SegContrast: 3D Point Cloud Feature Representation Learning through Self-supervised Segment Discrimination

Lucas Nunes Rodrigo Marcuzzi Xieyuanli Chen Jens Behley Cyrill Stachniss

***Abstract*—Semantic scene interpretation is essential for autonomous systems to operate in complex scenarios. While deep learning-based methods excel at this task, they rely on vast amounts of labeled data that is tedious to generate and might not cover all relevant classes sufficiently. Self-supervised representation learning has the prospect of reducing the amount of required labeled data by learning descriptive representations from unlabeled data. In this paper, we address the problem of representation learning for 3D point cloud data in the context of autonomous driving. We propose a new contrastive learning approach that aims at learning the structural context of the scene. Our approach extracts class-agnostic segments over the point cloud and applies the contrastive loss over these segments to discriminate between similar and dissimilar structures. We apply our method on data recorded with a 3D LiDAR. We show that our method achieves competitive performance and can learn a more descriptive feature representation than other state-of-the-art self-supervised contrastive point cloud methods.**

***Index Terms*—Semantic Scene Understanding, Deep Learning Methods, Representation Learning**

I. Introduction

FINE-GRAINED scene interpretation is crucial to understand the surroundings of a robot or autonomous vehicle. Given the extensive research effort on convolutional neural networks (CNNs) [23], [56], [5], 2D image tasks have been defined and adapted to this scope, e.g., instance segmentation [33], [53] or panoptic segmentation [32], [57]. For 3D scene understanding, we can employ semantic segmentation [58], [43] to achieve a pixel-level classification of an image. Such semantic information is important for autonomous systems to interpret and safely interact with their surroundings.

Image data however does not provide straightforward 3D information, which is also essential in this domain. LiDAR sensors provide 3D information in the form of spatial positions and intensity values as point-wise information. Such data is often more challenging to interpret than images due to the lack of color information and the sparser representation of the objects. Therefore, many recent studies [34], [38], [40], [19], [10], [44] focus on convolution operations applied to point clouds to boost their performance.

Manuscript received: Sep 8, 2021; Revised: Nov 29, 2021; Accepted: Dec 31, 2021. This paper was recommended for publication by Editor Markus Vincze upon evaluation of the Associate Editor and Reviewers' comments.

This work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, EXC-2070–390732324–PhenoRob. by the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017008 (Harmony). All authors are with the University of Bonn, Germany.

Digital Object Identifier (DOI): see top of this page.

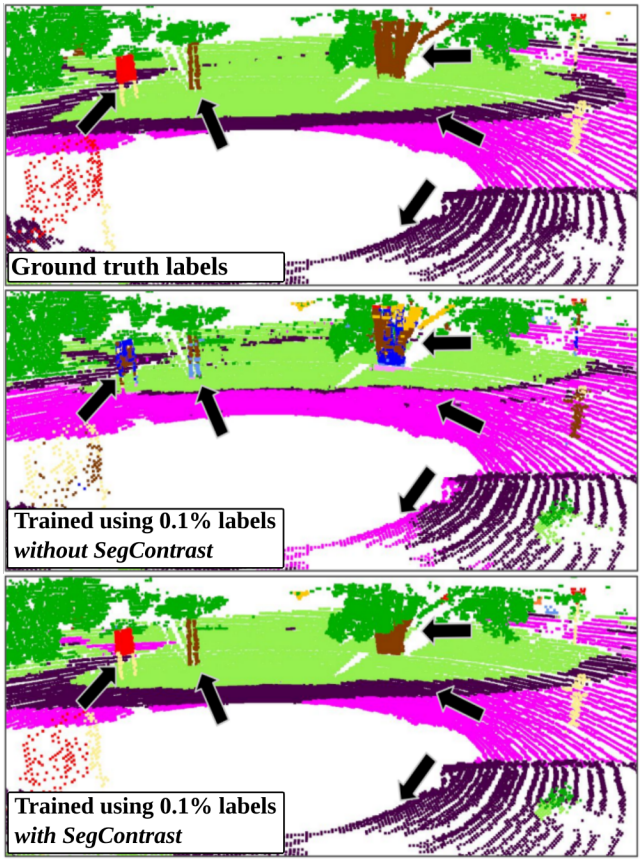

Fig. 1: Results trained with only 0.1% of the labels, trained from scratch (middle), i.e., without any pre-training, and pre-training with our approach SegContrast (bottom). With SegContrast pre-training, the semantic segmentation can better delineate the structural information in the scene and better classify the more fine-grained objects, like trees, traffic signs and poles, highlighted by black arrows.

Supervised methods need a substantial amount of labeled data to achieve high performance, which is particularly hard to acquire for LiDAR data. Data annotation is expensive [3] due to the density of measurements, which are sensor-specific and also depend on the sensor mounting. For semantic segmentation, point-wise labeling is required, making it even harder to annotate. There are also fewer labeled LiDAR datasets available compared to image datasets [11], [30]. Given the sensor-specific characteristics, learned features from one dataset cannot be easily transferred to a dataset with a different sensor setup. Therefore, recent work in this domain address label efficiency [51], [27], [47] or transferability across datasets [28], [29], [52], [1], [35].

Self-supervised representation learning has the prospect of using the data in an unsupervised way to learn a robust feature presentation that can be transferred to different downstream supervised tasks. One of the self-supervised learning methods that received increasing attention recently is contrastive learning [22], [6], [7], [20], [8]. These methods take advantage of data augmentation to generate augmented versions of one anchor sample, then learn a representation to approximate these augmented samples with similar features in the feature space and put it apart from different samples. Based on the pre-trained networks using this paradigm, the network is fine-tuned for different downstream tasks.

The main contribution of this paper is a contrastive representation learning method for 3D LiDAR point clouds that is able to learn structural information. Our method extracts segments from the LiDAR data and uses contrastive learning to discriminate between similar and dissimilar structures. The learned feature representation is then used as a starting point for supervised fine-tuning, reducing the number of labeled training data needed. Our results suggest that our method can better learn the structural information (see Fig. 1) and a more descriptive feature representation during the self-supervised pre-training, surpassing previous point cloud-based contrastive methods in different evaluations. We explicitly show that, compared to the state of the art, our approach (i) is more efficient when using fewer labels, (ii) can better describe fine-grained structures, and (iii) is more transferable between different datasets.

The implementation of our approach is publicly available at https://github.com/PRBonn/segcontrast.

II. Related Work

Contrastive learning methods are receiving increasing attention in the vision community. Such methods improve the performance of different image-based classification methods using fewer or even no labels [45], [6], [7], [22], [20], [54], [8]. Contrastive learning takes advantage of data augmentation to generate two distinct versions of one anchor sample, creating a positive pair. Then, they train the network to learn a feature representation that maximizes the similarity between this positive pair and minimizes the similarity with other so-called negative samples.

Recent contrastive learning works aimed at fine-grained tasks, such as semantic segmentation [46], or object detection [24]. These methods apply the contrastive loss over image segments extracted using a class-agnostic segmentation approach. Such works improved the network performance using the contrastive loss for pre-training [24], [46] or as an auxiliary supervised loss [48], [37].

Point cloud data has been the source of extensive research in the field of autonomous vehicles [21]. Much effort has been dedicated to applying convolutional operations over point cloud data either by projecting the point cloud to images [34], [32], [49], [12] or by defining 3D convolution operations [38], [40], [19], [10], [44]. Between both approaches, 3D convolutions gained recently more visibility on LiDAR data due to their performance on different tasks, e.g., semantic segmentation [58], [43] and panoptic segmentation [16], [57].

The compelling results of contrastive learning methods on 2D image data drew attention to the application of such techniques on 3D data, especially in autonomous driving, for example, for semantic segmentation [58], [43] and object detection [9], [39] tasks. Xie et al. [50] address the point cloud contrastive learning using a point-wise loss. However, the point-wise contrastive loss depends on a involved pre-processing to generate a map of correspondent points between sequential scans. Hou et al. [26] add more context to the contrastive pre-training by dividing the scene into different spatial partitions, learning both the feature representation and the spatial partitioning of the points. Zhang et al. [55] propose a more global contrastive loss. For each scan a pair of augmented views are generated, i.e., the positive pair. Then, the other augmented scans are taken as the negative samples, computing the contrastive loss over extracted features from the scans. However, their method relies on a multi-branch architecture using two backbones, one for points and one for voxels. Despite promising results, none of the methods focus on point clouds generated by an automotive LiDAR sensor.

Our contribution is a contrastive representation learning method for outdoor LiDAR data used in autonomous driving. We extract class-agnostic segments from the point cloud and propose a contrastive loss to be applied over the extracted segments. Distinct from prior work, our representation learning method learns more contextualized information by discriminating the segmented structures on the point cloud and it learns a more robust and descriptive embedding space.

III. Our Approach

Fig. 2 provides an overview of our approach. We rely on a class-agnostic point cloud segmentation to segment the structures in the scene. Unlike most prior work, our method uses a single backbone and does not require a point-wise mapping. We use a momentum encoder network and a feature bank [22] during training to increase the number of negatives samples and contrast structures segmented over different scans. Then, point-wise features are computed for the whole point cloud and the mapping of the segment is used to extract the segment-wise points and features. We apply dropout [42] and global max pooling for each segment and compute a feature vector using an MLP projection head [6]. After that, we calculate the contrastive loss over the segments and update the feature bank with these segments features. In the next sections, we provide more details on the individual parts of our method.

A. Unsupervised Segment Extraction

Our method relies on segments of different structures in the point cloud. In outdoor LiDAR data, the different objects in a scene are mainly connected by the ground and usually better separated compared to indoor scenes. Given this characteristic, it is comparably easy to extract segments without labels in two steps by removing first the ground and then clustering the remaining points [4], [25], [18].

Given a point cloud $\mathcal{P} = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_N\}$ with $|\mathcal{P}| = N$ points $\boldsymbol{p}_i \in \mathbb{R}^3$, we can fit the ground plane and partition the point cloud into ground $\mathcal{G}$ and non-ground points $\mathcal{P}'$,

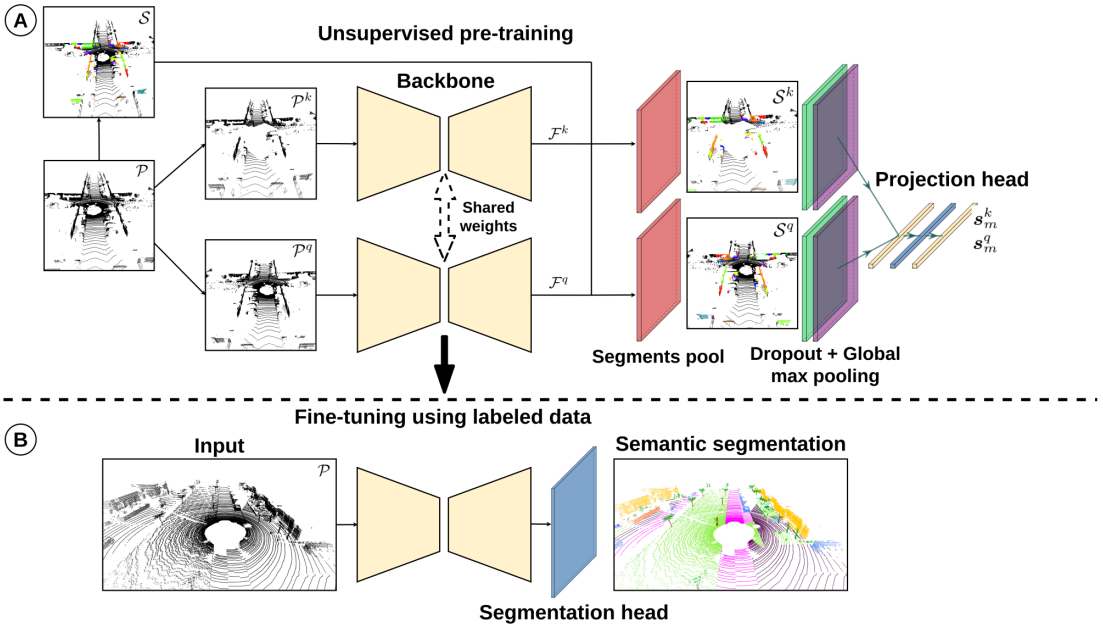

Fig. 2: (A) From a point cloud $\mathcal{P}$, we generate the augmented views $\mathcal{P}^q$ and $\mathcal{P}^k$ by data augmentation and extract class-agnostic segments $\mathcal{S}$ from $\mathcal{P}$. We compute the point-wise features $\mathcal{F}^q$ and $\mathcal{F}^k$ and determine the augmented segments $\mathcal{S}^q$ and $\mathcal{S}^k$ with its point-wise features using the point indexes of $\mathcal{S}$ extracted from $\mathcal{P}$. Then, we apply dropout followed by global max pooling over each segment, and we project the segment feature vectors using the projection head to get the final features $\boldsymbol{s}_m^q$ and $\boldsymbol{s}_m^k$ from the M segments and compute the contrastive loss. (B) After the pre-training, we fine-tune the pre-trained backbone for the downstream task, i.e., semantic segmentation.

such that $\mathcal{P} = \mathcal{P}' \cup \mathcal{G}$ and $\mathcal{P}' \cap \mathcal{G} = \emptyset$, similarly to prior approaches [4], [25]. Then, using a clustering algorithm, we can divide $\mathcal{P}'$ into M segments $\mathcal{S}_m$, such that $\mathcal{P}' = \bigcup_{m=1}^{M} \mathcal{S}_m$ and $\bigcap_{m=1}^{M} \mathcal{S}_m = \emptyset$, i.e., the segments are mutually exclusive. Each segment $\mathcal{S}_m$ in this partition represents a different structure from the original point cloud.

More specifically, we use RANSAC [15] to fit the ground plane and define the ground $\mathcal{G}$ and non-ground points $\mathcal{P}'$. Given the fitted plane and a distance threshold α, the inliers (ground) and outliers (non-ground) points are partitioned. We use DBSCAN [14] to cluster the non-ground points $\mathcal{P}'$ and determine the segments $\mathcal{S}_m$.

A common problem of such class-agnostic segmentation is the aspect of over- and under-segmentation [4]. To overcome this and extract representative segments, we define a minimum number of points ε in a cluster to be considered a segment. Moreover, since we will have a different number of segments for every point cloud segmentation, we select the δ segments with the highest number of points to avoid memory overflow during training. The remaining segments are also added to the set of ground points $\mathcal{G}$.

Despite its simplicity, this segmentation method can divide the scene into distinct structures, as illustrated in the Fig. 3. To save computations while training, we cache the segments after the first pass over the point clouds.

B. Segment Augmentation

With each point assigned to a segment, we apply data augmentations to generate the segments augmented pairs $\mathcal{S}^q$ and $\mathcal{S}^k$. We extract random views, $\mathcal{P}^q$ and $\mathcal{P}^k$, by cropping a random cuboid region from the anchor point cloud $\mathcal{P}$ [55]. Then, we apply random augmentations individually over the

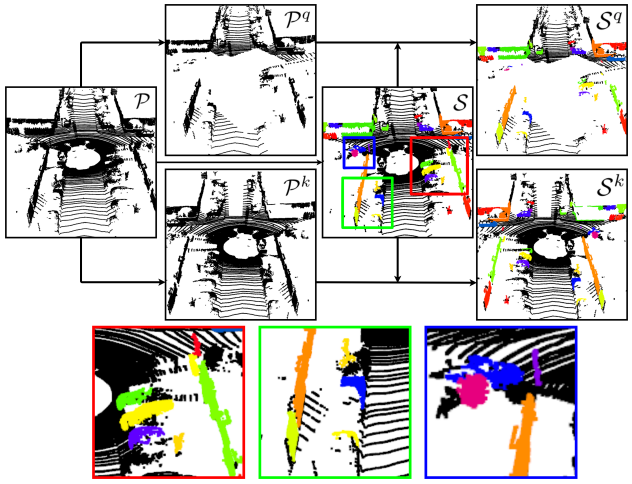

Fig. 3: From $\mathcal{P}$, we extract the segments $\mathcal{S}$ and generate the augmented views $\mathcal{P}^q$ and $\mathcal{P}^k$. With the augmented views and the segments given as point indexes, we can extract the set of augmented segments $\mathcal{S}^q$ and $\mathcal{S}^k$. We highlight some of the structures for a better visualization of the segments (solid colored squares).

point clouds $\mathcal{P}^q$ and $\mathcal{P}^k$. We use random rotation around the z axis, random scale, random flip, random cuboid dropout [55], point jittering, and rotation perturbations around x, y and z axes to augment the views. All augmentations are combined and applied once to each augmented view $\mathcal{P}^q$ and $\mathcal{P}^k$.

By extracting a pair of views and applying those augmentations on the point clouds, we indirectly apply augmentations over the extracted segments, see Fig. 3 for an illustration. Since we maintain the point segment assignments via the point indexes during the augmentation, we can easily extract the augmented segments $\mathcal{S}^q$ and $\mathcal{S}^k$ from the augmented views and compute the contrastive loss.

C. Segment Contrastive Loss

The contrastive loss function is designed to discriminate between the positive and negative pairs. We use in this paper the InfoNCE loss [45], together with the momentum encoder and the feature bank proposed by He et al. [22]. For each augmented sample feature vector $\boldsymbol{q}$ and its positive pair $\boldsymbol{k}$ in the batch, the loss is calculated as:

$$\mathcal{L}_q = -\log \frac{\exp\left(\boldsymbol{q}^\top \boldsymbol{k}/\tau\right)}{\exp\left(\boldsymbol{q}^\top \boldsymbol{k}/\tau\right) + \sum_{i=1}^{K} \exp\left(\boldsymbol{q}^\top \boldsymbol{f}_i/\tau\right)}, \quad (1)$$

where τ is a normalization temperature parameter and $\mathcal{Q} = \{\boldsymbol{f_1}, \ldots, \boldsymbol{f_K}\}$ is the feature bank with $|\mathcal{Q}| = K$ implemented as FIFO queue, which are used as negative samples.

In our case, we have two augmented views, $\mathcal{P}^q$ and $\mathcal{P}^k$, from the anchor point cloud $\mathcal{P}$. We compute the point-wise features $\mathcal{F}^q$ and $\mathcal{F}^k$ from both augmented views and extract the augmented segments $\mathcal{S}^q$ and $\mathcal{S}^k$ from them. Then, we pass the segments through the projection head to compute the segment-wise feature vectors $\boldsymbol{s}_m^q$ and $\boldsymbol{s}_m^k$. Therefore, we define our contrastive loss as a segment discrimination:

$$\mathcal{L}_q = -\sum_m \log \frac{\exp\left(\boldsymbol{s}_m^{q\top} \boldsymbol{s}_m^k/\tau\right)}{\exp\left(\boldsymbol{s}_m^{q\top} \boldsymbol{s}_m^k/\tau\right) + \sum_{i=1}^{K} \exp\left(\boldsymbol{s}_m^{q\top} \boldsymbol{f}_i/\tau\right)}, \quad (2)$$

where, as in Eq. (1), τ is a normalization temperature parameter and $\boldsymbol{f}_k$ are the features from the feature bank as before. Note that the number of segments M may vary for different point clouds but is the same for the positive pairs $\mathcal{S}^q$ and $\mathcal{S}^k$.

At the end of each iteration, the feature bank is updated with the segment features from the current batch, maintaining only the last K segments seen by the network.

D. Pre-training Pipeline

Given one input point cloud $\mathcal{P}$ and its augmented pair $\mathcal{P}^q$ and $\mathcal{P}^k$, we use the backbone to compute the point-wise features $\mathcal{F}^q$ and $\mathcal{F}^k$. We use the entire point cloud during the backbone forward pass to learn the relationship between the segments and the scene. Then, we extract the augmented segments $\mathcal{S}^q$ and $\mathcal{S}^k$ from the point cloud with its point-wise features. Next, we apply dropout and global max-pooling over each segment to compute a feature vector. This feature vectors are then passed through the projection head to get $\boldsymbol{s}_m^q$ and $\boldsymbol{s}_m^k$ and calculate the contrastive loss.

IV. Implementation and Experimental Setup

Our experimental setup follows the usual evaluations on contrastive learning methods. First, we pre-train the backbone using our contrastive method. Then, we fine-tune it on different setups for a more in-depth ablation. We used the SemanticKITTI [3], [2], [17] and SemanticPOSS [36] datasets for the self-supervised pre-training, both collected in an outdoor urban environment. We compare our approach to PointContrast [50] and DepthContrast [55], using their official implementations for pre-training and data pre-processing. We use MinkUNet [10] as the backbone, which employs sparse convolutions for 3D processing.

For the class-agnostic segmentation, see Sec. III-A, we set $\alpha = 0.25\,\text{cm}$, $\varepsilon = 20$ and $\delta = 50$, which are the RANSAC distance threshold, the minimum number of points per segment and the maximum number of segments extracted per point cloud, respectively.

For our pre-training, we use the stochastic gradient descent (SGD) optimizer with a momentum of 0.9 and set the learning rate to 0.12 and the weight decay to 0.0004. We use a cosine annealing learning rate schedule [31] with a minimum learning rate of 0.00012, following the pre-training scheme used by Zhang et al. [55]. For all the methods, we randomly sample $20,000$ points from the entire point cloud after data augmentation to limit the number of points per sample, and we pre-train the backbone for 200 epochs. Given the size of SemanticKITTI, we use $K = 8,152$ for the feature bank of the DepthContrast method. For our approach, we set it to $K = 65,536$, with the temperature parameter $\tau = 0.1$. It is important to highlight that we have a bigger feature bank since we save the M segments features extracted from the point cloud instead of the complete point cloud. We use the two linear layers proposed by Chen et al. [6] for our projection head, which is also used in DepthContrast, and we set $p = 0.4$ for the dropout layer. For our method and DepthContrast, we use batch size 8, and for PointContrast, we use batch size 16 to maintain the batch size 4 per GPU as in the original paper. Given the PointContrast method characteristics, it is known that batch size change may affect the approach performance. However, we used the same cluster with four NVIDIA GTX1080TI 12 GB GPUs for all evaluated methods to make the comparisons as fair as possible.

For fine-tuning, we use the same datasets used for pre-training, i.e., SemanticKITTI and SemanticPOSS. For the semantic segmentation experiments, we use SGD with momentum of 0.9, learning rate of 0.24, and weight decay of 0.0004. We also use a cosine annealing scheduler with a minimum learning rate equal to 0.00024, following the semantic segmentation setup used by Tang et al. [43]. We set the batch size to 2 and randomly sample $80,000$ points per point cloud during training. For the object detection experiment, we use PointRCNN [41] as the base detector, and the same 5% label set used by Zhang et al. [55] for a fair comparison. For all the experiments, we use one NVIDIA RTX2080 Super 8 GB GPU. The experimental results are collected in the validation sequences from both datasets, i.e., sequence 8 for SemanticKITTI and sequences 4 and 5 for SemanticPOSS. In both datasets, the validation sequences were not used for pre-training or fine-tuning.

V. Experimental Evaluation

We present our experiments to show the capabilities of our method. We compare our method to the state of the art and show that our approach (i) is more efficient when using fewer labels, (ii) can better describe fine-grained structures and (iii) is more transferable between different datasets.

A. Label Efficiency

The first experiment evaluates the robustness of the features learned from the different representation learning methods by fine-tuning it to the semantic segmentation task using

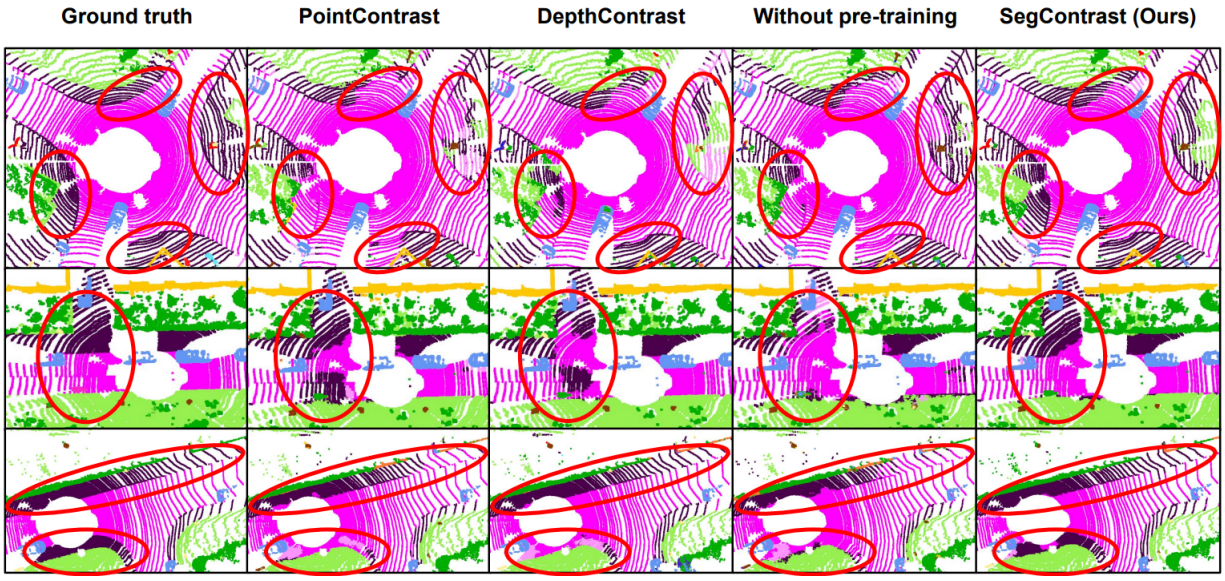

Fig. 4: Qualitative results on three different validation scans (rows). We show results of the networks fine-tuned with only 0.1% of the labels that are pre-trained with different contrastive methods or trained from scratch (without pre-training). We compare the results from PointContrast [50], DepthContrast [55] and our method, SegContrast. With our pre-training, SegContrast, the fine-tunned network can better distinguish the division between different structures, i.e., sidewalk and road, as shown by the highlighted areas (solid red circles).

TABLE I: Number of training epochs used for different label regimes on SemanticKITTI

Label regime	0.1%	1%	10%	50%	100%
Number of epochs	300	120	40	20	15

TABLE II: Fine-tuning at different label regimes on SemanticKITTI for semantic segmentation (mIoU)

	0.1%	1%	10%	50%	100%
Without pre-training	25.59	41.70	53.87	58.34	59.63
PointContrast [50]	28.52	43.40	53.79	57.30	59.77
DepthContrast [55]	33.51	46.41	**56.29**	**58.54**	59.88
SegContrast (Ours)	**34.78**	**47.41**	55.21	58.33	**60.53**

the SemanticKITTI dataset. Since it is a larger dataset, we can divide it into different label percentage regimes and increasingly compare it to support our first claim. We define five different label percentage regimes, i.e., using 0.1%, 1%, 10%, 50%, and 100% of the labeled training data, where we select a fixed subset of scans from the dataset given the regime percentage. Every subset is chosen from the entire dataset, such that all the classes are present. When training with fewer labels, the total number of training iterations will decrease accordingly. Therefore, we increase the number of epochs as we reduce the number of training scans to achieve convergence at every label regime (see Tab. I).

Fig. 4 gives a qualitative comparison between the different contrastive methods and the network without pre-training when training with 0.1% of the labels. From the top views, we observe that the network trained from scratch could not learn much structural information, leading to a noisy division between different structures. This same noisy division can be seen in the other contrastive methods. Our approach better learns the structural information of the point clouds, which leads to better boundaries between different structures in the scene. This improvement can also be seen in more fine-grained classes, e.g., pole and traffic signs, shown in Fig. 1.

Tab. II shows the results on the different label regimes. All methods perform better than training from scratch, i.e., without pre-training, and the gap between the results diminishes as the number of scans used for training increases. Our method is better at lower label regimes. As the amount of labels increases, DepthContrast achieves a comparable performance. At the full label training, all the methods converge to a similar result as training from scratch. This is expected since the data used for pre-training, and fine-tuning are the same. Moreover, Tab. III presents the per-class IoU at the lowest label regime. It is possible to see that our method performs better in the per-class IoU, outperforming previous approaches in the majority of the classes. This evaluation shows that our approach performs better when using fewer labels compared to the other contrastive learning methods, being able to better describe the different structures in the point cloud.

For a more complete evaluation, we also compare the methods fine-tuning to object detection. Tab. IV presents our results on the object detection task on the KITTI dataset [17]. In this experiment, we use the same 5% labels set used by Zhang et al. [55]. With 100% of labels, the comparison between the network without pre-training and the pre-trained models shows no significant difference. Since we use KITTI for pre-training and fine-tuning, this is expected since no new data is seen during pre-training. With 5% of labels, it is possible to show the gain of the pre-trained network. Except for the pedestrian class, all the methods outperform the network without pre-training by a large margin. In the car and cyclist classes, our method surpasses previous methods in almost all the difficulties. These results indicate that our approach can learn a robust feature representation, outperforming previous methods on different tasks when using fewer labels.

TABLE III: Per-class IoU fine-tuning with 0.1% labels

Method	mIoU	road	sidewalk	parking	building	car	vegetation	trunk	terrain	fence	pole	traffic-sign
Without pre-training	25.59	72.46	46.53	7.51	78.43	78.79	82.15	26.01	63.48	17.68	9.02	1.29
PointContrast [50]	28.52	72.71	50.24	9.06	80.20	85.78	83.29	39.29	68.18	19.56	19.68	7.61
DepthContrast [55]	33.51	79.13	58.85	12.91	**82.41**	**89.18**	**84.82**	45.07	**71.52**	21.34	50.80	17.30
SegContrast (Ours)	**34.78**	**80.32**	**61.53**	**14.70**	81.99	89.05	84.18	**49.34**	70.25	**21.53**	**50.89**	**29.93**

TABLE IV: Fine-tuning with 5% and 100% of labels on KITTI for object detection (mAP with 40 recall positions)

	Without pre-training			PointContrast [50]			DepthContrast [55]			SegContrast (ours)		
	100% labels											
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
Car	91.46	81.77	**79.92**	91.44	**81.83**	79.85	91.79	81.48	79.90	**91.97**	80.17	79.79
Pedestrian	68.29	**63.76**	58.07	**69.81**	63.56	**58.13**	66.39	60.44	55.41	67.46	62.71	56.42
Cyclist	90.65	73.35	70.05	90.48	73.87	69.81	**92.45**	74.69	**71.18**	90.71	**75.38**	70.79
	5% labels											
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
Car	79.33	63.83	58.92	80.83	66.52	61.70	82.32	67.61	62.59	**82.62**	**68.27**	**64.41**
Pedestrian	69.34	61.72	54.49	69.75	**62.77**	**55.71**	**69.81**	62.37	55.13	69.10	61.59	54.49
Cyclist	84.48	61.82	57.80	**88.82**	67.53	63.17	87.52	66.91	62.50	88.70	**68.48**	**64.35**

TABLE V: Linear evaluation mIoU and per-class IoU (%) pre-trained and evaluated on SemanticKITTI

Method	mIoU	road	sidewalk	parking	building	car	vegetation	trunk	terrain	fence	pole	traffic-sign
Without pre-training	4.29	30.61	2.72	0.0	5.56	0.02	42.39	0.0	0.05	0.24	0.0	0.0
PointContrast [50]	**24.02**	**70.61**	**37.00**	0.0	**83.37**	**88.93**	**75.10**	33.26	**53.93**	8.66	5.55	0.0
DepthContrast [55]	15.91	40.68	5.35	0.0	54.81	59.43	65.30	24.55	17.56	13.50	20.47	0.0
SegContrast (Ours)	23.36	59.02	30.83	**0.5**	71.68	80.23	73.13	**34.56**	37.51	**15.68**	**40.59**	**0.19**

TABLE VI: Pre-training with SemanticKITTI and fine-tuning on SemanticPOSS with different label regimes (mIoU)

	0.1%	1%	10%	50%	100%
Without pre-training	33.09	43.14	57.27	63.34	64.22
Supervised pre-training	42.88	54.40	60.22	64.26	64.54
PointContrast [50]	31.78	49.06	56.49	62.93	64.30
DepthContrast [55]	41.94	52.66	59.27	64.09	64.65
SegContrast (Ours)	**43.69**	**55.21**	**60.33**	**64.58**	**64.86**

B. Linear Evaluation

A typical experiment used for image-based contrastive methods is the so-called linear evaluation. This evaluation freezes the pre-trained backbone and trains only a linear layer on top of it to compare how well the feature representation can describe the different classes even without fine-tuning the whole network. Our experiment follows the same setup, the pre-trained backbone weights are frozen, and we train only a linear segmentation head. The result of this experiment supports our claim that our method can learn a feature representation that better describe fine-grained structures.

Tab. V displays the results of the linear evaluation over the different self-supervised contrastive methods and over the randomly initialized network without pre-training for a lower bound on the performance. DepthContrast shows the worst performance in this evaluation, which indicates that the method cannot learn a feature representation as descriptive as the other methods. PointContrast shows a better performance, since the method uses a point-wise contrastive loss. Furthermore, when looking at the underrepresented classes, e.g., parking, trunk, fence, pole or traffic sign, our method outperforms the other methods. Thus, PointContrast achieves a higher mIoU by learning the more represented classes, e.g., road, building. In contrast, our method seems better suited to represent the fine-grained classes.

C. Feature Representation Transferability

In this third experiment, we evaluate the transferability of our learned feature representation. The results support our claim that our method is more transferable between different datasets. We use SemanticKITTI for unsupervised pre-training of the backbone with different contrastive methods or use the commonly used supervised pre-training. Then, we fine-tune the differently pre-trained networks on the SemanticPOSS dataset and compare their performance. SemanticPOSS is smaller than SemanticKITTI, which gives a setup aligned to standard image-based setting, where a larger dataset, e.g., ImageNet [13], is used for pre-training and then fine-tuning is performed on smaller datasets.

Fig. 5 displays the network performance in different training epochs during fine-tuning. Here, we see a comparable performance of the unsupervised and supervised pre-training. As shown, after only one training epoch, our method shows a considerably better performance than the other contrastive methods and achieves the same performance as the supervised pre-training. Even though our approach does not use any labels, our method learns a feature representation comparable with the supervised pre-training on SemanticKITTI. This result suggests that our method can learn a more general feature representation than previous methods and it seems to be more suitable for fine-tuning on a different dataset.

TABLE VII: Linear evaluation on SemanticPOSS with pre-training on SemanticKITTI (mIoU)

Method	mIoU (%)	accuracy (%)
Without pre-training	6.32	42.46
PointContrast [50]	24.79	72.55
DepthContrast [55]	16.41	63.09
SegContrast (Ours)	**31.51**	**73.51**

TABLE VIII: Fine-tuning to SemanticPOSS with pre-training on SemanticKITTI and SemanticPOSS using our method (mIoU)

Dataset	Linear evaluation	Fine-tune
Without pre-training	6.32	64.22
SemanticPOSS	29.68	64.31
SemanticKITTI	**31.51**	**64.86**

Tab. VI displays the results of the different contrastive pre-training methods and the supervised pre-training on SemanticKITTI. The pre-training methods improve the network performance at the lower label regimes, but the previous contrastive methods cannot surpass the supervised pre-training. However, our method outperforms both self-supervised and supervised pre-training at all different label regimes. This experiment indicates the robustness of the learned feature representation when fine-tuning to a different LiDAR data, exceeding even the supervised pre-training.

In Tab. VII, we show the linear evaluation over the SemanticPOSS with the network pre-trained on SemanticKITTI. DepthContrast shows the lowest performance in this evaluation, showing that the feature representation is less descriptive when transferring to a different dataset compared to the other methods. Our method surpasses the other approaches, outperforming them by a large margin. These results suggest that our approach can learn a point cloud representation transferable across different datasets and LiDAR sensors.

Finally, we evaluate the self-supervised pre-training on SemanticKITTI and SemanticPOSS, fine-tuning it on SemanticPOSS. In Tab. VIII, we show both the linear evaluation and the fine-tuning. As we can see, the performance of the pre-training on SemanticKITTI is better on both experiments. We can see that we obtain a performance gain using a large dataset for self-supervised pre-training. This also highlights the generalization of the feature representation achieved by our method. The pre-training on SemanticKITTI performed better than on SemanticPOSS, even though both datasets were collected with a different LiDAR sensors and different sensor mounting positions.

VI. Conclusion

In this paper, we present a novel representation learning approach for LiDAR point clouds in outdoor environments. Our approach exploits the characteristics of outdoor LiDAR data to extract class-agnostic segments and applies the contrastive loss over these segments. We evaluate our strategy on different datasets and provide comparisons with other state-of-the-art feature representation learning approaches. The experiments suggest that our approach can learn a more robust feature representation than previous works outperforming them on different downstream tasks. Furthermore, our self-supervised feature representation seems to be more transferable when fine-tuning on a different target dataset outperforming even the supervised pre-training.

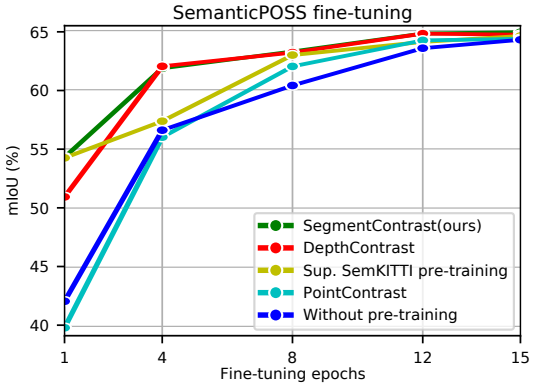

Fig. 5: Comparison between contrastive pre-trained networks fine-tuned to the SemanticPOSS dataset with the network trained from scratch. At the beginning of training, our method shows a comparable performance to supervised pre-training on SemanticKITTI, evidencing that our learned feature representation is as robust as the supervised pre-training.

References

[1] I. Achituve, H. Maron, and G. Chechik. Self-supervised learning for domain adaptation on point clouds. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2021.

[2] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, and C. Stachniss. Towards 3d lidar-based semantic scene understanding of 3d point cloud sequences: The semantickitti dataset. *Intl. Journal of Robotics Research (IJRR)*, 40(8-9):959–967, 2021.

[3] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[4] J. Behley, V. Steinhage, and A. Cremers. Laser-based Segment Classification Using a Mixture of Bag-of-Words. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[5] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, and J. Sun. You Only Look One-Level Feature. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[6] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2020.

[7] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G.E. Hinton. Big Self-Supervised Models are Strong Semi-Supervised Learners. In *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[8] X. Chen and K. He. Exploring Simple Siamese Representation Learning. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[9] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-View 3D Object Detection Network for Autonomous Driving. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] C. Choy, J. Gwak, and S. Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[11] M. Cordts, S.M. Omran, Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] T. Cortinhal, G. Tzelepis, and E. Aksoy. SalsaNext: Fast Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving. *arXiv preprint*, 2003.03653v1, 2020.

[13] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, June 2009.

[14] M. Ester, H. Kriegel, J. Sander, and X.Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the Conf. on Knowledge Discovery and Data Mining (KDD)*, 1996.

[15] M. Fischler and R. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24(6):381–395, 1981.

[16] S. Gasperini, M.N. mahani, A. Marcos-Ramiro, N. Navab, and F. Tombari. Panoster: End-To-End Panoptic Segmentation of LiDAR Point Clouds. *IEEE Robotics and Automation Letters (RA-L)*, 2021.

[17] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.

[18] Z. Gojcic, O. Litany, A. Wieser, L.J. Guibas, and T. Birdal. Weakly Supervised Learning of Rigid 3D Scene Flow. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[19] B. Graham and L. van der Maaten. Submanifold Sparse Convolutional Networks. *arXiv preprint*, 2017.

[20] J.B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[21] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 2020.

[22] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[23] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[24] O.J. Hénaff, S. Koppula, J.B. Alayrac, A. van den Oord, O. Vinyals, and J. Carreira. Efficient Visual Pretraining with Contrastive Detection. *arXiv preprint*, 2021.

[25] M. Himmelsbach, F.v. Hundelshausen, and H.J. Wuensche. Fast Segmentation of 3D Point Clouds for Ground Vehicles. In *Proc. of the IEEE Vehicles Symposium (IV)*, 2010.

[26] J. Hou, B. Graham, M. Niessner, and S. Xie. Exploring Data-Efficient 3D Scene Understanding With Contrastive Scene Contexts. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[27] Q. Hu, B. Yang, G. Fang, Y. Guo, A. Leonardis, N. Trigoni, and A. Markham. SQN: Weakly-Supervised Semantic Segmentation of Large-Scale 3D Point Clouds with 1000x Fewer Labels. *arXiv preprint*, abs/2104.04891, 2021.

[28] P. Jiang and S. Saripalli. LiDARNet: A Boundary-Aware Domain Adaptation Model for Point Cloud Semantic Segmentation. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[29] F. Langer, A. Milioto, A. Haag, J. Behley, and C. Stachniss. Domain Transfer for Semantic Segmentation of LiDAR Data using Deep Neural Networks. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[30] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick. Microsoft COCO: Common Objects in Context. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 740–755, 2014.

[31] I. Loshchilov and F. Hutter. SGDR: Stochastic Gradient Descent with Restarts. *arXiv preprint*, abs/1608.03983, 2016.

[32] A. Milioto, J. Behley, C. McCool, and C. Stachniss. LiDAR Panoptic Segmentation for Autonomous Driving. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[33] A. Milioto, L. Mandtler, and C. Stachniss. Fast Instance and Semantic Segmentation Exploiting Local Connectivity, Metric Learning, and One-Shot Detection for Robotics. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[34] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. In *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[35] A. Nekrasov, J. Schult, O. Litany, B. Leibe, and F. Engelmann. Mix3D: Out-of-Context Data Augmentation for 3D Scenes. In *Proc. of the Intl. Conf. on 3D Vision (3DV)*, 2021.

[36] Y. Pan, B. Gao, J. Mei, S. Geng, C. Li, and H. Zhao. SemanticPOSS: A Point Cloud Dataset with Large Quantity of Dynamic Instances. *arXiv preprint:2002.09147*, 2020.

[37] T. Park, A.A. Efros, R. Zhang, and J.Y. Zhu. Contrastive Learning for Unpaired Image-to-Image Translation. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[38] C.R. Qi, H. Su, K. Mo, and L.J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[39] C.R. Qi, O. Litany, K. He, and L.J. Guibas. Deep Hough Voting for 3D Object Detection in Point Clouds. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[40] C. Qi, K. Yi, H. Su, and L.J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[41] S. Shi, X. Wang, and H. Li. PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[42] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal on Machine Learning Research (JMLR)*, 15:1929–1958, 2014.

[43] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[44] H. Thomas, C. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[45] A. van den Oord, Y. Li, and O. Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv preprint*, 2018.

[46] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, and L. Van Gool. Unsupervised Semantic Segmentation by Contrasting Object Mask Proposals. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[47] H. Wang, Y. Cong, O. Litany, Y. Gao, and L.J. Guibas. 3DIoUMatch: Leveraging IoU Prediction for Semi-Supervised 3D Object Detection. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[48] W. Wang, T. Zhou, F. Yu, J. Dai, E. Konukoglu, and L. Gool. Exploring Cross-Image Pixel Contrast for Semantic Segmentation. *arXiv preprint*, 2021.

[49] B. Wu, A. Wan, X. Yue, and K. Keutzer. SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[50] S. Xie, J. Gu, D. Guo, C.R. Qi, L. Guibas, and O. Litany. PointContrast: Unsupervised Pre-training for 3D Point Cloud Understanding. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[51] X. Xu and G.H. Lee. Weakly Supervised Semantic Point Cloud Segmentation: Towards 10x Fewer Labels. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[52] L. Yi, B. Gong, and T. Funkhouser. Complete & Label: A Domain Adaptation Approach to Semantic Segmentation of LiDAR Point Clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[53] X. Yuan, A. Kortylewski, Y. Sun, and A. Yuille. Robust Instance Segmentation Through Reasoning About Multi-Object Occlusion. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[54] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. *arXiv preprint*, abs/2103.03230, 2021.

[55] Z. Zhang, R. Girdhar, A. Joulin, and I. Misra. Self-Supervised Pretraining of 3D Features on any Point-Cloud. *arXiv preprint*, 2021.

[56] L. Zheng, M. Tang, Y. Chen, G. Zhu, J. Wang, and H. Lu. Improving Multiple Object Tracking With Single Object Tracking. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[57] Z. Zhou, Y. Zhang, and H. Foroosh. Panoptic-PolarNet: Proposal-Free LiDAR Point Cloud Panoptic Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[58] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin. Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.