ProposalContrast: Unsupervised Pre-training for LiDAR-based 3D Object Detection

Junbo Yin[1†], Dingfu Zhou[2,3], Liangjun Zhang[2,3], Jin Fang[2,3,4],
Cheng-Zhong Xu[4], ✉Jianbing Shen[4], and ✉Wenguan Wang[5]

[1]School of Computer Science, Beijing Institute of Technology   [2]Baidu Research
[3]National Engineering Laboratory of Deep Learning Technology and Application, China
[4]SKL-IOTSC, CIS, University of Macau   [5]ReLER, AAII, University of Technology Sydney
{yinjunbocn,wenguanwang.ai}@gmail.com
https://github.com/yinjunbo/ProposalContrast

Abstract. Existing approaches for unsupervised point cloud pre-training are constrained to either scene-level or point/voxel-level instance discrimination. Scene-level methods tend to lose local details that are crucial for recognizing the road objects, while point/voxel-level methods inherently suffer from limited receptive field that is incapable of perceiving large objects or context environments. Considering region-level representations are more suitable for 3D object detection, we devise a new unsupervised point cloud pre-training framework, called *ProposalContrast*, that learns robust 3D representations by contrasting region proposals. Specifically, with an exhaustive set of region proposals sampled from each point cloud, geometric point relations within each proposal are modeled for creating expressive proposal representations. To better accommodate 3D detection properties, *ProposalContrast* optimizes with both inter-cluster and inter-proposal separation, *i.e.*, sharpening the discriminativeness of proposal representations across semantic classes and object instances. The generalizability and transferability of *ProposalContrast* are verified on various 3D detectors (*i.e.*, PV-RCNN, CenterPoint, PointPillars and PointRCNN) and datasets (*i.e.*, KITTI, Waymo and ONCE).

**Keywords:** 3D Object Detection, Unsupervised Point Cloud Pre-training

1   Introduction

3D object detection from LiDAR point clouds has received great interest in recent years due to its significance to self-driving vehicles. Most existing 3D detectors are trained on massive labeled data. However, annotating point clouds is expensive and time-consuming. On the other hand, unlabeled point cloud data can be easily generated by self-driving vehicles: it is estimated that a self-driving car could collect 200k point cloud frames within only 8 working hours.

✉: Corresponding author.   †: Work done during an internship at Baidu Research.

LiDAR Point Cloud   **Point-wise Pre-training**   **Proposal-wise Pre-training**
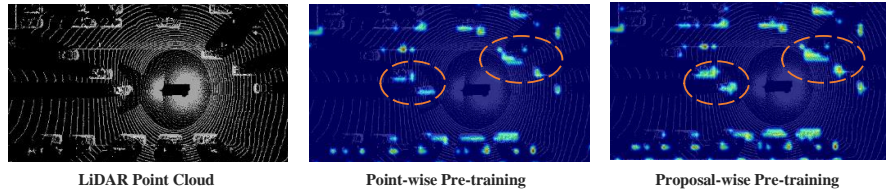
(a) Compared with our proposal-level pre-training, point-level pre-training tends to produce incomplete object representations (as indicated by the dotted circle).

LiDAR Point Cloud   **Point-wise Pre-training**   **Proposal-wise Pre-training**
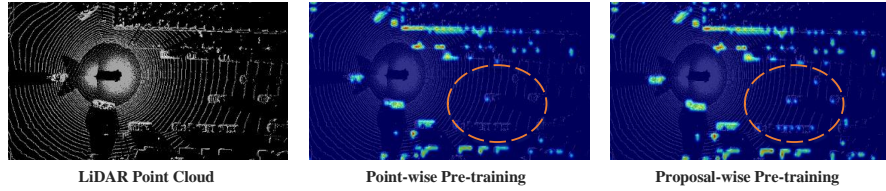
(b) By exploring larger context, our proposal-level pre-training successfully perceives objects that are missed by the point-wise method (as indicated by the dotted circle).

Fig. 1. Comparison of VoxelNet representations learned from PointContrast [38] and our *ProposalContrast*.

Self-supervised learning (SSL) [5,12,13,20,46] provides a feasible way to make use of unlabeled data. SSL methods typically define a pretext task, where free supervision signals can be derived from the data itself for representation learning. With a few labeled data of downstream tasks, the learnt representations can be fine-tuned and show excellent performance. Recent advances in SSL can be largely ascribed to contrastive learning [5, 6, 13, 14]. Contrastive learning explores the pretext task of instance discrimination, *i.e.*, maximizing the agreement of the feature embeddings between two differently augmented views of the same data instance, and minimizing the agreement between different instances. The instances are often defined as images in the 2D domain. Intuitively, the instance discrimination pretext relies on an *object-centric* assumption [4, 48], where the object of interest should lie in the center of an image such that different augmentations can be applied to achieve cross-view consistency. Although this is hold for some image datasets like ImageNet [9], it is suboptimal for LiDAR point cloud datasets; taking Waymo [33] as an example, objects are of smaller sizes (*e.g.*, $4 \times 2m^2$ for car), and are unevenly distributed across a considerably wide range (*e.g.*, $150 \times 150m^2$ for a point cloud scene).

Therefore, how to define the instances is crucial for the adoption of contrastive SSL techniques in point clouds. However, previous approaches for point cloud SSL either directly contrast different views of a whole point cloud scene [15, 47], or merely focus on point-/voxel-level instance discrimination [21, 38]. The scene-level methods struggle to describe the locality of the road objects, while the point-/voxel-level methods overemphasize fine-grained details, lacking object-

level characteristics. Each point cloud consists of several objects and the objects such as vehicles typically contain numerous points and span several voxels. Thus previous approaches take little consideration of the properties of point data, hurting the utility of the learned representations in downstream tasks, such as 3D object detection. In light of the analysis above, we believe learning point representations on the proposal-level is more desired (see Fig. 1 for more evidence).

In this work, we propose a proposal-level point cloud SSL framework, named *ProposalContrast*, that conducts proposal-wise contrastive pre-training for 3D detection-aligned representation learning. In particular, for each point cloud region proposal, its representation is designed to explicitly encode the geometrical relations of the points inside the proposal. This is achieved by a cross-attention based encoding module, which attends each point to its neighbors [22]. To better align proposal-level pre-training with the nature of 3D detection, we propose to jointly optimize two pretext tasks, *i.e.*, inter-proposal discrimination (IPD) and inter-cluster separation (ICS). IPD is for instance-discriminative representation learning. Through minimizing a proposal-wise contrastive loss, proposal representations are encouraged to gather instance-specific characteristics, eventually benefiting the localization of objects. Differently, ICS is for class-discriminative representation learning. Since class label is not available during pre-training, ICS conducts cluster-based contrastive learning. It groups proposals into clusters and enforces consistency between cluster predictions (*i.e.*, pseudo and soft class labels) of different views of each proposal. In this way, ICS encourages the proposal representations to abstract instance-invariant, common patterns, hence facilitating the semantic recognition of objects.

We comprehensively demonstrate the generalizability of our *ProposalContrast* on prevalent 3D detection network architectures, *i.e.*, PV-RCNN [30], Center-Point [45], PointPillars [19], and PointRCNN [31], as well as empirically validate the transferability on three popular self-driving point cloud datasets, *i.e.*, Waymo [33], KITTI [11], and ONCE [24]. Moreover, *ProposalContrast* can significantly save the annotation cost, *e.g.*, with only half annotations, our pre-trained PV-RCNN outperforms the scratch model trained with full annotations.

2   Related Work

3D Object Detection. Existing solutions for 3D object detection can be broadly grouped into two classes in terms of point representations, *i.e.*, grid-based and point-based. Specifically, grid-based methods [19, 39, 43, 44, 49] typically first discretize the raw point clouds into regular girds (*e.g.*, voxels [39] or pillars [19]), which can then be processed by 3D or 2D convolutional networks. Point-based methods [25, 26, 28, 31, 40] directly extract features and predict 3D objects from raw point clouds. In general, grid-based approaches are efficient but suffer from information loss in the quantification process. Point-based methods yield impressive results, but the computational cost is high for large-scale point clouds. Hence some recent detectors [30], built upon the hybrid of voxel-and point-based architectures, are developed to enjoy the best of both worlds.

In this work, we show that a wide range of modern 3D detectors can benefit from our self-supervised pre-training algorithm, which can learn meaningful and transferable representations from large-scale unlabeled LiDAR point clouds.

Self-supervised Learning (SSL) in Point Cloud. SSL [14,16,18,27,41,46] is to learn expressive feature representation without manual annotations. Recently, contrastive learning based SSL algorithms [5, 13, 14, 36, 42] proved impressive results on various downstream tasks, even surpassing the supervised alternatives. In this article, we follow this paradigm with a proposal-level pretraining method specifically designed for the task of point cloud object detection. Concurrent to our study, PointContrast [38], DepthContrast [47], GCC-3D [21], and STRL [15] also exploit the potential of contrastive SSL in point cloud pretraining. Though impressive, these methods have a few limitations. First, [15, 47] takes the whole point cloud scene as the instance, neglecting the underlying *object-centric* assumption [4, 48], since self-driving point cloud scenes typically comprise of multiple object instances. Second, other methods like [21, 38] only take into account instance discrimination at the point-/voxel-level. Thus they are hard to acquire object-level representations that are compatible with 3D object detection. Third, [15, 38, 47] overlook the semantic relations among instances, focusing on modeling low-level characteristics instead of high-level informative patterns. In [21], although an extra self-clustering strategy is introduced for capturing semantic properties, it only provides supervisory signals on moving voxels which are too sparse to cover potential object candidates, and leads to a complicated two-stage pipeline that trains the 3D encoder and 2D encoder separately. Differently, our point cloud pretraining method encourages discrimination between proposal instances and clusters, hence comprehensively capturing the properties of point cloud data and well aligning with 3D detection. Although RandomRooms [29] also considers region-level representations, it refers to synthetic CAD objects in indoor environments. In contrast, we automatically mine potential object instances from raw point data in self-driving scenes.

3   Approach

In Sec. 3.1, we first briefly describe our proposal-level pre-training method specifically designed for 3D object detection. Then, we detail the crucial components, *i.e.*, region proposal encoding module (in Sec. 3.2) and joint optimization of inter-proposal discrimination and inter-cluster separation (in Sec. 3.3).

3.1   Overview of *ProposalContrast*

Due to the specific characteristics of 3D LiDAR point cloud data, such as the irregular and sparse structures across large perception ranges, simply applying 2D pre-training techniques to point clouds cannot get satisfactory results. This calls for better adapting existing pre-training techniques to the inherent structures of point cloud data. Rather than current point cloud pretraining methods investigating the unsupervised representation learning on the scene-/point-/voxel-
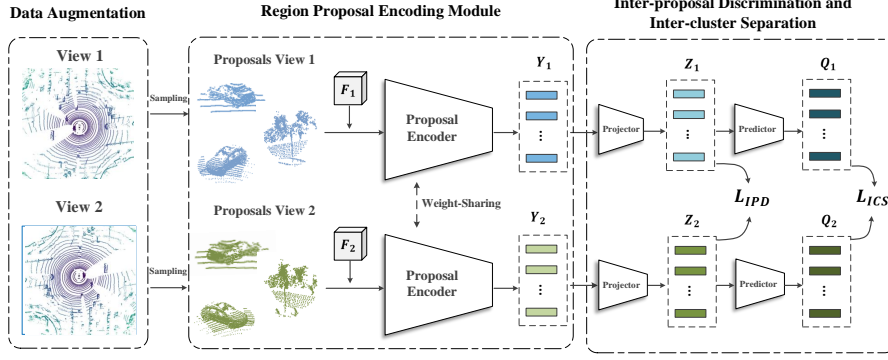
Fig. 2. Illustration of our *ProposalContrast* framework. Given augmented point cloud with different views, we first sample paired region proposals and then extract the features with a region proposal encoding module. After that, inter-proposal discrimination and inter-cluster separation are enforced to optimize the whole network.

level [15, 38, 38, 47], our *ProposalContrast* learns representations by contrasting directly on region proposals.

As shown in Fig. 2, to achieve proposal-level pre-training, *ProposalContrast* has five core components: data augmentation, correspondence mapping, region proposal generation, region proposal encoding, as well as a joint optimization of inter-proposal discrimination (IPD) and inter-cluster separation (ICS).

Data Augmentation. Let $X_0 \in \mathbb{R}^{L_0 \times 3}$ denote an input point cloud with L_0 points (here we describe point cloud with 3D coordinates for simplicity). We apply two different data augmentation operators $\mathcal{T}_1, \mathcal{T}_2$ on X_0 to produce two augmented views X_1, X_2:

$$X_1 = \mathcal{T}_1(X_0) \in \mathbb{R}^{L_1 \times 3}, \quad X_2 = \mathcal{T}_2(X_0) \in \mathbb{R}^{L_2 \times 3}, \tag{1}$$

where L_1 and L_2 are number of points of X_1 and X_2. The data augmentation strategies include random rotation, flip, scaling and random point drop out (see Sec. 4.1 for detailed definition). Note that random rotation is only applied on the upright axis since we are aware of self-driving scenarios.

Correspondence Mapping. Before generating proposals, we first get the correspondence mapping M between point sets in X_1 and X_2. This can be easily achieved by recording the index of each point in the original view X_0. M is later used for sampling and grouping region proposals.

Region Proposal Generation. Some SSL methods for 2D representation learning make use of image proposals in the form of 2D bounding boxes [4]. For point clouds, straightforwardly representing proposals as 3D bounding boxes, however, is not a feasible choice, due to the significantly enlarged space of object candidates in 3D scenarios. In addition, generating a dense set of 3D bounding boxes will lead to unacceptable high computational cost. These considerations motivate us to adopt spherical proposals, instead of 3D bounding box proposals.
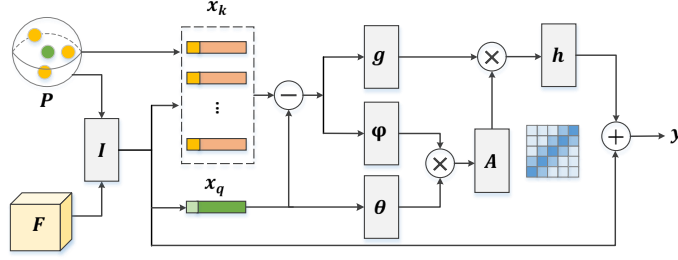
Fig. 3. Illustration of the region proposal encoding module (Sec. 3.2). The encoding module adopts a cross-attention mechanism that attends the center query point with its neighbor key points, for collecting expressive proposal representations.

Specifically, given the input point cloud X_0, we first abandon the road plane points so as to sample less from the background [1]. Then we perform farthest point sampling (FPS) [28] on X_0 to sample a total of N points as the centers of N spherical proposals (the corresponding samples in X_1 and X_2 can also be identified according to the correspondence M). FPS encourages the sampled points to be away from each other and thus guarantees the diversity of the sampled proposals. Next, spherical proposals are generated by searching K points around each sampled center point within a pre-defined radius r. Finally, we get two sets of spherical proposals $P_1, P_2 \in \mathbb{R}^{N \times K \times 3}$ from the two views, *i.e.*, $P_1 \in X_1, P_2 \in X_2$.

Region Proposal Encoding. An encoding module is further adopted to extract expressive proposal representations, by considering the geometric relations of the points inside each proposal. Given the scene-level representations, *i.e.*, $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$, of X_1 and X_2, extracted by the backbone network, the proposal encoding module outputs geometry-aware proposal representations $\boldsymbol{Y}_1, \boldsymbol{Y}_2 \in \mathbb{R}^{N \times C}$ for P_1 and P_2:

$$\boldsymbol{Y}_1 = f_{\text{En}}(P_1, \boldsymbol{F}_1), \quad \boldsymbol{Y}_2 = f_{\text{En}}(P_2, \boldsymbol{F}_2), \tag{2}$$

where the encoding module f_{En} is achieved by a neural attention mechanism, which will be detailed in Sec. 3.2.

Joint Optimization. The proposal features $\boldsymbol{Y}_1, \boldsymbol{Y}_2$ from different views are learned by enforcing both cluster-based class consistency and instance-wise discrimination. This endows $\boldsymbol{Y}_1, \boldsymbol{Y}_2$ with desired properties for 3D object detection. The detailed design is presented in Sec. 3.3.

3.2   Region Proposal Encoding Module

The relations among points in a proposal provide crucial geometry information for describing the proposal. To extract better proposal representations, we leverage the cross-attention [10] to capture the geometric relations between points.

For a point cloud scene X and corresponding proposals P, we extract its global scene-wise representation through a backbone network, *i.e.*, $\boldsymbol{F} = f_{\text{Bbone}}(X)$.

Popular 3D backbones like VoxelNet [39] and PointNet++ [28] can be used to instantiate f_{Bbone}.

Then we obtain initial representation $\boldsymbol{P} \in \mathbb{R}^{N \times K \times C}$ for all the proposals P by applying bilinear interpolation function $I(\cdot)$ over $\boldsymbol{F}$, *i.e.*, $\boldsymbol{P} = I(P, \boldsymbol{F})$, where N is the proposal number of one view, K is the point number inside a proposal and C is the backbone channel number.

After that, we capture the geometrical relation between points via cross attention, *i.e.*, attending a query point with other key points. Let $\boldsymbol{p} \in \boldsymbol{P}$ with $K \times C$ size denote the initial representation of a proposal $p \in P$. As illustrated in Fig. 3, we set the center point feature $\boldsymbol{x}_q \in \mathbb{R}^{1 \times C}$ of proposal p, *i.e.*, $\boldsymbol{x}_q \in \boldsymbol{p}$ as the query, since the center point is more informative [45]. Next, we get the neighbor features $\boldsymbol{x}_k \in \mathbb{R}^{K \times C}$, where $\boldsymbol{x}_k \in \boldsymbol{p}$, and use the difference between $\boldsymbol{x}_k$ and $\boldsymbol{x}_q$ as keys to encode the asymmetric geometry relation as recommended by [37]. The point coordinate of $\boldsymbol{x}_q$ and $\boldsymbol{x}_k$ are also integrated to provide explicit position information.

Formally, the $\boldsymbol{x}_q$ and $\boldsymbol{x}_k$ are first projected to query, key, and value vectors through:

$$\boldsymbol{w}_q = \theta(\boldsymbol{x}_q), \quad \boldsymbol{w}_k = \phi(\boldsymbol{x}_k - \boldsymbol{x}_q), \quad \boldsymbol{w}_v = g(\boldsymbol{x}_k - \boldsymbol{x}_q), \tag{3}$$

where $\theta(\cdot)$, $\phi(\cdot)$, and $g(\cdot)$ are linear layers. Afterwards, we compute the attention weight $A \in [0,1]^K$ based on normalized similarities between the query $\boldsymbol{w}_q$ and each key $\boldsymbol{w}_k$. The attention weight is then applied on the value vector $\boldsymbol{w}_v$ to aggregate information from all the keys:

$$\boldsymbol{w}_o = \sum_{\boldsymbol{w}_k} A(\boldsymbol{w}_q, \boldsymbol{w}_k) \cdot \boldsymbol{w}_v, \quad A(\boldsymbol{w}_q, \boldsymbol{w}_k) = \frac{\boldsymbol{w}_q^\top \boldsymbol{w}_k}{\sum_{\boldsymbol{w}_k} \boldsymbol{w}_q^\top \boldsymbol{w}_k}. \tag{4}$$

As seen, the attention weight encodes the geometry relation of each point pair under a metric space, which makes the center point aware of the informative neighborhoods and thus encourages effective information exchange between points inside a proposal. Finally, the representation of proposal p is given as:

$$\boldsymbol{y} = \boldsymbol{x}_q + h(\boldsymbol{w}_o), \tag{5}$$

where $h(\cdot)$ is a linear layer. In this way, by applying Eqs. (3-5) on each the region proposal in X_1 and X_2, we got all the proposal representations, *i.e.*, $\boldsymbol{Y}_1, \boldsymbol{Y}_2 \in \mathbb{R}^{N \times C}$, which will be trained by joint optimization of inter-proposal discrimination and inter-cluster separation.

3.3   Joint Optimization of Inter-proposal Discrimination and Inter-cluster Separation

A good point cloud representation for 3D detection is desired to be discriminative across both instances and classes; object instance-sensitive representation benefits the localization of 3D objects, while class-discriminative representation is crucial for the recognition of object categories. We therefore propose to optimize with both inter-proposal discrimination (IPD) and inter-cluster separation

(ICS) simultaneously. Because annotations for object bounding box and semantics labels are not given during pretraining, IPD is designed to contrast the representations on the proposal-level, while ICS discovers stable and informative 3D patterns through grouping the proposals into clusters.

Inter-proposal discrimination (IPD). For object instance-sensitive representation learning, we conduct proposal-level contrastive learning [5]. Specifically, given the proposal representations, $i.e.$, $\boldsymbol{Y}_1, \boldsymbol{Y}_2 \in \mathbb{R}^{N \times C}$, from the two different views X_1, X_2. We aim to pull positive pairs ($e.g.$, the same proposal with different views) close, as well as push negative pairs ($e.g.$, the different proposals) apart.

Following the common practice in contrastive learning, we first adopt a projection layer $f_{\mathrm{Proj}}(\cdot)$ to project each proposal representation $\boldsymbol{y} \in \{\boldsymbol{Y}_1, \boldsymbol{Y}_2\}$ to a ℓ_2-normalized embedding space:

$$\boldsymbol{z} = \frac{f_{\mathrm{Proj}}(\boldsymbol{y})}{||f_{\mathrm{Proj}}(\boldsymbol{y})||_2} \in [0,1]^C. \tag{6}$$

Given the sets of normalized proposal embeddings $\boldsymbol{Z}_1, \boldsymbol{Z}_2 \in [0,1]^{N \times C}$, the IPD loss is designed in a form of the InfoNCE loss [27]:

$$\mathcal{L}_{\mathrm{IPD}} = \frac{1}{N} \sum_{\boldsymbol{z}_1^n \in \boldsymbol{Z}_1} -\log \frac{\exp(\boldsymbol{z}_1^{n\top} \cdot \boldsymbol{z}_2^{n'}/\tau)}{\sum\limits_{\boldsymbol{z}_2^m \in \boldsymbol{Z}_2} \exp(\boldsymbol{z}_1^{n\top} \cdot \boldsymbol{z}_2^m/\tau)} + \frac{1}{N} \sum_{\boldsymbol{z}_2^m \in \boldsymbol{Z}_2} -\log \frac{\exp(\boldsymbol{z}_2^{m\top} \cdot \boldsymbol{z}_1^{m'}/\tau)}{\sum\limits_{\boldsymbol{z}_1^n \in \boldsymbol{Z}_1} \exp(\boldsymbol{z}_2^{m\top} \cdot \boldsymbol{z}_1^n/\tau)}, \tag{7}$$

where τ is a temperature hyper-parameter. For $\boldsymbol{z}_1^n$, $\boldsymbol{z}_2^{n'}$ refers to a positive sample; $\boldsymbol{z}_1^n$ and $\boldsymbol{z}_2^{n'}$ correspond to a same proposal in X_0. Similarly, $\boldsymbol{z}_2^m$ is a negative sample for $\boldsymbol{z}_1^n$, $i.e.$, $\boldsymbol{z}_2^m \neq \boldsymbol{z}_2^{n'}$.

Inter-cluster Separation (ICS). For class-discriminative representation learning, we group the spherical proposals into different clusters, which can be viewed as pseudo class labels. Inspired by recent clustering-based contrastive SSL [2], ICS is designed to maximize the agreement between cross-view cluster assignments.

For each normalized proposal feature $\boldsymbol{z}_1^n \in \boldsymbol{Z}_1$ from view X_1, we first apply a predictor $f_{\mathrm{Pred}}(\cdot)$ to map it to vector $\boldsymbol{q}_1^n \in \mathbb{R}^O$ that represents its pseudo-class embedding, such that $\boldsymbol{q}_1^n = f_{\mathrm{Pred}}(\boldsymbol{z}_1^n)$, where the output channel dimension O of $\boldsymbol{q}_1^n$ refers to the number of pseudo classes. To get cluster assignment, $i.e.$, $\hat{\boldsymbol{q}}_1^n \in [0,1]^O$, we stop gradient on $\boldsymbol{q}_1^n$ and adopt Sinkhorn-Knopp algorithm [7] to group all the proposals in each training batch into O clusters. Then the training target of ICS is defined as:

$$\mathcal{L}_{\mathrm{ICS}} = \frac{1}{N} \sum_{\boldsymbol{q}_1^n} -\hat{\boldsymbol{q}}_1^n \log \sigma(\boldsymbol{q}_2^{n'}) + \frac{1}{N} \sum_{\boldsymbol{q}_2^m} -\hat{\boldsymbol{q}}_2^m \log \sigma(\boldsymbol{q}_1^{m'}), \tag{8}$$

where $\sigma(\cdot)$ refers to the softmax function that maps the pseudo-class embedding into class probability distribution. $\mathcal{L}_{\mathrm{ICS}}$ encourages the cross-view clustering consistency of each proposal, $i.e.$, use the cluster assignment $\hat{\boldsymbol{q}}_1^n$ (resp. $\hat{\boldsymbol{q}}_2^m$) of view X_1 (resp. X_2) as pseudo grouptruth to supervise the class probability distribution $\sigma(\boldsymbol{q}_2^{n'})$ (resp. $\sigma(\boldsymbol{q}_1^{m'})$). Note that $\hat{\boldsymbol{q}}_1^n$ and $\boldsymbol{q}_2^{n'}$ correspond to a same proposal in X_0.

Finally, the overall self-supervised learning loss is defined as:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{IPD}} + \beta \mathcal{L}_{\text{ICS}}, \tag{9}$$

where α and β are the balancing coefficients, respectively. In Sec. 4.3, we provide analysis on the effectiveness of the two optimization targets.

4   Experimental Results

4.1   Pre-training Settings

Datasets. We adopt the common experimental protocol of SSL, *i.e.*, first pre-training a backbone network with large-scale unlabeled data and then fine-tuning it on downstream tasks with much fewer labeled data. Some previous 3D SSL methods make use of ShapeNet [3] and ScanNet [8] datasets to pre-train the 3D backbones, thus they only focus on the indoor setting and suffer from large domain gap when transferring to the self-driving setting. In our experiments, we adopt Waymo Open Dataset [33] for the self-supervised pre-training. The Waymo dataset contains 798 scenes (158,361 frames) for training and 202 scenes (40,077 frames) for validation; it is 20× larger than KITTI [11]. We adopt the whole training set to pre-train various 3D backbones without using the labels.

Network Architectures. For thorough examination of the efficacy and versatility of our approach, we investigate the performance of *ProposalContrast* on diverse 3D backbone architectures, including grid-based, *i.e.*, VoxelNet [39] and PointPillars [19], as well as point-based, *i.e.*, PointNet++ [28]. The projection layer, $f_{\text{Proj}}(\cdot)$, in Sec. 3.3 is implemented as two linear layers, with the first layer followed by a batch normalization (BN) layer and a ReLU. The channel dimension of the output is set as $C = 128$. The predictor $f_{\text{Pred}}(\cdot)$, implemented as a linear layer, outputs a 128-d vector as the pseudo-class embedding, *i.e.*, $O = 128$. All the functions in the proposal encoding module are instantiated by linear layers with 128 channels. The attention linear head $h(\cdot)$ transforms the attended features to the original backbone channels.

Implementation Details. We empirically consider four types of data augmentations to generate different views, including random rotation ($[-180°, 180°]$), random scaling ($[0.8, 1.2]$), random flipping along X-axis or Y-axis, and random point drop out. For random point drop out, we sample 100k points from the original point cloud for each of the two augmented views. 20k points are chosen from the same indexes to ensure a 20% overlap for the two augmented views, while the other 80k points are randomly sampled from the remained point clouds. We sample $N = 2048$ spherical proposals for every point cloud frame; each proposal contains $K = 16$ points within $r = 1.0$ m radius. The parameters for the VoxelNet [39] backbone are the same as the corresponding 3D object detectors. The temperature parameter τ in the IPD loss $\mathcal{L}_{\text{IPD}}$ (Eq. 7) is set to 0.1. The coefficients α and γ in Eq. 9 are both set to 1 empirically. We pre-train the models for 36 epochs, and use Adam optimizer [17] to optimize the network. Cosine learning rate schedule [23] is adopted with warmup strategy in the first 5 epochs. The maximum learning rate is set to 0.003.

Table 1. Data-efficient 3D Object Detection on KITTI. We pre-train the backbones of PointRCNN [31] and PV-RCNN [30] on Waymo and transfer to KITTI 3D object detection with different label configurations. Consistent improvements are obtained under each setting. Our approach outperforms all the concurrent self-supervised learning methods, *i.e.*, DepthContrast [47], PointContrast [38], GCC-3D [21], and STRL [15].

Fine-tuning with various label ratios	Detector	Pre-train. Schedule	mAP (Mod.)	Car Easy	Mod.	Hard	Pedestrian Easy	Mod.	Hard	Cyclist Easy	Mod.	Hard
20% ($\sim$ 0.7k frames)	PointRCNN	Scratch	63.51	88.64	75.23	72.47	55.49	48.90	42.23	85.41	66.39	61.74
		Ours	**66.20**$_{+2.69}$	88.52	**77.02**$_{+1.79}$	72.56	58.66	**51.90**$_{+3.00}$	44.98	90.27	**69.67**$_{+3.28}$	65.05
	PV-RCNN	Scratch	66.71	91.81	82.52	80.11	58.78	53.33	47.61	86.74	64.28	59.53
		Ours	**68.13**$_{+1.42}$	91.96	**82.65**$_{+0.13}$	80.15	62.58	**55.05**$_{+1.72}$	50.06	88.58	**66.68**$_{+2.40}$	62.32
50% ($\sim$ 1.8k frames)	PointRCNN	Scratch	66.73	89.12	77.85	75.36	61.82	**54.58**	47.90	86.30	67.76	63.26
		Ours	**69.23**$_{+2.50}$	89.32	**79.97**$_{+2.12}$	77.39	62.19	54.47$_{-0.11}$	46.49	92.26	**73.25**$_{+5.69}$	68.51
	PV-RCNN	Scratch	69.63	91.77	82.68	81.9	63.70	57.10	52.77	89.77	69.12	64.61
		Ours	**71.76**$_{+2.13}$	92.29	**82.92**$_{+0.24}$	82.09	65.82	**59.92**$_{+2.82}$	55.06	91.87	**72.45**$_{+3.33}$	67.53
100% ($\sim$ 3.7k frames)	PointRCNN	Scratch	69.45	90.02	**80.56**	78.02	62.59	55.66	48.69	89.87	72.12	67.52
		DepthCon. [47]	70.26$_{+0.81}$	89.38	80.32$_{-0.24}$	77.92	65.55	57.62$_{+1.96}$	50.98	90.52	72.84$_{+0.72}$	68.22
		Ours	**70.71**$_{+1.26}$	89.51	80.23$_{-0.33}$	77.96	66.15	**58.82**$_{+3.16}$	52.00	91.28	**73.08**$_{+0.96}$	68.45
	PV-RCNN	Scratch	70.57	-	84.50	-	-	57.06	-	-	70.14	-
		GCC-3D [21]	71.26$_{+0.69}$	-	-	-	-	-	-	-	-	-
		STRL [15]	71.46$_{+0.89}$	-	84.70$_{+0.20}$	-	-	57.80$_{+0.74}$	-	-	71.88$_{+1.74}$	-
		PointCon. [38]	71.55$_{+0.98}$	91.40	84.18$_{-0.32}$	82.25	65.73	57.74$_{+0.68}$	52.46	91.47	72.72$_{+2.58}$	67.95
.		Ours	**72.92**$_{+2.35}$	92.45	**84.72**$_{+0.22}$	82.47	68.43	**60.36**$_{+3.30}$	55.01	92.77	**73.69**$_{+3.55}$	69.51

4.2   Transfer Learning Settings and Results

In this paper, we investigate self-supervised pre-training in an autonomous driving setting. We evaluate our approach on several widely used LiDAR point cloud datasets, *i.e.*, KITTI [11], Waymo [33] and ONCE [24]. Specifically, we compare our *ProposalContrast* with other pre-training methods by fine-tuning the detection models. Different amounts of labeled data are used for fine-tuning to show the data-efficient ability. Besides, various modern 3D object detectors are involved to demonstrate the generalizability of our pre-trained models.

KITTI Dataset. KITTI 3D object benchmark [11] has been widely used for 3D object detection from LiDAR point cloud. It contains 7,481 labeled samples, which are divided into two groups, *i.e.*, a training set (3,712 samples) and a validation set (3,769 samples). Mean Average Precision (mAP) with 40 recall positions are usually adopted to evaluate the detection performance, with a 3D IoU thresholds of 0.7 for cars and 0.5 for pedestrians and cyclists.

We assess the transferability of our pre-trained model by pre-training on Waymo then fine-tuning on KITTI. Two typical 3D object detectors, *i.e.*, PointR-CNN [31] and PV-RCNN [30], are used as the baselines. The two detectors are based on different 3D backbones (*i.e.*, point-wise or voxel-wise networks), covering most cases of mainstream 3D detectors. A crucial advantage of self-supervised pre-training is to improve data efficiency for the downstream task with limited annotated data. To this end, we evaluate the data-efficient 3D object detection. In particular, the training samples are split into three groups, with each containing 20% (0.7k), 50% (1.8k) and 100% (3.7k) labeled samples, respectively. The experimental results are shown in Table 1. On both 3D detectors, our self-supervised pre-trained model effectively improves the performance in comparison to the model trained from scratch. Our model also outperforms several concurrent works. For example, based on PV-RCNN detector, our model

Table 2. Comparisons between our model and other self-supervised learning methods on Waymo. All the detectors are trained by 20% training samples following the OpenPCDet [34] configuration and evaluated on the validation set. Both PV-RCNN [30] and CenterPoint [45] are used as beseline detectors.

3D Object Detector	Transfer Paradigm	Overall AP/APH	Vehicle AP/APH	Pedestrian AP/APH	Cyclist AP/APH
SECOND [39]	Scratch	55.08/51.32	59.57/59.04	53.00/43.56	52.67/51.37
Part-A^2-Anchor [32]	Scratch	60.39/57.43	64.33/63.82	54.24/47.11	62.61/61.35
PV-RCNN [30]	Scratch	59.84/56.23	64.99/64.38	53.80/45.14	60.72/59.18
GCC-3D (PV-RCNN) [21]	Fine-tuning	61.30/58.18$_{(+1.46/+1.95)}$	65.65/65.10	55.54/48.02	62.72/61.43
Ours (PV-RCNN)	Fine-tuning	**62.62/59.28**$_{(+2.78/+3.05)}$	66.04/65.47	57.58/49.51	64.23/62.86
CenterPoint [45]	Scratch	63.46/60.95	61.81/61.30	63.62/57.79	64.96/63.77
GCC-3D (CenterPoint) [21]	Fine-tuning	65.29/62.79$_{(+1.83/+1.84)}$	63.97/63.47	64.23/58.47	67.68/66.44
Ours (CenterPoint)	Fine-tuning	**66.42/63.85**$_{(+2.96/+2.90)}$	64.94/64.42	66.13/60.11	68.19/67.01
CenterPoint-Stage2 [45]	Scratch	65.29/62.47	64.70/64.11	63.26/58.46	65.93/64.85
GCC-3D (CenterPoint-Stage2) [21]	Fine-tuning	67.29/64.95$_{(+2.00/+2.48)}$	66.45/65.93	66.82/61.47	68.61/67.46
Ours (CenterPoint-Stage2)	Fine-tuning	**68.06/65.69**$_{(+2.77/+3.22)}$	66.98/66.48	68.15/62.61	69.04/67.97

exceeds STRL [15] and GCC-3D [21] by 1.46% and 1.66%, respectively. Our model also surpasses DepthContrast [47] and PointContrast [38], thanks to the proposal-level representation. Besides, the classes with fewer labeled instances (*e.g.*, pedestrian and cyclist) are improved a lot, showcasing the ability to address imbalanced class distribution. More importantly, with our pre-trained model, PointRCNN and PV-RCNN using half annotation achieve comparative performance compared with the counterparts with full annotations. This also suggests the potential of our approach in reducing the heavy annotation burden.

Waymo Open Dataset. Waymo dataset [33] contains three classes: vehicles, pedestrians, and cyclists. 3D Average Precision (AP) and Average Precision with Heading (APH) are defined as the evaluation metrics for all classes. The AP and APH are based on IoU thresholds of 0.7 for vehicles and 0.5 for pedestrians and cyclists. Two difficulty levels, *i.e.*, LEVEL_1 and LEVEL_2 are defined according to the points number in the bounding boxes, where we mainly consider the LEVEL_2 metric.

We follow the schedule of OpenPCDet [34] to fine-tune the detectors on 20% training samples for 30 epochs. In particular, we first report the training-from-scratch results of SECOND [39], Part-A^2-Anchor [32], CenterPoint [45] (VoxelNet version) and PV-RCNN [30], in terms of GCC3D [21]. After that, we apply our *ProposalContrast* on two strong baselines, CenterPoint [45] and PV-RCNN [30], to verify our model. According to Table 2, with our self-supervised pre-training, the performances of popular 3D detectors are substantially improved. For PV-RCNN [30], we improve the model training from scratch by 3.05% APH, as well as outperform GCC-3D [21] by 1.10% APH on average. We further evaluate our pre-trained model on CenterPoint with VoxelNet backbone. The experimental results show that our approach improves 2.9% and 1.06% APH, compared with training from scratch and GCC-3D [21], respectively. Furthermore, based on the two-stage CenterPoint, our model reaches 65.69% APH, improving the model trained from scratch by 3.22%.

Table 3. Data-efficient 3D object detection on Waymo dataset. Our *ProposalContrast* consistently improves the performance of modern 3D object detectors, especially when only limited labeled data are available. In each row, we present the results trained from scratch on the top and show the fine-tuning results at the bottom.

Fine-tuning with various label ratios	Detector	Relative Gain	3D AP/APH (LEVEL 2)			
			Overall	Vehicle	Pedestrian	Cyclist
1% (∼ 0.8k frames)	PointPillars [19]	**+8.60/+8.26**	23.05/18.08	27.15/26.17	30.31/18.79	11.68/9.28
			31.65/26.34	35.88/35.08	37.61/25.22	21.47/18.73
	VoxelNet [39]	**+17.48/+16.95**	20.88/17.83	21.95/21.45	27.98/20.52	12.70/11.53
			38.36/34.78	37.60/36.91	39.74/31.70	37.74/35.73
10% (∼ 8k frames)	PointPillars [19]	**+2.33/+2.85**	51.75/46.58	54.94/54.32	54.01/41.53	46.31/43.88
			54.08/49.43	57.54/56.93	56.97/45.25	47.74/46.1
	VoxelNet [39]	**+4.96/+5.06**	54.04/51.24	54.37/53.74	51.45/45.05	56.30/54.93
			59.00/56.30	58.83/58.23	57.75/51.75	60.42/58.91
50% (∼ 40k frames)	PointPillars [19]	**+1.06/+1.08**	59.77/55.58	61.89/61.32	61.89/51.26	55.54/54.16
			60.83/56.66	63.01/62.44	62.57/52.02	56.91/55.53
	VoxelNet [39]	**+1.14/+1.05**	63.51/61.05	63.18/62.66	63.35/57.67	63.99/62.82
			64.65/62.10	64.07/63.54	64.64/58.71	65.24/64.04
100% (∼ 80k frames)	PointPillars [19]	**+0.54/+0.40**	61.68/57.92	63.95/63.39	62.91/53.38	58.17/57.01
			62.22/58.32	64.05/63.85	63.51/53.69	58.66/57.41
	VoxelNet [39]	**+0.55/+0.61**	64.84/62.29	64.38/63.86	66.05/60.06	64.09/62.95
			65.39/62.90	64.67/64.16	66.52/60.65	64.97/63.88

Table 4. 3D Object Detection Performance on ONCE validation set. The improved CenterPoint achieves the best performance among these SOTA detectors.

Methods	mAP	Orientation-aware AP		
		Vehicle	Pedestrian	Cyclist
PointPillars [19]	44.34	68.57	17.63	46.81
SECOND [39]	51.89	71.19	26.44	58.04
PV-RCNN [30]	53.55	77.77	23.50	59.37
CenterPoint [30]	60.05	66.79	49.90	63.45
PointPainting [35]	57.78	66.17	44.84	62.34
CenterPoint* [45]	64.24	75.26	51.65	65.79
Ours (CenterPoint*)	**66.24**$_{+2.00}$	**78.00**$_{+2.74}$	**52.56**$_{+0.91}$	**68.17**$_{+2.38}$

We also evaluate our model under a data-efficient 3D object detection setting. To be specific, we split the Waymo training set into two groups, with each group containing 399 scenes (∼80k frames). We first conduct pre-training on one group without using the labels, and then fine-tune the pre-trained model with labels on another group. During fine-tuning, various fractions of training data are uniformly sampled: 1% (0.8k frames), 10% (8k frames), 50% (40k frames) and 100% (80k frames). Two different backbones, *i.e.*, PointPillars [19] and VoxelNet [39] based on CenterPoint, are involved to measure the performance of our pre-trained model. The detection model trained from random initialization is viewed as the baseline. The advantage of our pre-trained model over the baseline is presented in Table 3. In essence, our pre-trained model can consistently promote detection performance on both backbones, especially when the labled data is scarce, *i.e.*, improving 8.26% and 16.95% APH with 1% labeled data. Our model also outperforms the baselines under all label settings.

Table 5. Ablation studies on different granularities of instance features.

Methods	Initialization	Overall (mAP/mAPH)	Vehicle	Pedestrian	Cyclist
Baseline	Random Initialization	59.63/56.96	58.96/58.40	56.72/50.77	63.20/61.70
PointContrast [38]	Point-level Pre-train	60.32/57.75$_{(+0.69/+0.79)}$	60.47/59.85	56.67/50.78	63.82/62.63
DepthContrast [47]	Scene-level Pre-train	60.71/57.96$_{(+1.08/+1.00)}$	60.11/59.62	58.42/52.27	63.59/62.00
ProposalContrast	Proposal-level Pre-train	**62.75/60.13**$_{(+3.12/+3.17)}$	61.20/60.64	60.75/54.83	66.29/64.93

Table 6. Ablation studies on each module of *ProposalContrast*.

Modules	Aspect	Param.	mAP/mAPH
Baseline	Random Init.	-	59.63/56.96
MaxPooling	N=2048, r=1.0	-	62.09/59.32
Attentive Proposal Encoder	Proposal Number (N)	1024	62.10/59.47
		2048	**62.75/60.13**
		4096	62.54/59.91
	Spherical Radius (r)	0.5	62.43/59.81
		1.0	**62.75/60.13**
		2.0	62.36/59.55

Modules	Aspect	Param.	mAP/mAPH
Baseline	Random Init.	-	59.63/56.96
IPD task	*w/o* ICS	-	62.17/59.56
ICS task	#Cluster (*w/o* IPD)	64	61.47/58.87
		128	61.77/59.16
		256	61.36/58.91
	w/o SKC	128	57.29/54.52
IPD + ICS	#Cluster	128	**62.75/60.13**

ONCE Dataset. ONCE [24] is a newly released dataset for 3D object detection in autonomous driving. It involves 5k frames for training or fine-tuning and 3k frames for validation. An orientation-aware AP is introduced to account for objects with opposite orientations. ONCE evaluates 3 categories for 3D object detection: vehicle, pedestrian and cyclist. The official ONCE benchmark provides some results from popular 3D detectors on the validation set. Since our implemented CenterPoint* achieves much better results than the official version, we use it as the baseline. We pre-train the VoxelNet backbone of CenterPoint* on Waymo with our *ProposalContrast* and fine-tune it with the ONCE training set. Then we give a performance comparison with several 3D detectors on the validation set. As shown in Table 4, our model improves the baseline by 2.00% mAP, achieving better performance on the validation set. This gives another evidence of the transferability of our pre-trained model.

4.3   Ablation Study

In this section, we examine our *ProposalContrast* model in depth. We conduct each group experiment by pre-training the VoxelNet backbone on the full Waymo training set in an unsupervised manner, and evaluate the performance by fine-tuning the detector on Waymo 20% training data. CenterPoint trained from random initialization is viewed as the baseline. 1× schedule (12 epochs) in [45] is used to save computation on the large-scale Waymo.

Comparison to Point-/Scene-level Pre-training. The main contribution of this work is to propose a proposal-wise pre-training paradigm. To show its merits over previous contrastive learning methods in point cloud, we re-implement the pioneer works like PointContrast [38] and DepthContrast [47] based on the VoxelNet backbone of CenterPoint. As seen in Table 5, *ProposalContrast* achieves

much better results when transferring to 3D object detection in large-scale Li-DAR point clouds, thanks to the more suitable representation.

Effectiveness of the Proposal Encoding Module. Next, we ablate the design choices in the proposal encoding module. We first apply a heuristic method (MaxPool) that directly pools the point features inside a proposal and uses liner layers for embedding. As shown in Table 6, this has obtained improvement over the baseline due to operating on more informative candidate proposals. After being equipped with the attentive proposal encoder, better results are achieved (+0.81 mAPH). This shows the importance of modeling the geometry structures of proposals. After that, we also search the proposal number N and proposal radius r. It turns out that $N = 2048$ and $r = 1.0$ m give better results. We infer that more proposals or a larger proposal radius will cause overlaps between neighbor proposals, which may confuse the instance discrimination process.

Effectiveness of the Joint Optimization Module. We further investigate the effect of the two learning targets in Table 6. We first consider only the inter-proposal discrimination (IPD) task for self-supervised learning, which improves the baseline by 2.60% mAPH. Then, we evaluate the pre-training model with only the inter-cluster separation (ICS) task and check the effect of class (cluster) number, *i.e.*, the output dimensions of the predictor. After that, we examine the importance of Sinkhorn-Knopp clustering (SKC). As seen, the performance drops a lot (-2.44 mAPH) without SKC. Finally, the joint learning of the two tasks further improves the performance.

5   Conclusion

This paper presented *ProposalContrast*, a proposal-wise pre-training framework for LiDAR-based 3D object detection. Despite the previous works for scene-level or point/voxel-level instance discrimination, we argue that the proposal-level representation is more suitable for 3D object detection in the large-scale LiDAR point cloud, which is not well addressed by previous works. To achieve proposal-wise contrastive learning, we carefully designed a proposal generation module, a region proposal encoding module and a joint optimization module. In particular, the proposal generation module samples dense and diverse spherical proposals with different augmented views. The proposal encoding module abstracts proposal features to model the intrinsic geometry structure of proposals by considering relationships of points inside proposals. To further build a comprehensive representation, we proposed to jointly optimize an inter-proposal discrimination task and an inter-cluster separation task. Extensive experiments on diverse prevalent 3D object detectors and datasets show the effectiveness of our model. We expect this work will encourage the community to explore the unsupervised pre-training paradigm in driving scenarios.

References

1. Bogoslavskyi, I., Stachniss, C.: Efficient online segmentation for sparse 3d laser scans. Journal of Photogrammetry, Remote Sensing and Geoinformation Science **85**(1), 41–52 (2017)
2. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: NeurIPS (2020)
3. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. In: arXiv preprint (2015)
4. Chen, K., Hong, L., Xu, H., Li, Z., Yeung, D.Y.: Multisiam: Self-supervised multi-instance siamese representation learning for autonomous driving. In: ICCV (2021)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML (2020)
6. Chen, X., He, K.: Exploring simple siamese representation learning. In: CVPR (2021)
7. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: NeurIPS (2013)
8. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR (2017)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
11. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR (2012)
12. Gidaris, S., Bursuc, A., Komodakis, N., Pérez, P., Cord, M.: Learning representations by predicting bags of visual words. In: CVPR (2020)
13. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020)
14. Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization. In: ICLR (2019)
15. Huang, S., Xie, Y., Zhu, S.C., Zhu, Y.: Spatio-temporal self-supervised representation learning for 3d point clouds. In: ICCV (2021)
16. Jing, L., Tian, Y.: Self-supervised visual feature learning with deep neural networks: A survey. TPAMI **43**(11), 4037–4058 (2020)
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
18. Kingma, D.P., Mohamed, S., Jimenez Rezende, D., Welling, M.: Semi-supervised learning with deep generative models. In: NeurIPS (2014)
19. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: CVPR (2019)
20. Li, X., Liu, S., De Mello, S., Wang, X., Kautz, J., Yang, M.H.: Joint-task self-supervised learning for temporal correspondence. In: NeurIPS (2019)
21. Liang, H., Jiang, C., Feng, D., Chen, X., Xu, H., Liang, X., Zhang, W., Li, Z., Van Gool, L.: Exploring geometry-aware contrast and clustering harmonization for self-supervised 3d object detection. In: ICCV (2021)

22. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021)
23. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. In: ICLR (2017)
24. Mao, J., Niu, M., Jiang, C., Liang, H., Chen, J., Liang, X., Li, Y., Ye, C., Zhang, W., Li, Z., et al.: One million scenes for autonomous driving: Once dataset. In: NeurIPS Datasets and Benchmarks (2021)
25. Meng, Q., Wang, W., Zhou, T., Shen, J., Jia, Y., Van Gool, L.: Towards a weakly supervised framework for 3d point cloud object detection and annotation. TPAMI (2021)
26. Meng, Q., Wang, W., Zhou, T., Shen, J., Van Gool, L., Dai, D.: Weakly supervised 3d object detection from lidar point cloud. In: ECCV (2020)
27. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. In: arXiv preprint (2018)
28. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: NeurIPS (2017)
29. Rao, Y., Liu, B., Wei, Y., Lu, J., Hsieh, C.J., Zhou, J.: Randomrooms: Unsupervised pre-training from synthetic shapes and randomized layouts for 3d object detection. In: ICCV (2021)
30. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: CVPR (2020)
31. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: CVPR (2019)
32. Shi, S., Wang, Z., Shi, J., Wang, X., Li, H.: From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. TPAMI **43**(8), 2647–2664 (2020)
33. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR (2020)
34. Team, O.D.: Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet (2020)
35. Vora, S., Lang, A.H., Helou, B., Beijbom, O.: Pointpainting: Sequential fusion for 3d object detection. In: CVPR (2020)
36. Wang, W., Zhou, T., Yu, F., Dai, J., Konukoglu, E., Van Gool, L.: Exploring cross-image pixel contrast for semantic segmentation. In: ICCV (2021)
37. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions On Graphics **38**(5), 1–12 (2019)
38. Xie, S., Gu, J., Guo, D., Qi, C.R., Guibas, L.J., Litany, O.: Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In: ECCV (2020)
39. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
40. Yang, Z., Sun, Y., Liu, S., Jia, J.: 3dssd: Point-based 3d single stage object detector. In: CVPR (2020)
41. Ye, M., Shen, J.: Probabilistic structural latent representation for unsupervised embedding. In: CVPR (2020)
42. Yin, J., Fang, J., Zhou, D., Zhang, L., Xu, C.Z., Shen, J., Wang, W.: Semi-supervised 3d object detection with proficient teachers. In: ECCV (2022)
43. Yin, J., Shen, J., Gao, X., Crandall, D., Yang, R.: Graph neural network and spatiotemporal transformer attention for 3d video object detection from point clouds. TPAMI (2021)

44. Yin, J., Shen, J., Guan, C., Zhou, D., Yang, R.: Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention. In: CVPR (2020)
45. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: CVPR (2021)
46. Zhai, X., Oliver, A., Kolesnikov, A., Beyer, L.: S4l: Self-supervised semi-supervised learning. In: CVPR (2019)
47. Zhang, Z., Girdhar, R., Joulin, A., Misra, I.: Self-supervised pretraining of 3d features on any point-cloud. In: ICCV (2021)
48. Zhao, Y., Wang, G., Luo, C., Zeng, W., Zha, Z.J.: Self-supervised visual representations learning by contrastive mask prediction. In: ICCV (2021)
49. Zhou, D., Fang, J., Song, X., Liu, L., Yin, J., Dai, Y., Li, H., Yang, R.: Joint 3d instance segmentation and object detection for autonomous driving. In: CVPR (2020)