This supplementary material contains the following four sections. In Section A, we show that point-to-pixel level contrastive losses like PPKT [17] can also benefit from the proposed semantically tolerant loss. In Section B, we visualize the class imbalance in nuScenes dataset on the superpixel level and then report the per-class fine-tuning performance of SLidR and ST-SLidR represenations on nuscenes dataset. In Section C, we visualize the superpixel-to-superpixel similarity across a range of 2D self-supervised pretrained models. Finally, In Section D, we discuss the limitations of ST-SLidR.

A. Semantically Tolerant PPKT

We conduct an experiment to evaluate whether pixel-to-pixel semantic similarity can be used to improve the quality of learned representations of pixel-to-point contrastive losses like PPKT [17]. The main challenge of utilizing pixel-to-pixel similarity to infer false negative pixels, is the high level of noise compared to superpixel-to-superpixel similarity. Starting with the implementation of PPKT provided by SLidR's code base [21], we implement $\mathcal{L}_{knn}$ on the pixel level. We run two sets of experiments using 4096 and 8192 point-pixel pairs per batch. We use a batch size of 16. Here, we report the average of 3 runs for each experiment. In Table 7, we observe that semantically tolerant PPKT loss provides a modest improvement over PPKT [17]. We also experimented with balancing PPKT using aggregate pixel-to-pixels similarity but due to the high level of noise in pixel level similarity, we did not observe any significant improvement.

Method	Number of Samples	nuScenes Lin. Prob 100%	nuScenes Finetune 1%
PPKT	4096	35.90	37.52
ST-PPKT	4096	**36.70**	**38.32**
Improvement		*+0.80*	*+0.80*
PPKT	8192	35.57	38.01
ST-PPKT	8192	**36.64**	**38.60**
Improvement		*+1.07*	*+0.59*

Table 7. Pixel-to-Pixel feature similarity used to remove the closest k nearest negative pixels identified as false negatives. Here, we show results for 4096 and 8192 pixel-point contrastive pairs per batch. We report semantic segmentation results on nuScenes.

B. Class Imbalance

B.1. Superpixel Class Imbalance

In Figure 4, we show the distribution of classes in the nuScenes [6] dataset at the superpixel level. To determine the class of a superpixel, we first project the LiDAR point cloud onto the 2D image. The class of a superpixel is given by ground truth LiDAR point-wise labels of the points within the superpixel of interest. Specifically, its class is the same as its LiDAR points' label. In the cases where LiDAR points of multiple classes occur within a superpixel, we assign the class of the superpixel to be the mode of the points' LiDAR labels. We exclude the superpixels without LiDAR points as they are not used in pretraining. Note that the "others" category on the pie chart includes movable objects such as traffic cones and barriers. We observe that only 8.9% of the superpixels cover moving objects like vehicles and pedestrians, while a large portion of the superpixels correspond to static classes like driveable surface, vegetation, and manmade. Thus, the pretraining loss of PPKT and SLidR is dominated by gradients from over-represented classes. It is important to note that accurately segmenting moving objects is critical for autonomous driving agents as they share the same environment and their actions will affect the agent. ST-SLidR specifically improves the quality of representations of minority classes which includes moving objects (see Table 4 and Table 8).

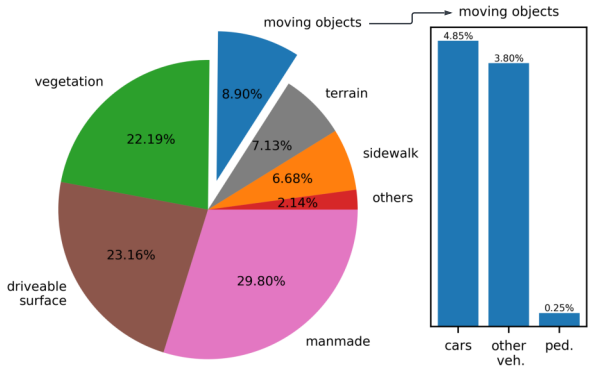

Figure 4. Class distribution of nuScenes dataset at superpixel level.

B.2. Per-class Performance

Table 8 shows the average per-class performance finetuning SLidR and ST-SLidR on 1% of nuscenes dataset. We observe that 3D representations learned by ST-SLidR significantly improve performance on minority classes like moving objects. For instance, we see an improvement of +5.5% IoU on motorcyclists which consists of less than 0.04% of the superpixels, +3.7% IoU on pedestrians which consists of 0.25% of the superpixels and 5.7% IoU on trucks which consists of 2.11% of the superpixels.

Method	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	driv. surf.	other flat	sidewalk	terrain	manmade	vegetation	mIoU
Random	0.0	0.0	8.1	65.0	0.1	6.6	21.0	9.0	9.3	25.8	89.5	14.8	41.7	48.7	72.4	73.3	30.3
SLidR	0.0	1.8	15.4	73.1	1.9	19.9	47.2	17.1	14.5	34.5	92.0	27.1	53.6	61.0	79.8	82.3	38.8
ST-SLidR	0.0	**2.7**	**16.0**	**74.5**	**3.2**	**25.4**	**50.9**	**20.0**	**17.7**	**40.2**	92.0	**30.7**	**54.2**	61.1	**80.5**	**82.9**	**40.7**
Improvement	*+0.0*	*+0.8*	*+0.6*	*+1.4*	*+1.3*	*+5.5*	*+3.7*	*+2.9*	*+3.1*	*+5.7*	*+0.0*	*+3.6*	*+0.6*	*+0.0*	*+0.7*	*+0.6*	*+1.9*

Table 8. Per-class 3D semantic segmentation using 1% of labelled data for fine-tuning on nuscenes dataset on official validation set. We report the mean performance of 3 pretrained SLidR and ST-SLidR models.

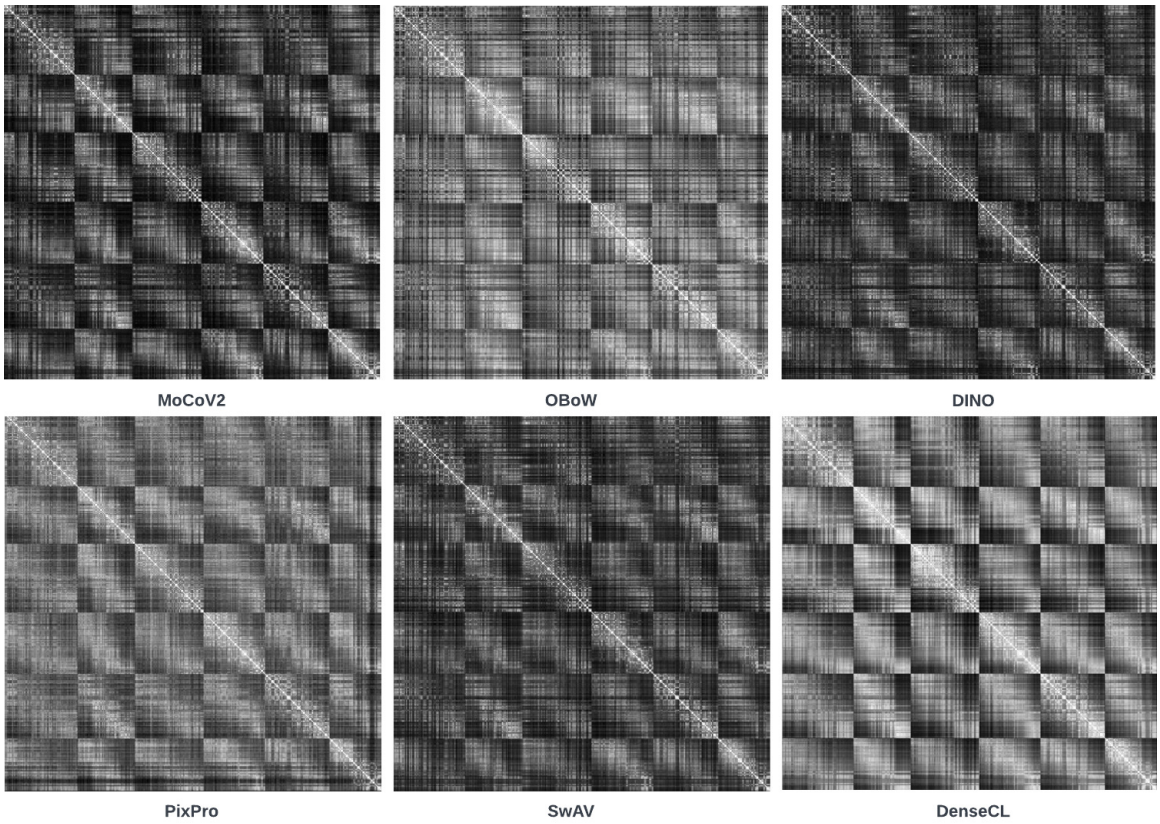

Figure 5. Superpixel-to-superpixel cosine similarity for an entire scene consisting of 6 cameras. Here, we show the similarity estimated using different self-supervised 2D pretrained models.

C. Superpixel Semantic Similarity

In Figure 5, we use pretrained weights from multiple 2D SSL frameworks to extract superpixel features from the 6 cameras covering a single scene from nuscenes dataset. Then, we compute the superpixel-to-superpixel cosine similarity ranging from 0.0 (black) to 1.0 (white). Figure 5 shows that 2D SSL frameworks learn different representations, however, we can see that similarity patterns appear to be consistent across different frameworks. ST-SLidR assumes that the value of cosine similarity can be different across different pretrained models, but the relative of order of similarity with respect to an anchor is more consistent. This is demonstrated in Table 2, where ST-SLidR provides significant gain over SLidR across multiple 2D pretrained models.

D. Limitations

D.1. Fixed Number of Negative Samples

We address the issue of contrasting semantically similar point and image regions by excluding a subset of the closest negative samples to the anchor from the pool of negative samples. Since the K nearest neighbours are excluded,

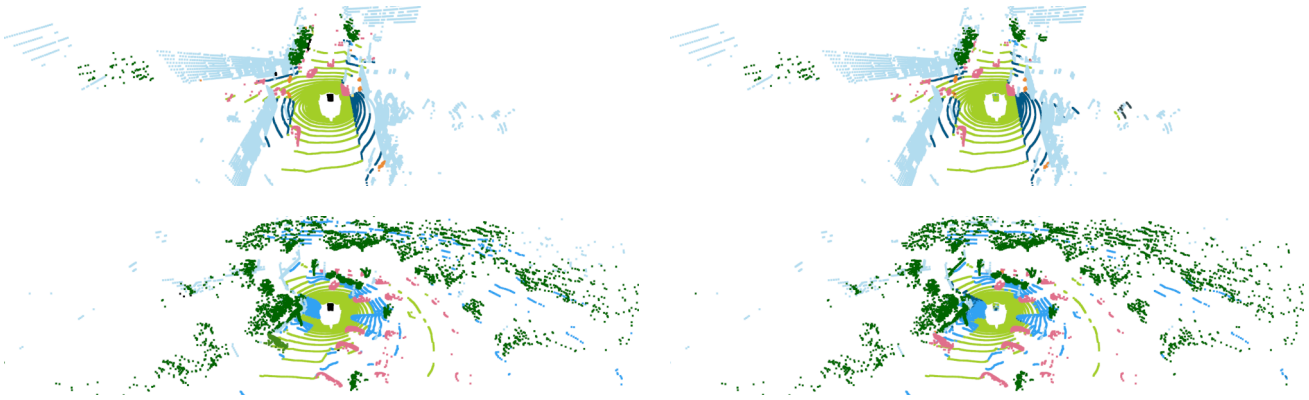

Figure 6. Ground truth (left) and ST-SLidR (right) segmentation results on the nuScenes dataset. ST-SLidR is finetuned on 1% of the data.

a fixed number of false negative samples are identified for each anchor. However, Figure 2 shows that the number of semantically similar samples greatly vary based on the semantic class of the anchor. For instance, the number of samples similar to a road or a vegetation anchor is much larger than the number of samples similar to a car or a pedestrian anchor. This is mainly due to the severe class imbalance in autonomous driving datasets. A potential solution for future work is to design an adaptive K nearest neigbour loss, where the value of K is a function of the aggregate sample-to-samples similarity. Over-represented anchors are similar to many negative samples in a batch and therefore the value of K should be higher for these anchors than under-represented anchors.

D.2. Frozen Image Encoder

Authors in SLidR [21] observe that backpropagating gradients to the image encoder can result in degenerate solutions, where the contrastive loss is easily minimized without learning useful 3D representations for downstream tasks. One of the advantages of updating the image encoder parameters initialized by ImageNet pretrained weights, is to learn optimal 2D features for autonomous driving scenes. To prevent degenerate solutions, the image encoder can be first initialized with ImageNet pretrained weights, and any 2D SSL framework can be used to learn optimal 2D representations for autonomous driving images. Finally, the image encoder is frozen and then ST-SLidR can be used to transfer knowledge from the 2D features to the point cloud encoder.