COˆ3: Cooperative Unsupervised 3D Representation Learning for Autonomous Driving

Runjian Chen
The University of Hong Kong
rjchen@connect.hku.hk

Yao Mu
The University of Hong Kong
muyao@connect.hku.hk

Runsen Xu
Zhejiang University
runsenxu@zju.edu.cn

Wenqi Shao
The Chinese University of Hong Kong
weqish@link.cuhk.edu.hk

Chenhan Jiang
Huawei Noah's Ark Lab
jiang.chenhan@huawei.com

Hang Xu
Huawei Noah's Ark Lab
xu.hang@huawei.com

Zhenguo Li
Huawei Noah's Ark Lab
li.zhenguo@huawei.com

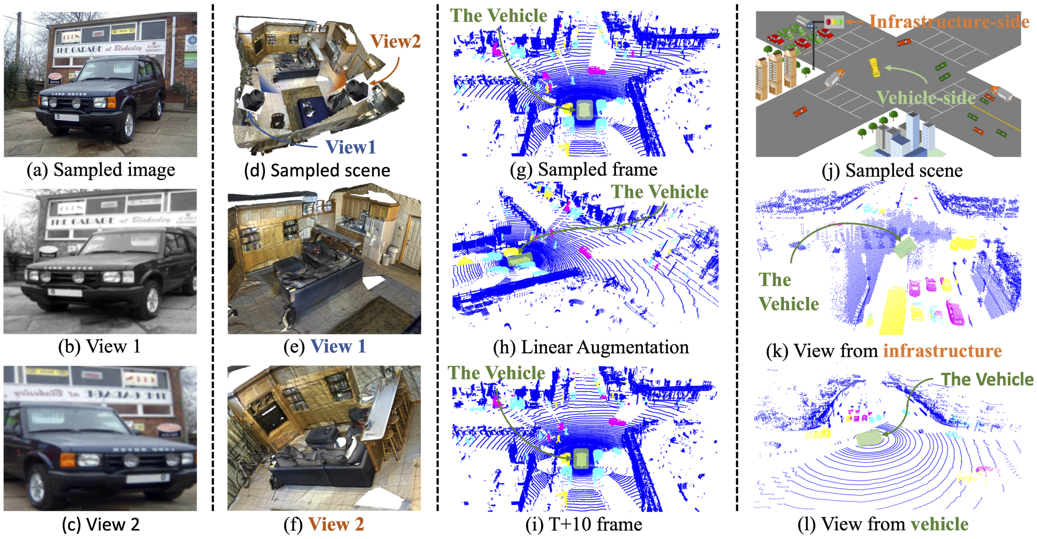
Ping Luo
The University of Hong Kong
pluo@cs.hku.hk

Abstract: Unsupervised contrastive learning for indoor-scene point clouds has achieved great successes. However, unsupervised learning point clouds in outdoor scenes remains challenging because previous methods need to reconstruct the whole scene and capture partial views for the contrastive objective. This is infeasible in outdoor scenes with moving objects, obstacles, and sensors. In this paper, we propose **COˆ3**, namely **Co**operative **Co**ntrastive Learning and **Co**ntextual Shape Prediction, to learn 3D representation for outdoor-scene point clouds in an unsupervised manner. **COˆ3** has several merits compared to existing methods. (1) It utilizes LiDAR point clouds from vehicle-side and infrastructure-side to build views that differ enough but meanwhile maintain common semantic information for contrastive learning, which are more appropriate than views built by previous methods. (2) Alongside the contrastive objective, shape context prediction is proposed as pre-training goal and brings more task-relevant information for unsupervised 3D point cloud representation learning, which are beneficial when transferring the learned representation to downstream detection tasks. (3) As compared to previous methods, representation learned by **COˆ3** is able to be transferred to different outdoor scene dataset collected by different type of LiDAR sensors. (4) **COˆ3** improves current state-of-the-art methods on both *Once* and *KITTI* datasets by up to 2.58 mAP. Codes and models will be released here. We believe **COˆ3** will facilitate understanding LiDAR point clouds in outdoor scene.

Keywords: 3D Representation Learning, Autonomous Driving, Contrastive Learning, Shape Context

1 Introduction

As the most reliable sensor in outdoor environments, LiDAR is able to precisely measure 3D location of objects and the computer vision community has shown strong interest on perception tasks on LiDAR point clouds, including 3D object detection, segmentation and tracking. Up to now, randomly initializing and directly *training from scratch* on detailed annotated data still dominates this field. Embraced by MOCO [1], recent research efforts [1, 2, 3, 4, 5] focus on unsupervised representation learning with contrastive objective on different views built from images (the first column of Figure 1 shows example views built by [1, 2, 3, 4, 5]). They pre-train the 2D backbone in an unsupervised manner and achieve significant performance improvement over *training from scratch* in various



(a) Sampled image (d) Sampled scene (g) Sampled frame (j) Sampled scene

(b) View 1 (e) View 1 (h) Linear Augmentation (k) View from infrastructure

(c) View 2 (f) View 2 (i) T+10 frame (l) View from vehicle

Figure 1: Example views built by different methods in contrastive learning. (a), (b) and (c) are sampled image and views (different augmentations of the original image) used in [1, 2, 3, 4, 5]. (d), (e) and (f) show an example of two views built in PointContrast [14], which are captured at different poses and differ much but meanwhile still maintain enough common semantic information including the same sofa and table. (g) and (h) are views for outdoor-scene point cloud from [17]. (g) is the original frame of point cloud and authors in [17] apply point cloud augmentation to (g) for (h), which can be implemented with a simple linear transformation. [18] use point cloud at different timestamps as views for contrastive learning, which is indicated in (g) and (i). During the sampled period, the autonomous vehicle is waiting at the crossing and some other cars and pedestrians are moving around. The autonomous vehicle has no access to how the environment changes and in different timestamps, there are possibly different objects at the same position in its own coordinate, which makes it hard to find accurate correspondence.

downstream tasks such as 2D object detection [6, 7, 8] and image segmentation [6, 9]. Inspired by these successes together with the abundant unlabelled data available from self-driving vehicles, we explore unsupervised representation learning for *outdoor scene* point clouds to improve the performance on 3D object detection tasks.

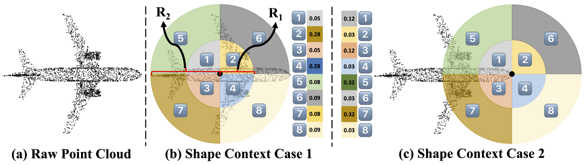
In the past decade, learning 3D representation from unlabelled data has achieved great success in *single-object* and *indoor-scene* point clouds. For point clouds of *single objects* such as CAD models, previous works pre-train 3D encoders with various goals including reconstruction [10, 11], orientation estimation [12] and minimizing contrastive loss [13] and the 3D encoders extract meaningful **global** representations of the point clouds for low-level downstream tasks like object classification and registration. To extend this idea to high-level perception tasks for *indoor-scene* point clouds, PointContrast [14] propose to reconstruct the point clouds of the whole indoor scenes, collect partial point clouds from two different poses and utilize them as two views in contrastive learning to learn **dense** (point-level or voxel-level) representation. More recent works such as [15] and [16] also need to reconstruct the whole indoor scenes and this naturally brings the assumption that the environment should be static. Figure 1 (d), (e) and (f) show an example of two views built in PointContrast [14]. We can see that these two views differ a lot because they are captured in different directions but meanwhile, they still contain enough common semantic information such as the same sofa and table.

However, outdoor scenes are dynamic and large-scale, making it impossible to reconstruct the whole scenes for building views. Thus, methods in [14, 15, 16] cannot be directly transferred but there exists two possible alternatives to build views. The first idea, embraced by [17], is to apply data augmentation to single frame of point cloud and treat the original and augmented versions as different views, as indicated by Figure 1 (g) and (h). However, all the augmentation of point clouds, including random drop, jittering, rotation and scaling, can be implemented in a linear transformation and views constructed in this way do not differ enough. The second one is to consider point clouds at different timestamps as different views, represented by [18]. Yet the moving objects and obstacles would make it hard to find correct correspondence for contrastive learning. See Figure 1 (g) and (f), while the autonomous vehicle is waiting for the traffic light to turn green, other cars and pedestrians are moving and the autonomous vehicle has no idea about how they move, making it impossible to find correct correspondence. Due to these limitations, pre-trained 3D encoders in [17, 18] cannot achieve noticeable improvement when transferring to datasets collected by different LiDAR sensors or large-scale labelled datasets with the same kind of LiDAR sensors.

To overcome these limitations, we propose **CO**operative **CO**ntrastive Learning and **CO**ntextual Shape Prediction, **COˆ3**, to learn representation for outdoor-scene point clouds in an unsupervised manner. **COˆ3** mainly contains two components, as described below.

Cooperative Views for Contrastive Learning. To build views of LiDAR point clouds that differ enough and share adequate semantic information, we propose to utilize a recently released infrastructure-vehicle-cooperation dataset called DAIR-V2X [19] and build views for contrastive representation learning using point clouds respectively from infrastructure LiDAR and vehicle LiDAR. As shown in (j), (k) and (l) in Figure 1, views built in this way differ a lot because they are captured at different positions and they share enough information because they are captured at the same timestamp. With the raw input point clouds from the vehicle and infrastructure, we further fuse the point cloud from both sides at the same timestamp and use the fusion point cloud and point cloud from vehicle-side as two views in contrastive representation learning.

Contextual Shape Prediction. As proposed in [20], representation learned from purely contrastive learning is not able to capture task-relevant information and a reconstruction objective can be implemented alongside to compensate this limitation. Detailed experiments on image representation learning have been conducted in [20] to demonstrate this statement and we want to borrow this idea to 3D point cloud. However, it can be extremely difficult to reconstruct the whole scene with point-level or voxel-level representations. Instead, we propose a pre-training goal to reconstruct local distribution of neighboring points using the dense representations. In practice, we use shape context to describe the local distribution of each point's neighborhood, which has been demonstrated as a useful local distribution descriptor in previous works [15, 21, 22, 23]. Figure 2 shows two examples of shape context with 8 bins. The neighborhood of the query point (marked as a larger black point) is first divided into 8 bins and we compute an 8-dimensional distribution with the numbers of points in these bins. The pre-training task is to predict local distributions of each point or voxel with the extracted point-level or voxel-level representation. Note that the number of bins can be changed as needed. This refined reconstruction pre-training task introduces more task-relevant information and helps learn much better representations.



(a) Raw Point Cloud (b) Shape Context Case 1 (c) Shape Context Case 2

Figure 2: Two examples of shape context.

The contributions of this work can be summarized as follows. (1) **COˆ3** is proposed to utilize the recently proposed vehicle-infrastructure cooperative dataset to build adequate views for unsupervised contrastive learning and learn good 3D representations for *outdoor-scene* point clouds. (2) A shape-context prediction task is proposed alongside the contrastive objective and inject more task-relevant information in the learned representation, which is beneficial for downstream tasks. (3) The learned 3D representations can be well transferred to datasets collected by different LiDAR sensors on 3D object detection tasks. (4) Extensive experiments demonstrate the effectiveness of **COˆ3**. For example, **COˆ3** improves Second, PV-RCNN, CenterPoints on *Once* [19] by 1.07, 0.62 and 2.58 respectively.

2 Related Works

3D Object Detection. Figure 3 summarizes current 3D object detection methods in autonomous driving scenes. The raw LiDAR point clouds is first passed through a 3D encoder and transferred

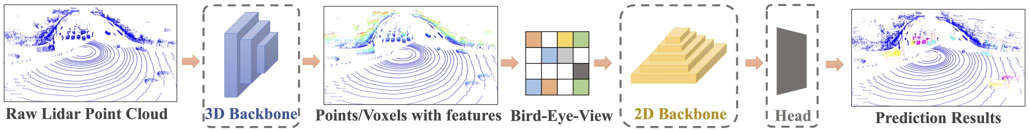


Figure 3: Summary of current 3D object detectors. The raw input LiDAR point cloud is first processed by the 3D encoder and per-point or per-voxel representation is generated. After that, these dense representation is mapped onto the ground plane and transformed into the Bird-Eye-View (BEV) map. Finally, a 2D backbone is used to encode the BEV map and afterwards a detection head is stacked to generate the final detection results.

into per-point or per-voxel representation. Then these dense representation is projected onto the ground plane and we get Bird-Eye-View (BEV) map. After that, the BEV map is encoded by a 2D backbone followed with a detection head and the final detection results are generated. Up to now, 3D object detectors can be divided into three main streams due to different 3D encoders they used: (1) point-based methods [24, 25, 26] produce per-point representation. (2) voxel-based methods [27, 28, 29, 30, 31, 32] generally transform point cloud into voxel grids and process them using 3D volumetric convolutions. (3) point-voxel-combined methods [33, 34, 35] utilize features from both (1) and (2). Among all these methods, [27, 31, 34] are the most widely used detectors and achieve state-of-the-art performance, all of which have a 3D voxel encoder. Also, all these methods rely on sufficient training labels and precise 3D annotations. Thus in this paper, we propose **COˆ3** to pre-train voxel encoders without labels for outdoor-scene LiDAR point clouds and the representation learned in this unsupervised manner can be transferred to different downstream datasets collected by different LiDAR sensors.

3D Unsupervised Representation Learning for LiDAR Point Clouds. On the contrary to 2D unsupervised pre-training and 3D unsupervised representation learning for *object-based* or *indoor-scene* point clouds, which have demonstrated promising performance on downstream tasks [1, 2, 3, 4, 5, 14, 15, 16, 12, 10, 11, 13], training from scratch still dominates 3D vision field on *outdoor scene* point clouds. PointContrast [14] is the pioneering work for unsupervised contrastive learning on *indoor-scene* point clouds and shows performance gain on diverse indoor scene understanding tasks. Together with the following works including [15, 16], they rely on the assumption of static scenes that have been registered for constructing adequate views. To extend their ideas to outdoor-scene LiDAR point clouds, [17] proposes to augment single frame of point cloud for building views in contrastive learning and [18] utilizes point clouds at different timestamps as different views for unsupervised representation learning. However, all the augmentation of point cloud can be implemented in a single linear transformation and views built in this manner do not differ enough. Meanwhile, [17] needs to pre-train both the 3D backbone and 2D backbone of the detectors but there exists different 2D backbones in current 3D object detection methods, making this methods less scalable across different detectors. For the views built in [18], it is very difficult to find correct correspondence because the outdoor scene is dynamic. These limitations make representations learned in [17, 18] unable to transfer to large-scale datasets or different datasets collected by various LiDAR sensors. In this work, we propose to use vehicle-side and fusion (vehicle and infrastructure) point clouds as two views in contrastive learning. Besides the contrastive objective, a shape-context prediction pre-training goal is proposed to learn better 3D representation.

3 Methods

In this section, we introduce the proposed **COˆ3** for unsupervised representation learning on LiDAR point clouds in outdoor scenes. As detailed in Figure 4, **COˆ3**has two pre-training objectives: (a) a cooperative contrastive learning goal on dense (point-level or voxel-level) representations between vehicle-side and fusion point clouds, which provides adequate views for contrastive learning. (b) a contextual shape prediction loss to bring in more task-relevant information. To start with, we discuss the problem formulation and overall pipeline of **COˆ3** in Section 3.1. Then we respectively introduce the cooperative contrastive objective and contextual shape prediction goal in Section 3.2 and Section 3.3. Finally in Section 3.4, we provide detailed implementation of **COˆ3**.

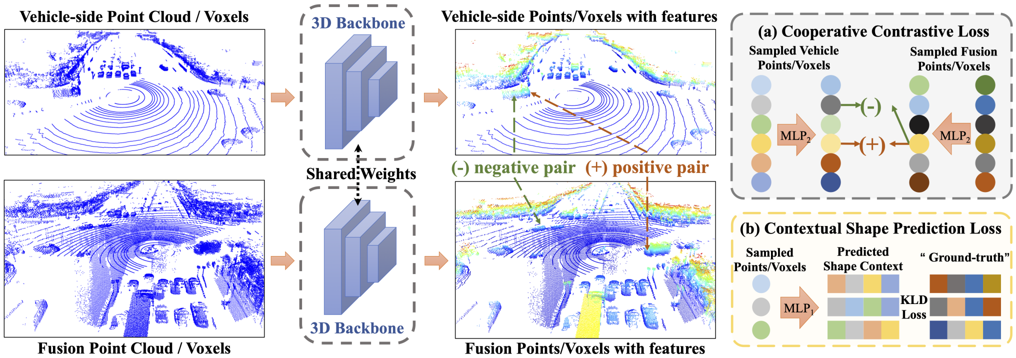

Vehicle-side Point Cloud / Voxels
3D Backbone
Shared Weights
3D Backbone
Fusion Point Cloud / Voxels
Vehicle-side Points/Voxels with features
(-) negative pair
(+) positive pair
Fusion Points/Voxels with features
(a) Cooperative Contrastive Loss
Sampled Vehicle Points/Voxels
Sampled Fusion Points/Voxels
MLP$_2$
(-)
(+)
MLP$_2$
(b) Contextual Shape Prediction Loss
Sampled Points/Voxels
Predicted Shape Context
" Ground-truth"
MLP$_1$
KLD Loss

Figure 4: The pipeline of **CO^3**. With vehicle-side and infrastructure-side LiDAR point clouds as inputs, we first transform the infrastructure-side LiDAR point cloud into vehicle-side coordinate and fuse them to form the fusion point cloud. Then vehicle-side and fusion point clouds are processed by the 3D backbone to generate point/voxel-level representations. With these dense representations, we propose two pre-training objectives: (a) Cooperative Contrastive Loss, which introduces adequate views of outdoor scene LiDAR point clouds for contrastive learning. (b) Contextual Shape Prediction Loss, which brings in task-relevant information and makes the learned representation better for downstream tasks.

3.1 Problem Formulation and Pipeline

To begin with, we define the raw LiDAR point clouds from vehicle-side and infrastructure-side respectively as $\mathbf{P}_{\text{veh}} = [\mathbf{P}^{\text{xyz}}_{\text{veh}}, \mathbf{P}^{\text{feat}}_{\text{veh}}]$ and $\mathbf{P}_{\text{inf}} = [\mathbf{P}^{\text{xyz}}_{\text{inf}}, \mathbf{P}^{\text{feat}}_{\text{inf}}]$, where $\mathbf{P}^{\text{xyz}}_{\text{v/i}} \in \mathbb{R}^{N^{\text{p}}_{\text{v/i}} \times 3}$ and $\mathbf{P}^{\text{feat}}_{\text{v/i}} \in \mathbb{R}^{N^{\text{p}}_{\text{v/i}} \times d}$. Here $N^{\text{p}}_{\text{v/i}}$ denotes the number of points (or voxels) in vehicle-side and infrastructure-side respectively and $d = 1$ is always the case to represent the intensity of each point (or voxel). **Note** here that as vehicle/infrastructure/fusion point clouds may sometimes go through the same process, we will change the notation to v/i/f to indicate the same processing on respective point cloud for convenience. When collecting the cooperative dataset, each pair of vehicle-side and infrastructure-side point clouds is associated with a transformation $T^{\text{veh}}_{\text{inf}}$ indicating the relationship between vehicle-side coordinate and infrastructure-side coordinate.

With $\mathbf{P}_{\text{veh}}$, $\mathbf{P}_{\text{inf}}$ and $T^{\text{veh}}_{\text{inf}}$ as inputs, **CO^3** first transforms the infrastructure point cloud into vehicle-side coordinate, that is $\mathbf{P}'_{\text{inf}} = [T^{\text{veh}}_{\text{inf}}(\mathbf{P}^{\text{xyz}}_{\text{inf}}), \mathbf{P}^{\text{feat}}_{\text{inf}}]$, and concatenates the transformed infrastructure point cloud and the vehicle point cloud into fusion point cloud $\mathbf{P}_{\text{fusion}} = [\mathbf{P}_{\text{veh}}, \mathbf{P}'_{\text{inf}}]$, where $\mathbf{P}_{\text{fusion}} \in \mathbb{R}^{(N^{\text{p}}_{\text{veh}} + N^{\text{p}}_{\text{inf}}) \times (3+d)}$. Then $\mathbf{P}_{\text{veh}}$ and $\mathbf{P}_{\text{fusion}}$ are embedded by the 3D encoder f^{enc}

$$\mathbf{P}^{\text{enc}}_{\text{v/f}} = f^{\text{enc}}(\mathbf{P}_{\text{v/f}}) \tag{1}$$

where $\mathbf{P}^{\text{enc}}_{\text{v/f}} = [\mathbf{P}^{\text{xyz}^{\text{enc}}}_{\text{v/f}}, \mathbf{P}^{\text{feat}^{\text{enc}}}_{\text{v/f}}]$ and $\mathbf{P}^{\text{xyz}^{\text{enc}}}_{\text{v/f}} \in \mathbb{R}^{N^{\text{p}^{\text{enc}}}_{\text{v/f}} \times 3}$, $\mathbf{P}^{\text{feat}^{\text{enc}}}_{\text{v/f}} \in \mathbb{R}^{N^{\text{p}^{\text{enc}}}_{\text{v/f}} \times d^{\text{enc}}}$. $N^{\text{p}^{\text{enc}}}_{\text{v/f}}$ is the number of points (or voxels) after encoding. As there exists pooling operations in 3D encoders, the number of points (or voxels) may change when processed by the 3D encoders. d^{enc} is the number of feature channels after encoding. To guide the 3D encoder to learn good representations in an unsupervised manner, we propose a cooperative contrastive loss L_{CO_2} and a contextual shape prediction loss L_{CSP} for optimization. The overall loss function can be written as:

$$L = \sum_{\mathbf{P}_{\text{v/f}} \in \{\mathcal{P}_{\text{v/f}}\}} L_{\text{CO}_2}\{f^{\text{enc}}(\mathbf{P}_{\text{v/f}})\} + w_{\text{CSP}} \times L_{\text{CSP}}\{f^{\text{enc}}(\mathbf{P}_{\text{v/f}}), \mathbf{P}^{\text{xyz}}_{\text{veh}}, \mathbf{P}^{\text{xyz}}_{\text{fusion}}\} \tag{2}$$

where $\mathcal{P}_{\text{v/f}}$ denote a batch of vehicle and fusion point clouds. As described in this equation, L_{CO_2} takes as inputs the encoded vehicle and fusion point clouds and applies contrastive learning on the features of these two views. Meanwhile, L_{CSP} introduces more task-relevant information into f^{enc} by using the encoded features to predict contextual shape whose ground truth is obtained by $\mathbf{P}^{\text{xyz}}_{\text{veh}}$ and $\mathbf{P}^{\text{xyz}}_{\text{fusion}}$. w_{CSP} is a weighting constant that makes the magnitudes of the two loss similar. Details about L_{CO_2} and L_{CSP} will be discussed respectively in Section 3.2 and Section 3.3.

3.2 Cooperative Contrastive Objective

Unsupervised contrastive learning has been demonstrated successful in image domain [1, 2, 3, 4, 5] and indoor-scene point clouds [14, 15, 16]. However, when it turns to outdoor-scene LiDAR point clouds, building adequate views, which share common semantics while differing enough, for contrastive learning. To tackle this issue, we utilize a recently released vehicle-infrastructure-cooperation dataset called DAIR-V2X [19] and use vehicle-side point clouds and fusion point clouds as views for contrastive representation learning. The loss is defined as follows:

$$L_{\text{CO}_2} = \frac{1}{N_1}\sum_{n=1}^{N_1} -\log\left(\frac{\exp(z^n_{\text{veh}} \cdot z^n_{\text{inf}}/\tau)}{\sum_{i=1}^{N_1}\exp(z^i_{\text{veh}} \cdot z^i_{\text{inf}}/\tau)}\right) \quad \text{with}$$
$$\{z^n_{\text{v/f}}\}_{n=1}^{N_1} \overset{\text{sample}}{\sim} Z_{\text{v/f}} \quad ; \quad Z_{\text{v/f}} = \text{normalize}(\text{MLP}_1(\mathbf{P}^{\text{feat}^{\text{enc}}}_{\text{v/f}})) \tag{3}$$

where the embedded features of vehicle and infrastructure point clouds, $\mathbf{P}^{\text{feat}^{\text{enc}}}_{\text{veh}}$ and $\mathbf{P}^{\text{feat}^{\text{enc}}}_{\text{inf}}$, are first projected into a common feature space by a Multi-Layer-Perceptron MLP_1 and then normalized. $Z_{\text{v/f}} \in \mathbb{R}^{N^{\text{p}^{\text{enc}}}_{\text{v/f}} \times d_1}$ is the projected features of vehicle point cloud and infrastructure point cloud, where d_1 indicates the dimension of the common feature space and $N^{\text{p}^{\text{enc}}}_{\text{v/f}}$ are the point/voxel numbers of encoded vehicle and infrastructure point clouds respectively. We then sample N_1 pairs of features from $Z_{\text{v/f}}$ for contrastive learning. According to our empirical observation, ground points have a great negative effect on contrastive learning. Thus we mark those points with z value lower than a threshold z_{thd} as ground points and filter them out when sampling. After filtering, we randomly sample N_1 points from the vehicle point cloud and find their corresponding points (or voxels) in the fusion point cloud to form N_1 pairs of points (or voxels). We treat corresponding points (or voxels) as positive pairs and otherwise negative pairs for contrastive learning and the final loss function is shown in the first line of Equation (3), where τ is the temperature parameter.

3.3 Contextual Shape Prediction

CO^3 aims to learn representations applicable to various downstream tasks. But it cannot be guaranteed that task-relevant information is extracted by contrastive loss in Eqn.(3) [20]. Instead, [20] shows that an additional reconstruction objective alongside contrastive loss can bring more task-relevant information. However, it is extremely difficult to reconstruct the whole scene with point/voxel-level representations on outdoor-scene LiDAR point clouds. To mitigate this issue, we propose to reconstruct the neighborhood of each point/voxel with its representation. To this end, a contextual shape prediction loss is designed as written by

$$L_{\text{CSP}} = \frac{1}{N_2}\sum_{n=1}^{N_2}\sum_{m=1}^{N_{\text{bin}}} p_{n,m}\log\frac{p_{n,m}}{q_{n,m}} \quad \text{with}$$
$$\{p_{n,*}\}_{n=1}^{N_2} \overset{\text{sample}}{\sim} P \quad ; \quad \{q_{n,*}\}_{n=1}^{N_2} \overset{\text{sample}}{\sim} Q \quad ; \quad P = \text{softmax}(\text{MLP}_2(\mathbf{P}^{\text{feat}^{\text{enc}}}_{\text{veh}})) \tag{4}$$

where the encoded features of vehicle point clouds, $\mathbf{P}^{\text{feat}^{\text{enc}}}_{\text{veh}}$, are first passed through another Multi-Layer-Perceptron MLP_2 and softmax operation is applied on the projected features to get a predicted local distribution of each point/voxel, that is $P \in \mathbb{R}^{N^{\text{p}^{\text{enc}}}_{\text{veh}} \times N_{\text{bin}}}$. $N^{\text{p}^{\text{enc}}}_{\text{veh}}$ is the number of vehicle-side points/voxels after embedded by the 3D encoder and N_{bin} is the number of bin we divide the local neighborhood of each point/voxel. We use $N_{\text{bin}} = 32$ in this paper and compute the 'ground truth' local shape context $Q \in \mathbb{R}^{N^{\text{p}}_{\text{veh}} \times N_{\text{bin}}}$ ($N^{\text{p}}_{\text{veh}}$ is the number of raw input vehicle points/voxels) beforehand, which will be discussed later. With P and Q, N_2 sampled points/voxels are drawn from P and Q. We have $p_{n,*} \in \mathbb{R}^{N_{\text{bin}}}$ and $q_{n,*} \in \mathbb{R}^{N_{\text{bin}}}$. Note that these sampled predicted contextual shape distributions are in pairs. Finally, as shown in the first line in Equation (4), L_{CSP} is a KL-divergence loss applied on $p_{n,*}$ and $q_{n,*}$, where the KL-divergence describes the distance between two probability distribution ($p_{n,*}$ and $q_{n,*}$).

To compute the "ground truth" local shape context Q for the i^{th} point/voxel, we first divide the neighborhood of the point/voxel into $N_{\text{bin}} = 32$ bins along x-y plane with $R_1 = 0.5m$ and $R_2 = 4m$. Then we compute the number of points/voxels in each bin and this results in $N_{\text{bin}} = 32$ numbers $Q^{\text{raw}}_{i,*} \in \mathbb{R}^{N_{\text{bin}}}$. After that $Q_{i,*}$ is finalized as below:

$$Q_{i,*} = \text{softmax}(\text{normalize}(Q^{\text{raw}}_{i,*}) \times SF_{\text{CSP}}) \tag{5}$$

where SF_{CSP} is a scaling factor to make the final distribution more sensible.

3.4 Detailed Implementation of COˆ3

3D Encoder. We use Sparse-Convolution as the 3D encoder which is a 3D convolutional network because it is widely used as 3D encoders in current state-of-the-art methods [27, 31, 33]. Thus it can be used to evaluate **COˆ3** in as many 3D detectors as possible.

4 Experiments

4.1 Experiment Setup

Datasets. We utilize the recently released vehicle-infrastructure-cooperation dataset called DAIR-V2X [19] to pre-train 3D sparse encoder with **COˆ3** and fine-tune the pre-trained encoder on two downstream datasets: Once [36] and KITTI [37]. *DAIR-V2X* [19] contains 38845 LiDAR frames (10084 in vehicle-side and 22325 in infrastructure-side) for cooperative-detection task and we utilize their dataset for pre-training 3D encoder in an unsupervised manner via the proposed **COˆ3**. The total number of provided frames in cooperative dataset is around 6670 and they are collected by 120-beam LiDAR. *Once* [36] is a large-scale autonomous dataset for evaluating self-supervised methods with 1 Million LiDAR frames and only 15k fully annotated scenes with 3 classes (Vehicle, Pedestrian, Cyclist). A 40-beam LiDAR is used in [36] to collect the point cloud data. We adopt common practice, including point cloud range and voxel size, in their public code repository[1] to evaluate the proposed **COˆ3**. *KITTI* [37] is another widely used self-driving dataset, where point clouds are collected by LiDAR with 64 beams. It contains around 15k samples for training and evaluation. For point cloud range and voxel size, we adopt common practice in current popular codebase like MMDet3D[2] and OpenPCDet[3]. **Note** that LiDAR sensors used in *Once* and *KITTI* are **different** than that used in *DAIR-V2X* [19].

Detectors. We select several current state-of-the-art methods implemented in the public repository of Once dataset [36] and OpenPCDet to evaluate the quality of representations learned by **COˆ3**, including Second [27], CenterPoint [31] and PV-RCNN [34]. Note that only CenterPoint in Once [36] has a slightly different 3D backbone and we make it the same as those of other methods. As the GPUs and PyTorch [38] versions used in the public code repositories are different from those in our experiments, we further tune the configurations in the repositories and make the performance of *training from scratch* match or even surpass their released results for fair comparisons.

Baselines. As introduced in Section 1, there exists two ways to directly transfer the idea in PointContrast [14] to outdoor-scene LiDAR point clouds, embraced respectively by GCC-3D [17] and STRL [18]. Thus we conduct experiments on these two methods. Also, [36] proposes several self-supervised learning methods following hints from previous works in image domain and indoor-scene point clouds, including Swav [3], Deep Cluster (short as D. Cl.) [39], BYOL [4] and Point Contrast (short as P.C.) [14]. As the same to **COˆ3**, we pre-train all these baseline unsupervised 3D representation learning methods on DAIR-V2X [19] and fine-tune the pre-trained encoders on Once [36] and KITTI [37] for comparisons. For STRL, we follow their public code repository for indoor scene point cloud and reproduce for outdoor scene point cloud, which is not published yet[4]. As for GCC-3D, the authors do not public their code due to privacy restriction so we write emails to them and they kindly pre-train the 3D encoders on DAIR-V2X dataset and give us the pre-trained models. **Note** that all methods compared in this section are pre-trained only once on DAIR-V2X [19] and then fine-tuned on different downstream detectors and datasets.

Evaluation Metrics. We use common evaluation metrics for both the two downstream datasets. For *Once* dataset, IoU thresholds 0.7, 0.3, 0.5 are respectively adopted for vehicle, pedestrian, cyclist. Then 50 score thresholds with the recall rates ranging from 0.02 to 1.00 (step size if 0.02) are computed and the 50 corresponding values are used to draw a PR curve, resulting in the final mAPs

[1] `https://github.com/PointsCoder/Once_Benchmark`

[2] `https://github.com/open-mmlab/mmdetection3d`

[3] `https://github.com/open-mmlab/OpenPCDet`

[4] `https://github.com/yichen928/STRL`

Init.	Det.	Vehicle			Pedestrian			Cyclist			mAP
		0-30m	30-50m	50m-	0-30m	30-50m	50m-	0-30m	30-50m	50m-	
Rand	Sec.	**84.35**	66.41	49.49	27.87	**23.24**	16.36	**69.92**	52.27	**35.25**	52.21
Swav		83.21	65.25	**50.32**	**31.55**	**26.18**	**17.45**	69.40	**53.60**	**35.91**	**53.03**$^{+0.82}$
D. Cl.		84.02	**67.51**	**50.26**	**29.21**	21.55	**17.39**	**69.86**	51.95	34.69	**52.30**$^{+0.09}$
BYOL		81.60	60.93	46.97	18.71	16.59	12.95	61.20	43.15	27.30	45.24
P.C.		84.19	62.66	46.32	21.55	17.70	14.05	64.98	47.25	28.81	47.64
GCC-3D		**85.43**	**67.88**	**51.64**	27.18	21.55	16.86	**72.15**	**52.28**	**35.27**	52.28
STRL		83.71	65.59	50.39	27.41	22.23	17.17	68.28	51.96	34.17	51.57
Ours		**84.62**	**67.11**	49.42	**33.64**	**28.00**	**17.61**	68.22	**52.89**	32.92	**53.28**$^{+1.07}$
Rand	PV	**88.01**	**72.15**	**58.93**	29.67	23.24	16.47	71.46	**54.61**	**36.60**	54.55
Swav		87.80	**71.81**	57.42	**29.86**	24.88	17.15	**72.56**	**54.25**	36.44	**54.89**$^{+0.34}$
D. Cl.		87.68	71.77	57.11	**32.20**	**26.00**	**18.28**	71.64	53.09	34.80	**54.91**$^{+0.36}$
BYOL		87.37	69.63	55.55	19.74	18.82	14.64	67.01	47.11	31.11	49.41
P.C.		87.72	70.42	55.43	20.52	18.93	16.76	68.58	49.55	33.59	50.49
GCC-3D		87.71	**72.20**	**59.42**	27.91	**25.96**	16.40	**72.59**	53.88	**37.58**	54.55
STRL		**89.39**	70.32	57.40	27.67	23.41	**17.55**	**72.05**	**54.21**	**36.85**	54.25
Ours		**87.85**	71.79	**57.46**	**32.75**	**26.57**	**17.29**	71.22	52.50	36.20	**55.17**$^{+0.62}$
Rand	Cen.	77.42	**54.68**	38.21	51.08	41.13	25.79	70.67	54.68	35.14	55.92
Swav		77.58	54.28	38.51	53.95	**42.52**	**28.04**	71.10	54.99	**37.93**	57.00
D. Cl.		77.35	55.12	38.91	**54.99**	**42.26**	**29.31**	**71.80**	**56.60**	37.05	**57.65**$^{+1.08}$
BYOL		76.56	53.61	37.79	46.48	31.73	18.72	67.55	49.65	27.67	52.17
P.C.		**77.64**	53.38	39.15	49.69	35.57	23.29	69.37	50.65	30.03	54.17
GCC-3D		**77.80**	**56.75**	**39.16**	**54.46**	40.11	25.61	**74.43**	**57.04**	**39.51**	**58.32**$^{+2.40}$
STRL		78.01	54.10	**39.32**	54.09	40.77	25.90	71.60	**56.56**	36.57	57.44
Ours		**78.02**	**56.13**	**39.94**	**55.09**	**42.34**	**27.44**	**74.17**	56.05	**38.16**	**58.50**$^{+2.58}$

Table 1: Results of 3D object detection on Once dataset [36]. We conduct experiments on 3 different detectors: Second [27] (short as Sec.), PV-RCNN [34] (short as PV) and CenterPoint [31] (short as Cen.) and 8 different initialization methods including random (short as Rand, i.e. training from scratch), Swav [3], Deep Cluster (short as D. Cl.) [39], BYOL [4], Point Contrast (short as P.C.) [14], GCC-3D [17] and STRL [18]. Results are mAPs in %. "0-30m", "30-50m" and "50m-" respectively indicate results for objects in 0 to 30 meters, 30 to 50 meters and 50 meters to infinity. The "mAP" in the final column is the overall evaluation and major metric for comparisons. We use bold font for top 3 mAP in each category in each range for better understanding.

(mean accurate precisions) for each category. We also further overage over the three categories and compute an 'Overall' mAP for evaluations. For *KITTI* dataset, all the results are evaluated by mAPs with three difficulty levels: Easy, Moderate and Hard. These three results are further average and an 'Overall' mAP is generated for comparisons.

4.2 Main Results

Once Detection. As shown in Table 1, when initialized by **CO^3** , all the three detectors achieve the best performance on the overall mAPs, which we value the most. When CenterPoint [31] is used as backbone, **CO^3** achieves the best performance among all detectors and all initialization methods with 2.58 improvement on mAPs. Meanwhile, the improvement on PV-RCNN [34] is only 0.62 in mAP. This is because PV-RCNN [34] has two 3D backbones, the point-based branch and voxel-based branch, and **CO^3** only pre-trains the voxel-based branch. Thus same phenomenon can be observed in other pre-training methods. When we look into detailed categories, it can be found that **CO^3** achieve consistent improvement on Pedestrian class and the highest mAP when CenterPoint [31] is used as the detector, which is important for the deployment of autonomous driving system in real world. For Cyclist class, CenterPoint [31] initialized by **CO^3** achieves the best performance among all the detectors and all the initialization methods. However, as we can see in the Vehicle class, the improvement achieved by **CO^3** is not that significant (same phenomenons in other initialization methods) and the reason might be that the performances of all the detectors on Vehicle class are already very high and there is little room for improvement.

KITTI Detection. As shown in Table 2, when initialized by **CO^3** , PV-RCNN [34] achieves the best performance on Easy and Hard (+1.19) level and third place on Moderate level among all the initialization schemes. Meanwhile Second [27] equipped with **CO^3** achieves the highest mAP on Easy (+1.11) and Moderate level (+1.22) and third place on Hard level among all the initialization schemes. The relatively lower improvements on KITTI dataset [37] stem from the smaller number of training samples (nearly half of that in Once dataset [36]) and this makes the detectors easily reach their capacity, where improvement is hard to achieve. This is also demonstrated by the consistent

Init.	Det.	Vehicle	Pedestrian	Cyclist	Overall		
					Easy	Moderate	Hard
Random	Sec.	77.45	48.71	63.32	73.29	63.16	60.34
Swav		**77.64**	**49.48**	64.95	73.23	**64.02**$^{+0.86}$	**60.93**$^{+0.59}$
D. Cl.		77.47	**49.46**	63.19	73.19	63.37	60.08
BYOL		76.89	43.29	60.99	71.05	60.39	56.98
P.C.		77.45	45.32	**65.44**	72.67	62.74	59.21
GCC-3D		**77.99**	47.92	64.45	**73.86**$^{+0.57}$	63.45	59.80
STRL		77.63	48.46	**65.52**	**73.95**$^{+0.66}$	**63.87**$^{+0.71}$	**60.93**$^{+0.59}$
Ours		**77.95**	**49.59**	**65.60**	**74.40**$^{+1.11}$	**64.38**$^{+1.22}$	**60.88**$^{+0.54}$
Random	PV	79.13	**53.43**	69.12	**78.54**	67.23	63.68
Swav		**79.35**	**52.92**	71.45	**78.43**$^{-0.11}$	**67.91**$^{+0.68}$	**64.60**$^{+0.92}$
D. Cl.		**79.22**	50.75	71.21	77.05	67.06	**64.50**$^{+0.82}$
BYOL		79.02	51.40	**72.07**	77.96	67.50	64.42
P.C.		**79.31**	51.66	**72.40**	77.62	**67.79**$^{+0.56}$	63.31
GCC-3D		79.16	50.66	69.95	77.07	66.59	63.67
STRL		79.15	51.71	67.78	77.10	66.21	62.90
Ours		79.05	**52.47**	**71.73**	**78.81**$^{+0.27}$	**67.75**$^{+0.52}$	**64.87**$^{+1.19}$

Table 2: Results of 3D object detection on KITTI dataset [37]. We conduct experiments on 3 different detectors: Second [27] (short as Sec.), PV-RCNN [34] (short as PV) and CenterPoint [31] (short as Cen.) and 8 different initialization methods including random (short as Rand, i.e. training from scratch), Swav [3], Deep Cluster (short as D. Cl.) [39], BYOL [4], Point Contrast (short as P.C.) [14], GCC-3D [17] and STRL [18]. Results are mAPs in %. "Easy", "Moderate" and "Hard" respectively indicate difficulty levels defined in KITTI dataset [37]. Results in each category are from moderate level. The "Overall" results in the final column is the major metric for comparisons. We use bold font for top 3 mAP in each category in each difficulty level for better understanding.

Init.	Once (CenterPoint)				KITTI (Second)			
	Vehicle	Pedestrian	Cyclist	Overall	Vehicle	Pedestrian	Cyclist	Overall
Random	62.85	45.52	59.39	55.92	77.45	48.71	63.32	63.16
Contextual Shape Prediction Only	62.86	**49.17**	59.86	57.30	77.75	49.16	63.18	63.36
Cooperative Contrastive Only	63.39	48.14	61.05	57.53	77.40	47.78	65.06	63.41
Ours	**64.50**	48.83	**62.17**	**58.50**	**77.95**	**49.59**	**65.60**	**64.38**

Table 3: Results of ablation study on Once [36] and KITTI [37]. We use CenterPoint [31] on Once and Second [27] on KITTI. Results are mAPs in %. For Once, results are average across different ranges. For KITTI, results are all in moderate level. We highlight the best performance in each category for better understanding.

results across different initialization schemes (relatively small improvements as compared to those in Once dataset [36]). When we look into detailed categories, **COˆ3** achieve consistent improvement on Pedestrian and Cyclist, which is essential for autonomous driving system.

Overall. **COˆ3** achieve consistent performance improvement on different detectors on different datasets while other initialization methods only occasionally make improvements but sometimes even make the performance worse. These demonstrate that the representation learned by **COˆ3**, which provides adequate views for contrastive learning and injects task-relevant information via contextual shape prediction, is able to be transferred to different datasets collected by different LiDAR sensors.

4.3 Ablation Study

We conduct ablation experiments to analyze the effectiveness of different components in **COˆ3**. We respectively pre-train the 3D encoder with cooperative contrastive objective and contextual shape prediction objective. Then we compare their performance in downstream tasks with those of training from scratch and pre-trained by **COˆ3**. As shown in Table 3, it can be found that each of the objective alone can achieve slight improvement, which demonstrates the effectiveness of each part of pre-training goal. Besides, once pre-trained by **COˆ3**, we achieve the best performance.

5 Conclusion

In this paper, we propose **COˆ3**, namely **Co**operative **Co**ntrastive Learning and **Co**ntextual Shape Prediction, for unsupervised 3D representation learning in outdoor scenes. The cooperative contrastive loss utilize the recently released vehicle-infrastructure-cooperation dataset DAIR-V2X [19] to build views for contrastive learning, which differ enough while sharing common semantics. The contextual shape prediction objective guides the 3D encoders to learn representations that are able to predict the neighborhood distribution of each point/voxel and provides task-relevant information for the 3D encoders. According to our experiments, **CO^3** is able to learn good representations and these representations can be transferred to downstream datasets collected by different LiDAR sensors to improve performance of different detectors. The performance gain surpasses previous unsupervised 3D representation learning methods for outdoor scene LiDAR point clouds, including GCC-3D [17] and STRL [18].

References

[1] K. He, H. Fan, Y. Wu, S. Xie, and R. 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] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[3] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9912–9924. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf.

[4] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf.

[5] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li. Dense contrastive learning for self-supervised visual pre-training. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2021.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[7] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[8] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[10] S. Chen, C. Duan, Y. Yang, D. Li, C. Feng, and D. Tian. Deep unsupervised learning of 3d point clouds via graph topology inference and filtering. *IEEE Transactions on Image Processing*, 29: 3183–3198, 2020. doi:10.1109/TIP.2019.2957935.

[11] M. Gadelha, R. Wang, and S. Maji. Multiresolution tree networks for 3d point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018.

[12] O. Poursaeed, T. Jiang, H. Qiao, N. Xu, and V. G. Kim. Self-supervised learning of point clouds via orientation estimation. In *2020 International Conference on 3D Vision (3DV)*, pages 1018–1028, 2020. doi:10.1109/3DV50981.2020.00112.

[13] L. Zhang and Z. Zhu. Unsupervised feature learning for point cloud understanding by contrasting and clustering using graph convolutional neural networks. In *2019 International Conference on 3D Vision (3DV)*, pages 395–404. IEEE, 2019.

[14] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany. Pointcontrast: Unsupervised pretraining for 3d point cloud understanding. In *European Conference on Computer Vision*, pages 574–591. Springer, 2020.

[15] J. Hou, B. Graham, M. Nießner, and S. 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.

[16] Y. Liu, L. Yi, S. Zhang, Q. Fan, T. Funkhouser, and H. Dong. P4contrast: Contrastive learning with pairs of point-pixel pairs for rgb-d scene understanding. *arXiv preprint arXiv:2012.13089*, 2020.

[17] H. Liang, C. Jiang, D. Feng, X. Chen, H. Xu, X. Liang, W. Zhang, Z. Li, and L. 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.

[18] S. Huang, Y. Xie, S.-C. Zhu, and Y. 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.

[19] H. Yu, Y. Luo, M. Shu, Y. Huo, Z. Yang, Y. Shi, Z. Guo, H. Li, X. Hu, J. Yuan, and Z. Nie. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[20] H. Wang, X. Guo, Z.-H. Deng, and Y. Lu. Rethinking minimal sufficient representation in contrastive learning. *arXiv preprint arXiv:2203.07004*, 2022.

[21] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4):509–522, 2002.

[22] M. Körtgen, G.-J. Park, M. Novotni, and R. Klein. 3d shape matching with 3d shape contexts. In *The 7th central European seminar on computer graphics*, volume 3, pages 5–17. Citeseer, 2003.

[23] S. Xie, S. Liu, Z. Chen, and Z. Tu. Attentional shapecontextnet for point cloud recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4606–4615, 2018.

[24] S. Shi, X. Wang, and H. Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[25] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[26] B. Yang, W. Luo, and R. Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[27] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.

[28] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019.

[29] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[30] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2647–2664, 2020.

[31] T. Yin, X. Zhou, and P. Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021.

[32] L. Fan, Z. Pang, T. Zhang, Y.-X. Wang, H. Zhao, F. Wang, N. Wang, and Z. Zhang. Embracing single stride 3d object detector with sparse transformer. *arXiv preprint arXiv:2112.06375*, 2021.

[33] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[34] S. Shi, L. Jiang, J. Deng, Z. Wang, C. Guo, J. Shi, X. Wang, and H. Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *arXiv preprint arXiv:2102.00463*, 2021.

[35] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1201–1209, 2021.

[36] J. Mao, M. Niu, C. Jiang, X. Liang, Y. Li, C. Ye, W. Zhang, Z. Li, J. Yu, C. Xu, et al. One million scenes for autonomous driving: Once dataset. 2021.

[37] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[39] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.