65	63.20	56.61
64	86.91	78.71	78.43	72.05	71.23	69.69
128	**88.80**	78.79	78.46	**74.02**	71.84	70.89
256	87.89	**80.94**	**80.31**	73.30	**73.40**	**71.67**

TABLE VII: **Dataset generalization gap studies.**

# obj	set	P	R	F
15	seen	83.04	87.29	84.09
	unseen	75.50	83.69	77.50
20	seen	88.89	89.30	88.69
	unseen	86.37	88.55	85.91
23	seen	93.45	91.29	91.94
	unseen	84.37	84.36	82.27

(a)

#data	#obj	P	R	F
2070	5	67.91	52.37	69.37
	25	80.83	67.93	71.61
6375	10	72.19	64.62	76.03
	25	84.96	76.72	79.26
17265	15	81.84	75.42	85.81
	25	86.94	83.09	90.31

(b)

train	Set	P	R	F
15-var1	seen	83.04	87.29	84.09
	unseen	75.50	83.69	77.50
15-var2	seen	82.44	84.96	82.32
	unseen	78.4	82.62	79.00
15-var3	seen	86.15	84.43	85.84
	unseen	80.14	84.97	80.86

(c)

train	Set	P	R	F
cubics	seen	90.50	77.84	81.31
	unseen	71.94	54.80	58.93
cont.	seen	91.50	80.41	83.55
	unseen	68.42	52.24	55.88
electr.	seen	79.91	66.20	68.43
	unseen	71.89	49.95	55.51
misc.	seen	82.26	78.35	77.35
	unseen	79.53	69.07	71.29

(d)

experiments on UOIS method. The closed unseen test includes images of only unseen classes as opposed to open-world split that can have both seen+unseen objects. We report Overlap scores.

1. Seen/unseen gap for different number of seen classes. We prepare three training splits by selecting images of 15/20/23 seen objects and the remaining (10/5/2) unseen. As shown in Tab. VII (a), there is a generalization gap between seen and unseen data. When there are more training classes, the performance on both seen and unseen splits are higher.

2. Number of train samples versus class variance. We prepare three training splits with a varying number of data samples (2070/6375/17265) and either have a set of seen classes(5/10/15) or include all 25 classes. We report the values on the public TOD test split, which includes objects from 25 classes. In Tab. VII (b), increasing class variance helps on generalizability.

3. Generalization gap for different class types. We prepare three training datasets of 15 seen classes, where each dataset has the same number of samples but different 15 categories. In Tab. VII (c), we report the generalization gaps between closed seen and unseen test splits. We see that types of classes matter, meaning that an ideal training dataset should contain a variety of objects to generalize better.

4. Proposed TOD-Z groups on generalizability. As explained in Section 4, the original TOD can be grouped into four groups after the similarity analysis. We prepare four training sets for each group and measure the generalization gap. In Tab. VII (d), electronics and miscellaneous objects are

more difficult to learn, as the seen performances are generally lower than the other categories. Yet, they perform better on unseen objects, indicating that training with these samples helps with generalizability.

VI. CONCLUSION AND FUTURE WORKS

In this work, we study the problem of ZSIS from RGB-D images for cluttered indoor environments in a class-agnostic manner. Towards that, we analyze the generalization gap on an RGB-D dataset TOD and nominate a novel zero-shot RGB-D split, TOD-Z. We propose SupeRGB-D tailored for ZSIS, which first extracts small patches from RGB-D images and then learns to merge them through local visual and 3D cues. Our approach outperforms the baselines on unseen instances and is very lightweight with a 0.4 MB memory requirement that can be used on mobile robots and embedded devices. Future work includes incorporating uncertainty estimates to reflect sparsity and noise characteristics of real depth maps when fusing two modalities to improve the accuracy of resulting segmentation.

REFERENCES

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.

[2] A. M. Hafiz and G. M. Bhat, "A survey on instance segmentation: state of the art," *IJMIR*, vol. 9, no. 3, pp. 171–189, 2020.

[3] Z. Deng, S. Todorovic, and L. Jan Latecki, "Unsupervised object region proposals for rgb-d indoor scenes," *Computer Vision and Image Understanding*, vol. 154, pp. 127–136, 2017.

[4] W. Wang, R. Yu, Q. Huang, and U. Neumann, "Sgpn: Similarity group proposal network for 3d point cloud instance segmentation," in *CVPR*, 2018.

[5] Z. Liu, H. Tang, Y. Lin, and S. Han, "Point-voxel cnn for efficient 3d deep learning," *NeurIPS*, 2019.

[6] E. P. Örnek, S. Mudgal, J. Wald, Y. Wang, N. Navab, and F. Tombari, "From 2d to 3d: Re-thinking benchmarking of monocular depth prediction," 2022. [Online]. Available: https://arxiv.org/abs/2203.08122

[7] Y. Wang, X. Chen, L. Cao, W. Huang, F. Sun, and Y. Wang, "Multimodal token fusion for vision transformers," in *CVPR*, 2022.

[8] E. Simsar, E. P. Örnek, F. Manhardt, H. Dhamo, N. Navab, and F. Tombari, "Object-aware monocular depth prediction with instance convolutions," *IEEE RA-L*, vol. 7, no. 2, pp. 5389–5396, 2022.

[9] C. Xie, Y. Xiang, A. Mousavian, and D. Fox, "Unseen object instance segmentation for robotic environments," *IEEE T-RO*, 2021.

[10] D. Kim, T.-Y. Lin, A. Angelova, I. S. Kweon, and W. Kuo, "Learning open-world object proposals without learning to classify," *IEEE RA-L*, 2022.

[11] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: theory and its application to image segmentation," *IEEE TPAMI*, vol. 15, no. 11, pp. 1101–1113, 1993.

[12] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE TPAMI*, vol. 22, no. 8, pp. 888–905, 2000.

[13] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, vol. 59, no. 2, pp. 167–181, 2004.

[14] P. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," in *Neurips*, 2015.

[15] Y. Xian, S. Choudhury, Y. He, B. Schiele, and Z. Akata, "Semantic projection network for zero- and few-label semantic segmentation," in *CVPR*, 2019.

[16] M. Bucher, T.-H. VU, M. Cord, and P. Pérez, "Zero-shot semantic segmentation," in *NeurIPS*, 2019.

[17] Y. Zhao, Z. Zhong, N. Sebe, and G. H. Lee, "Novel class discovery in semantic segmentation," in *CVPR*, 2022.

[18] Z. Tian, X. Lai, L. Jiang, S. Liu, M. Shu, H. Zhao, and J. Jia, "Generalized few-shot semantic segmentation," in *CVPR*, 2022.

[19] D. A. Ganea, B. Boom, and R. Poppe, "Incremental few-shot instance segmentation," in *CVPR*, 2021.

[20] Y. Zheng, J. Wu, Y. Qin, F. Zhang, and L. Cui, "Zero-shot instance segmentation," in *CVPR*, 2021.

[21] W. Wang, M. Feiszli, H. Wang, J. Malik, and D. Tran, "Open-world instance segmentation: Exploiting pseudo ground truth from learned pairwise affinity," *CVPR*, 2022.

[22] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, "Kinect-fusion: Real-time dense surface mapping and tracking." in *ISMAR*. IEEE Computer Society, 2011, pp. 127–136.

[23] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[24] X. Ren, L. Bo, and D. Fox, "Rgb-(d) scene labeling: Features and algorithms," *CVPR*, 2012.

[25] Z. Deng, S. Todorovic, and L. Jan Latecki, "Semantic segmentation of rgbd images with mutex constraints," in *ICCV*, 2015.

[26] X. Gong and J. Liu, "Rock detection via superpixel graph cuts," in *ICIP*, 2012.

[27] A. Richtsfeld, T. Mörwald, J. Prankl, M. Zillich, and M. Vincze, "Segmentation of unknown objects in indoor environments," in *IROS*, 2012.

[28] J. Hou, A. Dai, and M. Niessner, "3d-sis: 3d semantic instance segmentation of rgb-d scans," in *CVPR*, 2019.

[29] N. Zhao, T.-S. Chua, and G. H. Lee, "Few-shot 3d point cloud semantic segmentation," in *CVPR*, 2021.

[30] M. F. Naeem, E. P. Örnek, Y. Xian, L. Van Gool, and F. Tombari, "3D Compositional Zero-shot Learning with DeCompositional Consensus," in *ECCV*, 2022.

[31] B. Michele, A. Boulch, G. Puy, M. Bucher, and R. Marlet, "Generative zero-shot learning for semantic segmentation of 3D point cloud," in *3DV*, 2021.

[32] T. Luo, K. Mo, Z. Huang, J. Xu, S. Hu, L. Wang, and H. Su, "Learning to group: A bottom-up framework for 3d part discovery in unseen categories," in *ICLR*, 2020.

[33] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, "Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data," in *ICRA*, 2019.

[34] S. Back, J. Kim, R. Kang, S. Choi, and K. Lee, "Segmenting unseen industrial components in a heavy clutter using rgb-d fusion and synthetic data," in *ICIP*, 2020.

[35] Y. Xiang, C. Xie, A. Mousavian, and D. Fox, "Learning rgb-d feature embeddings for unseen object instance segmentation," in *CoRL*, 2020.

[36] M. Laskey, B. Thananjeyan, K. Stone, T. Kollar, and M. Tjersland, "Simnet: Enabling robust unknown object manipulation from pure synthetic data via stereo," in *CoRL*, 2021.

[37] C. Xie, A. Mousavian, Y. Xiang, and D. Fox, "Rice: Refining instance masks in cluttered environments with graph neural networks," in *CoRL*, 2021.

[38] S. Back, J. Lee, T. Kim, S. Noh, R. Kang, S. Bak, and K. Lee, "Unseen object amodal instance segmentation via hierarchical occlusion modeling," in *ICRA*, 2022.

[39] M. Durner, W. Boerdijk, M. Sundermeyer, W. Friedl, Z.-C. Marton, and R. Triebel, "Unknown object segmentation from stereo images," in *IROS*, 2021.

[40] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *ICCV*, 2021.

[41] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE TPAMI*, vol. 34, no. 11, pp. 2274–2282, 2012.

[42] A. Das, Y. Xian, Y. He, B. Schiele, and Z. Akata, "Sp2net for generalized zero-label semantic segmentation," in *Pattern Recognition*. Cham: Springer International Publishing, 2021, pp. 235–249.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[44] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," no. arXiv:1512.03012, 2015.

[45] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," *CVPR*, 2017.

[46] A. Bansal, K. Sikka, G. Sharma, R. Chellappa, and A. Divakaran, "Zero-shot object detection," in *ECCV*, 2018.

[47] A. Dave, P. Tokmakov, and D. Ramanan, "Towards segmenting everything that moves," *ICCV Workshops*, 2019.

[48] M. Suchi, T. Patten, D. Fischinger, and M. Vincze, "Easylabel: A semi-automatic pixel-wise object annotation tool for creating robotic RGB-D datasets," in *ICRA*, 2019.

[49] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, p. 5–32, 2001.