



Trail-Det: Transformation-Invariant Local Feature Networks for 3D LiDAR Object Detection with Unsupervised Pre-Training

Li Li¹ 

li.li4@durham.ac.uk

Tanqiu Qiao¹ 

tanqiu.qiao@durham.ac.uk

Hubert P. H. Shum¹ 

hubert.shum@durham.ac.uk

Toby P. Breckon^{1,2} 

toby.breckon@durham.ac.uk

¹ Department of Computer Science
Durham University
Durham, UK

² Department of Engineering
Durham University
Durham, UK

Abstract

3D point clouds are essential for perceiving outdoor scenes, especially within the realm of autonomous driving. Recent advances in 3D LiDAR Object Detection focus primarily on the spatial positioning and distribution of points to ensure accurate detection. However, despite their robust performance in variable conditions, these methods are hindered by their sole reliance on coordinates and point intensity, resulting in inadequate isometric invariance and suboptimal detection outcomes. To tackle this challenge, our work introduces **Transformation-Invariant Local (TraIL)** features and the associated TraIL-Det architecture. Our TraIL features exhibit rigid transformation invariance and effectively adapt to variations in point density, with a design focus on capturing the localized geometry of neighboring structures. They utilize the inherent isotropic radiation of LiDAR to enhance local representation, improve computational efficiency, and boost detection performance. To effectively process the geometric relations among points within each proposal, we propose a Multi-head self-Attention Encoder (MAE) with asymmetric geometric features to encode high-dimensional TraIL features into manageable representations. Our method outperforms contemporary self-supervised 3D object detection approaches in terms of mAP on KITTI (**67.8**, 20% label, moderate) and Waymo (**68.9**, 20% label, moderate) datasets under various label ratios (20%, 50%, and 100%).

1 Introduction

LiDAR-based point clouds, comprising 3D positions and LiDAR intensity/reflectivity [3, 10, 20, 21, 36], are essential for interpreting outdoor environments, particularly in the context of autonomous vehicle perception systems. The realm of 3D object detection has seen significant progress, with a variety of strategies aimed at predicting 3D bounding boxes. Recent approaches have employed color information [6, 26, 44], range imagery [29, 30], and

Birds Eye View (BEV) projections [6, 26] to devise multi-modal techniques that merge inputs from LiDAR and other sensors to improve feature representation and detection accuracy.

Despite these advancements, these methods commonly face challenges with isometric invariance due to their reliance primarily on coordinates and intensity data, often leading to suboptimal detection results [44, 48, 57]. These limitations are predominantly caused by inadequate translational invariance and issues such as occlusions or sparse observations at longer ranges [18], which adversely affect the spatial distribution of the data.

In this work, we aim to identify features that (1) capture the localized geometric structure of neighboring points, (2) are invariant to rotation and translation, and (3) function effectively in noisy LiDAR outdoor scenes. Although various methods meet some of these criteria individually [15, 25, 28], they often do not satisfy all simultaneously. Given the necessity for higher-level features that can encapsulate local geometry and potentially include LiDAR-specific attributes such as intensity and reflectivity, we focus on Pointwise Distance Distribution (PDD) features [41, 42]. PDD features are noted for their exceptional ability to provide robust and detailed geometric representations of point clouds, effectively maintaining both rotational and translational invariance while capturing intricate details of local geometry.

However, the direct use of PDD features is impractical due to their high dimensionality and the substantial memory and storage demands they impose on large-scale point clouds [22]. Additionally, PDD tends to overlook local features because the inclusion of distant points can diminish the emphasis on nearby neighborhoods.

To facilitate 3D object detection, we propose the **Transformation-Invariant Local** (TraIL) features and the associated TraIL-Det network. It leverages the robustness of TraIL features against rigid transformations and variability in point cloud density, focusing on extracting compact features within defined local neighborhoods. Our approach employs inherent LiDAR isotropic radiation and multi-head self-attention to improve the representation of local features while reducing computational overhead. To effectively handle the high dimensionality of TraIL features, we introduce a novel embedding method within our TraIL Proposal Multi-head self-Attention Encoding (TraIL MAE) module. Additionally, we enhance ability of the model to precisely localize individual objects and accurately identify different object categories through a joint optimization of discrimination and separation. This integration into the overall network is designed to elevate performance and expand generalization capabilities.

We conduct extensive experiments on KITTI [10] and Waymo [36] datasets, where our methodology outperforms existing state-of-the-art (SoTA) self-supervised methods in 3D object detection. Overall, our contributions are summarized as follows:

- A novel **Transformation-Invariant Local (TraIL)** feature for 3D object detection that ensures robustness to rigid transformations through isometry-invariant metrics.
- A novel method for **embedding TraIL with Multi-head self-Attention Encoder (MAE)** to capture the geometric relations between points, jointly attending to information from different representation subspaces at different positions.
- A novel **open-source pre-training architecture TraIL-Det**¹ and supporting training methodology for 3D object detection that outperform the recent contemporary approaches of ProposalContrast [52], DepthContrast [55], and PointContrast [46].

¹The code is publicly available at: https://github.com/l1997i/rapid_seg.

2 Related Work

3D LiDAR Object Detection: Initial approaches in the field convert LiDAR point clouds into 2D formats, specifically Bird Eye View (BEV) or range-view images [1, 6] to facilitate 3D object detection. More recent advancements have shifted towards using voxel-based sparse convolution techniques [9, 11, 17, 47] and point-based methodologies for set abstraction [32, 33, 49, 50] to create more effective detection frameworks. A common challenge with LiDAR data is its low resolution for objects at a distance, leading to sparse detection outcomes. To mitigate this, researchers have delved into multimodal 3D object detection, demonstrating that the integration of LiDAR with RGB image data improves detection performance. Initial strategies enrich LiDAR points with image data [35, 38, 39], while others pursue independent encoding of multimodal features, followed by their fusion either within the local Region of Interest (RoI) [7, 16] or on the BEV plane [27]. Recent advancements utilize virtual points for feature fusion [45, 54], which effectively improve the geometric representation of distant objects through depth estimation, showcasing significant promise for elevated detection performance. Nonetheless, virtual points introduce challenges related to their density and noise levels. VirConv [44] integrates RGB image data through virtual points and introduces StVD and NRConv as effective solutions to address the related challenges.

Pointwise Distance Distribution: Pointwise Distance Distribution (PDD) quantifies the local context of each point within a unit cell by sequentially measuring distances to nearby points. This isometry-invariant technique, developed by Widdowson & Kurlin [41], effectively addresses data ambiguity in periodic crystals, as demonstrated through detailed pairwise comparisons of atomic 3D clouds from structured periodic environments [41, 42, 43]. Despite its proven effectiveness in periodic crystals and atomic clouds, PDD has not yet been applied to outdoor 3D point clouds. In outdoor scenarios, commonly used invariant features [15, 23, 25, 28] often struggle with irregular and sparse data, compounded by increased noise and environmental complexity [23, 25]. Additionally, the computational intensity of these features limits their feasibility for large-scale outdoor applications [15, 28]. For instance, Melia *et al.* [28] report a rotation-invariant feature that has difficulty scaling across diverse point cloud densities and sizes due to its computational demand and vulnerability to outdoor interferences. Recognizing these limitations, we propose the exploration of PDD features in outdoor settings, where accurately representing the local context of points in a transformation-invariant and structurally sound manner is paramount. Leveraging the architectural advantages of PDD, we introduce the Trail feature, specifically designed for LiDAR-based point clouds to adeptly capture the local geometric configuration of neighboring structures.

Self-Supervised Learning Methods for Point Cloud: Self-supervised Learning (SSL) [5, 12, 13, 40, 51] have demonstrated exceptional performance on various tasks, at times outperforming supervised methods. This work presents a proposal-level pretraining approach specifically designed for point cloud object detection. Simultaneously, studies like PointContrast [46], DepthContrast [55], ProposalContrast [52], GCC-3D [24], and STRL [14] explore the utility of contrastive SSL in point cloud pretraining. However, these methods encounter several issues. Firstly, some [14, 55] treat the entire point cloud scene as a single instance, overlooking the multiple object instances typically present [4, 56]. Secondly, methods like [24, 46] focus on point-/voxel-level discrimination, which hinders the development of object-level representations crucial for 3D object detection. Thirdly, several approaches [14, 46, 55] ignore the semantic relationships between instances, concentrating instead on low-level details rather than more informative high-level patterns. While an additional self-clustering strategy is implemented to capture semantic features [24], it supervises only moving voxels, which are

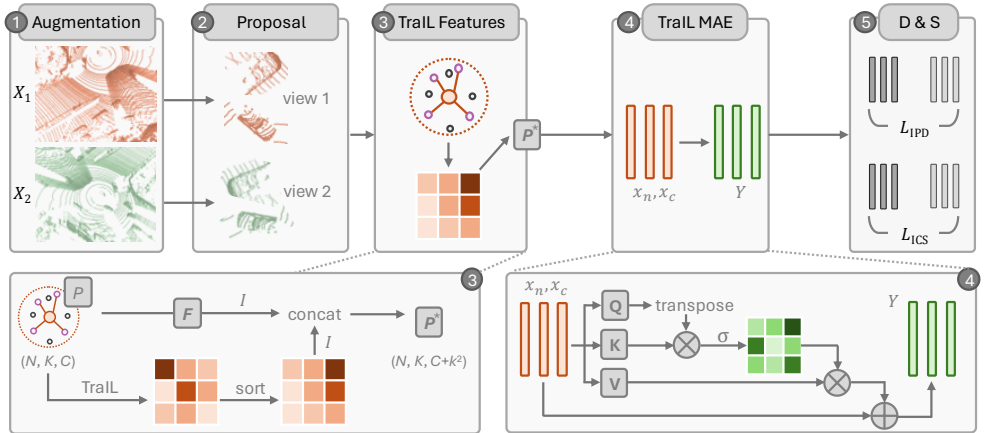


Figure 1: Our proposed **TraIL architecture for 3D object detection** leverages TraIL features from the point cloud. **1** We take point cloud inputs as input and augment them with differing views. **2** The augmented point clouds are sampled to the initial paired region proposals. **3** The encoding module (TraIL MAE) extracts expressive proposal representations by considering the geometric relations among points within each proposal. **4** We extract the concatenated features with the Multi-Head Attention Encoding Module (TraIL MAE). **5** Inter-Proposal Discrimination (IPD) and Inter-Cluster Separation (ICS), *i.e.* D&S module [52] are subsequently enforced to optimize the whole network.

too sparse to encompass all potential object candidates, resulting in a cumbersome two-stage training pipeline for separate 3D and 2D encoders [24].

3 TraIL-Det for 3D Object Detection Pre-Training

As shown in Fig. 1, we propose TraIL-Det architecture for 3D object detection which leverages TraIL features from the 3D point cloud. The proposed architecture generates the TraIL-based proposals from different augmented views (Sec. 3.1). The TraIL-based proposals are further embedded from TraIL MAE (Sec. 3.2) to process the geometric relations among points within each proposal.

3.1 Transformation-Invariant Local Feature (TraIL) Overview

As illustrated in Fig. 1 **1****2****3**, we present an overview of our TraIL features in terms of their geometric descriptor (Sec. 3.1.1) and the augmented TraIL proposal (Sec. 3.1.2).

3.1.1 TraIL Descriptor

Spatial transformations, including translation and rotation, are ubiquitous in real-world scenes, necessitating rotation-invariant representation of 3D point clouds. While translation-invariance can be achieved through weight sharing in 2D image understanding, 3D rotation-invariance remains a challenge due to the complexity of 3D geometry.

To achieve a 3D **T**ransformation-**I**nvariant **L**ocal (TraIL) representation, we consider a point cloud patch $X \in \mathbb{R}^{3 \times K}$ with K points, where each point x_i ($i = 1, \dots, K$) represents the 3D coordinates in Euclidean space. We define a transformation-invariant mapping TraIL :

$\mathbb{R}^{3 \times K} \rightarrow \mathbb{R}^{C \times K}$, where $C \in \mathbb{N}^+$, to yield a consistent descriptor for geometrically identical point clouds under different orientations. The mapping should satisfy Eq. (1):

$$\text{TraIL}(X) = \text{TraIL}(RX + T), \quad (1)$$

where $\text{TraIL}(\cdot)$ denotes the translation-invariant operation, and $\text{TraIL}(X)$ is the invariant descriptor of X .

The difficulty in achieving 3D rotation-invariance has led us to seek alternative approaches that can effectively capture the intrinsic properties of 3D point clouds. Inspired by the concept of isometric invariance, which states that the properties of an object remain unchanged under rigid transformations, we hypothesize that features invariant to translation and rotation can be extracted from the geometric structure of point clouds. This idea is motivated by the fact that the distances between adjacent points in a point cloud remain constant regardless of the object orientation or position in 3D space. Therefore, we propose to exploit the Point Distance Distribution (PDD) [41, 42] as a translation-invariant feature, which measures the distribution of distances between adjacent points in a point cloud.

PDD is defined for a point cloud patch X with K points where $K > k$ and k is the count of the nearest neighbours of a point, forming an $K \times k$ matrix $\text{PDD}(X; k)$. Each row i of this matrix includes the ordered distances from the i -th point in X to its k nearest neighbours. Although the points in X and rows of $\text{PDD}(X; k)$ are unordered, they are stored in lexicographic order to keep points and PDD matrix permutation-invariant (refer to the *Supplementary Materials* for more details on computing PDD of point cloud and the corresponding *sort* method).

Considering PDD properties under rotations R and translations T , the distances between points in a point cloud remain unchanged. Subsequently, for a transformed cloud X expressed as $RX + T$, the internal distances between points in X and $RX + T$ are identical, preserving the Euclidean distance invariance under rotation and translation. Consequently, the ordered distances from any point in X to its k nearest neighbours remain the same in both X and $RX + T$, which leads to Eq. (2):

$$\text{TraIL}(X) = \text{TraIL}(RX + T) = \text{PDD}(X; k), \quad (2)$$

where $\text{TraIL}(X)$ effectively captures the invariant spatial relationships within the cloud. Thus, $\text{PDD}(X; k)$ is a suitable candidate for $\text{TraIL}(X)$, fulfilling the requirements for a transformation-invariant 3D data representation regardless of point cloud orientation or position in space.

3.1.2 TraIL Proposal

In 2D representation learning, some SSL methods utilize image proposals delineated by 2D bounding boxes. However, directly applying 3D bounding boxes to represent proposals in point clouds is impractical due to the vast candidate space and high computational cost of 3D spatial operations. Instead, we opt for spherical proposals.

In Fig. 1, starting with the initial point cloud X_0 , we remove road plane points to minimize background sampling [2]. We then apply farthest point sampling (FPS) [31] to select N distinct points from X_0 , which serve as the centers for N spherical proposals. Each proposal is formed by gathering K nearby points within a predetermined radius r , ensuring proposal diversity. It results in two sets of spherical proposals P_1 and P_2 , derived from two augmented views of X_0 , represented as $P_1 \in X_1$ and $P_2 \in X_2$. We compute the TraIL features U_1 and U_2 inside P_1 and P_2 , i.e., $U_1 = \text{TraIL}(P_1)$, and $U_2 = \text{TraIL}(P_2)$. Since P_1 and P_2 may contain different numbers of points, the resulting TraIL matrices U_1 and U_2 may also vary in size. Following the approach used for image matrices, we scale U_1 and U_2 to a predefined fixed

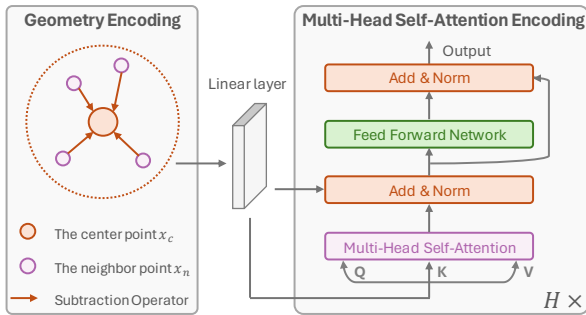


Figure 2: **Multi-attention geometric encoding.** The asymmetric geometric features are computed from the proposal P^* , specifically the center and neighbor points, through a subtraction operator. The geometric features are further refined by a proposal-aware encoding module that utilizes a multi-head self-attention mechanism.

size using bilinear interpolation $I(\cdot)$, making them suitable for input into subsequent neural networks.

3.2 TraIL Multi-Head Self-Attention Encoding (TraIL MAE)

For the point cloud scene X and corresponding proposals P , we first derive a global scene-wise representation using a backbone network, *e.g.*, VoxelNet [57] or PointNet++ [31], denoted as $\mathbf{F} = f_{\text{Bbone}}(X)$. Initial representations for the proposals $P^* \in \mathbb{R}^{N \times K \times C}$, are obtained by applying a bilinear interpolation function $I(\cdot)$ over \mathbf{F} , formulated as $P^* = I(P, \mathbf{F}) \oplus I(P, \mathbf{U})$, where \oplus is the concatenate operator, N is the number of proposals per view, K is the number of points within a proposal, and C is the channel number from the backbone network.

As shown in Fig. 1 ④, we employ the multi-head attention mechanism to process the geometric relations among points within each proposal. For each proposal $\mathbf{p} \in P^*$, with the size of $K \times C$, we designate the center point feature $\mathbf{x}_c \in \mathbb{R}^{1 \times C}$ of the proposal \mathbf{p} as the query, recognizing its informativeness. Neighbor features $\mathbf{x}_n \in \mathbb{R}^{K \times C}$, derived from \mathbf{p} , serve as keys, with their differences to \mathbf{x}_c encoding the asymmetric geometric relations. Mathematically, the \mathbf{x}_c and \mathbf{x}_n are projected to query \mathbf{Q} , key \mathbf{K} , and value \mathbf{V} embeddings:

$$\mathbf{Q} = \delta(\mathbf{x}_c), \quad \mathbf{K} = \theta(\mathbf{x}_n - \mathbf{x}_c), \quad \mathbf{V} = \gamma(\mathbf{x}_n - \mathbf{x}_c), \quad (3)$$

where δ , θ , and γ represent the linear transformations.

The embeddings \mathbf{Q} , \mathbf{K} , and \mathbf{V} are then processed by multi-head self-attention mechanism. In a H -head attention situation, \mathbf{Q} , \mathbf{K} , and \mathbf{V} are further divided into $\mathbf{Q} = [\mathbf{Q}_1, \dots, \mathbf{Q}_H]$, $\mathbf{K} = [\mathbf{K}_1, \dots, \mathbf{K}_H]$, and $\mathbf{V} = [\mathbf{V}_1, \dots, \mathbf{V}_H]$. For each h ranging from 1 to H , $\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h \in \mathbb{R}^{N \times D'}$ with $D' = D/H$. The output of the multi-head self-attention is computed as follows:

$$S^{(\text{att})}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left(\frac{\mathbf{Q}_h \mathbf{K}_h^\top}{\sqrt{D'}} \right) \cdot \mathbf{V}. \quad (4)$$

As shown in Fig. 2, a simple Feed-Forward Network (FFN) and residual operator are then adopted to obtain proposal representations as:

$$\mathbf{Y} = S^{(\text{emb})}(\mathbf{x}_c, \mathbf{x}_n) = \mathcal{Z} \left(\mathcal{N} \left(\mathcal{Z} \left(S^{(\text{att})}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \right) \right) \right) \quad (5)$$

where $\mathcal{Z}(\cdot)$ denotes add and normalization operator, $\mathcal{N}(\cdot)$ denotes a FFN with $2 \times$ linear layers and $1 \times$ ReLU activation. We observe that a stack of 3 identical self-attention encoding modules (*i.e.*, $H = 3$) is ideal for our TraIL-Det framework.

By implementing these operations across each region proposal in X_1 and X_2 , we obtain refined proposal representations, $\mathbf{Y}_1, \mathbf{Y}_2 \in \mathbb{R}^{N \times C}$, which are further optimized through joint training for both inter-proposal discrimination and inter-cluster separation. This integration of multi-head attention not only enriches the geometric understanding of the proposals but also enhances the model capacity to represent complex spatial dependencies effectively.

We follow ProposalContrast [52], taking \mathbf{Y}_1 and \mathbf{Y}_2 to the D&S module (Fig. 1 ⑤), to optimize with inter-proposal discrimination (IPD) and inter-cluster separation (ICS) simultaneously in a self-supervised manner for better overall discriminative and classificatory capabilities. The hyper-parameter settings (*e.g.*, α , β , and τ) for the D&S module are consistent with those used in ProposalContrast [52].

4 Evaluation

We follow the standard SSL experimental framework, which involves pretraining a backbone network on extensive unlabeled data and subsequently fine-tuning this pretrained model on downstream tasks using a smaller set of labeled data. Unlike some previous 3D SSL approaches that utilize the ShapeNet [19] and ScanNet [8] datasets for pretraining—thereby concentrating exclusively on indoor environments and encountering significant domain gaps when applied to self-driving scenarios—we employ a different strategy to mitigate this limitation.

4.1 Experimental Setup

Datasets: We evaluate the transferability of our pre-trained model by pre-training on Waymo Open Dataset (WOD) [36] then fine-tuning on KITTI dataset [10]. WOD comprises 798 training scenes (158,361 frames) and 202 validation scenes (40,077 frames), which is about $20\times$ larger than KITTI. We leverage the entire WOD training set for pretraining various 3D backbone architectures, explicitly avoiding the use of labels. KITTI contains 7,481 labeled samples, which are divided into two groups, *i.e.* a training set (3,712 samples) and a validation set (3,769 samples).

Evaluation Protocol: Mean Average Precision (mAP) and Mean Average Precision weighted by Heading (mAPH) with 40 recall positions (R_{40}) are employed to evaluate detection performance. We report results on the two difficulty levels and 3 classes, with 3D Intersection over Union (IoU) thresholds set at 0.7 for cars and 0.5 for pedestrians and cyclists.

Implementation Details: We consider four types of widely-used data augmentations to generate different views, *i.e.*, random rotation ($[-\pi, +\pi]$), random scaling ($[0.5, 1.5]$), random flipping (X -axis, Y -axis), and point-wise random drop out. All experiments are conducted on $4\times$ NVIDIA A100 GPUs ($1\times$ for inference). Except for the parameters mentioned in Sec. 4.3, we follow the configurations from ProposalContrast [52] to facilitate comparison with SoTA approaches.

4.2 Experimental Results of Transfer Learning

We explore the effectiveness of self-supervised pre-training within the context of autonomous driving. We evaluate our methodology on multiple popular LiDAR point cloud datasets for autonomous driving, *i.e.*, KITTI [10] and WOD [36]. Our evaluation involves a comparative analysis of our TraIL-Det against SoTA pre-training strategies [46, 52, 55] by fine-tuning detection models on these datasets. We utilize varying amounts of labeled data for fine-tuning to demonstrate the data efficiency of our approach. We employ a range of contemporary 3D object detectors as well to illustrate the broad applicability and generalizability of our pre-trained models.

Table 1: **Data-efficient 3D Object Detection on KITTI.** We pre-train the backbones of PointRCNN [32] and PV-RCNN [33] on Waymo and transfer to KITTI 3D object detection with different label configurations. Consistent improvements are obtained under each setting. Our approach outperforms all the concurrent self-supervised learning methods, *i.e.*, DepthContrast [55], PointContrast [46], ProposalContrast [52], GCC-3D [24], and STRL [14].

Fine-tuning with various label ratios	Detector	Pre-train. Schedule	mAP (Mod.)	Car			Pedestrian			Cyclist		
				Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
20% (~ 0.7k frames)	PointRCNN	Scratch	63.51	88.64	75.23	72.47	55.49	48.90	42.23	85.41	66.39	61.74
		Prop.Con. [52]	66.20	88.52	77.02	72.56	58.66	51.90	44.98	90.27	69.67	65.05
	★ Ours	67.80	89.07	78.86	73.63	59.12	53.37	46.11	92.95	71.16	66.12	
	PV-RCNN	Scratch	66.71	91.81	82.52	80.11	58.78	53.33	47.61	86.74	64.28	59.53
Prop.Con. [52]		68.13	91.96	82.65	80.15	62.58	55.05	50.06	88.58	66.68	62.32	
50% (~ 1.8k frames)	PointRCNN	Scratch	66.73	89.12	77.85	75.36	61.82	54.58	47.90	86.30	67.76	63.26
		Prop.Con. [52]	69.23	89.32	79.97	77.39	62.19	54.47	46.49	92.26	73.25	68.51
	★ Ours	69.77	90.47	81.23	76.82	64.15	54.79	47.28	91.16	73.29	71.13	
	PV-RCNN	Scratch	69.63	91.77	82.68	81.90	63.70	57.10	52.77	89.77	69.12	64.61
Prop.Con. [52]		71.76	92.29	82.92	82.09	65.82	59.92	55.06	91.87	72.45	67.53	
100% (~ 3.7k frames)	PointRCNN	Scratch	69.45	90.02	80.56	78.02	62.59	55.66	48.69	89.87	72.12	67.52
		DepthCon. [55]	70.26	89.38	80.32	77.92	65.55	57.62	50.98	90.52	72.84	68.22
	Prop.Con. [52]	70.71	89.51	80.23	77.96	66.15	58.82	52.00	91.28	73.08	68.45	
	★ Ours	71.41	90.82	81.95	77.85	66.28	58.73	53.96	92.41	73.55	71.53	
PV-RCNN	Scratch	70.57	-	84.50	-	-	57.06	-	-	70.14	-	
	GCC-3D [24]	71.26	-	-	-	-	-	-	-	-	-	
	STRL [14]	71.46	-	84.70	-	-	57.80	-	-	71.88	-	
	PointCon. [46]	71.55	91.40	84.18	82.25	65.73	57.74	52.46	91.47	72.72	67.95	
★ Ours	Prop.Con. [52]	72.92	92.45	84.72	82.47	68.43	60.36	55.01	92.77	73.69	69.51	
	★ Ours	73.89	92.10	85.39	84.12	68.01	61.25	54.29	93.46	75.04	72.49	

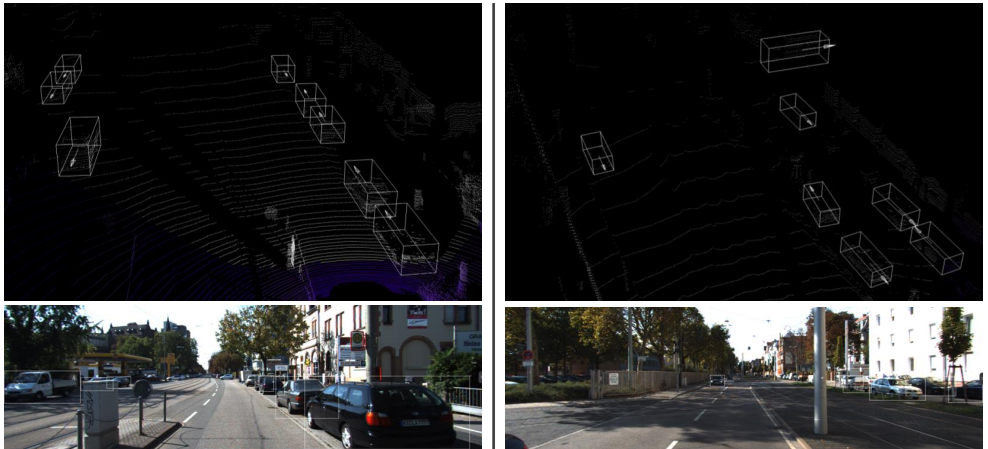


Figure 3: The qualitative results of 3D object detection with our Trail-Det on the KITTI dataset. The predicted 3D bounding boxes are marked within the point cloud frame, while the corresponding 2D bounding boxes are highlighted in the RGB images. In the point cloud visualization, white points represent those within the camera field of view (FOV), whereas purple points indicate those outside the camera FOV. Best viewed in color.

Table 2: **Comparisons between our model and other self-supervised learning methods on WOD.** All the detectors are trained by 20% training samples following the OpenPCDet [37] configuration and evaluated on the validation set. Both PV-RCNN [33] and CenterPoint [53] are used as baseline detectors.

3D Object Detector	Transfer Paradigm	Overall AP/APH	Vehicle AP/APH	Pedestrian AP/APH	Cyclist AP/APH
SECOND [47]	Scratch	59.00/54.97	63.81/63.24	56.77/46.66	56.42/55.02
Part-A ² -Anchor [34]	Scratch	64.81/61.63	69.04/68.49	58.21/50.56	67.19/65.84
PV-RCNN [33]	Scratch	60.88/57.20	66.12/65.50	54.73/45.92	61.77/60.20
+ GCC-3D [24]	Fine-tuning	61.30/58.18 _(+0.42/+0.98)	65.65/65.10	55.54/48.02	62.72/61.43
+ Prop.Con. [52]	Fine-tuning	62.62/59.28 _(+1.74/+2.08)	66.04/65.47	57.58/49.51	64.23/62.86
+ TraIL-Det (★ Ours)	Fine-tuning	64.16/60.62 _(+3.28/+3.42)	67.26/66.73	59.31/51.13	65.90/64.01
CenterPoint [53]	Scratch	64.56/62.01	62.88/62.36	64.72/58.79	66.09/64.87
+ GCC-3D [24]	Fine-tuning	65.29/62.79 _(+0.73/+0.78)	63.97/63.47	64.23/58.47	67.68/66.44
+ Prop.Con. [52]	Fine-tuning	66.42/63.85 _(+1.86/+1.84)	64.94/64.42	66.13/60.11	68.19/67.01
+ TraIL-Det (★ Ours)	Fine-tuning	66.55/ 63.94 _(+1.99/+1.93)	65.72/65.20	65.49/59.53	68.43/67.09
CenterPoint-Stage2 [53]	Scratch	66.41/63.54	65.81/65.21	64.34/59.46	67.06/65.96
+ GCC-3D [24]	Fine-tuning	67.29/64.95 _(+0.88/+1.41)	66.45/65.93	66.82/61.47	68.61/67.46
+ Prop.Con. [52]	Fine-tuning	68.06/65.69 _(+1.65/+2.15)	66.98/66.48	68.15/62.61	69.04/67.97
+ TraIL-Det (★ Ours)	Fine-tuning	68.88/66.42 _(+2.47/+2.88)	68.21/67.67	69.50/63.76	68.92/67.83

4.2.1 Results on KITTI Dataset

We evaluate the transferability of our pre-trained model by initially pre-training on Waymo and then fine-tuning on KITTI, using PointRCNN [32] and PV-RCNN [33] as baseline detectors. These detectors employ distinct 3D backbones (point-wise and voxel-wise), representing common types of 3D detectors. A key benefit of self-supervised pre-training is enhanced data efficiency, especially with limited annotated data – we thus train the model with 20% ($\sim 0.7k$), 50% ($\sim 1.8k$) and 100% ($\sim 3.7k$) labeled samples. Tab. 1 demonstrates that our pre-trained model boosts performance across both detectors compared to training from scratch and outperforms multiple existing methods. For instance, under the 50% label setting, our model achieves 73.24% mAP on moderate difficulty using PV-RCNN backbone, surpassing both the baseline and ProposalContrast [52] by 3.61% and 1.48% respectively. It also significantly outperforms DepthContrast [55] and PointContrast [46] in detecting cars (+1.40, average, 50% moderate), pedestrians (+2.92, average, 50% moderate) and cyclists (+3.30, average, 50% moderate), highlighting its superior proposal-level representation and ability to handle imbalanced class distributions. Furthermore, we present supporting qualitative results in Fig. 3 (more visualization results in the supplementary materials).

4.2.2 Results on Waymo Open Dataset (WOD)

We follow the widely-used OpenPCDet [37] protocol, fine-tuning the detectors on 20% of the training data for 30 epochs. Tab. 2 shows the results on Level-2 to other SoTA pre-training methods: GCC3D [24] and ProposalContrast [52]. Initially, we report results for training from scratch with different detectors, *i.e.*, SECOND [47], Part-A²-Anchor [34], PV-RCNN [33], and CenterPoint [53] (VoxelNet version), benchmarked against GCC3D [24]. Subsequently, we deploy our TraIL-Det model with two widely-used detectors, *i.e.*, CenterPoint [53] and PV-RCNN [33] for evaluation.

As demonstrated in Tab. 2, our self-supervised pre-training significantly enhances the performance of popular 3D detectors. For PV-RCNN [33], our approach increases the APH

by 3.42% over training from scratch and surpasses SoTA ProposalContrast [52] by 1.34% APH on average. Additionally, we apply our pre-training to CenterPoint [53] equipped with a VoxelNet backbone. The results indicate an improvement of 1.93%. Moreover, utilizing our model with the two-stage CenterPoint architecture achieves an APH of 67.67%, marking a 2.88% increase over the model trained from scratch.

4.3 Ablation Studies

In Tab. 3, we ablate each component of our TraIL-Det in depth. We pre-train the VoxelNet [57] backbone on the full WOD [36] training set in an unsupervised manner, and evaluate the performance by finetuning the detector on WOD 20% training data. We choose CenterPoint [53] which is trained from random initialization as the baseline.

Effectiveness of TraIL Feature: In Tab. 3, we validate the efficacy of the proposed TraIL feature. The use of the TraIL feature result in a 60.52 mAP (+2.1). We further analysis how different neighbor sizes k (Eq. (2)) affected the TraIL, finding $k = 7$ optimal with a 62.29 mAP (+3.87). A larger k can dilute local feature preservation by incorporating distant point distance, while a smaller k risks losing important geometric relationships between neighboring points.

Effectiveness of TraIL MAE: In Tab. 3, we evaluate the performance with (w/) and without (w/o) our proposed multi-head self-attention in our TraIL MAE. Employing the above mechanism, the results show a significant improvement of +1.77 mAP and +1.70 mAPH, which demonstrate the efficacy of the proposed TraIL MAE with multi-head mechanism.

Table 3: Component-wise ablation of our TraIL-Det.

Module	Aspect	Param.	mAP/mAPH	Δ
Baseline	–	–	58.42/55.64	
TraIL Feature	Neighbor size (k)	5	59.48/57.19	(+1.06/+1.55)
		7	62.29/59.89	(+3.87/+4.25)
		10	60.94/58.60	(+2.52/+2.96)
TraIL MAE	Multi-head attention	w/	62.29/59.89	(+3.87/+4.25)
		w/o	60.52/58.19	(+2.10/+2.55)

5 Conclusion

In conclusion, our **T**ransformation-**I**nvariant **L**ocal (TraIL) Features within the TraIL-Det architecture effectively address the limitations of traditional 3D LiDAR object detection by focusing on localized geometry and relationships of points inside proposals. The introduction of the Multi-head self-Attention Encoder (MAE) efficiently processes and encodes high-dimensional TraIL features, leveraging inherent LiDAR isotropic radiation for enhanced representation and computational efficiency. Experimentally, our approach outperforms existing self-supervised methods on KITTI [10] and Waymo [36] datasets, demonstrating its effectiveness in advancing 3D object detection pre-training.

Our features are also highly effective for 3D semantic segmentation [22]. Future research directions may include exploring more downstream tasks and applying the proposed features to domain adaptation. These features enhance robustness to transformations, making them well-suited for complex tasks such as domain adaptation. Additionally, their effectiveness in 3D semantic segmentation [22] suggests they could improve performance across a broader range of downstream tasks.

References

- [1] Jorge Beltrán, Carlos Guindel, Francisco Miguel Moreno, Daniel Cruzado, Fernando Garcia, and Arturo De La Escalera. Birdnet: A 3d object detection framework from lidar information. In *Int. Conf. Intell. Transp. Syst.*, pages 3517–3523. IEEE, 2018.
- [2] Igor Bogoslavskyi and Cyrill Stachniss. Efficient online segmentation for sparse 3D laser scans. *J. Photogramm. Remote Sens. Geoinformation Sci.*, 85(1):41–52, 2017.
- [3] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. Nusences: A multimodal dataset for autonomous driving. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11618–11628, 2020.
- [4] Kai Chen, Lanqing Hong, Hang Xu, Zhenguo Li, and Dit-Yan Yeung. Multisiam: Self-supervised multi-instance siamese representation learning for autonomous driving. In *Int. Conf. Comput. Vis.*, 2021.
- [5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Int. Conf. Mach. Learn.*, 2020.
- [6] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-View 3D Object Detection Network for Autonomous Driving. In *IEEE Conf. Comput. Vis. Pattern Recog.* IEEE, 2017.
- [7] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. Futr3d: A unified sensor fusion framework for 3d detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 172–181, 2023.
- [8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5828–5839, 2017.
- [9] 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 *Proc. AAAI Conf. Artif. Intell.*, volume 35, pages 1201–1209, 2021.
- [10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3354–3361, 2012.
- [11] Chenheng He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11873–11882, 2020.
- [12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.
- [13] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *Int. Conf. Learn. Represent.*, 2019.

- [14] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3D point clouds. In *Int. Conf. Comput. Vis.*, 2021.
- [15] Mingyang Jiang, Yiran Wu, Tianqi Zhao, Zelin Zhao, and Cewu Lu. PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation, 2018.
- [16] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *Int. Conf. Intell. Robots Syst.*, pages 1–8. IEEE, 2018.
- [17] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.
- [18] Enxu Li, Sergio Casas, and Raquel Urtasun. MemorySeg: Online LiDAR Semantic Segmentation with a Latent Memory. In *Int. Conf. Comput. Vis.*, pages 745–754, 2023.
- [19] Guozhong Li, Byron Choi, Jianliang Xu, Sourav S. Bhowmick, Kwok-Pan Chun, and Grace Lai-Hung Wong. ShapeNet: A Shapelet-Neural Network Approach for Multivariate Time Series Classification. In *Conf. AAAI Artif. Intell.*, volume 35, pages 8375–8383, 2021.
- [20] Li Li, Khalid N. Ismail, Hubert P. H. Shum, and Toby P. Breckon. DurLAR: A High-Fidelity 128-Channel LiDAR Dataset with Panoramic Ambient and Reflectivity Imagery for Multi-Modal Autonomous Driving Applications. In *Int. Conf. 3D Vis.*, pages 1227–1237, 2021.
- [21] Li Li, Hubert P. H. Shum, and T. P. Breckon. Less is More: Reducing Task and Model Complexity for Semi-Supervised 3D Point Cloud Semantic Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.
- [22] Li Li, Hubert P. H. Shum, and Toby P. Breckon. RAPID-Seg: Range-Aware Pointwise Distance Distribution Networks for 3D LiDAR Segmentation. In *Eur. Conf. Comput. Vis. Springer*, 2024.
- [23] Xianzhi Li, Ruihui Li, Guangyong Chen, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. A Rotation-Invariant Framework for Deep Point Cloud Analysis. *IEEE Trans. Vis. Comput. Graph.*, 28(12):4503–4514, 2022.
- [24] 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 *Int. Conf. Comput. Vis.*, 2021.
- [25] Jian Liang, Rongjie Lai, Tsz Wai Wong, and Hongkai Zhao. Geometric understanding of point clouds using Laplace-Beltrami operator. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 214–221, 2012.
- [26] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep Continuous Fusion for Multi-Sensor 3D Object Detection. In *Eur. Conf. Comput. Vis. IEEE*, 2018.

- [27] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *Int. Conf. Robot. Autom.*, pages 2774–2781. IEEE, 2023.
- [28] Owen Melia, Eric Jonas, and Rebecca Willett. Rotation-Invariant Random Features Provide a Strong Baseline for Machine Learning on 3D Point Clouds. *Trans. Mach. Learn. Res.*, 2023.
- [29] Gregory P. Meyer, Jake Charland, Darshan Hegde, Ankit Laddha, and Carlos Vallespi-Gonzalez. Sensor Fusion for Joint 3D Object Detection and Semantic Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.* IEEE, 2019.
- [30] Gregory P. Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K. Wellington. LaserNet: An Efficient Probabilistic 3D Object Detector for Autonomous Driving. In *IEEE Conf. Comput. Vis. Pattern Recog.* IEEE, 2019.
- [31] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Adv. Neural Inform. Process. Syst.*, volume 30, 2017.
- [32] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.
- [33] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10529–10538, 2020.
- [34] 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 Trans. Pattern Anal. Mach. Intell.*, 43(8):2647–2664, 2020.
- [35] Vishwanath A Sindagi, Yin Zhou, and Oncel Tuzel. Mvx-net: Multimodal voxelnet for 3d object detection. In *Int. Conf. Robot. Autom.*, pages 7276–7282. IEEE, 2019.
- [36] Pei Sun, Henrik Kretschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2443–2451, 2020.
- [37] OpenPCDet Development Team. OpenPCDet: An open-source toolbox for 3D object detection from point clouds, 2020.
- [38] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.
- [39] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11794–11803, 2021.

- [40] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *Int. Conf. Comput. Vis.*, 2021.
- [41] Daniel Widdowson and Vitaliy Kurlin. Resolving the data ambiguity for periodic crystals. In *Adv. Neural Inform. Process. Syst.*, volume 35, pages 24625–24638, 2022.
- [42] Daniel Widdowson and Vitaliy Kurlin. Recognizing Rigid Patterns of Unlabeled Point Clouds by Complete and Continuous Isometry Invariants With No False Negatives and No False Positives. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1275–1284, 2023.
- [43] Daniel Widdowson, Marco M. Mosca, Angeles Pulido, Andrew I. Cooper, and Vitaliy Kurlin. Average minimum distances of periodic point sets – foundational invariants for mapping periodic crystals. *MATCH Commun. Math. Comput. Chem.*, 87(3):529–559, 2022.
- [44] Hai Wu, Chenglu Wen, Shaoshuai Shi, Xin Li, and Cheng Wang. Virtual Sparse Convolution for Multimodal 3D Object Detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 21653–21662. IEEE, 2023.
- [45] Xiaopei Wu, Liang Peng, Honghui Yang, Liang Xie, Chenxi Huang, Chengqi Deng, Haifeng Liu, and Deng Cai. Sparse fuse dense: Towards high quality 3d detection with depth completion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5418–5427, 2022.
- [46] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas J Guibas, and Or Litany. PointContrast: Unsupervised pre-training for 3D point cloud understanding. In *Eur. Conf. Comput. Vis.*, 2020.
- [47] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.
- [48] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D: Self-training for Unsupervised Domain Adaptation on 3D Object Detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.
- [49] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Int. Conf. Comput. Vis.*, pages 1951–1960, 2019.
- [50] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11040–11048, 2020.
- [51] Junbo Yin, Jin Fang, Dingfu Zhou, Liangjun Zhang, Cheng-Zhong Xu, Jianbing Shen, and Wenguan Wang. Semi-supervised 3D object detection with proficient teachers. In *Eur. Conf. Comput. Vis.*, 2022.
- [52] 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. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Eur. Conf. Comput. Vis.*, volume 13699, pages 17–33. Springer Nature Switzerland, 2022.
- [53] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

- [54] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Multimodal virtual point 3d detection. *Adv. Neural Inf. Process. Syst.*, 34:16494–16507, 2021.
- [55] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pre-training of 3d features on any point-cloud. In *Int. Conf. Comput. Vis.*, 2021.
- [56] Yucheng Zhao, Guangting Wang, Chong Luo, Wenjun Zeng, and Zheng-Jun Zha. Self-supervised visual representations learning by contrastive mask prediction. In *Int. Conf. Comput. Vis.*, 2021.
- [57] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4490–4499, 2018.

Trail-Det: Transformation-Invariant Local Feature Networks for 3D LiDAR Object Detection with Unsupervised Pre-Training

Supplementary Material

Li Li¹

li.li4@durham.ac.uk

Tanqiu Qiao¹

tanqiu.qiao@durham.ac.uk

Hubert P. H. Shum¹

hubert.shum@durham.ac.uk

Toby P. Breckon^{1,2}

toby.breckon@durham.ac.uk

¹ Department of Computer Science

Durham University

Durham, UK

² Department of Engineering

Durham University

Durham, UK

In this documentation, we supplement additional materials to support our findings, observations, and experimental results. Specifically, it is organized as follows:

- Sec. **A** supplements more details on the 3D object detection benchmarks we are using.
- Sec. **B** supplements more details on Transformation-Invariant Local Feature (Trail).
- Sec. **C** acknowledges the public resources used during the course of this work.
- Sec. **D** attaches additional qualitative results, *i.e.*, the 3D object detection visualizations.

A Details on 3D Object Detection Benchmark

The KITTI 3D object detection task (Sec. **A.1**) trains detectors to identify classes such as `Car`, `Pedestrian`, and `Cyclist`, requiring both 2D and 3D bounding boxes along with confidence scores. The KITTI dataset emphasizes accurate projections and filtering of objects not visible in image planes. The Waymo Open Dataset (Sec. **A.2**) enhances autonomous vehicle technologies with high-resolution images and detailed 3D data from multiple LiDARs, capturing objects with unique tracking IDs and specific bounding box criteria. It also includes “No Label Zones” to indicate areas without labels, focusing on detailed spatial awareness and precision.

A.1 KITTI Dataset

The goal in the KITTI 3D object detection task is to train object detectors for the classes `Car`, `Pedestrian`, and `Cyclist`. The object detectors must provide BOTH the 2D 0-based bounding box in the image as well as the 3D bounding box (in the format specified above, *i.e.*, 3D dimensions and 3D locations) and the detection score/confidence. Note that the 2D bounding box should correspond to the projection of the 3D bounding box - this is required to filter objects larger than 25 pixel (height). We also note that not all objects in the point clouds have been labeled. To avoid false positives, detections not visible on the image plane

should be filtered (the evaluation does not take care of this). Similar to the 2D object detection benchmark, we do not count Van as false positives for Car or Sitting Person as false positive for Pedestrian. Evaluation criterion follows the 2D object detection benchmark (using 3D bounding box overlap).

A.2 Waymo Open Dataset (WOD)

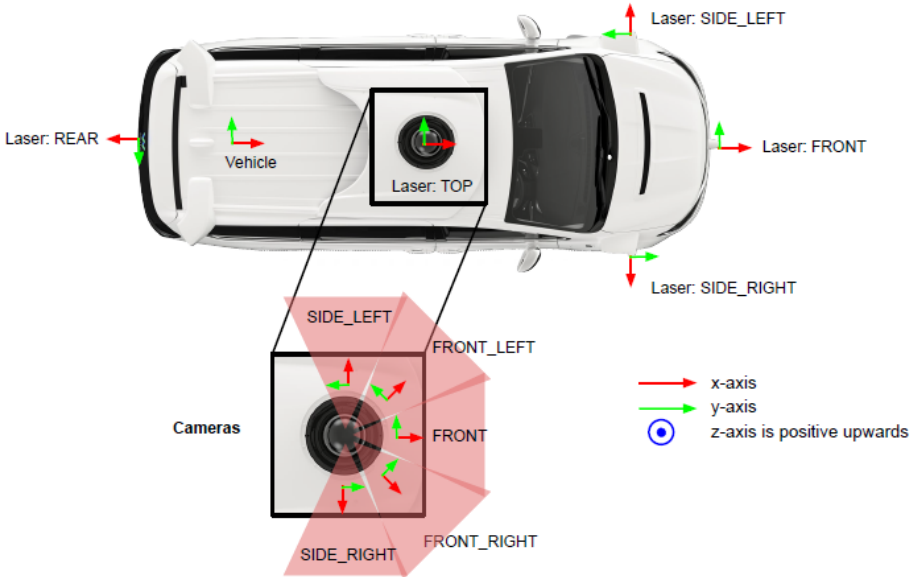


Figure A1: **The sensor setup and configuration on Waymo’s autonomous vehicle.** The positions of various laser sensors (REAR, TOP, SIDE_LEFT, SIDE_RIGHT, FRONT) and camera coverage areas (FRONT_LEFT, FRONT, FRONT_RIGHT, SIDE_LEFT, SIDE_RIGHT) are shown. The coordinate systems are shown with red arrows indicating the x-axis, green arrows indicating the y-axis, and a blue circle indicating the z-axis, which is positive upwards.

Fig. A1 shows Waymo sensor setup and sensor configuration on Waymo’s autonomous vehicle. Top LiDAR covers a vertical field of view (VFOV) from -17.6 to 2.4 degrees, and its range is 75 meters and covers 360 degrees horizontally. Front, side left, side right, and rear LiDARs covers a relatively smaller area than the top LiDAR. They all include a vertical field of view (VFOV) from -90 to 30 degrees, and their range is 20 meters, which is smaller than the top LiDAR. The following objects have 3D labels: vehicles, pedestrians, cyclists, signs. 3D bounding box labels in LiDAR data. The LiDAR labels are 3D 7-DOF bounding boxes in the vehicle frame with globally unique tracking IDs. The bounding boxes have zero pitch and zero roll. Heading is the angle (in radians, normalized to $[-\pi, \pi]$) needed to rotate the vehicle frame +X axis about the Z axis to align with the vehicle’s forward axis. Each scene may include an area that is not labeled, which is called a “No Label Zone” (NLZ). NLZs are represented as polygons in the global frame. These polygons are not necessarily convex. In addition to these polygons, each LiDAR point is annotated with a boolean to indicate whether it is in an NLZ or not.

The dataset contains data from five LiDARs (TOP = 1; FRONT = 2; SIDE_LEFT = 3; SIDE_RIGHT = 4; REAR = 5) - one mid-range LiDAR (top) and four short-range LiDARs (front, side left, side right, and rear). The point cloud of each LiDAR is encoded as a range image. Two range images are provided for each LiDAR, one for each of the two strongest returns. It has 4 channels:

- **Channel 0:** range (see spherical coordinate system definition)
- **Channel 1:** LiDAR intensity channel
- **Channel 2:** LiDAR elongation
- **Channel 3:** is_in_nlz (1 = in, -1 = not in)

B Details of Transformation-Invariant Local Feature (TraIL)

Given a fixed integer $k > 0$ denoting the number of nearest point neighbors, and a point cluster P containing at least k points, the Transformation-Invariant Local Feature (TraIL) is a $u \times k$ matrix preserving spatial distances among points in P .

The k -point TraIL matrix is formally defined as follows:

$$\text{TraIL}(P; k) = \text{sort} \left(\left[\text{sort} \left(\boldsymbol{\rho}_{j,1}, \dots, \boldsymbol{\rho}_{j,k} \right) \right]_{j=1}^u \right), \quad (\text{B1})$$

where $\boldsymbol{\rho}_{j,l}$ represents the distances from the j -th point in P to its k nearest neighbors. Each row of the TraIL matrix corresponds to one point in P and contains the distances to its k nearest neighbors. For convenience to facilitate comparison of various TraIL matrices, we arrange TraIL lexicographically by sorting Eq. (B1), where $\text{sort}(\cdot)$ on the inner and outer brackets sorts the elements $\boldsymbol{\rho}_{j,l}$ within each row j , and the sorted rows based on their first differing elements, both in ascending order.

The distance between points is defined as:

$$\boldsymbol{\rho}_{j,l} = \|\boldsymbol{p}_j - \boldsymbol{p}_{j,l}\|_2, \quad \forall l \in \{1, \dots, k\}, j \in \{1, \dots, u\}, \quad (\text{B2})$$

where \boldsymbol{p}_j and $\boldsymbol{p}_{j,l}$ denote the 3D coordinates of the j -th point and its l -th nearest neighbor within P , respectively. $\|\cdot\|_2$ is the Euclidean norm to compute the spatial distance.

C Public Resources Used

We acknowledge the use of the following public resources, during the course of this work:

- nuScenes¹ CC BY-NC-SA 4.0
- nuScenes-devkit² Apache License 2.0
- The KITTI Vision Benchmark Suite³ CC BY-NC-SA 4.0
- ProposalContrast⁴ MIT License
- VoxSeT⁵ MIT License

¹<https://www.nuscenes.org/nuscenes>.

²<https://github.com/nuTonomy/nuscenes-devkit>.

³<https://www.cvlibs.net/datasets/kitti/>.

⁴<https://github.com/yinjunbo/ProposalContrast>.

⁵<https://github.com/skyhehe123/VoxSeT>.

- SpConv⁶ Apache License 2.0
- Average-Minimum-Distance⁷ CC BY-NC-SA 4.0
- PyTorch-Lightning⁸ Apache License 2.0

D More Qualitative Results

TraIL Visualization of 3D Object Detection: We present supporting qualitative results on 3D Object Detection in Figs. **D1** and **D2**. As shown in Figs. **D1** and **D2**, our approach achieves excellent performance in 3D object detection. Although some vehicles are occluded in the RGB images, our method can still rely on the TraIL features from the point cloud to address the issue of occlusion to some extent.

References

(*NOT THE END*; visualization images follow)

⁶<https://github.com/traveller59/spconv>.

⁷<https://github.com/dwiddo/average-minimum-distance>.

⁸<https://github.com/Lightning-AI/lightning>.



Figure D1: Qualitative results of 3D object detection with our TraIL-Det on KITTI dataset. The predicted 3D bounding box is labeled in the LiDAR point cloud, while its corresponding 2D bounding box is labeled in the RGB image. In the point cloud, white points represent points within the camera field of view (FOV), and purple points indicate points outside the camera FOV. Best viewed in color.

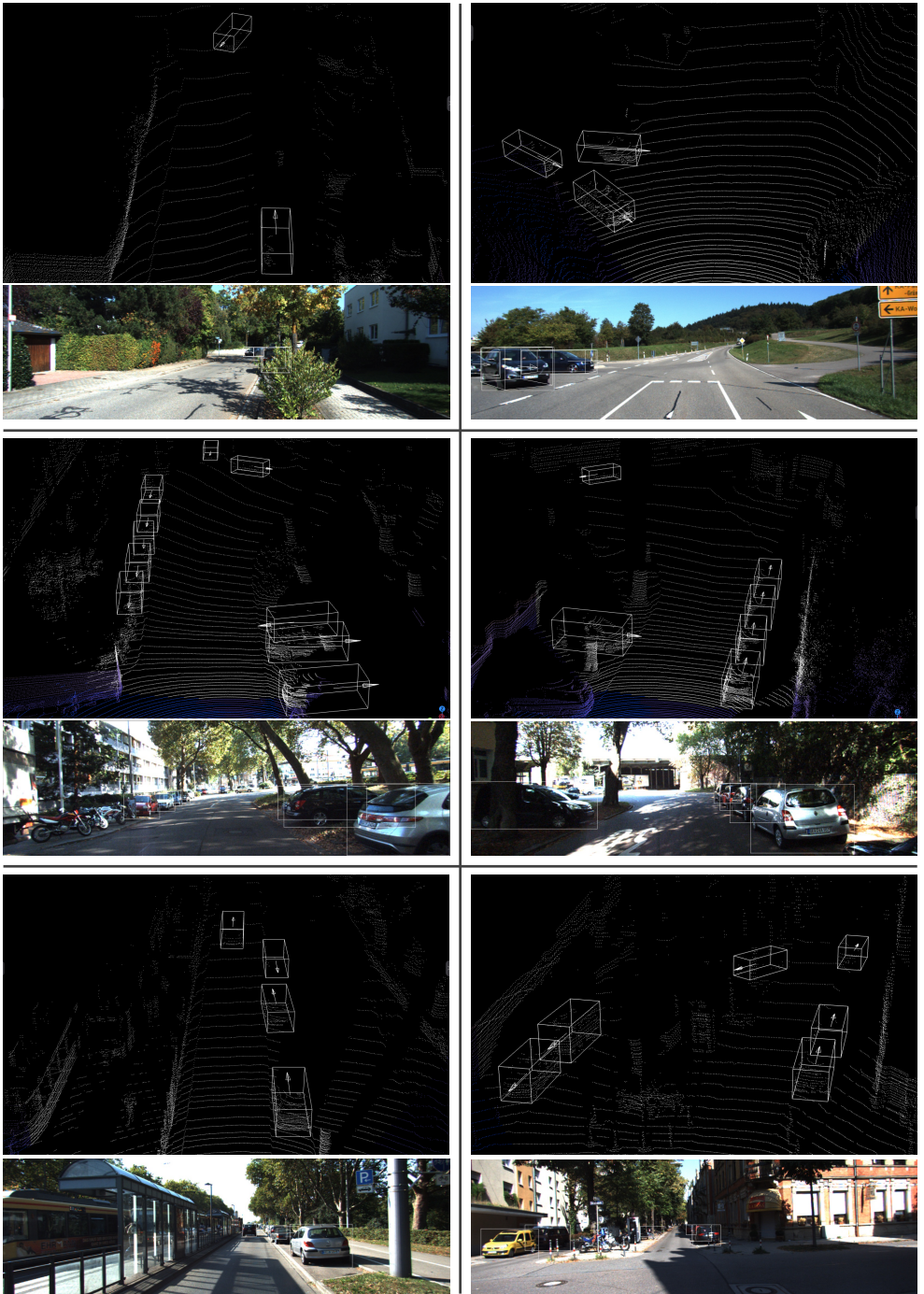


Figure D2: Qualitative results of 3D object detection with our TRAIL-Det on KITTI dataset. The predicted 3D bounding box is labeled in the LiDAR point cloud, while its corresponding 2D bounding box is labeled in the RGB image. In the point cloud, white points represent points within the camera field of view (FOV), and purple points indicate points outside the camera FOV. Best viewed in color.