

A Supplementary Materials for Attention-Guided Self-Supervised Distinctive Region Detection in Point Clouds

This document provides supplementary information and detailed results to complement our paper "Attention-Guided Self-Supervised Distinctive Region Detection in Point Clouds". We present extended analyses, ablation studies, and implementation details that further support and expand on the main findings reported in the primary manuscript. Our supplementary material includes: (1) a comprehensive description and visualization of our benchmark with ground-truth distinctive region labels, (2) detailed experimental protocols and implementation specifics, (3) additional quantitative and qualitative comparisons with state-of-the-art methods, (4) extended ablation studies. This supplementary information aims to offer a comprehensive understanding of our DRAG approach and its advancements in detecting distinctive regions in point clouds.

A.1 Datasets description

As described in the main paper, we employed a diverse range of point cloud datasets, encompassing both synthetic CAD models and real-world scans. For each dataset, we created splits of K related classes with noticeable distinguishing regions. In most cases $K = 2$, but we also experimented with $K = 3, 22$. Figure 7 illustrates representative point clouds from each class across the dataset splits used in our experiments. To establish per-point ground-truth labels for each point cloud in the dataset, we leveraged the existing part segmentation labels, carefully examining and selecting distinct regions. We specifically chose parts that uniquely differentiate the object class from similar classes within the dataset. For distinctive points, we assigned $\mathbf{y}_{ij} = 1$, while non-distinctive points were labeled as $\mathbf{y}_{ij} = 0$. Figure 8 provides a visual representation of the per-point ground truth distinctiveness labels $\mathbf{y}_i$ that we generated for representative point clouds from each class across the dataset splits.

Since the number of distinctive points relative to the total number of points in the point cloud varies across different dataset splits, we selected a unique threshold for each split to derive the predicted per-point distinctiveness labels from the attention weights. This threshold helps convert the attention weights, which are scalar values in the range $\mathbf{w}_i = \{w_{ij} \in [0, 1]\}_{j=1}^{N_i}$, into the predicted labels $\hat{\mathbf{y}}_i = \{\hat{y}_{ij} \in \{0, 1\}\}_{j=1}^{N_i}$, which are binary indicators of whether a point is considered distinct by the model. We evaluated different percentages of points considered distinct (e.g., top 10% of $\mathbf{w}_i$ values assigned $\hat{y}_{ij} = 1$, others 0). Specifically, we examined each percentage from 5 to 70, in increments of 5. We selected the percentage that yielded the highest mIoU on a validation set for each dataset split for both our model and that of [13]. We labeled all points in the top attention weight value percentile as distinct (i.e., $\hat{y}_{ij} = 1$), and all other points as non-distinct (i.e., $\hat{y}_{ij} = 0$). Table 13 describes the percentage of distinctive points relative to the whole point cloud.

A.2 Implementation details

We implemented our approach using PyTorch [18]. Data augmentation techniques included random scaling, rotation, translation, jittering, and dropout. For the hyperparameters in

Dataset	Split	Pct. (%)
ShapeNet-Part	4 vs. tail engine planes	10
	1 vs. 2 wing rockets	30
	handbags vs. bags	5
	0 vs. 4 wheels skateboards	10
	chairs vs. tables	25
PartNet	mugs vs. bottles	10
COSEG	bottles vs. jars	30
	vases vs. mugs	5
	chairs vs. stools	50
EngravedCubes	apples vs. hearts	55
	bats vs. cars vs. horseshoes	65
	all classes	70
ScanObjectNN	chairs vs. tables	55

Table 13 Dataset split thresholds. Percentage of distinctive points chosen for each dataset split for the mIoU calculation.

Equations (14) and (17), we used $\alpha_1 = 0.05$, $\alpha_2 = 2.0$, and $\alpha_4 = 100.0$. α_3 was set to 0 for the initial 50% of training and 0.1 for the remainder. We experimented with several hyperparameter values, using mIoU as our selection criterion. We employed the AdamW optimizer with a CosineAnnealingLR scheduler, using an initial learning rate of $5 \cdot 10^{-5}$. Training was conducted on an NVIDIA RTX A6000 GPU with 48 GB memory. Training duration varied across dataset splits, typically lasting several hours.

A.3 Experimental setup

In our self-supervised and few-shot experiments, each configuration was tested once due to time constraints, precluding multiple runs and averaging. For experiments utilizing our distinctiveness scores to guide sampling in classification, retrieval, and reconstruction tasks, we performed three trials per experiment and reported the mean and standard deviation of the results.

A.4 Additional results

Few-shot setup. We conducted an extended evaluation of DRAG’s performance as more ground-truth labels are added in the few-shot learning setup across all dataset splits. Figure 9 visualizes these results. The findings indicate that additional ground-truth labels generally improve performance, with even a small number of labels often sufficient for the model to generalize effectively to new examples.

Self-supervised vs. 1-shot. Figure 10 shows a qualitative comparison between our model’s performance in the self-supervised and 1-shot configurations. As illustrated in the figure, the distinctive regions identified by our model in the 1-shot setup are more precise. This demonstrates how providing even a single example per class helps improve the model’s performance.

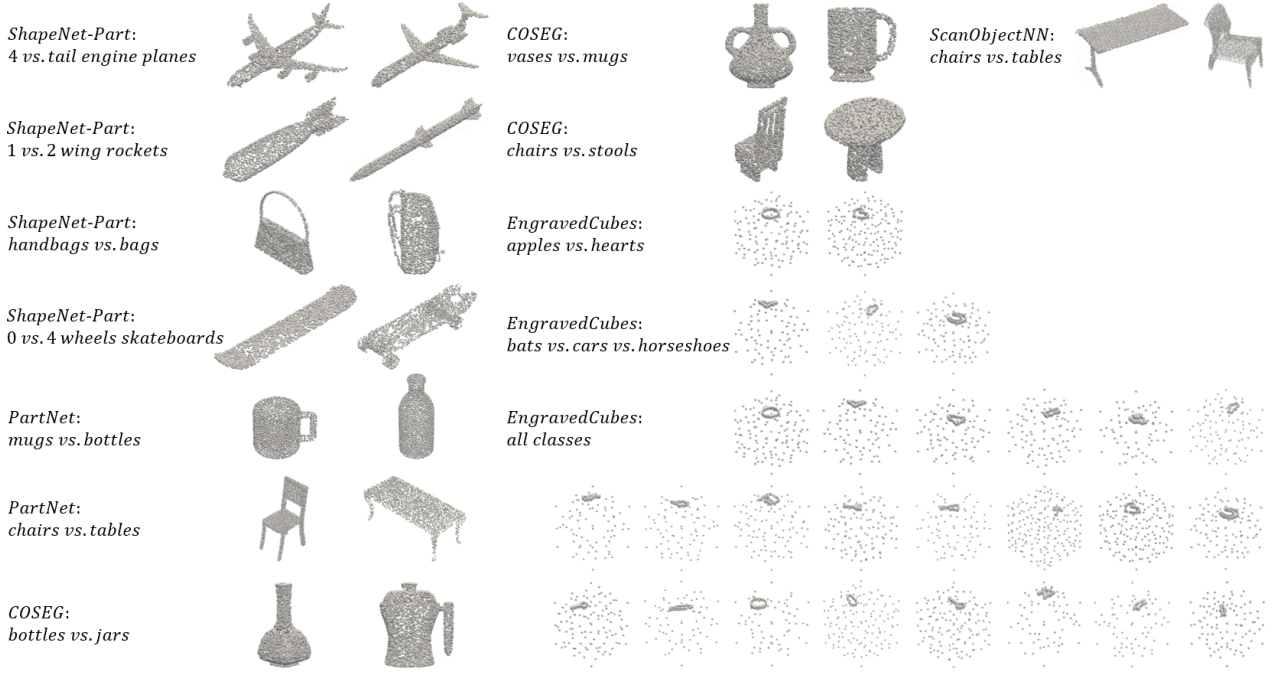


Fig. 7 Visualization of dataset splits. Representative objects from each dataset split utilized in our study.

$\mathcal{L}_{CC}$	$\mathcal{L}_{C_1}$	$\mathcal{L}_{C_2}$	$\mathcal{L}_{SC}$	$\mathcal{L}_{FS}$	Setup	mIoU
✓					SSL	0.41
	✓					0.42
		✓				0.42
✓		✓				0.45
	✓	✓	✓			0.52
✓	✓	✓				0.52
✓	✓	✓	✓			0.55
✓	✓	✓	✓	✓	1-shot	0.51
✓	✓	✓	✓	✓		0.57

Table 14 Ablation studies. Impact of various method components. Experiments conducted using the mugs vs. bottles split from the PartNet dataset.

Dataset	Split	mIoU	
		KLDiv	CosSim
ShapeNet-Part	4 vs. tail engine planes	0.48	0.55
	handbags vs. bags	0.56	0.69
	chairs vs. tables	0.47	0.50
PartNet	mugs vs. bottles	0.55	0.57
COSEG	vases vs. mugs	0.49	0.58
	chairs vs. stools	0.47	0.58
	apples vs. hearts	0.66	0.78
EngravedCubes	bats vs. cars vs. horseshoes	0.69	0.77
	all classes	0.59	0.69
ScanObjectNN	chairs vs. tables	0.50	0.56

Table 15 Few-shot loss selection. Quantitative comparison between KL divergence and cosine similarity metrics for the few-shot loss term in 1-shot distinctive region detection.

A.5 Additional ablation study results

Contribution analysis of each loss component. To supplement the ablation study in the main paper and show the consistency of its results across different dataset splits, Table 14 presents an additional ablation study on the PartNet dataset’s mugs vs. bottles split. This study assesses the contribution of each loss component in DRAG to the mIoU metric. We tested various loss function combinations.

The findings indicate that incorporating all components yields the highest mIoU, while omitting any loss reduces performance. Notably, using only the few-shot loss is less effective than employing all self-supervised losses without it, emphasizing the crucial role of self-supervised losses in this task.

Few-shot loss selection. To determine the optimal few-shot loss function for our model, we evaluated different loss terms, including KL divergence and cosine similarity. Results presented in Table 15 demonstrate that cosine similarity con-

sistently outperforms alternatives, yielding superior performance for our distinctive region detection task.

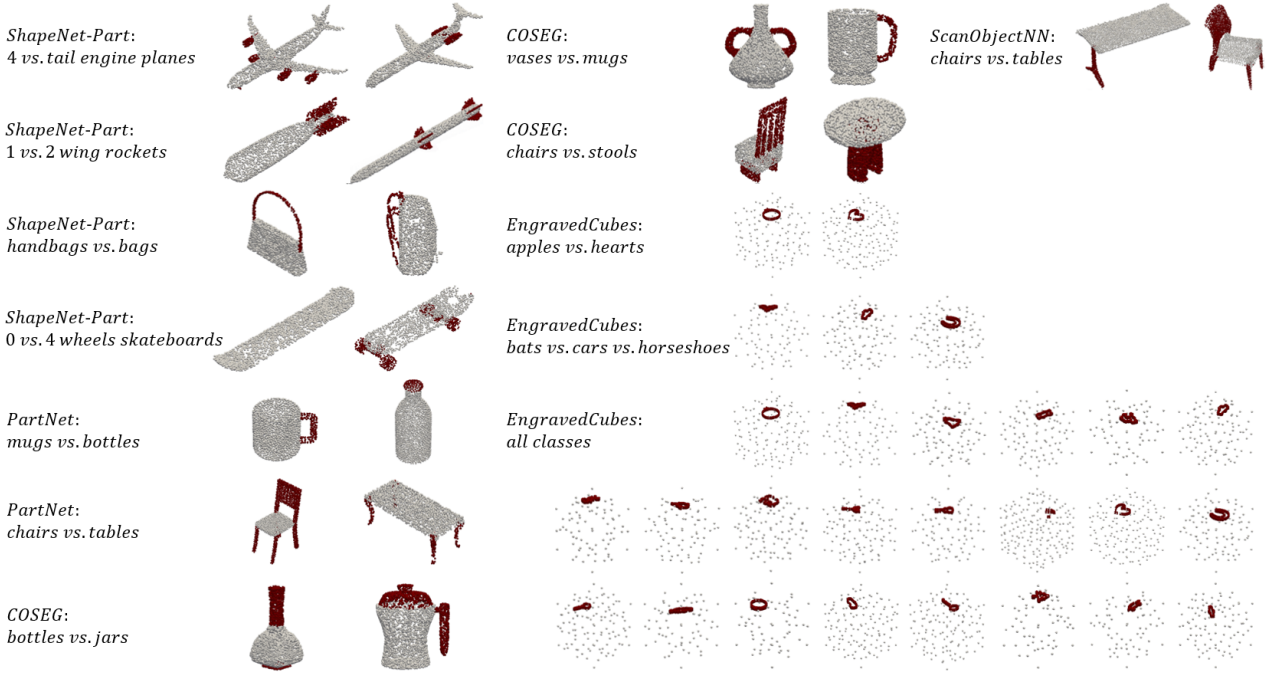


Fig. 8 Visualization of ground-truth labels. Illustration of the per-point ground truth distinctiveness labels Y_i for representative objects from each dataset split employed in our study.

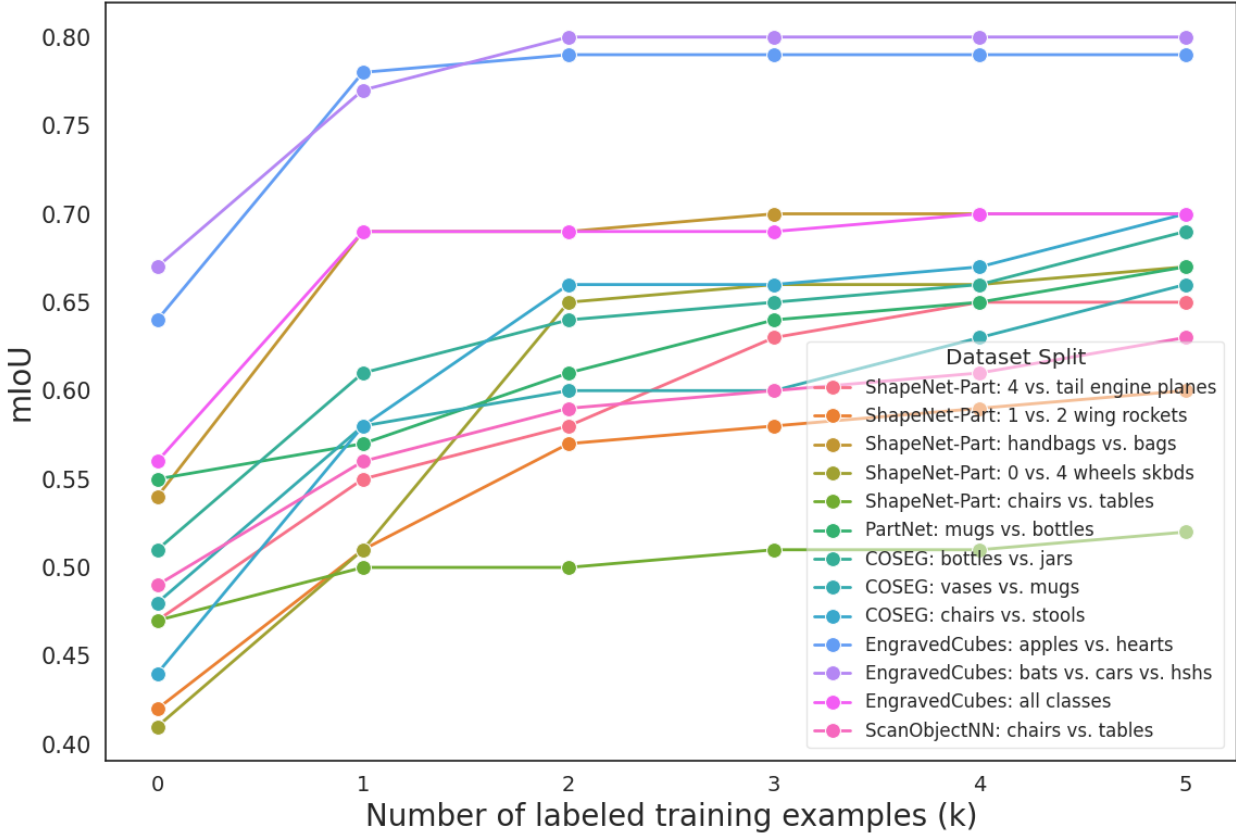


Fig. 9 Few-shot learning setup. The plot shows DRAG's performance relative to the number of labeled examples in a few-shot learning context. The $k = 0$ case represents the self-supervised setup.

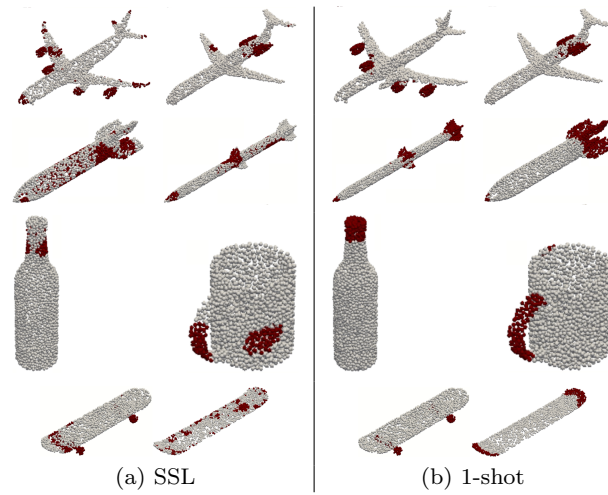


Fig. 10 Self-supervised vs. 1-shot. Qualitative comparison between our model’s detection results using self-supervised (a) and 1-shot learning (b) approaches across different dataset splits. Distinctive points are highlighted in maroon.