CO3: COOPERATIVE UNSUPERVISED 3D REPRESENTATION LEARNING FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

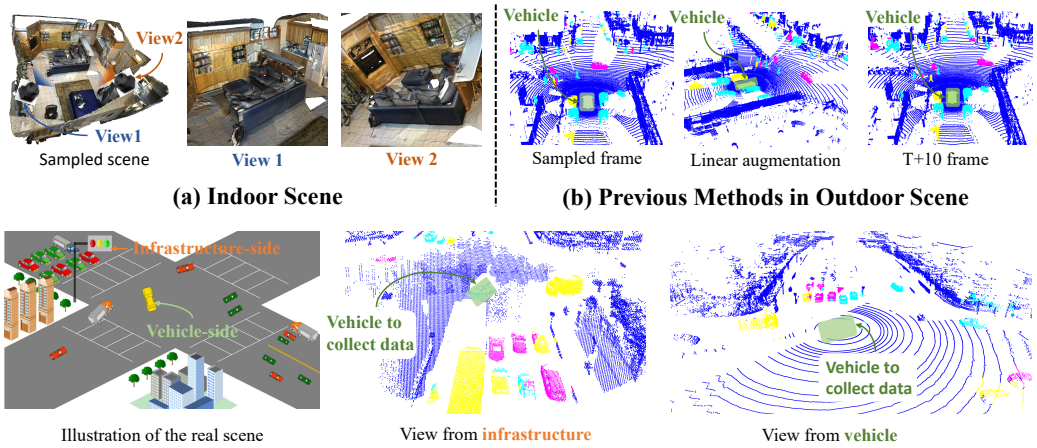
ABSTRACT

Unsupervised contrastive learning for indoor-scene point clouds has achieved great successes. However, unsupervised representation learning on outdoor-scene point clouds 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 CO3, namely **Co**operative **Co**ntrastive Learning and **Co**ntextual Shape Prediction, to learn 3D representation for outdoor-scene point clouds in an unsupervised manner. CO3 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, we propose contextual shape prediction to bring more task-relevant information for unsupervised 3D point cloud representation learning and we also provide a theoretical analysis for this pre-training goal. (3) As compared to previous methods, representation learned by CO3 is able to be transferred to different outdoor scene datasets collected by different type of LiDAR sensors. (4) CO3 improves current state-of-the-art methods on both *Once*, *KITTI* and *NuScenes* datasets by up to 2.58 mAP in 3D object detection task and 3.54 mIoU in LiDAR semantic segmentation task. Codes and models will be released. We believe CO3 will facilitate understanding LiDAR point clouds in outdoor scene.

1 INTRODUCTION

LiDAR is the most reliable sensor in outdoor environments and both of the machine learning and computer vision communities have 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. On the contrary, recent research efforts (He et al., 2020; Tian et al., 2019; Caron et al., 2020; Grill et al., 2020; Wang et al., 2021) in image domain focus on unsupervised representation learning with contrastive objective on different views built from different augmentation of images. They pre-train the 2D backbone with a large-scale dataset like ImageNet (Deng et al., 2009) in an unsupervised manner and use the pre-trained backbone to initialize downstream neural networks on different datasets and tasks, which achieve significant performance improvement in 2D object detection and semantic segmentation (Girshick et al., 2014; Lin et al., 2017; Ren et al., 2015). Inspired by these successes, we explore unsupervised representation learning for *outdoor scene* point clouds to learn general representations for different architectures on various downstream datasets and 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 to predict a **global** representation by minimizing contrastive loss (Zhang & Zhu, 2019) and aim at low-level downstream tasks including object classification and registration. To extend this idea to high-level perception tasks for *indoor-scene* point clouds, PointContrast (Xie et al., 2020) propose to reconstruct 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 (Hou et al., 2021) and (Liu et al., 2020) also need the reconstruction and this naturally assumes that the environment is static.



(a) Indoor Scene **(b) Previous Methods in Outdoor Scene**

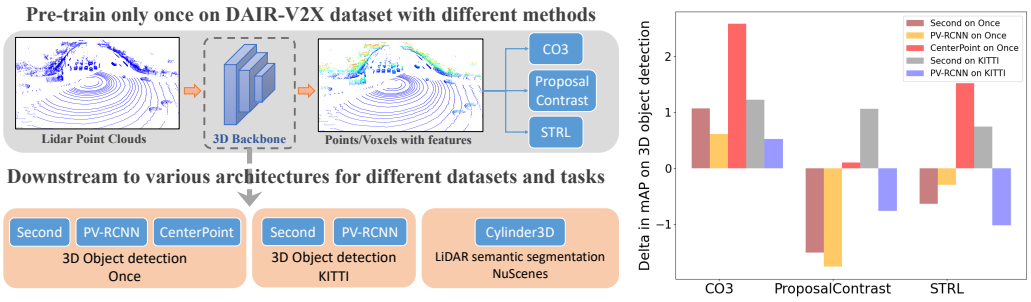
(c) Our CO3: Cooperative View from Infrastructure and Vehicle

Figure 1: Example views built by different methods in contrastive learning, including (a) previous indoor-scene methods (b) previous outdoor-scene methods and (c) the proposed CO3. (a) PointContrast capture point clouds at different poses in the reconstructed point cloud. Such views differ much and meanwhile share enough semantics including the same sofa and table. (b) GCC-3D and ProposalContrast apply linear augmentations to build views as shown in the first and second images. STRL utilize point cloud captured at different timestamps. As shown in the first and third images in (b), moving objects make it hard to find accurate correspondence. Our CO3 use point cloud from infrastructure and vehicle side for unsupervised representation learning as indicated in (c). An illustration of the scene is shown on the left and followings are point clouds from infrastructure-side and vehicle-side respectively. Compared to previous methods, CO3 can build two views that differ a lot as point clouds are captured in different sides and share adequate common semantics because they capture the same scene at the same time.

Fig. 1 (a) shows an example of views in PointContrast (Xie et al., 2020). We can see that the views differ a lot because they are captured from different poses but meanwhile, they still contain enough common semantic information such as the same sofa and table. These are demonstrated important properties of views in contrastive learning in (Tian et al., 2020).

However, outdoor scenes are dynamic and large-scale, making it impossible to reconstruct the whole scenes for building views. Thus, methods in (Xie et al., 2020; Hou et al., 2021; Liu et al., 2020) cannot be directly transferred but there exists two possible alternatives to build views. The first idea, embraced by (Liang et al., 2021; Yin et al., 2022), is to apply data augmentation to single frame of point cloud and treat the original and augmented versions as different views, which are indicated by the first and second pictures in Fig. 1 (b). However, all the augmentation of point clouds, including random drop, rotation and scaling, can be implemented in a linear transformation and views constructed in this way do not differ enough. The second idea is to consider point clouds at different timestamps as different views, represented by (Huang et al., 2021). Yet the moving objects would make it hard to find correct correspondence for contrastive learning. See the first and third pictures in Fig. 1 (b), while the autonomous vehicle is waiting at the crossing, other cars and pedestrians are moving around. The autonomous vehicle has no idea about how they move and is not able to find correct correspondence (common semantics). Due to these limitations, it is still challenging when transferring the pre-trained 3D encoders to datasets collected by different LiDAR sensors. *Could we find better views to learn general representations for outdoor-scene LiDAR point clouds?*

In this paper, we propose **CO**operative **CO**ntrastive Learning and **CO**ntextual Shape Prediction, namely CO3, to explore the potential of utilizing vehicle-infrastructure cooperation dataset for building adequate views in unsupervised 3D representation learning. As shown in (c) in Fig. 1, a recently released infrastructure-vehicle-cooperation dataset called DAIR-V2X (Yu et al., 2022) is utilized to learn general 3D representations. Point clouds from both vehicle and infrastructure sides are captured at the same timestamp thus views share adequate common semantic. Meanwhile infrastructure-side and vehicle-side point clouds differ a lot. These properties make views constructed in this way appropriate in contrastive learning. Besides, as proposed in (Wang et al., 2022), representations learned by pure contrastive learning lack task-relevant information. Thus we further add a pre-training goal called contextual shape prediction to reconstruct local point distribution, encouraging our proposed CO3 to capture more task-relevant information.



(a) Pipeline for unsupervised 3D representation learning

(b) Performance change after initialized by different methods

Figure 2: (a) shows the pipeline of unsupervised 3D representation learning pipeline. We aim to transfer the pre-trained 3D backbone to various architectures and downstream datasets. (b) presents the performance changes compared to training from scratch after pre-training with different methods. Our CO3 achieves consistent improvement across various detectors and downstream datasets.

Our contributions can be summarized as follows. (1) CO3 is proposed to utilize the vehicle-infrastructure-cooperation dataset to build adequate views for unsupervised contrastive 3D representation learning on *outdoor-scene* point clouds. (2) A shape-context prediction task is proposed alongside to inject task-relevant information, which is beneficial for downstream 3D detection and LiDAR semantic segmentation tasks. (3) The learned 3D representations is generic enough to be well transferred to datasets collected by different LiDAR sensors. (4) Extensive experiments demonstrate the effectiveness of CO3. For example, on 3D object detection task, CO3 improves Second, CenterPoints on *Once* dataset by 1.07 and 2.58 mAPs respectively. As for LiDAR semantic segmentation task, CO3 improves Cylinder3D on *NuScenes* dataset by 3.54 mIoUs.

2 Related Works

3D Perception Tasks. *3D object detection* aims to predict 3D boundary boxes for different objects in the LiDAR point clouds. Current 3D detectors can be divided into three main streams due to the 3D backbones they use: (1) point-based methods (Shi et al., 2019; Chen et al., 2017; Yang et al., 2018) use point-based 3D backbone. (2) voxel-based methods (Zhou & Tuzel, 2018; Lang et al., 2019; Su et al., 2015; Shi et al., 2020b; Yin et al., 2021; Fan et al., 2021) generally transform point cloud into voxel grids and process them using 3D volumetric convolutions. (3) point-voxel-combined methods (Shi et al., 2020a; 2021; Deng et al., 2021) utilize features from both (1) and (2). *LiDAR Semantic Segmentation* aims to predict per-point label for LiDAR point clouds. Cylinder3D (Zhu et al., 2021) and PVKD (Hou et al., 2022) are current SOTA methods for LiDAR Semantic Segmentation.

Unsupervised 3D Representation Learning. As shown in (a) of Fig. 2, unsupervised 3D representation learning aims to pre-train only for one time and downstream to different architectures on various datasets and tasks to achieve performance gain. PointContrast (Xie et al., 2020) is the pioneering work for unsupervised contrastive learning on *indoor-scene* point clouds, which relies on the reconstructed point clouds for constructing adequate views. To extend their ideas to outdoor-scene point clouds, GCC-3D and ProposalContrast (Liang et al., 2021; Yin et al., 2022) augment single frame of point cloud to build views and STRL (Huang et al., 2021) utilizes point clouds at different timestamps as views for contrastive learning. However, as discussed in Sec. 1, the views built in GCC-3D and ProposalContrast are too similar while those in STRL are hard to find correct correspondence, which violates the property for adequate views in contrastive learning as proposed in (Tian et al., 2020) and make their representations unable to transfer to datasets collected by different LiDAR sensors. In this work, we propose to use point clouds from vehicle and infrastructure to construct views in contrastive learning and introduce task-relevant information with a shape-context prediction pre-training goal to learn better 3D representation. The performance change after initialized with different methods are shown in (b) of Fig. 2. CO3 generally improve the performance and achieve more performance gains than other pre-training methods for different detectors on different datasets.

3 Methods

In this section, we introduce the proposed CO3 for unsupervised representation learning on LiDAR point clouds in outdoor scenes. As detailed in Fig. 3, CO3 has two pre-training objectives: (a) a

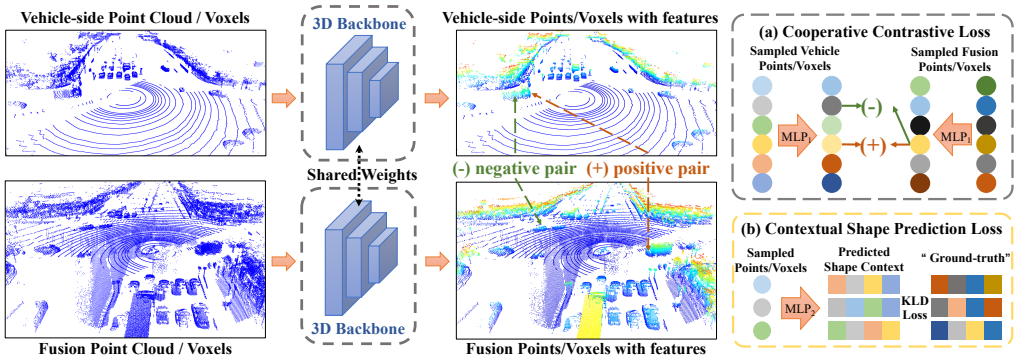


Figure 3: The pipeline of CO3. The infrastructure point cloud is transformed and fused with vehicle point cloud beforehand. With vehicle-side and fusion point clouds as inputs, we first process them with the 3D backbone to generate point/voxel-level representations. With these dense representations, we propose two pre-training objectives: (a) Cooperative Contrastive Loss, which introduces adequate views for contrastive learning. (b) Contextual Shape Prediction Loss, to bring in task-relevant information that are benefitial for downstream tasks.

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 in Sec. 3.1. Then we respectively introduce the cooperative contrastive objective and contextual shape prediction goal in Sec. 3.2 and Sec. 3.3.

3.1 Problem Formulation and Pipeline

Notation. To begin with, we denote LiDAR point clouds $\mathbf{P} \in \mathbb{R}^{N\times(3+d)}$ as the concatenation of the xyz-coordinate $\mathbf{C} \in \mathbb{R}^{N\times 3}$ and the point features $\mathbf{F} \in \mathbb{R}^{N\times d}$, which leads to $\mathbf{P} = [\mathbf{C}, \mathbf{F}]$. Here N denotes the number of points (or voxels) and d is the number of point feature channels. For raw LiDAR point clouds, $d = 1$ represents the intensity of each point. Moreover, we use the subscripts (e.g. 'v' and 'i') to indicate point clouds from different sources. For example, $\mathbf{P}_{\rm v} = [\mathbf{C}_{\rm v}, \mathbf{F}_{\rm v}] \in \mathbb{R}^{N_v\times(3+d)}$ and $\mathbf{P}_{\rm i} = [\mathbf{C}_{\rm i}, \mathbf{F}_{\rm i}] \in \mathbb{R}^{N_i\times(3+d)}$ respectively represent $N_{\rm v}$ and $N_{\rm i}$ points (or voxels) in vehicle and infrastructure sides. Besides, each pair of vehicle-side and infrastructure-side point clouds is associated with a transformation $\mathcal{T}$ mapping infrastructure-side coordinate to that of vehicle-side.

Pre-processing and Encoding. For cooperative learning, we need to first align the point clouds from both sides in the same coordinate. Hence, we transform the infrastructure-side point clouds to $\mathbf{P}'_{\rm i} = [\mathbf{C}'_{\rm i}, \mathbf{F}_{\rm i}]$ where $\mathbf{C}'_{\rm i} = \mathcal{T}(\mathbf{C}_{\rm i})$. However, we empirically find that the contrastive learning built upon vehicle-side point cloud $\mathbf{P}_{\rm v}$ and transformed infrastructure-side point clouds $\mathbf{P}'_{\rm i}$ only achieve marginal performance gains than training from scratch on downstream ONCE dataset (see Table 7 in Appendix D). This stems from the sparsity of LiDAR point clouds, which sometimes make it difficult to find good positive pairs to perform contrastive learning. To mitigate this problem, we concatenate the LiDAR point clouds from both sides to fusion point clouds $\mathbf{P}_{\rm f} = [\mathbf{P}_{\rm v}, \mathbf{P}'_{\rm i}] \in \mathbb{R}^{(N_{\rm v}+N_{\rm i})\times(3+d)}$, which is used as the contrastive view of vehicle-side point clouds $\mathbf{P}_{\rm v}$ as shown in Fig.3. By default of notations, we can express the coordinate and feature of fusion point clouds as $\mathbf{C}_{\rm f} = [\mathbf{C}_{\rm v}, \mathbf{C}'_{\rm i}]$ and $\mathbf{F}_{\rm f} = [\mathbf{F}_{\rm v}, \mathbf{F}_{\rm i}]$, respectively. Then $\mathbf{P}_{\rm v}$ and $\mathbf{P}_{\rm f}$ are embedded by the 3D encoder $f^{\rm enc}$ to obtain their 3D representations

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

where $\hat{\mathbf{P}}_{\rm v/f} \in \mathbb{R}^{\hat{N}_{\rm v/f}\times(3+\hat{d})}$. Here we use $\hat{N}_{\rm v/f}$ to denote the number of points after encoding because pooling operation often exists in 3D encoders (Graham et al., 2018) and changes the number of voxels/points. Moreover, $\hat{d}$ is the number of feature channels after encoding.

Loss Function. To guide the 3D encoder to learn good representations in an unsupervised manner, our proposed CO3 consists of a cooperative contrastive loss $\mathcal{L}_{\rm CO_2}$ and a contextual shape prediction loss $\mathcal{L}_{\rm CSP}$. The overall loss function is given by:

$$\mathcal{L} = \frac{1}{|\mathcal{P}_{\rm v/f}|} \sum_{\mathbf{P}_{\rm v/f}\in\{\mathcal{P}_{\rm v/f}\}} \mathcal{L}_{\rm CO_2}\{\hat{\mathbf{P}}_{\rm v}, \hat{\mathbf{P}}_{\rm f}\} + w \times \mathcal{L}_{\rm CSP}\{\hat{\mathbf{P}}_{\rm v}, \hat{\mathbf{P}}_{\rm f}, \mathbf{C}_{\rm f}\} \tag{2}$$

where $\mathcal{P}_{\text{v/f}}$ denote a batch of vehicle and fusion point clouds and $|\mathcal{P}_{\text{v/f}}|$ indicates the batch size. $\mathcal{L}_{\text{CO}_2}$ applies contrastive learning on the encoded vehicle and fusion pointclouds. Meanwhile, $\mathcal{L}_{\text{CSP}}$ introduces more task-relevant information into f^{enc} by using the encoded features to predict contextual shape obtained by the coordinate of fusion point clouds $\mathbf{C}_{\text{f}}$. w is a weight to balance the losses.

3.2 Cooperative Contrastive Objective

Unsupervised contrastive learning has been demonstrated successful in image domain (He et al., 2020; Tian et al., 2019) and indoor-scene point clouds (Xie et al., 2020). However, when it turns to outdoor-scene LiDAR point clouds, building adequate views, which share common semantics while differing enough, for contrastive learning is difficult. To tackle this challenge, we utilize a recently released vehicle-infrastructure-cooperation dataset called DAIR-V2X (Yu et al., 2022) and use vehicle-side point clouds and fusion point clouds as views for contrastive representation learning. 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. More details about view building in contrastive learning can be found in Appendix A.

Contrastive Head. Following BYOL (Grill et al., 2020), we construct contrastive head of cooperative contrastive objective by a Multi-Layer-Perceptron (MLP) layer and a ℓ_2-normalization. Specifically, the embedded features of vehicle and fusion point clouds, $\hat{\mathbf{F}}_{\text{v}}$ and $\hat{\mathbf{F}}_{\text{f}}$, are first projected by a MLP layer denoted as MLP_1. Then an ℓ_2-normalization is applied along feature dimension. This process is described below, where $\mathbf{Z}_{\text{v/f}} \in \mathbb{R}^{\hat{N}_{\text{v/f}} \times d_z}$ and d_z is the output feature dimension.

$$\hat{\mathbf{Z}}_{\text{v/f}} = \text{MLP}_1(\hat{\mathbf{F}}_{\text{v/f}}), \quad \mathbf{Z}_{\text{v/f}} = \hat{\mathbf{Z}}_{\text{v/f}} / \|\hat{\mathbf{Z}}_{\text{v/f}}\|_2 \tag{3}$$

Cooperative Contrastive Loss. For cooperative contrastive learning, we obtain positive pairs by exploring the correspondence between coordinates $\hat{\mathbf{C}}_{\text{v}}$ and $\hat{\mathbf{C}}_{\text{f}}$. In detail, we uniformly sample N_1 points from the vehicle point cloud and find their corresponding points in the fusion point cloud to form N_1 pairs of features $(\{z^n_{\text{v/f}}\}_{n=1}^{N_1})$ from $\mathbf{Z}_{\text{v/f}}$ for contrastive learning, where $z^n_{\text{v/f}} \in \mathbb{R}^{1 \times d_z}$. We treat corresponding points (or voxels) as positive pairs and otherwise negative pairs for contrastive learning. Example pairs are shown in Fig. 3. The final loss function is formulated below,

$$\mathcal{L}_{\text{CO}_2} = \frac{1}{N_1} \sum_{n=1}^{N_1} -\log\left(\frac{\exp(z^n_{\text{v}} \cdot z^n_{\text{f}}/\tau)}{\sum_{i=1}^{N_1} \exp(z^n_{\text{v}} \cdot z^i_{\text{f}}/\tau)}\right) \tag{4}$$

where τ is the temperature parameter. The bottom right box in Fig. 3 shows examples for this loss. As ground points only contain background information that is not related to perception tasks, we mark those points with height value lower than a threshold z_{thd} as ground points and filter them out when sampling.

3.3 Contextual Shape Prediction

CO3 aims to learn representations applicable to various downstream datasets. However, as demonstrated in (Wang et al., 2022), pure contrastive loss in Eqn. (4) would result in the lack of task-relevant information in the learned representations, making it hard to generalize across different architectures and datasets. Meanwhile, an additional reconstruction objective could increase the mutual information between the representations and the input views, which would bring in task-relevant information. Please refer to detailed explanations about this in Appendix B. For outdoor-scene point clouds, it is difficult to reconstruct the whole scene with point/voxel-level representations. To mitigate this issue, we propose to reconstruct the neighborhood of each point/voxel with its representation.

Local Distribution. 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 (Hou et al., 2021; Belongie et al., 2002; Körtgen et al., 2003; Xie et al., 2018). Fig. 4 shows two examples of shape context

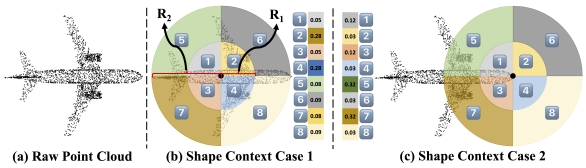


Figure 4: Two examples of shape context.

with 8 bins (the number of bins can be changed) around the query point, which is marked as a larger black point. In practice, we divide the neighborhood of each point in fusion point clouds into $N_{\text{bin}} = 32$ bins along xyz-plane with $R_1 = 0.5m$ and $R_2 = 4m$. Then we compute the "raw" shape context $Q' \in \mathbb{R}^{N_{\text{f}} \times N_{\text{bin}}}$, which describe the number of points in each bin around every fusion point. Next, we apply ℓ_2-normalization to Q' and a consecutive softmax to obtain the ground-truth shape context $Q \in \mathbb{R}^{N_{\text{f}} \times N_{\text{bin}}}$ as described below, where $Q_{i,*} \in \mathbb{R}^{1 \times N_{\text{bin}}}$ describes the ground-truth local distribution for i-th point.

$$Q_{i,*} = \text{softmax}(Q'_{i,*}/\|Q'_{i,*}\|_2) \tag{5}$$

Prediction Loss. We use the encoded features of both vehicle and fusion point clouds, i.e. $\hat{\mathbf{F}}_{\text{v/f}}$, to predict the local distribution. To be specific, $\hat{\mathbf{F}}_{\text{v/f}}$ is first passed through a MLP layer MLP_2 and softmax operation is applied on the projected features to get the predicted local distribution.

$$P_{\text{v/f}} = \text{softmax}(\text{MLP}_2(\hat{\mathbf{F}}_{\text{v/f}})) \tag{6}$$

where $P_{\text{v/f}} \in \mathbb{R}^{\hat{N}_{\text{v/f}} \times N_{\text{bin}}}$. Then we uniformly sample N_2 predictions ($\{p^n_{\text{v/f}}\}_{n=1}^{N_2}$) from $P_{\text{v/f}}$ and find their corresponding "ground truth" in Q by coordinate correspondence, where we have $\{q^n_{\text{v/f}}\}_{n=1}^{N_2}$. The dimensions of each prediction and "ground truth" are $p^n_{\text{v/f}} \in \mathbb{R}^{1 \times N_{bin}}$ and $q^n_{\text{v/f}} \in \mathbb{R}^{1 \times N_{bin}}$. Finally, the prediction loss with KL-divergence is given by,

$$\mathcal{L}_{\text{CSP}} = \frac{1}{N_2} \sum_{n=1}^{N_2} \sum_{m=1}^{N_{\text{bin}}} (p^{n,m}_{\text{v}} \log \frac{p^{n,m}_{\text{v}}}{q^{n,m}_{\text{v}}} + p^{n,m}_{\text{f}} \log \frac{p^{n,m}_{\text{f}}}{q^{n,m}_{\text{f}}}) \tag{7}$$

4 EXPERIMENTS

The goal for unsupervised representation learning is to learn general representation that can benefit different downstream architectures on different downstream datasets and tasks. In this section, we design experiments to answer the question whether CO3 learns such representation as compared to previous methods. We first provide experiment setups in Sec. 4.1 and then discuss main results in Sec. 4.2. We also conduct ablation study and qualitative visualization in Sec. 4.3 and 4.4.

4.1 EXPERIMENT SETUP

Pre-training Dataset. We utilize the recently released vehicle-infrastructure-cooperation dataset called *DAIR-V2X* (Yu et al., 2022) to pre-train the 3D encoder. *DAIR-V2X* is the first real-world autonomous dataset for vehicle-infrastructure-cooperative task. The LiDAR sensor at vehicle-side is 40-beam while a 120-beam LiDAR is utilized at infrastructure-side. There are 38845 LiDAR frames (10084 in vehicle-side and 22325 in infrastructure-side) for cooperative-detection task. The dataset contains around 7000 synchronized cooperative samples in total.

Implementation Details of CO3. 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 (Zhou & Tuzel, 2018; Yin et al., 2021; Shi et al., 2020a). We set the number of feature channels $d^{enc} = 64$, the temperature parameter in contrastive learning $\tau = 0.07$, the dimension of common feature space of vehicle-side and fusion point clouds $d_z = 256$ and the sample number in cooperative contrastive loss $N_1 = 2048$. For contextual shape prediction, we set the number of bins $N_{bin} = 32$, the sample number $N_2 = 2048$ and the weighting constant $w = 10$. The threshold for ground point filtering is $\text{z}_{\text{thd}} = 1.6\text{m}$.

Baselines. We implement STRL (Huang et al., 2021) and use the official code of ProposalContrast (Yin et al., 2022) as baselines. Besides, as proposed in (Mao et al., 2021), several methods in image domain and indoor-scene point clouds can be transferred to outdoor scene point clouds, including Swav (Caron et al., 2020), Deep Cluster (Caron et al., 2018), BYOL (Grill et al., 2020) and Point Contrast (Xie et al., 2020). ***Note*** that in order to make fair comparisons, all the pre-training methods are pre-trained on *DAIR-V2X*. Thus there might exist number discrepancy between the results and previous benchmarks.

Downstream Tasks. Two downstream tasks are selected for evaluation: 3D object detection and LiDAR semantic segmentation. 3D object detection task takes raw 3D point clouds as inputs and

Init.	Second			PV-RCNN			CenterPoint		
	Overall	0-30m	30-50m	**Overall**	0-30m	30-50m	**Overall**	0-30m	30-50m
Rand	52.21	60.71	47.31	54.55	63.05	50.00	55.92	66.39	50.16
Swav	53.03	61.38	48.34	54.89	63.41	50.31	57.00	67.54	50.60
Deep Cluster	52.30	61.03	47.00	54.91	63.84	50.29	57.65	68.05	51.04
BYOL	45.24	53.84	40.22	49.41	58.04	45.19	52.17	63.53	45.00
PointContrast	47.64	56.91	42.54	50.49	58.94	46.30	54.17	65.57	46.53
ProposalContrast	50.70	59.42	46.09	52.79	60.97	48.91	56.02	66.26	48.70
STRL	51.57	59.80	46.59	54.25	63.03	49.31	57.44	67.90	50.47
Ours	**53.28**$^{+1.07}$	62.16	49.33	**55.17**$^{+0.61}$	63.94	50.29	**58.50**$^{+2.58}$	69.09	51.51

Table 1: Results of 3D object detection on Once dataset (Mao et al., 2021). We conduct experiments on 3 different detectors: Second (Zhou & Tuzel, 2018) (short as Sec.), PV-RCNN (Shi et al., 2021) (short as PV) and CenterPoint (Yin et al., 2021) (short as Cen.) and 8 different initialization methods including random (short as Rand, i.e. training from scratch), Swav (Caron et al., 2020), Deep Cluster (short as D. Cl.) (Caron et al., 2018), BYOL (Grill et al., 2020), Point Contrast (short as P.C.) (Xie et al., 2020), GCC-3D (Liang et al., 2021) and STRL (Huang et al., 2021). Results are mAPs in %. "0-30m" and "30-50m" respectively indicate results for objects in 0 to 30 meters and 30 to 50 meters. The "Overall" metric highlighted in red is the overall mAP, which serves as major metric for comparisons. We use bold font for the best overall mAP of each detector for better understanding.

aims to output 3D boundary boxes of different objects in the scene. LiDAR semantic segmentation assign each 3D point a category label, including Car, Pedestrian, Bicycle, Truck and so on.

3D Object Detection. We select two downstream datasets: Once (Mao et al., 2021) and KITTI (Geiger et al., 2012). *Once* has 15k fully annotated frames of LiDAR scans. A 40-beam LiDAR is used to collect the point cloud data. We adopt common practice, including point cloud range and voxel size, in their public code repository. mAPs (mean accurate precisions) in different ranges and overall mAP are presented. *KITTI* is a widely used self-driving dataset, where point clouds are collected by LiDAR with 64 beams. It contains around 7k samples for training and another 7k for evaluation. All the results are evaluated by mAPs with three difficulty levels: Easy, Moderate and Hard. We select several current state-of-the-art methods implemented in the public repository of Once dataset (Mao et al., 2021)[1] and OpenPCDet[2] to evaluate the quality of representations learned by CO3, including Second (Zhou & Tuzel, 2018), CenterPoint (Yin et al., 2021) and PV-RCNN (Shi et al., 2020a).

LiDAR Semantic Segmentation. We select NuScenes (Caesar et al., 2020) as downstream dataset. There are 1000 scenes each of which lasts 20 seconds in *NuScenes*. It is collected with a 32-beam LiDAR sensor and the total number of frames is 40,000. We select Cylinder3D (Zhu et al., 2021) as downstream architecture. The full training is time-consuming (at least 4 days for one training) and we use a **1/8 training schedule** setting to evaluate whether CO3 is able to speed up training. We follow the conventional evaluation metrics. mAPs for detailed categories and mean intersection-over-union (mIoU) for overall evaluation are presented. Per-class mIoU is first computed as $\text{mIoU}_i = \frac{\text{TP}_i}{\text{TP}_i+\text{FP}_i+\text{FN}_i}$, where TP_i, FP_i and FN_i respectively represent true positive, false positive and false negative for class i. We then average over classes and get the final mIoU.

4.2 Main Results

Once Detection. As shown in Table 1, when initialized by CO3 , all the three detectors achieve the best performance on the overall mAPs, which we value the most, and CenterPoint (Yin et al., 2021) achieves the highest overall mAP (58.50) with 2.58 improvement. The improvement on PV-RCNN (Shi et al., 2021) is 0.62 in mAP (similar lower improvement with other pre-training methods) because PV-RCNN (Shi et al., 2021) has both the point-based and voxel-based 3D backbones, among which CO3 only pre-trains the voxel-based branch. It can be found that other baselines are not able to learn general representation for different architectures. For example, STRL achieves +1.52 mAPs improvements on CenterPoint but degrades the performance of Second and PV-RCNN. On the contrary, CO3 achieve consistent improvement over different detectors.

KITTI Detection. As shown in Table 2, when initialized by CO3 , PV-RCNN (Shi et al., 2021) achieves the best performance on Easy and Hard (+1.19) level and third place on Moderate level. Meanwhile, when Second (Zhou & Tuzel, 2018) is equipped with CO3, it achieves the highest mAPs on Easy level (+1.11), Moderate level (+1.22) and Hard level (+0.56). The lower gains on the KITTI

[1] https://github.com/PointsCoder/Once_Benchmark
[2] https://github.com/open-mmlab/OpenPCDet

Initialization	Second			PV-RCNN		
	Easy	Moderate	Hard	Easy	Moderate	Hard
Random	73.29	63.16	60.34	78.54	67.23	63.68
Swav	73.23	64.05	**60.90**	78.43	**67.91**	64.60
Deep Cluster	73.19	63.37	60.08	77.05	67.06	64.50
BYOL	71.05	60.39	56.98	77.96	67.50	64.42
PointContrast	72.67	62.74	59.21	77.62	67.79	63.31
ProposalContrast	74.23	64.22	60.88	77.28	66.47	63.22
STRL	73.95	63.90	**60.90**	77.10	66.21	62.90
Ours	$\mathbf{74.40}^{+1.11}$	$\mathbf{64.38}^{+1.22}$	$\mathbf{60.90}^{+0.56}$	$\mathbf{78.84}^{+0.30}$	$67.75^{+0.52}$	$\mathbf{64.87}^{+1.09}$

Table 2: Results of 3D object detection on KITTI dataset (Geiger et al., 2012). Results are mAPs in %. "Easy", "Moderate" and "Hard" respectively indicate difficulty levels defined in KITTI dataset. We use bold font for the best mAP of each detector in each difficulty level for better understanding.

Initialization	**mIoU**	Car	Truck	Con. Veh.	Ped.	Trailer	Bic.	M.C.	S.W.	Terrain	Veg.
Random	63.34	84.32	70.25	28.29	64.46	41.35	0.00	60.46	69.76	71.21	83.32
P.C.	64.31	84.70	74.36	30.81	63.52	47.13	10.16	55.55	70.38	71.43	84.65
STRL	64.71	84.66	76.65	27.30	63.29	52.76	12.79	60.11	70.27	71.70	84.60
Ours	$\mathbf{66.88}^{+3.54}$	**85.52**	**77.00**	**36.00**	**66.93**	**53.13**	**19.51**	**70.65**	**70.40**	**72.43**	**84.88**

Table 3: LiDAR semantic segmentation on NuScenes (Caesar et al., 2020). Results are mIoU and mAPs in %. We shows details results in each category. "Con. Veh.", "Ped.", "Bic.", "M.C.", "S. W." and "Veg." are abbreviations respectively for Construction Vehicle, Pedestrian, Bicycle, MotoCycle, Sidewalk and Vegetation. We use bold font for the best results in each column for better understanding.

Init.	**Overall**	Vehicle	Pedestrian	Cyclist
Random	55.92	62.85	45.52	59.39
Sup.	57.50	63.86	46.96	61.68
CO3	**58.50**	**64.60**	**48.83**	**62.17**

Table 4: Comparison to supervised initialization

Init.	**Overall**	Vehicle	Pedestrian	Cyclist
From Scratch	55.92	62.85	45.52	59.39
CO3 w. ground	57.37	63.19	48.16	60.76
CO3	**58.50**	**64.60**	**48.83**	**62.17**

Table 5: Ablation study on filtering out ground points.

dataset (Geiger et al., 2012) stem from the smaller number of training samples (half of that in Once (Mao et al., 2021)), which makes the detectors easily reach their capacity and improvement is hard to achieve. Consistent results across different initialization methods demonstrate this.

NuScene Semantic Segmentation. As shown in Table 3, CO3 improve Cylinder3D by 3.54 in mIoU and also achieves the best performance among the four initialization methods. Meanwhile, when initialized by CO3, Cylinder3D achieves the best mAPs among all the categories. On truck and construction vehicle, CO3 improve the performance of random initialization by 6.75 and 7.71 mAPs, which is very important in autonomous driving because correct segmentation of different vehicles can help benefit control and avoid accidents.

Comparison to Supervised Pre-training. We find backbone pre-trained on 3D detection task from the official codebase of DAIR-V2X dataset and use it to initialize CenterPoint, after which we fine-tune it on Once dataset. Results are shown in Table 4. It can be found that although supervised pre-training achieve improvement as compared to training from scratch, CO3 leads to the best results in all categories and the overall mAP. This is because supervised pre-training overfit to DAIR-V2X dataset, making the improvement lower than CO3 in 3D detection task on Once dataset.

Overall Evaluation. To summarize, CO3 achieves consistent performance gains over different architectures on different tasks (3D object detection and LiDAR semantic segmentation) and datasets (Once, KITTI and NuScenes) when pre-trained only on DAIR-V2X dataset. In comparison, other baseline pre-training methods only occasionally improve the performance and sometimes even bring degradation. These demonstrate CO3 learns general 3D representations.

4.3 Ablation Study

The influence of ground points in pre-training. We first conduct ablation experiments on filtering out ground points. We use CO3 to pre-train 3D backbone without filtering out ground points and downstream it to 3D object detection with CenterPoint on Once. Results are shown in Table 5. When pre-training without filtering out ground points, the performance of CO3 drop in each category and the overall mAP. This demonstrate the effectiveness of filtering out ground points because ground points contain background information that is not useful for 3D perception tasks.

Init.	Once (CenterPoint)				KITTI (Second)			
	Overall	Vehicle	Pedestrian	Cyclist	**Overall**	Vehicle	Pedestrian	Cyclist
Random	55.92	62.85	45.52	59.39	63.16	77.45	48.71	63.32
Contextual Shape Prediction Only	57.30	62.86	**49.17**	59.86	63.36	77.75	49.16	63.18
Cooperative Contrastive Only	57.53	63.39	48.14	61.05	63.41	77.40	47.78	65.06
CO3	**58.50**	**64.50**	48.83	**62.17**	**64.38**	**77.95**	**49.59**	**65.60**

Table 6: Results of ablation study on Once (Mao et al., 2021) and KITTI (Geiger et al., 2012). We use CenterPoint (Yin et al., 2021) on Once and Second (Zhou & Tuzel, 2018) 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 column for better understanding.

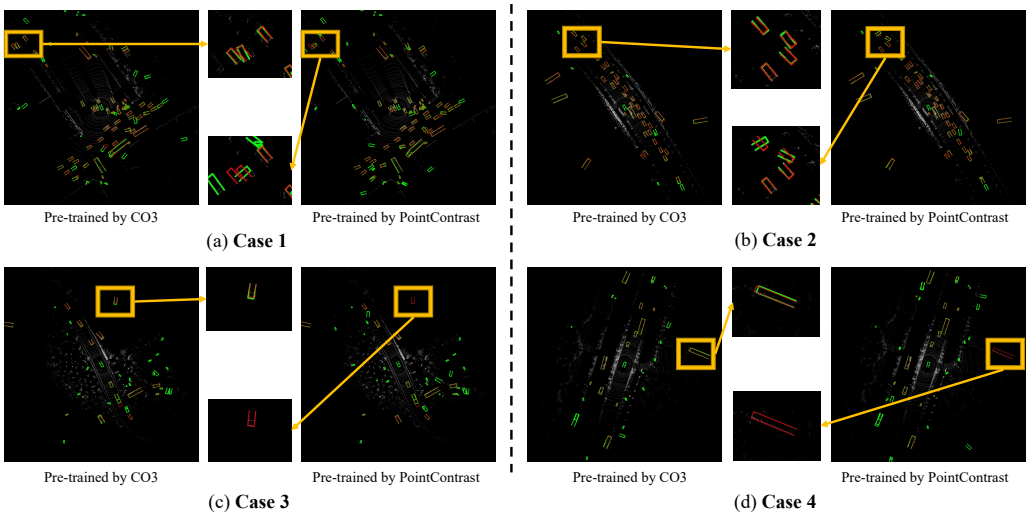


Figure 5: Visualization for detection results with CenterPoint. Green boxes are predicted ones and red boxes are the ground truth.

The effect of each component in CO3. We conduct ablation experiments to analyze the effectiveness of different components. We respectively pre-train the 3D encoder with cooperative contrastive objective and contextual shape prediction objective. As shown in Table 6, it can be found that each objective alone can achieve slight improvement, which demonstrates the effectiveness of either part. Besides, when pre-trained by CO3, we achieve the best performance on the overall mAPs.

4.4 Qualitative Experiment

We use the 3D backbone pre-trained by CO3 and PointContrast to initialize CenterPoint and train it on Once dataset. Then we visualize the detection results in Fig. 5, where predicted boxes are marked as green and the ground truth boxes are red. In Case 1 and 2, when the detector is initialized by CO3, the detection results are more correct in headings as shown in the zoom-in area. Correct heading prediction is important especially for control in autonomous driving. In Case 3 and 4, it can be found that CO3 helps the CenterPoint detect object with only a few points captured by the LiDAR sensor and meanwhile, PointContrast initialization fails to detect them. This is also essential in autonomous driving because detection failure can sometimes lead to disaster.

5 Conclusion and Future Work

In this paper, we propose CO3, namely **Co**operative **Co**ntrastive Learning and **Co**ntextual Shape Prediction, for unsupervised 3D representation learning in outdoor scenes. The recently released vehicle-infrastructure-cooperation dataset DAIR-V2X is utilized to build views for cooperative contrastive learning. Meanwhile the contextual shape prediction objective provides task-relevant information for the 3D encoders. Our experiments demonstrate that the representation learned by CO3 can be transferred to various architectures and different downstream datasets and tasks to achieve performance gain. Currently the size of the real cooperation dataset is relatively small and large-scale dataset has been demonstrated beneficial in unsupervised representation learning in previous works (He et al., 2020; Grill et al., 2020). Thus it will be interesting if larger cooperative datasets can be collected in the future and used for pre-training to have larger performance gain.

REFERENCES

Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4):509–522, 2002.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 132–149, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9912–9924. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf.

Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1907–1915, 2017.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 1201–1209, 2021.

Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing single stride 3d object detector with sparse transformer. *arXiv preprint arXiv:2112.06375*, 2021.

Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Elsevier, 2013.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal 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 (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf.

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

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

Yuenan Hou, Xinge Zhu, Yuexin Ma, Chen Change Loy, and Yikang Li. Point-to-voxel knowledge distillation for lidar semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8479–8488, 2022.

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

Marcel Körtgen, Gil-Joo Park, Marcin Novotni, and Reinhard Klein. 3d shape matching with 3d shape contexts. In *The 7th central European seminar on computer graphics*, volume 3, pp. 5–17. Citeseer, 2003.

J. Kreer. A question of terminology. *IRE Transactions on Information Theory*, 3(3):208–208, 1957. doi: 10.1109/TIT.1957.1057418.

Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12697–12705, 2019.

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

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

Yunze Liu, Li Yi, Shanghang Zhang, Qingnan Fan, Thomas Funkhouser, and Hao Dong. P4contrast: Contrastive learning with pairs of point-pixel pairs for rgb-d scene understanding. *arXiv preprint arXiv:2012.13089*, 2020.

Jiageng Mao, Minzhe Niu, Chenhan Jiang, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Jie Yu, Chunjing Xu, et al. One million scenes for autonomous driving: Once dataset. 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020a.

Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng 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, 2020b.

Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *arXiv preprint arXiv:2102.00463*, 2021.

Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pp. 945–953, 2015.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446–2454, 2020.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6827–6839. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/4c2e5eaae9152079b9e95845750bb9ab-Paper.pdf.

Haoqing Wang, Xun Guo, Zhi-Hong Deng, and Yan Lu. Rethinking minimal sufficient representation in contrastive learning. *arXiv preprint arXiv:2203.07004*, 2022.

Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2021.

Wikipedia contributors. Mutual information — Wikipedia, the free encyclopedia, 2022. URL https://en.wikipedia.org/w/index.php?title=Mutual_information&oldid=1089343634. [Online; accessed 21-June-2022].

Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. Attentional shapecontextnet for point cloud recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4606–4615, 2018.

Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European Conference on Computer Vision*, pp. 574–591. Springer, 2020.

Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7652–7660, 2018.

Junbo Yin, Dingfu Zhou, Liangjun Zhang, Jin Fang, Cheng-Zhong Xu, Jianbing Shen, and Wenguan Wang. Proposalcontrast: Unsupervised pre-training for lidar-based 3d object detection. 2022.

Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11784–11793, 2021.

Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, and Zaiqing 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.

Ling Zhang and Zhigang 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)*, pp. 395–404. IEEE, 2019.

Yin Zhou and Oncel 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*, pp. 4490–4499, 2018.

Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9939–9948, 2021.

A BACKGROUND ABOUT VIEW BUILDING IN CONTRASTIVE LEARNING

In this section, we discuss how to build proper views in contrastive learning. Firstly, we introduce the formulation of contrastive learning and some important properties of good views in contrastive learning as proposed in (Tian et al., 2020). Then we discuss view building for LiDAR point clouds.

View Building Contrastive Learning. Unsupervised representation learning aims to pre-train 2D/3D backbones on a dataset without labels, which can be transferred to downstream datasets and tasks to achieve performance improvement over training from scratch (random initialization). Recently, unsupervised contrastive learning achieves great success in image domain (He et al., 2020; Tian et al., 2019; Caron et al., 2020; Grill et al., 2020; Wang et al., 2021). Given a batch of images $\mathcal{X}$ as inputs, these works first apply two kinds of random augmentations for each image $x^n \in \mathcal{X}$ ($n = 1, 2, ..., N$ where N is the number of images in the batch) to get augmented images x_1^n and x_2^n, which are called different views of x^n. The main objective of contrastive learning is to pull together the representations of views of the same image in the feature space while pushing away representations of different images, as indicated in the equation below:

$$\begin{aligned} L_{\text{con}} &= \frac{1}{N}\sum_{n=1}^{N} -\log\left(\frac{\exp(z_1^n \cdot z_2^n/\tau)}{\sum_{i=1}^{N}\exp(z_1^n \cdot z_2^i/\tau)}\right) \quad \text{with} \\ z_{1,2}^n &= f_{\text{I}}^{\text{enc}}(x_{1,2}^n) \quad n = 1, 2, ..., N \end{aligned} \tag{8}$$

where $f_{\text{I}}^{\text{enc}}$ is the 2D backbone used to extract representations. $z_{1,2}^n$ is the encoded representations for views $x_{1,2}^n$. To apply contrastive loss in the first line in Eqn. (8), views of the same image are considered as positive pairs and other pairs are negative ones. The numerator indicates the similarity of positive pairs while the denominator sums up the positive similarity and the sum of similarity of negative pairs. τ is the temperature parameter. Minimizing this loss equals to maximize the similarity of positive pairs and minimize the similarity of negative pairs.

Authors in (Tian et al., 2020) discuss what property views should have to benefit contrastive learning via Information Theory and propose that mutual information (Shannon, 2001; Kreer, 1957; Wikipedia contributors, 2022) of views ,$I(x_1^n; x_2^n)$, can indicate the quality of the learned representations. Mutual information formally quantifies "how much information about one random variable we can obtain when observing the other one". Experiments on images suggest that there exists a "sweet spot" for $I(x_1^n; x_2^n)$ where the pre-trained backbone can achieve the most significant performance improvement in downstream tasks. This means the mutual information between views can neither be too low (sharing little semantics) nor too high (differing little). Further experiments in (Tian et al., 2020) indicate that the mutual information of different augmented images is high and reducing $I(x_1^n; x_2^n)$ by applying stronger augmentations is effective. The performance on downstream tasks increases at the beginning and then decrease when the augmentations are too strong.

Views Building for LiDAR Point Clouds. As discussed in the main paper, it is impossible for us to reconstruct the whole outdoor-scene for constrative learning, which is demonstrated useful in indoor-scene (Xie et al., 2020; Hou et al., 2021; Liu et al., 2020). And there exists two alternatives to build views for outdoor-scene LiDAR point clouds. The first one (Liang et al., 2021; Yin et al., 2022) is to apply data augmentation to single frame of point cloud and treat the original and augmented versions as different views, which is similar to what previous works do in image domain. However, the augmentations in image domain are highly non-linear while all the augmentation of point clouds, including random drop, rotation and scaling, can be implemented in a linear transformation. As claimed in (Tian et al., 2020), the highly non-linear augmentations on images already bring high mutual information between views. Thus views of LiDAR point clouds built in this way would have higher mutual information, which is not adequate for learning representations. The second intuitive idea to build views is to utilize point clouds at different timestamps, embraced by (Huang et al., 2021). However, outdoor-scenes are dynamic and the autonomous driving vehicle has no idea about how other objects (cars, pedestrians, etc.) move. Thus observing one view (timestamp t) bring in little information about the other one (timestamp t+10 for example), meaning that $I(x_1^n; x_2^n)$ can be extremely low and this can be harmful to the learned representations. Due to these limitations, pre-trained 3D encoders in (Liang et al., 2021; Huang et al., 2021) cannot achieve noticeable improvement when transferring to datasets collected by different LiDAR sensors. Thus, in this paper, we propose to utilize the vehicle-infrastructure-cooperation dataset (Yu et al., 2022), which capture the same scene from different view-point at the same timestamp, for contrastive representation learning. Views built

with this dataset neither share too much information (captured from different view-points) nor share too little information (captured at the same time, easy to find correspondence), which is adequate for contrastive learning.

B Reconstruction Objective for Task-relevant Information

In this section, we borrow the ideas from (Wang et al., 2022) to explain why pure contrastive learning bring less improvements as shown in Table 3 in the main paper. Firstly, we give the definition of sufficient representation and minimal sufficient representation in contrastive learning. Then, we present analysis on image classification problem as downstream task and we refer readers to (Wang et al., 2022) for other downstream tasks. Finally, we propose our pre-training objective for LiDAR point clouds.

Sufficient Representation and Minimal Sufficient Representation. Sufficient Representation $z^n_{1,\text{suf}}$ of x^n_1 contains all the information that is shared by x^n_1 and x^n_2, which means $z^n_{1,\text{suf}}$ can be used to express common semantics shared by these two views. Among all the sufficient representations for x^n_2, minimal sufficient representation $z^n_{1,\text{min}}$ contains the least information about x^n_1. The learned representation in contrastive learning is sufficient and almost minimal. Assuming that $\mathcal{Z}_{1,\text{suf}}$ is the set of all possible sufficient representations of view x^n_1, we can define these two concepts as belows,

Definition 1. *$z^n_{1,suf}$ of view x^n_1 is sufficient for x^n_2* ***if and only if*** *$I(z^n_{1,suf}, x^n_2) = I(x^n_1, x^n_2)$.*

Definition 2. *$z^n_{1,min} \in \mathcal{Z}_{1,suf}$ of view x^n_1 is minimal sufficient* ***if and only if*** *$I(z^n_{1,min}, x^n_1) \leq I(z^n_{1,suf}, x^n_1)$, $\forall z^n_{1,suf} \in \mathcal{Z}_{1,suf}$.*

Theorem. (1) $z_{1,suf}$ provides more information about the downstream task T than $z_{1,\text{min}}$. (2) The upper bound of error rates in downstream tasks using minimal sufficient representations are higher than that of sufficient representations. That is,

$$\begin{aligned} I(z_{1,\text{suf}}, T) &\geq I(z_{1,\text{min}}, T) \\ \sup\{P^e_{\text{suf}}\} &\leq \sup\{P^e_{\text{min}}\} \end{aligned} \tag{9}$$

This gap stems from the missing task-relevant information in $z_{1,\text{min}}$. To prevent this problem, authors in (Wang et al., 2022) propose to add a reconstruction objective (reconstruct x_1 using z_1) alongside the contrastive loss to increase $I(z_1, x_1)$, which indirectly increases $I(z_1, T|x_2)$ and brings improvement in downstream classification problem over pure contrastive learning.

Proof for Image Classification Problem. We denote the downstream classification task as T and the task-relevant information in minimal sufficient representation $z_{1,\text{min}}$ can be described as below:

$$\begin{aligned} I(z_{1,\text{suf}}, T) &= I(z_{1,\text{min}}, T) + [I(x_1, T|z_{1,\text{min}}) - I(x_1, T|z_{1,\text{suf}})] \\ &\geq I(z_{1,\text{min}}, T) \end{aligned} \tag{10}$$

To begin with, $z_{1,\text{suf}}$ and $z_{1,\text{min}}$ are sufficient representations and they contain two parts of information: shared information between x_1 and x_2, and extra information about x_1. Thus the mutual information $I(z_{1,\text{suf}}, T)$ can be decomposed into $I(z_{1,\text{min}}, T)$ and $[I(x_1, T|z_{1,\text{min}}) - I(x_1, T|z_{1,\text{suf}})]$, where $I(x_1, T|z_{1,\text{min}})$ indicates the information about T we can obtain by observing x_1 when $z_{1,\text{min}}$ is known. As $z_{1,\text{suf}}$ contains more information about x_1 than $z_{1,\text{min}}$, the second term is larger than zero and the right-hand-side of the first line in Eqn. (10) is larger than $I(z_{1,\text{min}}, T)$. This indicates that $z_{1,\text{suf}}$ contains more task-relevant information than $z_{1,\text{min}}$ and thus would have better performance in T. Then we consider using Bayes error rate P^e (Fukunaga, 2013), which is the lower-bound of achievable error for the classifier, to analyze performance of $z_{1,\text{suf}}$ and $z_{1,\text{min}}$ on downstream classification problem. We have

$$\begin{aligned} P^e_{\text{suf}} &\leq 1 - \exp[-H(T) + I(x_1, x_2, T) + I(z_{1,\text{suf}}, T|x_2)] \\ P^e_{\text{min}} &\leq 1 - \exp[-H(T) + I(x_1, x_2, T)] \end{aligned} \tag{11}$$

where $H(T)$ is the entropy of the task. Since $1 - \exp[-(H(T) + I(x_1, x_2, T) + I(z_{1,\text{suf}}, T|x_2)] \leq 1 - \exp[-H(T) + I(x_1, x_2, T)]$, the upper bound of Bayes error rate of minimal sufficient representation is larger than that of sufficient representations. This indicates that ideally $z_{1,\text{suf}}$ can achieve better performance than $z_{1,\text{min}}$ in classification problem.

Contextual Shape Prediction Objective. As it is impossible to reconstruct the whole scene point cloud with point-level or voxel-level representations. We propose an additional pre-training objective to predict distribution of local neighborhood using point/voxel-level representation. We use shape context to describe the local neighborhood distribution of a point/voxel, which has been demonstrated as a useful local distribution descriptor in previous works (Hou et al., 2021; Belongie et al., 2002; Körtgen et al., 2003; Xie et al., 2018). As demonstrated in our ablation study (Table 3 in main paper), this additional pre-training objective bring more significant performance improvement over pre-trained by pure contrastive loss. We also provide python-style code for computing shape context as followings

Algorithm 1 Implementation of Contextual Shape Computation in Python Style.

```
class Contextual_Shape(object):
   def __init__(self, r1=0.125, r2=2, nbins_xy=2, nbins_zy=2):
      self.r1 = r1
      self.r2 = r2
      self.nbins_xy = nbins_xy
      self.nbins_zy = nbins_zy
      self.partitions = nbins_xy * nbins_zy * 2

   def pdist_batch(rel_trans):
      D2 = torch.sum(rel_trans.pow(2), 3)
      return torch.sqrt(D2 + 1e-7)

   def compute_rel_trans_batch(A, B):
      return A.unsqueeze(1) - B.unsqueeze(2)

   def hash_batch(A, B, seed):
      mask = (A >= 0) & (B >= 0)
      C = torch.zeros_like(A) - 1
      C[mask] = A[mask] * seed + B[mask]
      return C

   def compute_angles_batch(rel_trans):
      angles_xy = torch.atan2(rel_trans[:, :, :, 1], rel_trans[:, :, :, 0])
      angles_xy = torch.fmod(angles_xy + 2 * math.pi, 2 * math.pi)
      angles_zy = torch.atan2(rel_trans[:, :, :, 1], rel_trans[:, :, :, 2])
      angles_zy = torch.fmod(angles_zy + 2 * math.pi, math.pi)
      return angles_xy, angles_zy

   def compute_partitions_batch(self, xyz_batch):
      rel_trans_batch = ShapeContext.compute_rel_trans_batch(xyz_batch,
          xyz_batch)
      # compute angles from different points to the query one
      angles_xy_batch, angles_zy_batch = ShapeContext.compute_angles_batch(
          rel_trans_batch)
      angles_xy_bins_batch = torch.floor(angles_xy_batch / (2 * math.pi /
          self.nbins_xy))
      angles_zy_bins_batch = torch.floor(angles_zy_batch / (math.pi / self.
          nbins_zy))
      angles_bins_batch = ShapeContext.hash_batch(angles_xy_bins_batch,
          angles_zy_bins_batch, self.nbins_zy)
      # compute distances between different points and the query one
      distance_matrix_batch = ShapeContext.pdist_batch(rel_trans_batch)
      dist_bins_batch = torch.zeros_like(angles_bins_batch) - 1
      # generate partitions for each points
      mask_batch = (distance_matrix_batch >= self.r1) & (
          distance_matrix_batch < self.r2)
      dist_bins_batch[mask_batch] = 0
      mask_batch = distance_matrix_batch >= self.r2
      dist_bins_batch[mask_batch] = 1
      bins_batch = ShapeContext.hash_batch(dist_bins_batch,
          angles_bins_batch, self.nbins_xy * self.nbins_zy)
      return bins_batch
```

C DATASETS DETAILS

In this section, we introduce details about different datasets used in the main paper for evaluation and also two more datasets in additional experiments.

DAIR-V2X. *DAIR-V2X* (Yu et al., 2022) is the first real-world autonomous dataset for vehicle-infrastructure-cooperative detection task. It covers various scenes, including cities and highways, and different whether condition including sunny, rainy and foggy days. A virtual world coordinate is used to align the vehicle LiDAR coordinate and infrastructure LiDAR coordinate. There are 38845 LiDAR frames (10084 in vehicle-side and 22325 in infrastructure-side) for cooperative-detection task. The dataset contains around 7000 synchronized cooperative samples in total and we utilize them to pre-train 3D encoder in an unsupervised manner via the proposed CO3. The LiDAR sensor at vehicle-side is 40-beam while a 120-beam LiDAR is utilized at infrastructure-side.

Once. *Once* (Mao et al., 2021) is a large-scale autonomous dataset for evaluating self-supervised methods with 1 Million LiDAR frames and only 15k fully annotated frames with 3 classes (Vehicle, Pedestrian, Cyclist). A 40-beam LiDAR is used in (Mao et al., 2021) to collect the point cloud data. We adopt common practice, including point cloud range and voxel size, in their public code repository[3]. As for the evaluation metrics, 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 (mean accurate precisions) for each category. We also further overage over the three categories and compute an 'Overall' mAP for evaluations.

KITTI. *KITTI* (Geiger et al., 2012) is a widely used self-driving dataset, where point clouds are collected by LiDAR with 64 beams. It contains around 7k samples for training and another 7k for evaluation. For point cloud range and voxel size, we adopt common practice in current popular codebase like MMDetection3D[4] and OpenPCDet[5]. 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.

D ADDITIONAL EXPERIMENT RESULTS

D.1 INFRASTRUCTURE AS VIEW IN CONTRASTIVE LEARNING

We also conduct ablation study on the fusion view. Instead of using fusion point clouds as view in contrastive learning, we directly use infrastructure side point cloud as another view. Results are shown in Table 7. It can be found that if we directly use infrastructure-side point cloud for contrastive learning, the performance improvement is very marginal. This stems from the sparsity of LiDAR point clouds, which sometimes make it difficult to find good positive pairs to perform contrastive learning.

Init.	Overall	Vehicle	Pedestrian	Cyclist
Random	55.92	62.85	45.52	59.39
Inf-view	56.45	62.71	46.63	60.02
CO3	58.50	64.60	48.83	62.17

Table 7: Resuls of directly using infrastructure-side point cloud as view in contrastive learning.

E IMPLEMENTATION DETAILS

In this section, we introduce some details about implementation in both pre-training stage and fine-tuning stage. Common settings of pre-training and fine-tuning are listed in Table 8 and we discuss other settings that vary in different detectors later.

[3] https://github.com/PointsCoder/Once_Benchmark
[4] https://github.com/open-mmlab/mmdetection3d
[5] https://github.com/open-mmlab/OpenPCDet

Configuration	Pre-training	KITTI	Once	Waymo	NuScenes
optimizer	AdamW	Adam	Adam	Adam	Adam
base learning rate	0.0001	0.003	0.003	0.003	0.003
weight decay	0.01	0.01	0.01	0.01	0.01
batch size	16	-	-	-	-
learning rate schedule	cyclic	cyclic	cyclic	cyclic	cyclic
GPU numbers	8	4	4	8	8
training epochs	***20***	80	80	30	-

Table 8: Details about implementations. "Pre-training" means settings in *DAIR-V2X* (Yu et al., 2022). "KITTI", "Once", "Waymo" and "NuScenes" respectively indicate settings for detection tasks in *KITTI* (Geiger et al., 2012), *Once* (Mao et al., 2021), *Waymo* (Sun et al., 2020) and *NuScenes* (Caesar et al., 2020). We list all common settings and discuss those vary in different detectors below, which are marked as "-" in this table.

Other Pre-training Settings. To accelerate the pre-training process, we utilize the "repeated dataset" in MMDetection3D and the schedule is set to 10 epochs, which equals to 20 epochs without "repeated dataset". Thus the number 20 for training epochs in Table 8 is tilt.

Other Fine-tuning Settings. Batch size settings for different detectors on different datasets are shown in Table 9. We train 20 epochs for CenterPoint (Yin et al., 2021) in NuScenes (Caesar et al., 2020) and 80 epochs for Second (Zhou & Tuzel, 2018) in Nuscenes (Caesar et al., 2020). We use different types of GPUs, different number of GPUs and different version of PyTorch (Paszke et al., 2019) as compared to those used in the codebases, which may lead to degrading when training from scratch. Thus these parameters are tuned based on the original settings from the codebases to make the performance of training from scratch match or even surpass the results they published.

Detectors	KITTI	Once	Waymo	NuScenes
Second	48	48	48	32
PV-RCNN	16	48	-	-
CenterPoint	-	32	32	32

Table 9: Details about batchsize settings for different detectors on different datasets. "-" means there is no configuration for the detector on the exact dataset in the codebase or we do not conduct the downstream experiments.

F TABLE OF NOTATION

Variable	Description	Note
$\mathbf{P}_{\text{v/f}}$	Raw vehicle/fusion LiDAR points.	
$\hat{\mathbf{P}}_{\text{v/f}}$	Point/Voxel-level features after embedded by the 3D backbone.	Indicating same process applied to vehicle/fusion point cloud.
$\hat{\mathbf{Z}}_{\text{v/f}}$	Projected point/voxel-level features in the common feature space.	
$\mathbf{Z}_{\text{v/f}}$	Normalized point/voxel-level features in the common feature space.	An L-2 normalization is applied on the feature dimension of $\hat{\mathbf{Z}}_{\text{v/f}}$. $z^n_{\text{v/f}}$ is sampled point/voxel feature for contrastive learning.
N_1	The number of samples for contrastive learning.	
τ	Temperature parameter in contrastive learning.	
Q'	A matrix in dimension $\mathbb{R}^{N_{\text{f}} \times N_{\text{bin}}}$ indicating the number of points in each bin in the neighborhood of each fusion point.	
Q	Apply ℓ_2-normalization and softmax to Q' then we get $Q \in \mathbb{R}^{N_{\text{f}} \times N_{\text{bin}}}$, which describes a local geometry distribution in the neighborhood of each fusion point.	$q^n_{\text{v/f}}$ is sampled "ground truth" distribution from Q
P	Predicted local geometry distribution from $\mathbf{P}^{enc}_{v/f}$.	$p^n_{\text{v/f}}$ is sampled distribution prediction from P.
N_2	The number of samples for contextual shape prediction.	

Table 10: Detailed description of variables