PSA-SSL: Pose and Size-aware Self-Supervised Learning on LiDAR Point Clouds

Supplementary Material

6.1. Sensitivity to hyperparameters

In this section, we study the sensitivity of PSA-SSL to LiDAR pattern augmentation probabilities and clustering ϵ. All models, unless mentioned otherwise, are pretrained on 10% Waymo for 30 epochs and fine-tuned on 1% of the labels for 15 epochs.

6.1.1. Sensitivity to pattern augmentation probabilities

The hyperparameters for LiDAR pattern augmentation are the probabilities with which we randomly sample different LiDAR configurations ('v32', 'v64', 'o64'). Tab. 8 shows semantic segmentation performance when the probabilities for randomly sampling LiDAR configurations are changed. For simplicity, we increase the probability for 'v32' from 0.3 to 0.6 and equally divide 1-prob(v32) between 'v64' and 'o64'. As expected, higher probabilities for v64 and o64 (*i.e.* prob(v32) $\in [0.3, 0.4]$) give better performance on dense LiDAR (Waymo and SemanticKITTI). Increasing v32 probability from 0.3 to 0.4 and 0.5 improves performance on nuScenes, but beyond 0.5, the performance on all datasets decreases, possibly due to a decrease in visible clusters during pretraining. Increasing the number of pretraining epochs from 30 to 200 allows the prob(v32)=0.6 to catch up to the performance of prob(v32)=0.4. Ablation studies in Tab. 7 also show that with prob(v32)=0.6, SegContrast benefits from LiDAR pattern augmentation across all datasets (see row 1 vs row 3). Hence, we conclude that our method is robust to the pattern augmentation probabilities.

Epochs	Prob. (v32)	1% Waymo	1% nuScenes	1% SemKITTI
Pretrain=30 Fine-tune=15	No pretraining	23.24	23.59	19.51
	0.3	37.8	32.76	**36.06**
	0.4	**39.79**	**33.80**	34.96
	0.5	37.06	33.04	34.46
	0.6	36.90	32.83	34.54
Pretrain=200 Fine-tune=100	0.4	**54.61**	37.56	51.71
	0.6	54.36	**37.89**	**52.11**

Table 8. Semantic Segmentation (mIoU) sensitivity to LiDAR pattern augmentation probabilities.

6.1.2. Sensitivity to clustering epsilon

Incorrectly setting the clustering ϵ may lead to over or under-segmentation for SegContrast, which can also affect the bounding box regression pretext task, resulting in lower semantic segmentation performance. Since SegContrast [18] does not study the effect of clustering ϵ on their method, we analyse the impact of changing ϵ on PSA-SC's segmentation performance. Table 9 shows that the performance of PSA-SC does not drop dramatically in the range of $\epsilon \in [0.2, 0.3]$. Hence, there is a range of ϵ, for which the performance is robust. We also note that PSA-SC outperforms training from scratch at all clustering ϵ and outperforms its baseline SC at the best $\epsilon = 0.2$. From the table, we can see that $\epsilon = 0.2$ is closest to the ideal ϵ, which matches the value we set by visually tuning on a few frames. Since visually tuning ϵ is trivial, we do not consider it as a limitation of our method.

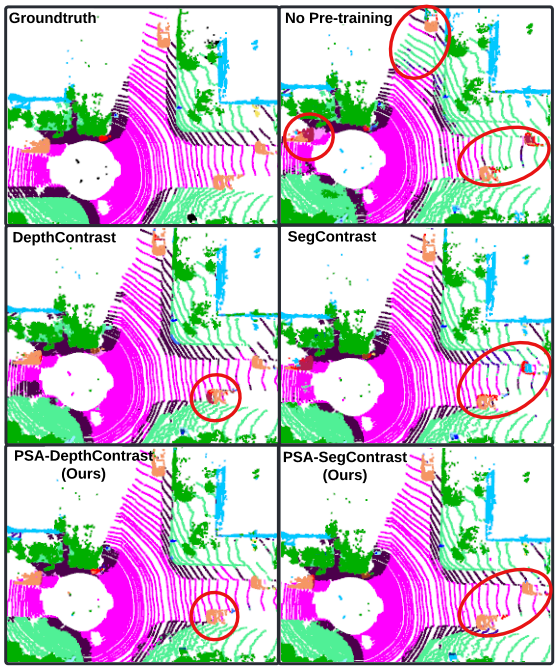

Figure 3. Comparison of qualitative semantic segmentation results of PSA-DepthContrast and PSA-SegContrast against their original baselines [18, 38] on SemanticKITTI validation scan.

Method	No Pretraining	SC	PSA-SC (Ours)			
ϵ	-	0.2	0.1	0.2	0.3	0.4
mIoU	23.24	38.50	36.71	**39.79**	38.22	36.06

Table 9. Semantic Segmentation (mIoU) sensitivity to clustering ϵ. Models are fine-tuned on 1% Waymo.

7. Qualitative Results

Figure 3 shows a qualitative comparison of semantic segmentation between our approach and baselines. As expected, training from randomly initialized backbone errs on

vehicles as well as leaks labels to neighbouring stuff classes. DepthContrast and SegContrast exhibit label confusion on vehicles but improve on stuff classes. Incorporating PSA, significantly minimizes the label confusion on vehicles, indicating the ability to learn features that can capture full extent of objects in the scene, resulting in better geomtery-aware class discrimination.