Spatiotemporal Self-supervised Learning for Point Clouds in the Wild

Yanhao Wu [1] Tong Zhang[2] Wei Ke[1] Sabine Süsstrunk[2] Mathieu Salzmann[2]
[1] School of Software Engineering, Xi'an Jiaotong University, China
[2] School of Computer and Communication Sciences, EPFL Switzerland

Abstract

Self-supervised learning (SSL) has the potential to benefit many applications, particularly those where manually annotating data is cumbersome. One such situation is the semantic segmentation of point clouds. In this context, existing methods employ contrastive learning strategies and define positive pairs by performing various augmentation of point clusters in a single frame. As such, these methods do not exploit the temporal nature of LiDAR data. In this paper, we introduce an SSL strategy that leverages positive pairs in both the spatial and temporal domain. To this end, we design (i) a point-to-cluster learning strategy that aggregates spatial information to distinguish objects; and (ii) a cluster-to-cluster learning strategy based on unsupervised object tracking that exploits temporal correspondences. We demonstrate the benefits of our approach via extensive experiments performed by self-supervised training on two large-scale LiDAR datasets and transferring the resulting models to other point cloud segmentation benchmarks. Our results evidence that our method outperforms the state-of-the-art point cloud SSL methods. [1]

1. Introduction

Semantic segmentation from LiDAR point clouds can be highly beneficial in practical applications, e.g., for self-driving vehicles to safely interact with their surroundings. Nowadays, state-of-the-art methods [13, 36, 46] achieve this with deep neural networks. While effective, the training of such semantic segmentation networks requires large amounts of annotated data, which is prohibitively costly to acquire, particularly for point-level LiDAR annotations [45]. By contrast, with the rapid proliferation of self-driving vehicles, large amounts of *unlabeled* LiDAR data are generated. Here, we develop a method to exploit such unlabeled data in a self-supervised learning framework.

Self-supervised learning (SSL) aims to learn features without any human annotations [1, 2, 22, 26, 33, 35, 40, 45] but so that they can be effectively used for fine-tuning on a downstream task with a small number of labeled samples. This is achieved by defining a pre-task that does not require annotations. While many pre-tasks have been proposed [27], contrastive learning has nowadays become a highly popular choice [30, 33, 40, 41, 45]. In general, it aims to maximize the similarity of positive pairs while potentially minimizing that of negative ones. In this context, most of

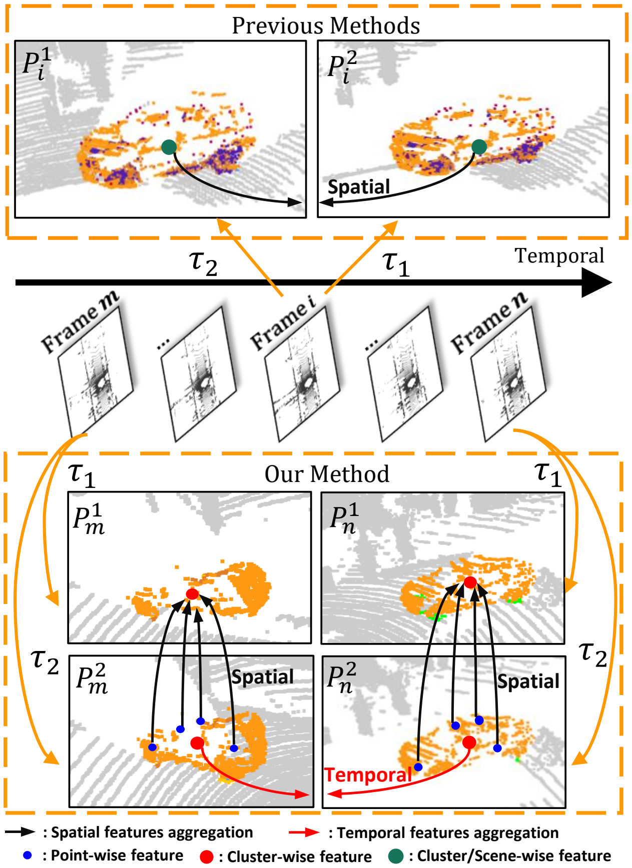

Figure 1. **Our method vs existing ones. (Top)** Previous methods create positive pairs for SSL by applying different augmentations, τ_1 and τ_2 (e.g., random flipping, clipping), to a single frame. **(Bottom)** By contrast, we leverage both spatial and temporal information via a point-to-cluster and an inter-frame SSL strategy. Points in the same color are from the same cluster in the latent space.

[1] Our code and pretrained models will be found at https://github.com/YanhaoWu/STSSL. Correspondence to Ke Wei.

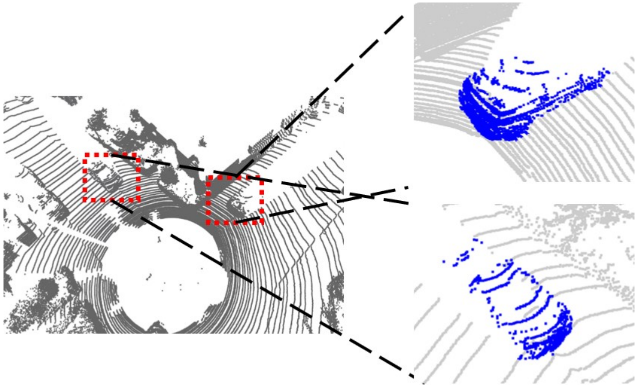

Figure 2. Cars in the **same frame but under different illumination angles**. Note that the main source of difference between the two instance point clouds arises from the different illumination angles.

the point cloud SSL literature focuses on indoor scenes, for which relatively dense point clouds are available. Unfortunately, for outdoor scenes, such as the ones we consider here, the data is more complex and much sparser, and creating effective pairs remains a challenge.

Several approaches [33, 45] have nonetheless been proposed to perform SSL on outdoor LiDAR point cloud data. As illustrated in the top portion of Fig. 1, they construct positive pairs of point clusters or scenes by applying augmentations to a single frame. As such, they neglect the temporal information of the LiDAR data. By contrast, in this paper, we introduce an SSL approach to LiDAR point cloud segmentation based on extracting effective positive pairs in both the spatial *and temporal* domain.

To achieve this without requiring any pose sensor as in [24, 40], we introduce (i) a **point-to-cluster (P2C)** SSL strategy that maximizes the similarity between the features encoding a cluster and those of its individual points, thus encouraging the points belonging to the same object to be close in feature space; (ii) a **cluster-level inter-frame self-supervised learning** strategy that tracks an object across consecutive frames in an unsupervised manner and encourages feature similarity between the different frames. These two strategies are depicted in the bottom portion of Fig. 1.

Note that the illumination angle of one object seen in two different frames typically differs. As shown in Fig. 2, this is also the main source of difference between two objects of the same class in the same frame. Therefore, our inter-frame SSL strategy lets us encode not only temporal information, but also the fact that points from different objects from the same class should be close to each other in feature space. As simulating different illumination angles via data augmentation is challenging, our approach yields positive pairs that better reflects the intra-class variations in LiDAR point clouds than existing single-frame methods [33, 45].

Our contribution can be summarized as follows:

- We introduce an SSL strategy for point cloud segmentation based only on positive pairs. It does not require any external information, such as pose, GPS, and IMU.
- We propose a novel Point-to-Cluster (P2C) training paradigm that combines the advantages of point-level and cluster-level representations to learn a structured point-level embedding space.
- We introduce the use of cluster-level inter-frame self-supervised leaning on point clouds generated by a LiDAR sensor, which introduces a new way to integrate temporal information into SSL.

Our experiments on several datasets, including KITTI [17], nuScene [5], SemanticKITTI [4] and SemanticPOSS [34], evidence that our method outperforms the state-of-the-art SSL techniques for point cloud data.

2. Related Work

Self-supervised learning for images. Self-supervised learning for images has developed at a fast pace in recent years [7–9, 11, 18, 21, 38]. Existing methods follow different paradigms, such as generation-based methods [32], clustering methods [6, 25, 42, 43] and contrastive learning methods [10, 12, 20]. Currently, BYOL [19], a self-supervised learning method that uses only positive pairs in its loss function, constitutes the state of the art. Intrigued by the success of such contrastive learning strategies, several works have studied the principles behind this approach, with a particular focus on the role of data augmentation [3, 23, 28, 37]. In [39], it was observed that data augmentation creates a certain degree of "chaos" between the intra-class samples that helps them to become more similar. Similarly, LoGo [44] also introduce local and global crops differently to handle the variance due to the augmentation. Our method is inspired by BYOL but targets 3D data. Because of the fundamentally different nature of 2D images and 3D point clouds, data augmentation designed for images does not directly apply to the 3D domain.

Self-supervised learning for 3D data. As in the image case, the number of self-supervised learning methods for 3D data has grown rapidly [1, 2, 22, 26, 33, 35, 40, 45], with examples such as DepthContrast [45], PointContrast [40], GCC-3D [30], ProposalContrast [41], STRL [24] and SegContrast [33]. Nevertheless, these methods still suffer from severe limitations. In particular, many methods [24, 40] need the camera pose in each frame to find correspondences to use as positive pairs. While effective for indoor scenes, the points in outdoor scenes are much sparser, and even with the ground-truth poses, correspondences between points are hard to obtain. By contrast, SegContrast, ProposalContrast, and DepthContrast [33, 41, 45] specifically tackle the outdoor scenario, without requiring camera poses. However, they aggregate features in each region through either max

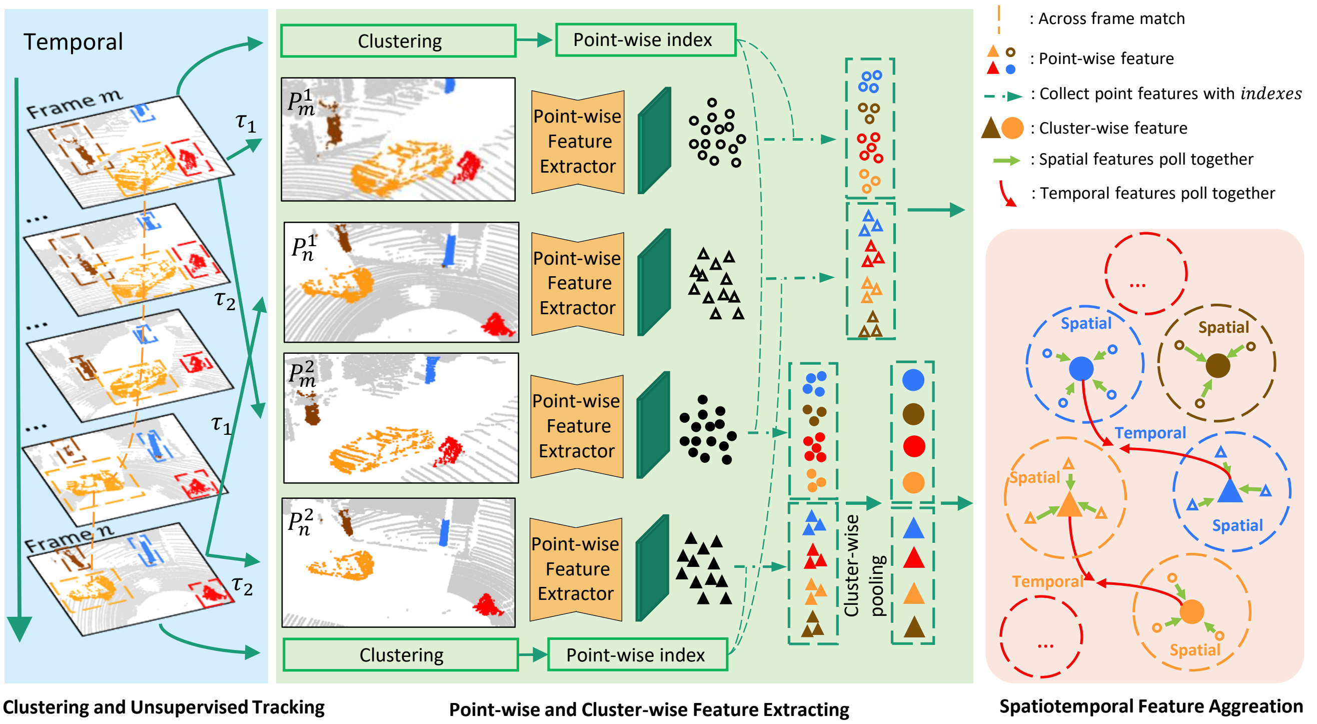
Temporal
Frame m
Frame n
τ_1
τ_2
Clustering
Point-wise index
P_m^1
P_n^1
P_m^2
P_n^2
Point-wise Feature Extractor
Cluster-wise pooling
: Across frame match
: Point-wise feature
: Collect point features with *indexes*
: Cluster-wise feature
: Spatial features poll together
: Temporal features poll together
Spatial
Temporal
...
Clustering and Unsupervised Tracking
Point-wise and Cluster-wise Feature Extracting
Spatiotemporal Feature Aggregation

Figure 3. **Overview of our STSSL.** Given a sequence of LiDAR point clouds, we first perform clustering and unsupervised tracking to associate clusters in different frames. At each training iteration, we select two frames and apply augmentations to generate two views for each frame (i.e., P_m^1, P_n^1, P_m^2, P_n^2). A feature extractor (Backbone) is then used to obtain point-wise features in the four views, and we collect the features belonging to each cluster. In P_m^2, P_n^2, we further apply a cluster-wise pooling layer to the features to generate cluster-wise features. Finally, we minimize the distance between the point features and the corresponding cluster features from P_m^1, P_n^1, and between the cluster features obtained from associated clusters in P_m^2, P_n^2. τ_1 and τ_2 are data augmentations, such as random flipping and random clipping.

or average pooling, and pull region-level features from different views together, therefore, it does not have constraint for each point. More importantly, they fail to find a way to associate the points in different time frames. By contrast, our method only utilizes point cloud data and does not rely on camera calibration to aggregate spatial and temporal features. Furthermore, we propose a point-to-cluster training paradigm that combines the advantages of point-level and cluster-level discrimination.

3. Method

The overall framework is depicted in Fig. 3 and contains three parts: clustering and unsupervised tracking, point-wise and cluster-wise feature extraction, and spatialtemporal feature aggregation. Below, we discuss these components in detail.

3.1. Clustering and Unsupervised Tracking

Let $P = \{P^1, P^2, ..., P^T\}$ denote a sequence of LiDAR point clouds with T frames, where $P^k = \{p_1^k, p_2^k, ..., p_{N_k}^k\}$ represents the k-th point cloud with N_k 3D points $p_i^k \in \mathbb{R}^3$. The segmentation map of each P^k is obtained by applying cluster to the non-ground points, where the ground points are eliminated by RANSAC [16]. Thanks to the over segmentation property of DBSCAN [15], each point has high possibility to represent the same semantic meaning with other points in the same cluster. This process yields a set of M_k clusters $S^k = \{S_1^k, S_2^k, ..., S_{M_k}^k\}$.

We will leverage these clusters to define a point-to-cluster loss for SSL, encoding a notion of spatial similarity. Furthermore, we will also exploit them to create temporal positive pairs for SSL via the unsupervised tracking strategy described below. Thus, the mechanism allows similar clusters to be merged into the same one in later stage to achieve final segmentation.

Specifically, unsupervised tracking is achieved by matching the clusters in two adjacent frames, *e.g.*, frames k with M_k clusters and $(k+1)$ with M_{k+1} clusters. To this end, we define a matching degree matrix $D \in R^{M_k \times M_{k+1}}$ as

$$D = D_{loc} + \alpha * D_{feat}, \tag{1}$$

where D_{loc} is the matrix of pairwise Euclidean distances between the cluster centers in the two frames, D_{feat} is the matrix of pairwise feature distance, and $\alpha \in (0, 1)$ is a weight balancing the two matrices. The center of cluster

j in frame k is taken as the average of all 3D points belong to this cluster. More details regarding the cluster features is provided in Section 3.2. We then use D to match the clusters in both frames using the Hungarian algorithm [29]. For the unmatched clusters, we will create trajectories for the one just appears in current frame, and abandon the trajectories of the clustering no longer exists. More details can be found in the supplementary

Thanks to the combination of 3D information and learned representations, this matching strategy allows us to robustly track a cluster across multiple frames. This lets us construct long-range positive pairs where a cluster is observed under different illumination angles, thus corresponding to a challenging positive sample for SSL. We will discuss how we exploit such pairs in Section 3.3.

3.2. Feature Extraction

As discussed above, we extract learned features from the input point clouds. Specifically, we extract two types of features: point-level ones and cluster-level ones. To this end, given an input point cloud P^k, we first apply data augmentation to obtain two view $\tilde{P}^k$ and $\bar{P}^k$. One view will be used to extract point-level features and the other for cluster-level features. This will let us create more challenging point-to-cluster pairs for the SSL strategy discussed in Section 3.3.

Point-level Features. Following the BYOL [19] format, let f denote the backbone encoder. In our case, f is MinkUnet [14]. We forward pass $\tilde{P}^k$ through the backbone encoder to obtain a feature vector $y_q^k = f(\tilde{p}_q^k)$ for every 3D point. We then group these representations according to the cluster to which each point belongs, giving us a set $F^k = \{F_1^k, F_2^k, ..., F_{M_k}^k\}$, where $F_i^k \in \mathbb{R}^{N_{k,i} \times d}$, with $N_{k,i}$ the number of points in cluster i from point cloud k, and d the feature dimension of each y_q^k.

Cluster-level Features. To extract cluster-level features, we first process $\bar{P}^k$ as above to extract d-dimensional point-level features. However, instead of simply grouping these features according to the clusters, we max-pool them according to the clusters. This yields a set of cluster-level features $C^k = \{c_1^k, c_2^k, ..., c_{M_k}^k\}$, where $c_i^k \in \mathbb{R}^{1 \times d}$.

3.3. Spatialtemporal Feature Aggregation

Let us now describe our spatiotemporal SSL framework. It relies on two loss functions encoding two goals: **i)** points from one object should be close in feature space; **ii)** points from the same class should be closer to each other than other classes. We materialize these two objectives via the **Point-to-Cluster** and **Cluster-level Inter-frame Self-supervised Leaning** strategies discussed below.

Point-to-Cluster Learning Strategy. To encourage points from the same object to be close to each other, we minimize the distance between the point features of $\tilde{P}^k$ and the corresponding cluster features in view $\bar{P}^k$. Given the features discussed in Section 3.2, this is achieved via the loss function

$$L_{p2c} = \sum_{i=1}^{M_k} \sum_{j=1}^{N_{k,i}} \left\| \frac{f_{i,j}^k}{\|f_{i,j}^k\|_2} - \frac{c_i^k}{\|c_i^k\|_2} \right\|_2^2 , \tag{2}$$

where $f_{i,j}^k \in \mathbb{R}^{1 \times d}$ denotes the feature vector from F_i^k corresponding to point j in cluster i. In essence, this encourages the network to learn similar features for all points in the same cluster while being robust to different views of the point cloud.

Cluster-level Inter-frame Self-supervised Leaning. To encourage points from the same class to be close to each other, we build on the observation that the main source of differences between two objects from the same class in point cloud data is the illumination angles under which they are observed. Thanks to the unsupervised tracking strategy, we can extract pairs of clusters in two distant frames, where the object is then seen under different illumination angles. Given two frames m and n, let N^{mn} denote the number of matched clusters across the two frames. Then, we use the cluster-level features to write the loss

$$L_{inter-frame} = \sum_{i=0}^{N^{mn}} \left\| \frac{c_i^m}{\|c_i^m\|_2} - \frac{c_i^n}{\|c_i^n\|_2} \right\|_2^2 , \tag{3}$$

where c_i^m and c_i^n are the cluster-level feature vectors of two matched clusters in frame m and n. Hence, the total loss can be written as

$$L_{total} = L_{p2c} + \lambda L_{inter-frame}, \tag{4}$$

where λ is a weight balancing the two loss terms. In practice, the inter-frame information can be better used with the feature of SSL on intra-frame. Thus, we choose a strategy of progressively increasing the λ.

4. Experiments

We first describe our experimental settings, including datasets, unsupervised tracking, and implementation details. Then, we demonstrate the benefits of our self-supervised pre-trained model on downstream tasks, and finally analyze different aspects of our method.

4.1. Experimental Settings

Datasets. We use the KITTI [17] and nuScene [5] datasets for pre-training, and SemanticKITTI [4] and SemanticPOSS [34] for the down-stream tasks.

KITTI [17] has 21 sequences, and its sampling rate is 10hz. Following [33], we use only the point clouds captured by the Velodyne LiDAR sensor rather than all the information obtained from the position sensors. The sequences 0-10 are used for pre-training, with the exception of sequence 8, which we use as validation data. nuScene [5] is much larger than KITTI. It comprises 1000 scenes and is divided

Method name	mIoU	car	road	sidewalk	building	fence	vegetation	terrain	parking	pole
From scratch	29.17	82.61	74.32	52.06	78.99	19.29	83.13	68.20	9.04	30.09
STRL [24]	16.64	47.66	56.17	23.17	58.63	13.68	69.96	41.91	0	3.12
DepthContrast [45]	30.91	88.80	69.51	49.87	82.67	22.70	83.36	67.38	9.32	48.69
SegContrast [33]	34.01	89.22	78.72	57.19	82.80	21.99	83.42	67.26	14.06	**50.91**
STSSL (ours)	**37.71**	**91.11**	**85.34**	**66.09**	**85.43**	**25.63**	**84.79**	**72.57**	**22.61**	48.67

Table 1. **Per-class IoU when fine-tuning with 0.1 % labels.**

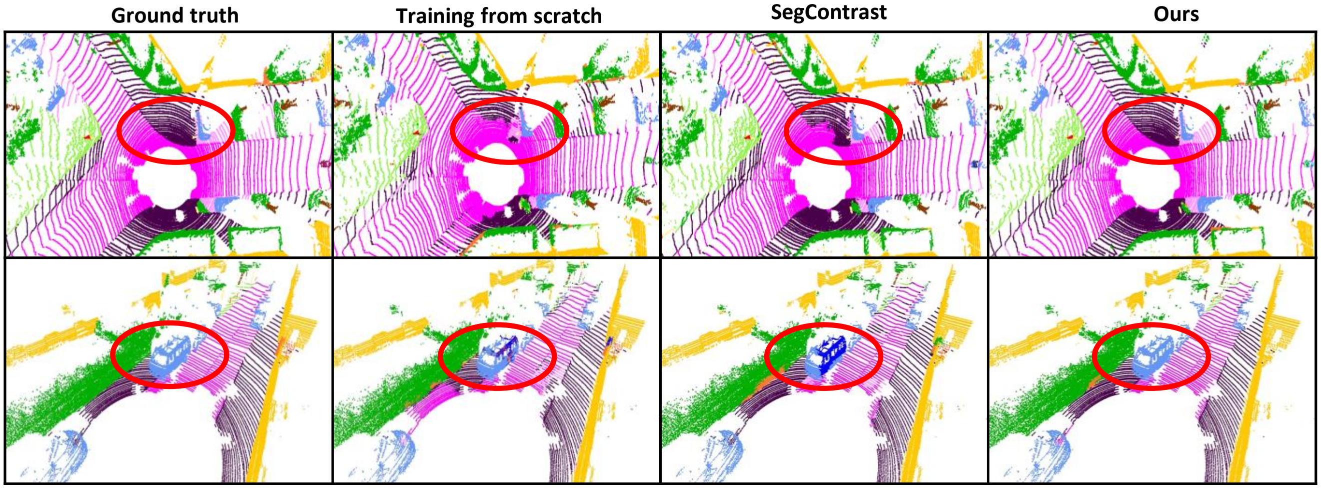

Figure 4. **Segmentation results on different frames (rows)**. The models are fine-tuned with 0.1% labels on KITTI. We compare SegContrast [33], STSSL (ours) and training from scratch (without pre-training). Our method better distinguishes the different structures shown in the highlighted area (red circle).

into 10 sequences. The LiDAR data is acquired at 20hz. Because of limited computational resources, we only use the point clouds captured by the Velodyne LiDAR sensor in sequence 1 and 2 (scenes 0 - 149) for pre-training. SemanticKITTI [4] provides dense point-wise annotations for almost every point in KITTI [4]. A total of 23,201 scans are annotated on sequences 0-10 of KITTI for training and validation. SemanticPOSS [34] is also used for semantic segmentation, and contains 2988 diverse and complicated LiDAR scans with 14 classes. The scans are divided into 6 splits, with 500 scans per split. Splits 4 and 5 are used for testing and the other ones are used for training.

Unsupervised Tracking. We set the relative weight between spatial distance and feature distance in Eq. (1) to $\alpha = 0.5$, and the threshold of RANSAC distance to 0.25 as in [33]. The threshold of DBSCAN distance is set to 0.25 in KITTI and 0.5 in nuScene. In each frame, we drop clusters with fewer than 200 points or more than 20000 points to filter out noise and retain up to 50 clusters.

Implementation Details. We compare our approach with DepthContrast [45], STRL [24], SegContrast [33], and training from scratch. We use MinkUnet [14] as backbone for all approaches and build our approach on the basis of BYOL [19]. We pre-train the backbone on KITTI and nuScene for 200 epochs using an SGD optimizer with a momentum of 0.9, and set the weight decay to 0.0004 following SegContrast [33]. The learning rate is initially set to 0.036 with a linear annealing scheme with a minimum learning rate equal to 0.009. In the early training stages, we set λ to 0, and to 4 in the later stages. When incorporating the inter-frame loss term, we re-initialize different MLPs after the backbone networks to avoid information leaks from spatial. The batch size is set to 8 for each GPU, and we use 8×GTX3090 GPUs to pre-train the models, which leads to the total batch size of 64.

For fine-tuning on the down-stream semantic segmentation task, we use an SGD optimizer with a cosine learning rate schedule. The fine-tuned models are evaluated on the validation sequences, i.e., sequence 8 for SemanticKITTI and sequences 4 and 5 for SemanticPOSS [33]. The batch size is set to 2 for each GPU, and 4×GTX2080Ti GPUs are used for the experiments.

Evaluation Metrics. We evaluate point cloud semantic segmentation using the mean intersection over union (mIoU) and the overall point classification accuracy (Acc).

4.2. Outdoor Scene Understanding

Label Efficiency. To assess the label efficiency of our STSSL approach, we fine-tune the model pre-trained on KITTI on SemanticKITTI. Following [33], SemanticKITTI is divided into different regimes corresponding to different

	0.1%	1%	10%	100%
From Scratch	29.17	48.11	51.00	56.14
STRL [24]	16.64	31.88	30.88	55.71
DepthContrast [45]	30.91	42.41	42.38	45.48
SegContrast [33]	34.01	48.02	52.26	55.45
STSSL (ours)	**37.71**	**52.60**	**54.51**	**57.33**

Table 2. Pre-training on **KITTI** and evaluating the fine-tuned models in different label regimes on **SemanticKITTI** for semantic segmentation. We report the mIoU.

mIoU / Acc	seq1	seq 1-2
From Scratch	29.17 / 82.57	29.17 / 82.57
STRL [24]	19.11 / 74.56	18.74 / 70.85
SegContrast [33]	33.91 / 84.88	34.28 / 85.20
STSSL (ours)	**34.43 / 85.34**	**35.08 / 85.75**

Table 3. Pre-training on **nuScene** and evaluating the fine-tuned models in the 0.1% label regime on **SemanticKITTI** for semantic segmentation. We report mIoU/Acc.

pre-train dataset	KITTI	nuScene(seq 1)
From Scratch	39.64 / 88.66	39.64 / 88.66
STRL [24]	38.43 / 88.32	36.63 / 87.53
SegContrast [33]	**43.88 / 89.64**	42.86 / 89.28
STSSL (ours)	43.84 / 89.47	**43.55 / 89.38**

Table 4. Pre-training on **KITTI and nuScene**, and evaluating the fine-tuned models on **SemanticPOSS** for semantic segmentation. We report the mIoU/Acc.

mIoU / Acc	0.1%	1%
From Scratch	29.17 / 82.57	48.11 / 89.94
STRL [24]	16.64 / 69.36	31.88 / 85.35
DepthContrast [45]	30.91 / 82.82	42.41 / 88.95
SegContrast [33]	34.01 / 84.72	48.02 / 88.84
P2C	**35.48 / 86.18**	**50.83 / 90.14**

Table 5. Ablation study on the **pre-training strategy** with 0.1% and 1% labels. We report mIoU/Acc.

percentages of labels. Specifically, we use 0.1%, 1%, 10%, and 100% of the training data to fine-tune the pre-trained model for semantic segmentation.

In Table 1, we compare the mIoU and per-class IoU of the proposed STSSL and of the state-of-the-art approaches when using 0.1% of the labels. Our method outperforms the baselines by an impressive margin, yielding an mIoU of 37.71%, which is 3.7% better than SegContrast [33] and 8.54% than training from scratch. Fig. 4 evidences that our method yields more complete and accurate segmentation masks than the other methods.

The per-class comparison shows that our approach greatly improves the network's performance on most classes when there are few annotations. Our STSSL yields much better results than the baselines, especially for car, building, vegetation, terrain, and parking. We attribute this to the fact that these classes have clearly different appearances under different illumination angles as shown in Fig.2, which is exactly the problem that our inter-frame self-supervised learning addresses.

The results obtained by fine-tuning with different percentages of training data are provided in Table 2. Our method consistently outperforms the state-of-the-art self-supervised approach for all label regimes. Specifically, the mIoU of our approach outperforms the SegContrast and From Scratch ones by 2.25% and 3.51%, with 10% labels.

Feature Representation Transferability. To confirm the transferability of the features learned by our approach, we pre-train our models on nuScene and design two settings for pre-training: i) only using sequence 1 (seq1), and ii) using sequence 1 and 2 (seq 1-2) with uniform down-sampling to keep the number of frames consistent with seq1 and the frame rate consistent with KITTI.

As shown in Table 3, our method outperforms training from scratch and Segcontrast [33] when using only seq 1 from nuScene for pre-training and fine-tuning on SemanticKITTI [4]. Our approach improves the segmentation performance by 5.26% in mIoU and 2.77% in Acc. When seq 1 and 2 are used for pre-training, our approach improves the segmentation performance by 5.91% in mIoU and 3.18% in Acc. We also fine-tune the models pre-trained on nuScene or KITTI to the semantic segmentation task using SemanticPOSS [34]. As SemanticPOSS is small, we fine-tune on the entire dataset. Table 4 shows that our method yields better mIoU results than training from scratch. When the network is pre-trained on KITTI, our approach improves the mIoU by 4.20% compared to the network without pre-training.

4.3. Analysis

Guaranteed over-segmentation assumptions. P2C SSL strategy relies on the over-segmentation assumptions and the hyper-parameter leading to the over-segmented clusters is easy to set. To show this, we performed the following experiment. We varied the DBSCAN distance threshold from 0.15 to 0.45 and measured the proportion of clusters having at least 90 % of their points from the same semantic class. The higher this proportion, the higher the chance that the clusters are over-segmented and not under-segmented. In the worst case, we found 73.45% of the clusters being over-segmented witch evidences the stability of the hyper-parameter of DBSCAN.

Performance of unsupervised tracking. We measured that 63.73% of the clusters are tracked for at least 3 frames, and 31.33% for at least 8 frames. Such tracking times are sufficient for us to create positive pairs of clusters observed under different illumination angles as shown in Fig. 7.

4.4. Ablation Study

Experiment in this section, if it is not specially stated, the model pre-training on KITTI, and fine-tuning and eval-

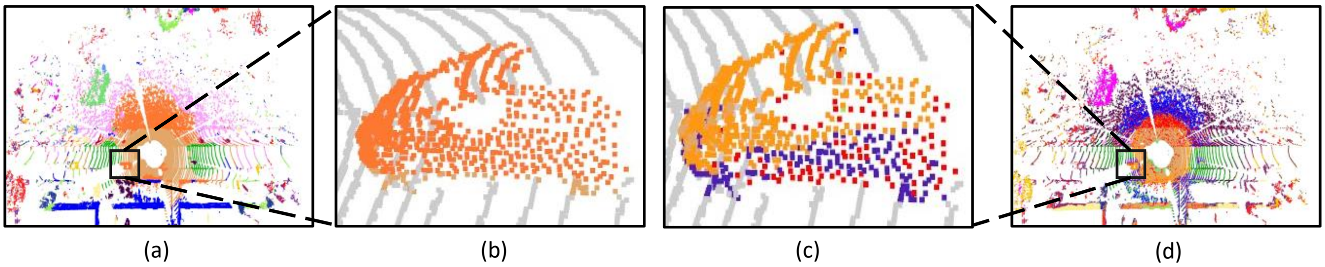

Figure 5. **Comparison of the features generated by different pre-trained models**. (a) Results with P2C. (b) Zoomed in car from (a). (c) Zoomed in car from (d). (d) Results with SegContrast [33]. The points with the same color are in the same cluster(clustering in feature space). The colors of (a, b) and (c, d) are independent and have no relationship with each other. For better visualization, we only colorize the car in (b) and (c).

Method name	mIoU	car	road	sidewalk	building	fence	vegetation	terrain	parking	pole
SegContrast [33]	34.01	89.22	78.72	57.19	82.80	21.99	83.42	67.26	14.06	50.91
SegContrast-BYOL	33.94	89.34	78.56	57.29	82.86	21.19	83.26	67.30	14.09	50.54
SegContrast-Inter	36.03	90.52	81.45	62.90	83.54	21.46	84.37	72.52	17.05	**51.78**
P2C	35.48	90.18	80.90	61.91	83.92	25.45	84.30	71.35	18.73	46.95
STSSL (ours)	**37.71**	**91.11**	**85.34**	**66.09**	**85.43**	**25.63**	**84.79**	**72.57**	**22.61**	48.67

Table 6. **Per-class IoU when fine-tuning with 0.1 % labels**. SegContrast-BYOL: SegContrast built on the basis of BYOL. SegContrast-Inter: SegContrast with a additional inter-frame self-supervised learning stage after the original SegContrast. P2C: Our approach with only the point-to-cluster loss function.

uating on SemanticKITTI.

Scene- vs. Cluster-level Pre-training. To demonstrate the effectiveness of the Point-to-Cluster learning strategy proposed in Section 3.3, we conduct an experiment with only the point-to-cluster loss function of Eq. (2) for pre-training. We dub this setting P2C. As shown in Table 5 and Table 6, a cluster-level pre-trained SegContrast performs better than scene-level pre-trained DepthContrast and STRL. Our proposed P2C pre-training method achieves better results than SegContrast. To better understand the advantages of P2C, we have designed the following visualization. We select a point cloud frame, use the pre-trained model (i.e., SegContrast and P2C) to extract features for each point, and use K-Means [31] to cluster these features. Specifically, we use 20 clusters, which corresponds to the number of categories in the annotations. In Fig. 5, we visualize the points by coloring them according to the clusters. With features extracted by the SegContrast pre-trained model, the car zoomed in in Fig. 5 (c) is divided into several colors. This indicates that the points from the same category can be distant in the learned feature space. By contrast, encouraging the points in the same cluster to have similar features, P2C yields feature such that most of the car points are in the same cluster, as shown in Fig. 5 (b), thus indicating that the features are more representative of the object instances.

Effectiveness of Inter-frame Self-supervised Learning. To better demonstrate the role of inter-frame self-supervised learning, we add a new stage after the original SegContrast, in which we activate our inter-frame loss function, as in our method. Note that we maintain the total number of training epochs to 200. We dub the resulting model SegContrast-inter, and show its per-class IoU in Table 6. For most classes, SegContrast-inter performs much better than SegContrast. However, SegContrast performs better in the fence class. Fence has almost the same appearance when they are viewed from different angles (the z axis remains unchanged). This experiment also supports our motivation that the illumination angle is an important factor, making the appearance of an object differ.

Interval between Two Frames. Since we choose positive pairs from different frames, the interval between the two selected frames is a hyper-parameter affecting the results. In our approach, we gradually increase the interval between two frames to avoid having to set such a hyper-parameter. Here we evaluate the impact of the interval on a model using a fixed interval between two frames. The experiments are repeated for 3 times to reduce randomness, and we report the average. As shown in Fig. 8, with the increase of interval, the performance first increases and then decreases. We think it is because greater interval can reduce the impact of more illumination angles, but the number of clusters that can be tracked across frames decreases as the interval increases. The best-performing model corresponds to an interval of 5, reaching an mIoU of 36.05% when it reaches the balance of illumination angles change and number of tracking clusters. Note that the worst performance (interval of 9) is still better than that of SegContrast.

In Fig. 6, we visualize a frame with multiple cars under different LiDAR illumination angles. We also render the points in the same cluster with the same color. Note that SegContrast and STSSL with interval=0 cannot extract sim-

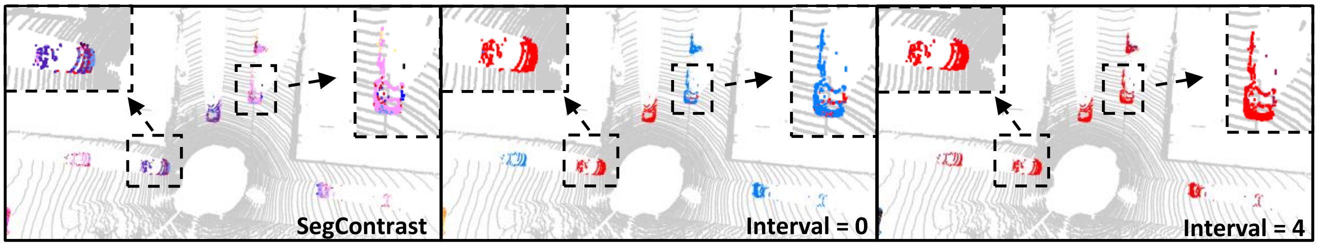

Figure 6. **Comparison of models pre-trained with different intervals between two frames**. Left: SegContrast. Middle: (STSSL, interval=0). Right: (STSSL, interval=4). The pre-trained models are used to extract point features for visualization, and the points with the same color are adjacent in feature space. For better visualization, we only colorize the points belonging to the specified clusters.

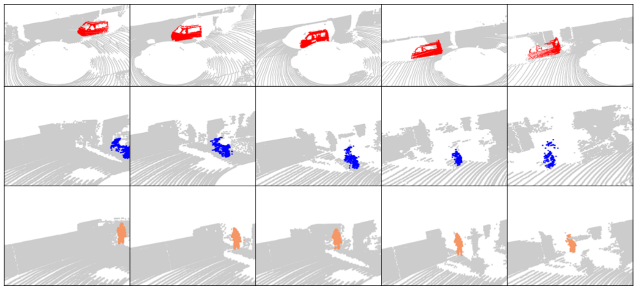

Figure 7. **Associated clusters in multiple frames**. Some clusters with clear semantic information are selected for display. The same color represents the same cluster. Red: car; Blue: motorcycle; Orange: person.

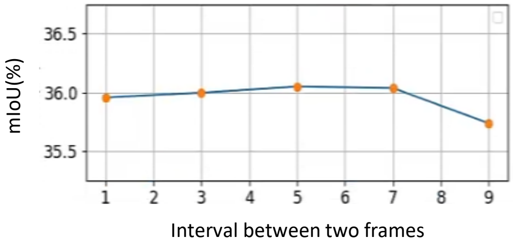

Figure 8. Comparison mIoU on SemanticKITTI dataset, which is fine-tuned by the models pre-trained with **different intervals** between two frames.

mIoU / Acc	0.1 %	1 %
From Scratch	29.17 / 82.57	48.11 / 89.94
SegContrast [33]	34.01 / 84.72	48.02 / 88.84
STSSL-n	36.01 / 87.44	51.06 / 89.68
STSSL	**37.71 / 87.67**	**52.60 / 90.24**

Table 7. **Ablation study on feature similarity for tracking**. STSSL-n only considers location similarity in tracking stage.

Method	mIoU (%)	Acc (%)
SegContrast [33]	34.01	84.72
SegContrast-BYOL	33.94	84.67
STSSL (Ours)	**37.71**	**87.67**

Table 8. **Ablation study on the framework**. SegContrast-BYOL: SegContrast built on the basis of BYOL.

ilar features for the cars under different illumination angles. By contrast, STSSL with interval=4 yields similar features for all cars, benefiting from inter-frame matching.

Tracking with vs. without a Model. In Eq. (1), we use both location and feature similarities to compute the matching degree matrix for tracking. To illustrate the effectiveness of the feature similarity, we re-conduct the tracking with only location similarity. This is indicated by STSSL-n in Table 7. Note that the resulting method still outperforms SegContrast but not our complete STSSL.

Effect of BYOL. To evidence that the benefits of our approach over SegContrast are not only due to our use of BYOL instead of MoCo but truly to our training formalism, we replace the MoCo in SegContrast with BYOL. As shown in Table 8, SegContrast-BYOL performs on par with the original SegContrast (with MoCo) in the 0.1% label regime. It implies that the networks are not the key factor in point clouds SSL. Due to the over-segmentation, we used in our method, where each cluster could belong to the same class, building negative pairs for each point in our point-to-cluster strategy will harm the optimization. By contrast, it is more intuitive for us to only build positive pairs to avoid pushing the nearby clusters away, which offers a chance to merge the cluster with the help of temporal information.

5. Conclusion and Limitation

In this paper, we have introduced an SSL strategy for point cloud segmentation without external supervision. It relies on a novel Point-to-Cluster (P2C) training paradigm to exploit spatial information, and further introduces an inter-frame self-supervised learning strategy to capture temporal information. Altogether, our approach provides a practical tool for pre-training with point clouds in the wild. Experiments evidence that our approach outperforms the state-of-the-art SSL techniques for point cloud in the wild. However, such an improvement does not materialize for objects whose appearance is invariant to viewpoint changes, such as fences. In the future, we will therefore focus on how to improve the segmentation ability of such objects.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grant No. 62006182 and the Swiss National Science Foundation via the Sinergia grant CRSII5-180359.

References

[1] Idan Achituve, Haggai Maron, and Gal Chechik. Self-supervised learning for domain adaptation on point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 123–133, 2021. 1, 2

[2] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International Conference on Machine Learning*, pages 40–49. PMLR, 2018. 1, 2

[3] Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. Scarf: Self-supervised contrastive learning using random feature corruption. *arXiv preprint arXiv:2106.15147*, 2021. 2

[4] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019. 2, 4, 5, 6

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 2, 4

[6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018. 2

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020. 2

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020. 2

[9] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems*, 33:22243–22255, 2020.

[10] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2

[11] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 2

[12] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021. 2

[13] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. 2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12547–12556, 2021. 1

[14] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 4, 5

[15] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996. 3

[16] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3

[17] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 4

[18] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the ieee/cvf International Conference on Computer Vision*, pages 6391–6400, 2019. 2

[19] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 2, 4, 5

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 2

[21] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020. 2

[22] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15587–15597, 2021. 1, 2

[23] Dapeng Hu, Shipeng Yan, Qizhengqiu Lu, HONG Lanqing, Hailin Hu, Yifan Zhang, Zhenguo Li, Xinchao Wang, and Jiashi Feng. How well does self-supervised pre-training perform with streaming data? In *International Conference on Learning Representations*, 2021. 2

[24] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6535–6545, 2021. 2, 5, 6

[25] Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering networks. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[26] Longlong Jing, Yucheng Chen, Ling Zhang, Mingyi He, and Yingli Tian. Self-supervised modal and view invariant feature learning. *arXiv preprint arXiv:2005.14169*, 2020. 1, 2

[27] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058, 2020. 1

[28] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021. 2

[29] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 4

[30] Hanxue Liang, Chenhan Jiang, Dapeng Feng, Xin Chen, Hang Xu, Xiaodan Liang, Wei Zhang, Zhenguo Li, and Luc Van Gool. Exploring geometry-aware contrast and clustering harmonization for self-supervised 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3293–3302, 2021. 1, 2

[31] J MacQueen. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297, 1967. 7

[32] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *International Conference on Machine Learning*, pages 2391–2400. PMLR, 2017. 2

[33] Lucas Nunes, Rodrigo Marcuzzi, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Segcontrast: 3d point cloud feature representation learning through self-supervised segment discrimination. *IEEE Robotics Autom. Lett.*, 7(2):2116–2123, 2022. 1, 2, 4, 5, 6, 7, 8

[34] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. Semanticposs: A point cloud dataset with large quantity of dynamic instances. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 687–693. IEEE, 2020. 2, 4, 5, 6

[35] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2

[36] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *European Conference on Computer Vision*, pages 685–702. Springer, 2020. 1

[37] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020. 2

[38] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1096–1103, 2008. 2

[39] Yifei Wang, Qi Zhang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap. *arXiv preprint arXiv:2203.13457*, 2022. 2

[40] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pretraining for 3d point cloud understanding. In *European Conference on Computer Vision*, pages 574–591. Springer, 2020. 1, 2

[41] Junbo Yin, Dingfu Zhou, Liangjun Zhang, Jin Fang, Cheng-Zhong Xu, Jianbing Shen, and Wenguan Wang. Proposalcontrast: Unsupervised pre-training for lidar-based 3d object detection. *arXiv preprint arXiv:2207.12654*, 2022. 1, 2

[42] Tong Zhang, Pan Ji, Mehrtash Harandi, Richard Hartley, and Ian Reid. Scalable deep k-subspace clustering. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part V 14*, pages 466–481. Springer, 2019. 2

[43] Tong Zhang, Pan Ji, Mehrtash Harandi, Wenbing Huang, and Hongdong Li. Neural collaborative subspace clustering. In *International Conference on Machine Learning*, pages 7384–7393. PMLR, 2019. 2

[44] Tong Zhang, Congpei Qiu, Wei Ke, Sabine Süsstrunk, and Mathieu Salzmann. Leverage your local and global representations: A new self-supervised learning strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16580–16589, 2022. 2

[45] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10252–10263, 2021. 1, 2, 5, 6

[46] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9939–9948, 2021. 1