Exploring Geometry-aware Contrast and Clustering Harmonization for Self-supervised 3D Object Detection

Hanxue Liang\textsuperscript{1,}\textsuperscript{*}, Chenhan Jiang\textsuperscript{2,}*\textsuperscript{†}, Dapeng Feng\textsuperscript{3}, Xin Chen\textsuperscript{4}, Hang Xu\textsuperscript{2}, Xiaodan Liang\textsuperscript{3\textdagger}, Wei Zhang\textsuperscript{2}, Zhenguo Li\textsuperscript{2}, Luc Van Gool\textsuperscript{1},

\textsuperscript{1}\text{ETH Zurich} \textsuperscript{2}\text{Huawei Noah’s Ark Lab} \textsuperscript{3}\text{Sun Yat-Sen University} \textsuperscript{4}\text{The University of Hong Kong}

Abstract

Current 3D object detection paradigms highly rely on extensive annotation efforts, which makes them not practical in many real-world industrial applications. Inspired by that a human driver can keep accumulating experiences from self-exploring the roads without any tutor’s guidance, we first step forwards to explore a simple yet effective self-supervised learning framework tailored for LiDAR-based 3D object detection. Although the self-supervised pipeline has achieved great success in 2D domain, the characteristic challenges (e.g., complex geometry structure and various 3D object views) encountered in the 3D domain hinder the direct adoption of existing techniques that often contrast the 2D augmented data or cluster single-view features. Here we present a novel self-supervised 3D Object detection framework that seamlessly integrates the geometry-aware contrast and clustering harmonization to lift the unsupervised 3D representation learning, named GCC-3D. First, GCC-3D introduces a Geometric-Aware Contrastive objective to learn spatial-sensitive local structure representation. This objective enforces the spatially close voxels to have high feature similarity. Second, a Pseudo-Instance Clustering harmonization mechanism is proposed to encourage that different views of pseudo-instances should have consistent similarities to clustering prototype centers. This module endows our model semantic discriminative capacity. Extensive experiments demonstrate our GCC-3D achieves significant performance improvement on data-efficient 3D object detection benchmarks (nuScenes and Waymo). Moreover, our GCC-3D framework can achieve state-of-the-art performances on all popular 3D object detection benchmarks.

1. Introduction

LiDAR-based 3D object detection has been a long-standing task in visual perception systems for autonomous driving, attracting increasing industry and research attention recently due to its great advantage of high 3D localization precision and complementary to the 2D perception [43, 2, 29, 47, 33]. Different from the 2D detection problem, 3D object detectors transform sparse and unorganized point clouds into the structured 3D bounding box representations, including shape, orientation and semantic class. Almost all recent 3D object detectors are built upon fully supervised frameworks, while obtaining such large-scale and precise annotations for numerous instances in diverse self-driving scenarios is labor-intensive and time-consuming, e.g., it takes hundreds of hours to annotate just one hour of driving scene data [31]. This hinders the model improvement and deployment over ever-changing self-driving environments for LiDAR-based 3D object detection. Thus, a desirable self/unsupervised 3D object detection framework that can effortlessly lift the 3D representation learning purely using raw data is highly demanded but rarely explored.

Nonetheless, in the area of 2D image recognition [30, 20, 14] and natural language understanding [12], self-supervised pre-training over unlabeled data has yielded a significant performance boosting in downstream tasks when the labeled data is scarce. Thus, it is interesting to ask a

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure.png}
\caption{We finetune CenterPoint-pp detector from scratch, with GCC-3D pretrain or with Click-supervised pretrain and report performance on Waymo and nuScenes dataset. Our GCC-3D model show consistent significant improvements over the scratch model and learn more robust feature than Click-supervised pre-train.}
\end{figure}

\textsuperscript{*}Both authors contributed equally to this work.

\textsuperscript{†}Corresponding Author: xdliang328@gmail.com
question: Does there also exist an effective self-supervised pre-training algorithm that can significantly alleviate the heavy annotation burden in 3D object detection by fully exploiting abundant unlabeled point cloud data? Existing works mostly focus on low-level task [50, 10, 16] (e.g., registration) and single object [17, 9, 1, 21] (like reconstruction, classification and part segmentation). Recently, PointContrast [44] demonstrated that unsupervised pretraining can boost the performance on indoor scene understanding tasks. However, several limitations of PointContrast hinder its direct adoption into LiDAR-based 3D detection: 1) Static Partial Views: Multiple partial views [53] setup is claimed to be a crucial component of [44], requiring the object/scene to be static. This is usually not available in outdoor scenes of autonomous driving scenarios. 2) Inconsistent Contrast: It assigns hard labels to matched and unmatched pairs, which is contradictory to the fact that the randomly sampled unmatched pairs can have very similar structure; 3) Lack of Semantic Information: Semantic representation is important for high-level scene understanding tasks like 3D object detection. This kind of representation is not modeled during pre-training.

To advance the research on LiDAR-based 3D object detection into an unsupervised/self-supervised era and resolve the above-mentioned issues in designing a proper self-supervised scheme, we present a novel self-supervised 3D detection framework that seamlessly integrates the geometric-sensitive and semantic-consistent representations, named GCC-3D. Our framework is the first one focusing on autonomous driving scenario without static partial views setup [53]. Firstly, to alleviate the inconsistent contrast problem, GCC-3D exploits an important property of 3D data: two spatially close voxels in 3D space are very likely to have similar local geometric structures or belong to the same object. We inject this prior to our learning objective and use the geometric distance between voxels to guide feature similarity during contrastive feature learning. This geometric-aware contrastive objective can help learn local structural features of point clouds properly. The voxel-level features with geometric information will be aggregated as the embedding of pseudo instances, which are obtained from the sequential information in datasets. Secondly, we endow our model semantic property by defining a clustering harmonization phase. During training, we assign labels to each instance embedding by using K-means clustering following [41]. However, the commonly used hard labeling strategy [41] violates that some prototypes can be similar or represent the same semantic classes, and neglects the heterogeneous similarities between embeddings of pseudo-instances and is prone to "class collision" [3] problem. To alleviate this problem, we introduce a new harmonization term that encourages different pseudo-instances views to have consistent similarities with clustering prototype centers. This term is easily injected into the current self-clustering framework. By integrating the geometry-aware contrast and pseudo-instance clustering harmonization, our GCC-3D can capture both local structure and context semantic information, which can improve our model’s localization and classification capacity.

To better validate the self-supervised capability of current models in Lidar-based 3D object detection, we conduct extensive experiments in popular 3D object detection benchmarks (Waymo [39], nuScenes [5]) with limited supervised data, called data-efficient benchmarks. The methods are required to first pre-train only on the unlabeled data and then fine-tune it using limited labeled data to reduce the annotation effort. Our unsupervised framework GCC-3D can achieve consistent significant improvement over random initialized models on the data-efficient benchmarks. Notably, our pre-trained CenterPoint-voxel model achieves 67.29% mAP on Waymo (with 20% labeled data) and 57.3% mAP on nuScenes, a separate 4.1% and 1.95% relative improvement over previous state-of-the-art method [49]. After transferring our pre-trained model on Waymo to KITTI [18], we witness a 2.1% relative improvement over KITTI state-of-the-art method [36]. With 5% labeled data, our self-supervised model demonstrates over 6.3% and 5.6% relative improvement in mAP compared to PointContrast [44] on Waymo and nuScenes. Our contributions can be summarized as follows:

• We make the first attempt to propose a simple yet effective self-supervised LiDAR-based 3D object detection framework for alleviating the demand for extensive human annotations, towards a more flexible and scalable self-driving system.

• We propose a novel GCC-3D that is the first self-supervised learning mechanism to integrate both geometry-aware structure contrast and harmonized semantic pseudo-instance clustering. This method successfully self-explores and enhances the 3D instance-level representation from both the geometry and semantic perspectives.

• Our GCC-3D framework can achieve state-of-the-art performances on all popular 3D object detection benchmarks, i.e., 67.29% mAP on Waymo (20% labeled data) and 57.3% mAP on nuScenes.

2. Related Work

LiDAR-based 3D Object Detection. This task’s objective is to detect objects of interest and localize their amodal 3D bounding boxes from sparse and unorganized point clouds. Some representative works [8, 26, 46] project point clouds to bird’s view and use 2D CNNs to learn the point cloud features. Some other works [51, 38] apply 3D
Voxelization $u$

View 2

Projection $g$

3D encoder $\phi$

Projection $h$

Transformation

Voxelization

View 1

2D encoder $\phi$

Figure 2. Overview of our GCC-3D self-supervised learning framework. The first key component is the Geometric-Aware Contrast module, where voxels from different views of the same scene are passed through encoder $\phi$ and we use the geometric distance between them to guide voxel-wise feature learning via a geometric-aware contrastive objective. In the second Harmonized Instance Clustering module, we exploit sequential information to generate pseudo instances in the scenes. The pre-trained voxel features located in each instance will be aggregated as instance embedding and passed through backbone $\phi$ for semantic clustering. A harmonization term is introduced to encourage that different views of pseudo-instances should have consistent similarities to clustering prototype centers. These two modules endow our model with both geometric structure and contextual semantic representation.

Self-supervised Learning. Visual representation learning with self-supervision has drawn massive attention in 2D vision tasks for its fantastic data efficiency and generalization ability. Image-based self-supervised methods design many pretext tasks that exploit their spatial structure [13], color information [11], illumination [15], and rotation [19]. Compared to 2D vision, the limits of big data are far from being explored in 3D. Recent works attempt to adapt the 2D pretext tasks to 3D, but mostly focus on low-level tasks [50, 10, 16] or single object classification tasks [17, 9, 1, 21, 22, 28, 34, 35]. A recent contrastive-learning based method PointContrast [44] demonstrates promising results on a set of indoor scene-level understanding tasks. However, the good performance of [44] depends on partial views setup, which is usually not available in the outdoor autonomous driving scenario. The simple point-level pre-training objective in [44] is not properly designed and can assign points with similar local structure as negative pair, thus raising an obstacle for good contrastive representation learning. And it also neglects semantic information which is important for high-level 3D scene understanding tasks. So in this work, we propose a properly-designed self-supervised learning framework that captures both spatial discriminative information and semantic representation.

CNNs over point cloud voxels to generate cuboids. However, these state-of-the-art methods rely on sufficient training labels and precise 3D annotations, which cost a heavy workforce. In this work, we suggest pre-training paradigm is helpful for real-life LiDAR-based 3D object detection and further reduce the pressure of labeling through the proposed self-supervised framework.

3. Method

In this section, we elucidate our proposal of the novel self-supervised 3D detection framework GCC-3D that seamlessly integrates the geometry-aware contrast and clustering harmonization, illustrated in Fig. 2. We will first introduce the Geometric-Aware Contrast module that enhances spatial-sensitive local structure representation, then elaborate on the Pseudo-Instance Clustering harmonization mechanism, which encourages different views of pseudo-instances to be consistent.

As with the typical pipeline of LiDAR-based 3D object detection tasks, a 3D encoder $\phi$ takes the point cloud of a scene as input and estimates 3D bounding box representations for objects, including information on shapes, orientations and semantic classes. The quantized representations
are then reshaped and fed to a 2D backbone $\varphi$ to produce a feature map $F$. A task-specific head $\psi$ takes $F$ and can either be a 2D anchor-based detector or an anchor-free one.

### 3.1. Geometric-Aware Contrastive Objective

It is crucial to learn meaningful local structural features of point clouds. Yet methods adopted by previous work [44] focus on hard labeling strategy in contrastive learning, which can be ineffective. This is because, in the hard labeling strategy, both positive and negative pairs can be formed by two voxels extracted from the same type of object, which can be confusing and hinder the network from learning good representations. However, based on the observation that spatially close voxels in the 3D world are more likely to have similar local geometric structures (or belong to the same object), we can use the geometric distance among voxels as a proxy for their feature similarity.

We illustrate our learning process in Fig. 2. Given an original point cloud scene $S$, we sample a random geometric transformation $\mathcal{T}$ to transform it into an augmented view $\tilde{S}$. We mainly consider similarity transformation including rotation, translation and scaling. We then voxelize these two scenes into regular voxels and feed them into a shared 3D encoder $\phi$ to get feature maps $h$. Then we calculate the Euclidean distance between the chosen voxel $v_i$ and all sampled voxels $\tilde{v}_j$ in augmented view to get weights $w_{i,j}(v_i, \tilde{v}_j)$:

$$w_{i,j}(v_i, \tilde{v}_j) = \frac{e^{-d_{i,j}}}{\sum_{(\cdot, \tilde{v}_k) \in M} e^{-d_{i,k}}}.$$ (1)

Then the weights $w_{i,j}$ are used to calculate the similarity among voxels by minimizing the soft InfoNCE loss [32]:

$$L(v_i) = - \sum_{(\cdot, \tilde{v}_j) \in M} w_{i,j}(v_i, \tilde{v}_j) \log \frac{e^{u_{i}^\top \tilde{u}_j / \tau}}{\sum_{(\cdot, \tilde{v}_k) \in M} e^{u_{i}^\top \tilde{u}_k / \tau}}.$$ (2)

Since the global transformation brings different offsets between original voxels and their augmented counterparts, instead of taking all matched pairs as equally positive, we use the distances among all the positive pairs to calculate the match confidence $\rho_{i,j} = \frac{e^{-d_{i,j}}}{\sum_{(v_i, v_j) \in M} e^{-d_{i,j}}}$ for positive pair $(v_i, \tilde{v}_j)$. The final loss is thus:

$$L_{\text{voxel}} = \sum_{(v_i, \cdot) \in M} \rho_{i,j} L(v_i).$$ (3)

By minimizing $L_{\text{voxel}}$, our 3D encoder $\phi$ can learn the voxel-wise geometric-aware representation of local structure with equivariance to different transformations.

### 3.2. Harmonized Pseudo-Instance Clustering

For complex 3D scene understanding task like 3D object detection, simply learning voxel-level geometric feature $u$ might not promise good performance. It is also important to learn contextual semantic information for the model to detect with better robustness. Nevertheless, learning such information requires bounding boxes with exact patches of objects and ground truth semantic label of different objects, both of which unavailable in unsupervised learning settings.

To tackle this problem, we introduce a Motion Pseudo-Instance Generation component into our pipeline. It utilizes sequential information in the dataset to propose pseudo instances. The categorical labels for these instances are obtained by using K-means clustering over instance-level features and we use these labels to further pre-train our model following [41]. However, the hard labeling strategy in [41] takes all the unassigned cluster centers (prototypes) as equally negative. As is discussed previously, it violates the fact that some prototypes can be similar or represent the same semantic classes, especially considering that our number of prototypes $C$ is much larger than the actual semantic class in the scene. Hence, it neglects the heterogeneous similarities between embeddings of pseudo-instances and can lead to the “class collision” [3] problem. Therefore, we propose a Clustering Harmonization mechanism to encourage that different views of pseudo-instances should have consistent feature similarities to clustering prototype centers. As is shown in Fig. 2, we keep using a multi-view setup to learn pseudo-instance representation that is equivariant to transformation and robust to noise.

**Motion Pseudo-Instance Generation.** In the self-driving environment, the ego-vehicle’s sensor status is available at every frame (50 fps). The intuition behind our design is to use sequential information to localize patches with moving objects. Note that moving objects can be recognized from stationary objects by observing the non-overlapping area among consecutive frames. Therefore, we can analyze the occupancy voxel point information in BEV view to identify possible moving voxels and then find out connected domains between neighboring moving voxels to obtain pseudo moving patches.

Specifically, given two consecutive LiDAR frame $p$ and $q$, the rigid transformation between their coordinates can be
written as:  \( T = T_{(lidar \rightarrow ego)}T_{(ego \rightarrow lidar)} \), where we align the frame \( q \) into the coordinate system of \( p \) by \( p' = T(q) \). We quantify LiDAR points \( p \) and \( p' \) into regular voxels and compute the average coordination of points in each voxel. Voxels with an average coordination distance between \( p \) and \( p' \) larger than a predefined threshold will be considered as moving voxels. Finally, we group these voxels as pseudo instances by eight-neighbor-hooding [4].

### Clustering Harmonization.
After passing the point cloud scene \( S \) through 3D encoder \( \phi \) and 2D backbone \( \varphi \), we obtain its feature map \( F \). Given the set of pseudo-instance positions \( P \) in scene \( S \) obtained in the instance generation module, we obtain instance-level embedding \( x \) by cropping features corresponding to each pseudo instance on feature map \( F \). These embeddings are RoIAligned and projected to a latent embedding space by an MLP \( g \) to obtain instance-level feature \( x = g(RoIAlign(F, P^m)) \) \( \in \mathbb{R}^{D_2} \) for pseudo instance \( m \). These instance-level features are clustered into \( C \) distinct groups based on a geometric criterion at the end of each epoch. A \( D_2 \times C \) prototype matrix \( Z \) and the cluster assignments \( \alpha \) for each instance are then obtained. These assignments \( \alpha \) will be used as pseudo-labels for training the pseudo-instance clustering network. With the multi-view setup, we obtain augmented views of instance-level feature \( \tilde{x} \) by passing augmented scene \( \tilde{S} \) through our network using transformed instance position \( \tilde{P} \), then crop the features on the corresponding feature map \( \tilde{F} \).

To capture similarities among embeddings of pseudo-instances, we propose a harmonic term that encourages different views of pseudo-instances to be consistent with their clustering prototype centers. Specifically, given a pseudo instance \( m \) and the prototype matrix \( Z \in \mathbb{R}^{D_2 \times C} \), we calculate the similarity between instance feature \( x \) and the prototypes \( z_i (i \in \{1, \ldots, C\}) \) as:

\[
J(i) = \frac{e^{x^\top z_i}}{\sum_{k=1}^{C} e^{x^\top z_k}}, \tag{4}
\]

where \( J(i) \) is the probability that embedding \( x \) is assigned to cluster center \( i \). Similarly, the probability of assigning the augmented view of instance feature \( \tilde{x} \) to this cluster is:

\[
H(i) = \frac{e^{\tilde{x}^\top z_i}}{\sum_{k=1}^{C} e^{\tilde{x}^\top z_k}}. \tag{5}
\]

### 4. Experiments

#### Pre-training Details.
In the Geometric-Aware Contrastive module, \( K = 1024 \) and \( D_1 = 64 \). We pre-train the model for 20 epochs and use Adam optimizer with the initial learning rate 0.001. The batch size is 6 and \( \tau = 1 \). In
Table 3. Comparison with 3D detection on

<table>
<thead>
<tr>
<th>Model</th>
<th>mAP</th>
<th>Car AP/APH</th>
<th>Pedestrian AP/APH</th>
<th>Cyclist AP/APH</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECOND [45]</td>
<td>55.08</td>
<td>59.57/59.04</td>
<td>53.00/43.56</td>
<td>52.67/51.37</td>
</tr>
<tr>
<td>PART2 [37]</td>
<td>60.39</td>
<td>64.33/63.82</td>
<td>54.24/47.11</td>
<td>62.61/61.35</td>
</tr>
<tr>
<td>PV-RCNN [36]</td>
<td>59.84</td>
<td>64.99/64.38</td>
<td>53.80/45.14</td>
<td>60.72/59.18</td>
</tr>
<tr>
<td>center-point-voxel [49]</td>
<td>63.46</td>
<td>61.81/61.30</td>
<td>63.62/57.79</td>
<td>64.96/63.77</td>
</tr>
<tr>
<td>center-point-voxel 2stage [49]</td>
<td>64.63</td>
<td>64.70/64.11</td>
<td>63.26/58.46</td>
<td>65.93/64.85</td>
</tr>
<tr>
<td>GCC-3D (PV-RCNN)</td>
<td>61.30</td>
<td>66.65/65.10</td>
<td>55.54/48.02</td>
<td>62.72/61.43</td>
</tr>
<tr>
<td>GCC-3D (center-point-voxel)</td>
<td>65.29</td>
<td>63.97/63.47</td>
<td>64.23/58.47</td>
<td>67.68/66.44</td>
</tr>
<tr>
<td>GCC-3D (center-point-voxel 2stage)</td>
<td>67.29</td>
<td>66.45/65.93</td>
<td>66.82/61.47</td>
<td>68.61/67.46</td>
</tr>
</tbody>
</table>

Table 4. Transfer pre-trained weights from dataset A (column) to the whole set B (row). We use CenterPoint-voxel for nuScenes and Waymo, and PV-RCNN for KITTI. We show results are evaluated on moderate difficulty mAP for KITTI, NDS for nuScenes and mAP under L2 difficulty case for Waymo.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waymo</td>
<td>WYSIWYG [25]</td>
<td>35.0</td>
</tr>
<tr>
<td></td>
<td>3DSSD [48]</td>
<td>42.6</td>
</tr>
<tr>
<td></td>
<td>HotSpotNet [7]</td>
<td>50.6</td>
</tr>
<tr>
<td></td>
<td>CBGS [52]</td>
<td>50.6</td>
</tr>
<tr>
<td>NuScenes</td>
<td>center-point-pp [49]</td>
<td>49.6</td>
</tr>
<tr>
<td></td>
<td>center-point-voxel [49]</td>
<td>56.2</td>
</tr>
<tr>
<td>All</td>
<td>50.8 (+1.2)</td>
<td>60.8 (+0.6)</td>
</tr>
<tr>
<td></td>
<td>57.3 (+3.1)</td>
<td>65.0 (+0.5)</td>
</tr>
</tbody>
</table>

4.1. Data-Efficient 3D Object Detection Benchmark

To formally explore data-efficient scene understanding in autonomous driving, we propose a 3D object detection benchmark with limited bounding box annotations. Specifically, for each dataset, only a limited fraction of scenes will be labeled and we consider the configurations including {0.05, 0.1, 0.2, 0.5} (1 represents the entire train set). We pre-train our model as initialization for fine-tune and compare against the baseline of train from scratch. The train-pre-train our model as initialization for fine-tuning stage follow Harmonized Pseudo-Instance Clustering module, we pre-train for 20 epochs with Adam optimizer. The initial learning rate is 0.0048 with a cosine decay. The prototype number C is 100, D_2 is 128 and α is 0.1. All experiments run on 8 NVIDIA V100 GPUs. We use VoxelNet and PointPillars in CenterPoint network [49] as backbones, denoted as CenterPoint-pp and CenterPoint-voxel separately. We conduct experiments on two most popular self-driving datasets: Waymo Open Dataset [39] and nuScenes Dataset [5].

4.2. Comparison with SOTA

We compare our method with other state-of-the-art models on LiDAR-based 3D Object Detection in Table 3. For
Figure 4. (a) Ablation studies on hyper-parameter $\alpha$ for our GCC-3D method. (*) indicates default value. (b) The recall and center distance curve between generated instance and groundtruth.

task-specific head of PV-RCNN [36] and follow training setups in PCDet [40]. When finetuned on nuScenes, each scene only uses one sweep of point cloud scene. More experiment setups can be found in the appendix.

Table 4 shows our GCC-3D reaches consistent improvement across different datasets than random initialization. We observe that 1) Leveraging pre-trained weights on large scale dataset then finetune on small datasets can bring more significant performance improvements; when pretrained on nuScenes and Waymo then finetuned on KITTI, we see +0.98% mAP and +1.49% mAP improvement respectively; 2) Model pre-trained on nuScenes and fine-tuned on itself show bigger improvement over baseline than model pretrained on Waymo (+0.14% vs. +0.10%). We conjecture this dilution of pretraining effectiveness arises from domain gaps between different point clouds (point clouds of Waymo are much denser than that of nuScenes).

4.4. Ablation Study

In this section, we conduct ablation study experiments to analyze the effectiveness of different modules and hyperparameters of GCC-3D.

Effect of Geometry-Aware Contrast and Harmonized Pseudo-instance Clustering. We pre-train separately with Geometry-aware Contrast module and Harmonized pseudo-instance Clustering module following the same pre-train setup as in GCC-3D. Then we fine-tune the models on Waymo and nuScenes datasets with limited annotations (5%) and evaluate on the full validation set. Results are in Table 5. While the Harmonized Pseudo-instance Clustering module achieves reasonable pre-train performance, the Geometry-Aware Contrast module boosts the fine-tune result more significantly, with +6.77% mAP and +2.59% mAP than baseline on nuScenes and Waymo separately. Meanwhile, combining these two modules improves the performance to 32.75% mAP on nuScenes and 52.92% mAP on Waymo. It demonstrates that our method of integrating geometry structure and semantic context representation helps the high-level 3D Object detection task.

### Effective of Hyper-parameters.

We study the effect of hyper-parameter introduced in GCC-3D: the coefficient of harmonization term $\alpha$. In Fig. 4, we show the effect of $\alpha$ on nuScenes with 100% annotation. It is finetuned with the CenterPoint-pp model for 20 epochs. The best coefficient is 0.1. We see that by using the harmonization term, the objective accuracy can be boosted from 50.48% mAP to 50.84% mAP. Increasing $\alpha$ to 0.5 and beyond can hurt the performance. We conjecture that this is because when $\alpha \geq 0.5$, the model is over-regularized by the harmonization term and loses some discrimination among categories.

4.5. Exploration on Click-supervised Pretraining

To alleviate the annotation burden in 3D object detection, some efforts click the object center to provide localization supervision for this task [42, 27, 31]. Inspired by these works, we propose a simple supervised pretraining baseline pre-trained with centers of 3D objects of full dataset and fine-tuned on a limited scale of fully-annotated data.

We compare the performance of our GCC-3D with the Click-supervision pre-train on 3D detection task based on CenterPoint-pp. The results are in Fig. 1. Although the Click-supervision method achieves good performance with a highly limited fraction of data, our GCC-3D method show better performance when we increase the finetuned data fraction. We believe it is because our method can learn more robust features than click-annotation pre-train. The latter enforces the network to focus on localization regression task, but our method can learn more robust embedding suitable for 3D object detection task, which requires both localization, classification and rotation representation.

4.6. Comparison with Other SSL Methods

GCC-3D is the first self-supervised pre-training framework tailored for LiDAR-based 3D Object Detection. Nevertheless, we re-implement and adopt previously published self-supervised learning models, including contrastive method (PointContrast [44]) and clustering-based methods (DeepCluster2 [41] and SwAV [6]) on 3D object detection tasks. These methods are actually closely related
to our framework. The PointContrast method can be implemented by revising the voxel-level contrastive learning in our Geometry-aware Contrast module with hard labeling. During pre-train, it is optimized following the same setup as in GCC-3D. For Deepcluster v2 and SwAV, instead of applying these cluster-based learning strategies on images, we use them on our proposed pseudo-instances. In Harmonized pseudo-instance Clustering module, getting rid of the harmonization term and choosing $\alpha$ to be 0 reduce to Deepcluster v2. SwAV is an adaptation of Deepcluster v2 with online clustering. During pre-train, their hyperparameters and optimization setup follow the same setting as in GCC-3D.

We compare with these self-supervised pretraining methods on Waymo and nuScenes dataset with finetuning on limited annotation (5%) based on CenterPoint-pp in Table 5. Our pre-training method outperforms all these pre-training strategies, achieving 32.75% mAP on nuScenes and 52.92% mAP on Waymo with only 5% annotations. Notably, the consistent improvement of Geometry-Aware Contrast over PointContrast (32.56% mAP vs. 30.79% mAP on nuScenes and 51.89% mAP vs. 50.10% mAP on Waymo) demonstrates the effectiveness of our geometry-aware design – using distance to guide feature similarity can alleviate inconsistent hard labeling problem in PointContrast. Harmonization Clustering produces better performance than Deepcluster v2 and SwAV, which proves the importance of our proposed harmonization term.

4.7. How Pre-training Affects 3D Object Detection

Speed Up Convergence with Better Performance. To investigate whether the advantage of pre-training recedes when we prolong the fine-tuning stage, we compare the train from scratch baseline with fine-tuned model initialized with Click-supervised pre-training in Fig. 6. The experiments are conducted on nuScenes dataset with 2.5% and 5% labeled data during fine-tune and 20 times bigger scale dataset during pre-train. We train until both models converge and find that the model with pre-training consistently outperforms the baseline. This observation is not consistent with [23], which claims that train-from-scratch baselines are no worse than their pre-trained counterparts. Moreover, it is observed that pretraining speeds up convergence by four times compared with baseline (20 epochs vs. 80 epochs). These observations motivate using pre-training to reduce resource consumption for real-world application.

Consistent Improvement with Large Finetune Scale by Unsupervised Pre-train. It is observed in some pre-train methods that as the fine-tuned data scale grows, the benefits of pre-training will decrease. We see a similar phenomenon on Click-supervised pre-train. However, in our 3D detection baseline, we can observe a consistent improvement of the pre-trained model over the baseline. We hypothesize that our unsupervised pre-train objective can learn more robust features and is less likely to overfit to specific tasks.

Alleviate False Positive. To compare the performance of different initializations, including train-from-scratch (tfs), supervised pretraining, and GCC-3D pretraining, we consider the Precision-Recall (PR) curve on Waymo dataset, which plots precision against the recall at different thresholds. Fig. 5 shows that pre-trained models outperform tfs at the same recall level, especially on far range detection. Tfs fails on recall > 0.7 while GCC-3D can help the model make a more precise prediction.

5. Conclusion

In this work, we focus on data-efficient LiDAR-based 3D object detection through a novel self-supervised framework that integrates Geometry-Aware Contrast and Harmonized Pseudo-Instance Clustering. It can capture spatial-sensitive representation and high-level context information. We show the effectiveness of pre-train and hope these findings can drive more research on unsupervised 3D representation learning and 3D scene understanding in the future.
References


3301


