# Improving 3D Vulnerable Road User Detection with Point Augmentation

Weihao Lu[1], Dezong Zhao[1], *Senior Member, IEEE*, Cristiano Premebida[2], *Member, IEEE*, Li Zhang[3], *Senior Member, IEEE*, Wenjing Zhao[4], Daxin Tian[5], *Senior Member, IEEE*

***Abstract*—Point clouds have been a popular representation to describe 3D environments for autonomous driving applications. Despite accurate depth information, sparsity of points results in difficulties in extracting sufficient features from vulnerable objects of small sizes. One solution is leveraging self-attention networks to build long-range connections between similar objects. Another method is using generative models to estimate the complete shape of objects. Both approaches introduce large memory consumption and extra complexity to the models while the geometric characteristics of objects are overlooked. To overcome this problem, this paper proposes Point Augmentation (PA)-RCNN, focusing on small object detection by generating efficient complementary features without trainable parameters. Specifically, 3D points are sampled with the guidance of object proposals and encoded through the 3D grid-based feature aggregation to produce localised 3D voxel properties. Such voxel attributes are fed to the pooling module with the aid of fictional points, which are transformed from sampled points considering geometric symmetry. Experimental results on Waymo Open Dataset and KITTI dataset show a superior advantage in the detection of distant and small objects in comparison with existing state-of-the-art methods.**

***Index Terms*—Autonomous driving, 3D object detection, light detection and ranging (LiDAR) point clouds, intelligent vehicles.**

## I. Introduction

3D object detection is irreplaceable in current research around intelligent vehicles. The aim is to equip the ego-vehicles with the capability of recognising and locating other road users in 3D environments. It provides a fundamental understanding of the surrounding of intelligent systems and facilitates the subsequent tasks in the perception workflow of autonomous driving [1], [2]. With greater weather-proof ability than camera systems, light detection and ranging (LiDAR) sensors are more widely deployed to acquire accurate depth measurements and to extract the geometry information with point clouds. Recent development in deep neural networks has further boosted the use of LiDAR sensors.

Different from images, point cloud processing is less straightforward because of its sparsity and irregularity [1]. To address this issue, researchers extract the high-dimensional features from point clouds in two main formats, which are point-based and voxel-based. Point-based methods encode point coordinates with a symmetry function and store the information at point locations [3]–[5], while voxel-based methods discretise the 3D scene and perform feature learning on the regular grids [6]–[8]. Voxelisation simplifies the nearest neighbour query by directly selecting the adjacent indexes on the grid map to increase sampling efficiency in the receptive field. However, locating features at the fictional voxel centres harms the accuracy of voxel-based encoders. In contrast, by inheriting accurate point locations throughout the information flow of the networks, point-based methods can locate rich features precisely in the scene. However, owing to the fact that searching for nearest neighbours among unordered points is time-consuming, a poor point sampling scheme may also limit the efficiency of point-based encoders, such as Set-Abstraction in [5]. Although remarkable performance is achieved in 3D car detection using point clouds, researchers tend to overlook the deficiency in detecting more vulnerable targets, such as pedestrians and cyclists. Such small or distant objects often attract fewer laser beams from a LiDAR sensor due to the sensor's nature (*i.e.* , point density shrinks as distance increases). Therefore it is crucial to consider raw point features for the detection network.

Down-sampling points is inevitable to increase the receptive field and maintain the input size with point-based methods. Thus, convolutions are performed around the selected key points. The commonly used schemes are farthest point sampling (FPS) and random point sampling (RPS). FPS may ensure the most coverage of the scene, while RPS may avoid overfitting. However, both cannot guarantee object points can survive the next stage of the network. Many irrelevant background points are included due to the imbalance number of foreground/background points. This leads to potential information loss, especially for distant and small objects, such as pedestrians and cyclists. Increasing the receptive field is relatively easy with a grid-like structure, usually by decreasing the voxel resolution. It is often difficult to balance the trade-off

*This work was sponsored in part by the Engineering and Physical Sciences Research Council of the UK under the EPSRC Innovation Fellowship (EP/S001956/2) and in part by the Newton Advanced Fellowship (UK-China International Cooperation and Exchange Project) jointly supported by the UK Royal Society (NAF\R1\201213)

[1] Weihao Lu and Dezong Zhao are with the School of Engineering, University of Glasgow, University Avenue, Glasgow G12 8QQ, UK w.lu.1@research.gla.ac.uk, dezong.zhao@glasgow.ac.uk

[2] Cristiano Premebida is with the Institute of Systems and Robotics, University of Coimbra, 3030-290 Coimbra Portugal cpremebida@deec.uc.pt

[2] Li Zhang is with the Department of Computer Science, Royal Holloway, University of London, Surrey, TW20 0EX, UK li.zhang@rhul.ac.uk

[4] Wenjing Zhao is with the Department of Civil Environmental Engineering, Hong Kong Polytechnic University, Hong Kong, China wenjing.zhao@polyu.edu.hk

[5] Daxin Tian is with the School of Transportation Science and Engineering, Beihang University, Haidian District, Beijing 100191, China dtian@buaa.edu.cn

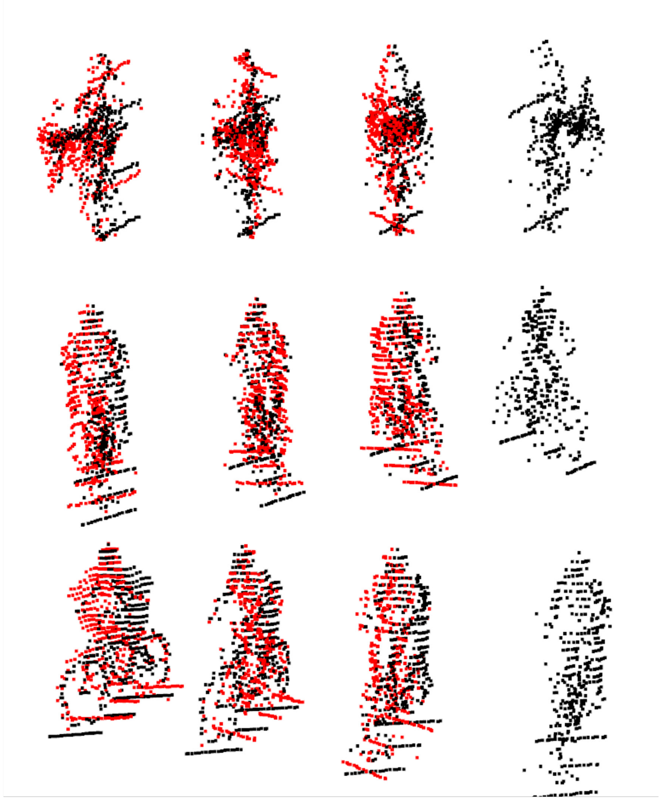


Fig. 1. Shape completion on a cyclist with Point Augmentation module utilising symmetry. From the top, there are one top view and two front views. From right to left, it includes Original, Mirroring (xz-plane), Mirroring (yz-plane), Rotation displays. Original and augmented points are in black and red respectively.

between memory usage (number of voxels) and performance, as larger voxels often neglect small objects.

Vision transformer (ViT), as a counterpart of convolutions, has played a noticeable part in current 2D object detection tasks [9]–[11]. The self-attention mechanism provides long-range connections between pixels which are close in high dimensions. This aids the discovery of small objects in the image. Researchers aim to transfer the success of ViT in 2D tasks into 3D tasks [7], [12]. Although the latency and performance can be improved, the transplant for the transformer can be expensive and bring extra complexity to detection networks.

Another approach to improve detection on occluded, distant or small instances is to reconstruct the points of missing shapes. With the aid of point completion algorithms, a generative module is trained to predict the full shape of objects from incomplete point sets in a self-contained manner or through external datasets [13]–[16]. The instances with incomplete geometry are replaced or augmented with the generated shapes to increase the confidence of predictions for small and occluded objects. As the predicted parts of objects are usually unseen or overlooked by the sensors, the point completion results may not always reconstruct the correct object surfaces, especially when testing outside the training domain. These methods tend to induce large memory usage and high computational cost due to the extra inference module, as well as a longer latency.

To solve the aforementioned issues, a new object detection framework with Point Augmentation, namely PA-RCNN is proposed, to facilitate efficient detection of small and distant objects, by integrating a proposal-guided sampling scheme and a simple yet effective object point augmentation module in a two-stage object detection architecture. The main contributions of this paper can be summarised as follows:

- To ensure accurate proposals are generated, a lightweight Attention-based Semantic Mining (ASM) module is adopted to yield the 2D feature map, considering both geometric and semantic information. Gradient degradation can be mitigated by fusing geometric information, which is relatively shallow. Compared to 3D transformer [7], the 2D attention algorithm consumes less memory, while achieving favourable results for detection proposals.
- To sample as many informative foreground points as possible for the second stage of the detector, key point sampling is guided by detection proposals. This effectively reduces background noises for the region of interest (RoI) pooling and bounding box refinement.
- To benefit from the complete object shapes, the RoI refinement stage comprises a point augmentation module (PAM) and local Grid-based Voxel-to-Point Feature Aggregation (GVPFA) module. The PAM has no trainable parameters and extracts all proposed object points from raw input and their associated features. To realise point cloud completion without trainable parameters, the generic geometric characteristics of symmetry are exploited, shown in Figure 1.

Thorough experiments prove that our proposed method exceeds the current state-of-the-art on Level_2 of Waymo Open Dataset and achieves the best results on KITTI cyclist category among methods without generative modules.

## II. Related Works

### A. *Convolution-based 3D Detection*

Most existing 3D detectors heavily rely on the advancement of convolutional neural networks [1], [2], [6]. Some detectors perform convolutions directly on raw points using, for instance, the Set Abstraction module from PointNet [1]. F-PointNet [17] crops the point cloud scene based on the proposals generated by a 2D detector from RGB images. The cropped point cloud can reduce the number of background points for bounding box refinement. Point-RCNN [5] improves proposal quality by adopting a 3D backbone to encode the entire scene to provide 3D proposals. 3DSSD [3] replaces the costly feature propagation layers with an advanced sampling technique to achieve single-stage anchor-free detection.

By discretising the 3D space into voxels, VoxelNet [18] can deploy 3D convolution directly on the regular grids. SECOND [19] improves the 3D CNN with sub-manifold and sparse convolutions, considering the sparsity of the point cloud. PointPillars [20] merges vertical voxels into pillars to form a pseudo-image, with the attempt to reduce computation burden. Voxel-RCNN [6] integrates a bounding box refinement stage to the SECOND backbone. PV-RCNN [2] associates voxel features to key point locations with the voxel set abstraction module. Li et al. [21] solve the IoU-misalignment issue with

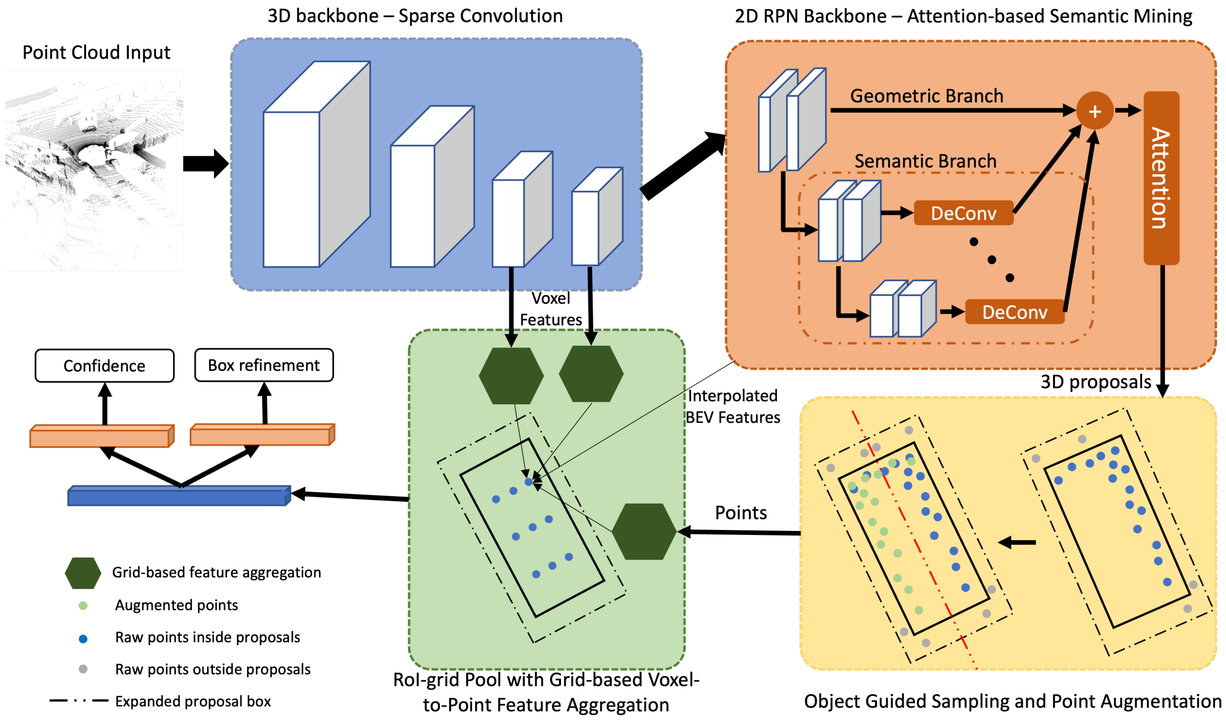

3D backbone – Sparse Convolution
2D RPN Backbone – Attention-based Semantic Mining
Point Cloud Input
Geometric Branch
Semantic Branch
+
Attention
DeConv
DeConv
Voxel Features
Interpolated BEV Features
3D proposals
Confidence
Box refinement
Points
Grid-based feature aggregation
Augmented points
Raw points inside proposals
Raw points outside proposals
Expanded proposal box
RoI-grid Pool with Grid-based Voxel-to-Point Feature Aggregation
Object Guided Sampling and Point Augmentation

Fig. 2. Overview of the PA-RCNN architecture. a) The input point cloud is first voxelised and encoded with the 3D sparse convolution backbone to output intermediate features as $L \times W \times H \times C$. The 3D voxel features are flattened as $L \times W \times (H \times C)$, followed by a 2D region proposal network (RPN). b) The 2D RPN comprises the proposed attention-based semantic mining network. The depth of the semantic branch depends on the architecture design for certain applications. c) 3D proposals and point semantics are given by the 2D backbone, and guide the point sampling and facilitate the point augmentation. The estimated object shapes are represented by the union of all complementary points and raw points, which are used in the RoI refinement stage. d) By encoding the point coordinates and aggregating voxel features with the GVPFA module, features are summarised to the RoI grid points. The RoI features are used to predict the final bounding box output and the corresponding confidence scores.

the redesigned box refinement module. However, such recent detectors mainly focus on the detection of targets such as cars or vehicles, while smaller vulnerable instances are often overlooked, due to the lack of both rich features and long-range connections to similar objects in the scene.

### B. Attention-Based 3D Detection

While being able to provide long-range connections, self-attention-based modules usually act as feature enhancers in 3D detectors in early work. DVFENet [22] enhances the features by adding the graph-attention-based branch in parallel with the baseline sparse convolution backbone. S-AT GCN [23] adds a spatial-attention module to PointPillars [20] to reduce partition effects. Pyramid-RCNN [8] improves the second-stage module by introducing the pyramid RoI head with conventional attention- and graph-based operators. VoTr [7] rebuilds the detector's backbone with a 3D voxel transformer with large memory consumption. CT3D [24] consists of a channel-wise transformer, which operates on raw 3D points. VoxSet [12] detects 3D objects with set-to-set translation, reducing memory usage and runtime. However, transformers are usually introduced to voxel-only networks. Increasing the usability of transformers on 3D raw points is not trivial, due to the unordered nature of point clouds.

### C. Generative Methods for 3D Detection

To solve the inconsistent point density, PC-RGNN [25] predicts the complete shapes of objects with a point cloud completion module. The point cloud completion module renders additional points to the proposals with a multi-resolution graph encoder and a point pyramid decoder. Associate-3Ddet [26] and AGO-Net [26] mimic the bio-model by learning to map incomplete perceived features of objects to more complete features of corresponding class-wise conceptual models. Such a generative feature enhancement scheme greatly improves the detection accuracy on distant objects with fewer numbers of points. SIENet [27] predicts the spatial shapes of foreground points in proposals, where the prediction module is trained with external data. Semantic point generation (SPG) [14] closes the domain gap by adopting an SPG module to recover the foreground points overlooked by the sensors. Btc-Det [13] predicts the occupancy map and estimates the complete object shapes that are occluded with prior learned information. SFD [15] generates pseudo point clouds by estimating depth on RGB images and extracting rich contextual and spatial features with attentive fusion with raw point clouds. Generative modules provide conceptual information that is not perceived by the sensors. Considering the advantages of generative modules, a more simple approach to estimate the complete object shapes is further investigated.

## III. PA-RCNN: Point Augmentation for Vulnerable Object Detection

This section introduces the proposed PA-RCNN detector. Based on PV-RCNN [2], a two-stage detection framework, the author explores the improvements of bird's eye view (BEV) encoders, point sampling strategy, deformable point-voxel feature aggregation and point augmentation. Figure 2 shows the layout of the network. The first stage of PA-RCNN encodes voxel features with the 3D backbone of sparse and sub-manifold convolutions and BEV features with the 2D backbone of an attention-based semantic mining module. BEV features are used to generate detection proposals. The proposals are then refined by the second stage to perform point augmentation and RoI-grid pooling to produce the final predictions.

### *A. Attention-based Semantic Mining Module*

2D BEV features are crucial in the 2-stage detection pipeline, as the detection proposals are generated solely from the BEV feature map. The quality of proposals directly affects the final results. In recent methods, flattened 2D BEV feature maps are processed with the widely used 2D backbone, consisting of a group of basic convolution layers for encoding and decoding. The features are less sensitive to small objects due to the partition effects, where small objects may be neglected or truncated through pooling.

To enhance the feature richness in the 2D backbone, CIA-SSD [28] builds a dual-branch encoding scheme and SMF-SSD [29] uses multi-scale 3D features. Inspired by [30], we consider both the depth and width of the features. The deeper features focus on the high-dimensional semantic information of the scenes, while the shallower features emphasise the intra- and inter-instance geometric relations. Similar to [28], a dual-branch feature encoder is constructed to avoid the shallower features being washed out in a deep neural network. On the short path, feature map resolution and the number of channels remain unchanged with fully connected layers $\phi$. While [28] uses a single semantic branch in the 2D backbone, ASM further exploits the high-dimensional semantics by adopting multiple information paths in the semantic branch. On this long path, strided convolutions $\psi$ are used to aggregate high dimensional semantics. Different from SMF-SSD [29], only features from the last layer of the 3D backbone are fed to ASM. The structure of ASM is shown in Figure 2(b). Given the flatten 3D feature map, $\mathbf{F}_{flat}$, the process can be summarised as:

$$\mathbf{F}_{bev,g} = \phi(\tau(\mathbf{F}_{flat})) \tag{1}$$

$$\mathbf{F}_{bev,s,i} = \psi(\tau(\mathbf{F}_{flat})), \tag{2}$$

**Algorithm 1** Object-guided Point Sampling for $N$ points and $M$ proposal boxes

```
Input: Point coordinates $p_i \in \mathbb{R}^3$, for $i \in [1, N]$
       Proposal centres $c_j \in \mathbb{R}^3$, for $j \in [1, M]$
       Proposal dimensions $d_j \in \mathbb{R}^3$, for $j \in [1, M]$
Output: key point candidates $\mathbf{P} \in \mathbb{R}^{n \times 3}$
        per-box key point list $\mathbf{Q} \in \mathbb{R}^{M \times n' \times 3}$
Create an empty list of key point candidates $\mathbf{P}$
for $i = 1$ to $N$ do
    for $j = 1$ to $M$ do
        create an empty list of key point sets $\mathbf{Q}_j \in \mathbb{R}^{n' \times 3}$
        if $||p_{i,x} - c_{j,x}|| < d_{j,x}$ & $||p_{i,y} - c_{j,y}|| < d_{j,y}$ &
            $||p_{i,z} - c_{j,z}|| < d_{j,z}$ do
            $\mathbf{Q}_j \leftarrow [\mathbf{Q}_j, p_i]$
            if $p_i$ not in $\mathbf{P}$ do
                $\mathbf{P} \leftarrow [P, p_i]$
            end if
        end if
    end for
    $\mathbf{Q} \leftarrow [\mathbf{Q}, \mathbf{Q}_j]$
end for
```

where $\tau$ is a shared bottom-up convolution layer, $\mathbf{F}_{bev,g}$ and $\mathbf{F}_{bev,s,i}$ are the features from the geometric and semantics branches respectively. Unlike dense connections [31], where features from different layers are stacked through concatenation, we follow SKNet [32] to use branch-wise attention for its advantages in filtering meaningful discriminative information from a sparse feature map. To match the feature map sizes of the branches, extra deconvolution layers are introduced to follow the strided convolutions to obtain $\mathbf{F}_{bev,s}$. We compute the intermediate features $\mathbf{z}$ with the channel-wise addition of $\mathbf{F}_{bev,g}$ and all $\mathbf{F}_{bev,s,i}$. The attention weights for each path in both branches can be denoted as $\Omega = \{\omega_g,\ \omega_{s,1},\ ...,\ \omega_{s,i}\}$. The weights can be given as:

$$\Omega = \mathrm{Softmax}(\mathbf{A} \cdot \mathrm{MLP}(\mathbf{z})), \tag{3}$$

$$\mathbf{z} = \mathbf{F}_{bev,g} \oplus \mathbf{F}_{bev,s,1} \oplus ... \oplus \mathbf{F}_{bev,s,i}, \tag{4}$$

where $\mathbf{A} = \{A_g,\ A_{s,1},\ ...,\ A_{s,i}\}$ are the attention embeddings for $\mathbf{F}_{bev,g}$ and each $\mathbf{F}_{bev,s,i}$ respectively. The aggregated BEV features can then be given as:

$$\mathbf{F}_{bev} = \omega_{\mathbf{g}} \cdot \mathbf{F}_{bev,g} + \sum_{i=1}^{l} \omega_{\mathbf{s,i}} \cdot \mathbf{F}_{bev,s,i}, \tag{5}$$

where $\omega_{\mathbf{g}} + \sum_{i=1}^{l} \omega_{\mathbf{s,i}} = 1$. $l$ is the number of layers on the semantic branch. The attention mechanism naturally gathers the related information from both the geometric and semantic feature maps.

### *B. RoI Refinement with Auxiliary Points*

*1) Object-guided Point Sampling:* The quality of point sampling greatly affects the efficiency of the refinement stage. Given the voxel features $\mathbf{F}_{voxel}$ and sampled key points $\mathbf{q} = \{q_i \,|\, i = 1, ..., N'\}$, the feature encoder aggregates the features of neighbouring voxels around each key points. To ensure the aggregated features are relevant to the target objects, it is essential to select as many foreground points as possible. Therefore the distraction from the background can be minimised when more foreground points are selected.

FPS is a popular approach in recent methods, which aims to cover the scene evenly by selecting the most distant points. FPS works well on detecting cars and vans, since the larger objects have more points. However, many background points

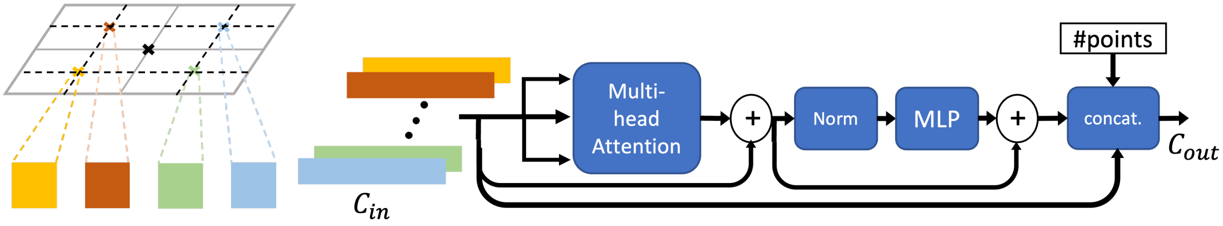


Fig. 3. Illustration of the grid-based voxel-to-point feature aggregation. With the grid drawn over the neighbourhood of a key point, local grid features are first aggregated to the local grid points. The local grid features are then processed with a grouped MLP layer to produce $C_{in}$ channels for each grid. The $C_{in}$ features are passed onto a multi-head attention layer, followed by a normalisation layer and MLP. The grid-distinct features with $C_{in}$ channels, attention features with $C_{out,att}$ and the number of points are concatenated to give a $C_{out}$ channel output, where $C_{out} = C_{in} + C_{out,att} + 1$.

are also selected. Semantic assisted FPS [4] introduces semantic weights to the distance between points, where foreground points have higher weights. This improves the sampling efficiency with a high computational cost. Moreover, small objects are more vulnerable to FPS and often overlooked. While aggregating voxel features to the key points, features related to a smaller object can be assigned to its nearest background point that has survived the sampling process. This leads to a mislocation of the features. The impact of feature mislocation is more significant due to smaller object sizes.

Proposals generated by the first stage provide guidance to the approximate locations and sizes of the target objects. By selecting the points within the proposals, one can improve the appearance of points from the small objects in the sampled point set. The procedure is shown in Algorithm 1. The proposal boxes are also enlarged to accommodate the imperfection and include the background points around the boxes, which possess important information to distinguish the object edges. Different from a whole scene sampling, the proposed method also creates a point set for each proposal. This helps separate points from different objects and facilitates point augmentation in the second stage.

*2) Grid-Based Voxel-to-Point Feature Aggregation:* The voxel representation is favoured for its regularity, which simplifies the neighbour quarrying process. Voxel neighbours around the key point can easily be found by indexing, while distance calculation and sorting are required to find point neighbours. Voxel centres can be calculated with voxel indexes $(i, j, k)$ by:

$$\mathbf{V_c} = (\frac{L}{N_x}(\frac{1}{2}+i), \frac{W}{N_y}(\frac{1}{2}+j), \frac{H}{N_z}(\frac{1}{2}+k)), \tag{6}$$

where $(L, W, H)$ are the scene size and $(N_x, N_y, N_z)$ are the numbers of voxels in each dimension. The voxelisation process tends to assign the encoded features to the voxel centres. Such process leads to the loss of fine-grained point details, since the precise point positions measured by a LiDAR sensor are not used. The actual precision of feature locations is greatly dependent on the degree of voxelisation, *i.e.* , voxel grid size. Smaller voxel grids produce more accurate locations for feature aggregation and remarkable results with only voxels on car detection are achieved [6]. However, cyclist and pedestrian targets are more prone to failure caused by inaccurate feature locations, since the bounding box sizes are significantly smaller than those of cars and vans.

To mitigate this inefficiency, a Grid-based Voxel-to-Point Feature Aggregation module (GVPFA) is proposed. Positional information is implanted by adding the relative coordinates of the neighbouring points. In addition, the point density information is also inserted by adding the number of points in the vicinity. Specifically, the space around a sampled key point $q_i$ is divided into local grids $\mathcal{G}_l$. Contradictory to the commonly used set abstraction [2], features are first aggregated within each local grid before being summarised to the key points. The features of local grid $\mathcal{G}_{l,i}$ can be expressed as:

$$f_{\mathcal{G}_{l,i}} = f(\mathcal{V}_i \,|\, \mathcal{V}_i \in \mathcal{N}(\mathcal{G}_{l,i})), \tag{7}$$

where $\mathcal{V}_i$ is one of the neighbouring voxels around the local grid centre. Inspired by [33], features of a key point $q_i$ can be generated by a grouped MLP by:

$$f_{q_i} = [\omega_{l,1} \times f_{\mathcal{G}_{l,1}}, \omega_{l,2} \times f_{\mathcal{G}_{l,2}}, ..., \omega_{l,n} \times f_{\mathcal{G}_{l,n}}], \tag{8}$$

where $\omega_{l,i}$ is the respective weight of kernel filters of the MLP and $n$ is the number of local grids around a key point. A grouped MLP can limit the influence between different groups by isolating the feature interaction. This allows the module to produce position-specific semantics. Memory consumption can also be reduced through the use of a grouped MLP by removing unused links.

However, objects can appear at any rotational angle on the ground surface. A complete detachment of features on different grids around the key point is insufficient to address this nature. The connection between these grids has to be built to realise the rotational invariance of the features. Inspired by [9], a lightweight self-attention module is deployed over the local grids to enable feature communication. Since the sparsity of points causes a more severe deficiency in detecting small objects, the number of points in the local neighbourhood is added to the feature map. The feature output can be summarised as,

$$\mathbf{F} = \mathrm{FC}([\mathrm{SA}(f_q),\ f_q,\ n]) \tag{9}$$

where $n$ is the number of points in each key point neighbourhood. The output of the self-attention (SA) module is concatenated with $n$ and key point features, followed by a fully connected (FC) layer to give the output feature dimension for subsequent processes.

The same strategy is applied to the RoI grid pool module, where keypoint features are aggregated to the box grid points instead of local grids. The self-attention layer also provides interdependence over individual bounding boxes.

*3) Point Augmentation:* Detection on only downsampled points results in a lower accuracy [3]. This is caused by insufficient information and further deteriorated by the ground truth ambiguities, occlusions and missing elements in the ground truth. As such, a potential solution is to utilise generative modules to predict the missing signals and provide the omitted semantic information [13], [14], [16]. However, the extra complexity and issues with domain adaption is often overlooked.

A simple and effective point augmentation module is built to recover the approximate shape of the object based on the pure geometric relation. By assuming approximate symmetry of the detection targets, the key points and the associated features of each proposal sampled by the object-guided point sampling module are processed with the operation $\mathcal{T}$. The augmented features can be generated by:

$$\mathbf{F}_{aug,j} = \mathcal{T}(\mathbf{F}_j, [p_x, p_y, p_z]), \;\; p \in \mathcal{B}_j, \tag{10}$$

where $\mathbf{F}_j$ is the features aggregated inside the proposal bounding box $\mathcal{B}_j$, $[p_x, p_y, p_z] \in \mathcal{R}^{N' \times 3}$ is the coordinates of $N'$ points inside $\mathcal{B}_j$. The operation $\mathcal{T}$ can be mirroring or rotating the points with reference to the bounding boxes. In our case, mirroring the points and duplicating features for the new points are used.

The enhanced features, as well as the original features, are fed to the RoI grid pool module, where features are gathered to the proposal box grid points accordingly. In addition, the coordinates of all raw points in the proposal boxes are processed with the GVPFA module to provide shallow and complete geometric information. The point sets for different proposals are given by the sampling layer.

With the help of the approximated object shapes and structure-sensitive features, the bounding box refinement layer can regress accurate bounding boxes based on enriched semantics from both perceptual and conceptual information, especially for small and vulnerable objects like cyclists and pedestrians.

## IV. Experiments

This section presents results from thorough experiments, and is formatted to provide: 1) a brief introduction to datasets and implementation details; 2) a comparison with other state-of-the-art methods and 3) an analysis of the effectiveness of each component in the architecture.

### *A. Dataset*

The proposed method is evaluated on the commonly used KITTI dataset [34] and Waymo Open dataset (WOD) [35].

**Waymo Open Dataset**. WOD is a significantly larger dataset with 798 training and 202 validation sequences, with around 160k and 40k point cloud samples respectively. The evaluation metric is calculated as the mean Average Precision (mAP) and the mean Average Precision weighted by Heading (mAPH). The 3D intersection-over-union (IoU) thresholds for the bounding boxes are $(0.7, 0.5, 0.5)$ for Car, Pedestrian and Cyclist categories. Depending on how the testing samples are split, the results can be formatted by difficulty levels and detection ranges. By difficulty levels, ground truth targets are divided into LEVEL_1 and LEVEL_2, which guarantees at least 5 and 1 laser points are reflected from the objects. By detection ranges, the ground truth targets are assigned to the groups of $0-30\text{m}$, $30-50\text{m}$ and $>50\text{m}$ from the sensor.

**KITTI dataset**. The KITTI 3D object detection dataset contains 7481 and 7518 samples for training and testing respectively. The training set is further divided into a 50/50 *train*/*val* splits with 3712 training and 3769 validation samples respectively. The official evaluation metric is the mAP calculated by the official evaluation tool with 40 points from the precision-recall curve on three difficulty levels. The 3D IoU thresholds are $(0.7, 0.5)$ for Car and Cyclist categories.

### *B. Implementation*

The proposed method is built based on the widely used OpenPCDet [40] codebase. Particularly, the 2D backbone of PV-RCNN [2] is replaced with our ASM module. The RoI refinement stage is also extended with our grid-based feature aggregation and point augmentation module, while keeping the rest of the network untouched.

For the ASM module, a 2-layer semantics mining submodule is used, which consists of a $3 \times 3$ convolution with a feature dimension of 128 for each layer. A deconvolution layer is added to each of the layers on the semantic branch to match the feature map shape of the geometric branch. The geometric branch comprises a single $3 \times 3$ convolution layer with 128 output channels. The features of different branches are summarised with an attention fusion module with 256 output channels.

For the second stage of RoI refinement, raw points that lie is the enlarged proposal boxes are sampled. The proposal boxes are enlarged by $0.2m$ in each axis. For aggregating features from the BEV and 3D backbone, each local grid has 32 output channels. The local grid features are processed with a transformer layer. The point augmentation is configured to mirror the raw points around the proposals to obtain the estimated shapes. For aggregating the shallow and complete geometric information from raw and augmented points, the author uses a two-scale approach with local grid sizes of $(2, 2, 2)$, $(3, 3, 3)$ for Waymo Open dataset, and $(3, 3, 3)$, $(4, 4, 4)$ for KITTI dataset.

The model is trained with the ADAM optimiser on 4 RTX 2080 Ti GPUs. With respect to the KITTI dataset, the model is trained with a batch size of 8 and a learning rate of 0.007 for 80 epochs. For obtaining the results on the *val* set, the models are trained on the 50/50 *train*/*val* split. For reporting the results on the test server, a 80/20 *train*/*val* split is used. On Waymo Open dataset, the model is trained with a batch size of 8 and a learning rate of 0.007 for 40 epochs. A 20% train-split training option is also provided, where training scenes are sampled uniformly and evaluation is performed on the full validation set. Following OpenPCDet [40], the cosine annealing learning rate decay strategy is adopted and the same data augmentation scheme is used.

TABLE I
PERFORMANCE COMPARISON ON THE WAYMO OPEN DATASET WITH 202 VALIDATION SEQUENCES BY OBJECT TYPES. *: RESULT ON 20% TRAINING SPLIT. †: RESULTS FROM [36].

| Methods | Vehicle LEVEL_1 | | Vehicle LEVEL_2 | | Pedestrian LEVEL_1 | | Pedestrian LEVEL_2 | | Cyclist LEVEL_1 | | Cyclist LEVEL_2 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | mAP | mAPH | mAP | mAPH | mAP | mAPH | mAP | mAPH | mAP | mAPH | mAP | mAPH |
| SECOND† [19] | 72.27 | 71.69 | 63.85 | 63.33 | 68.7 | 58.18 | 60.72 | 51.31 | 60.62 | 59.28 | 58.34 | 57.05 |
| PointPillars [20] | 56.62 | - | - | - | 59.25 | - | - | - | - | - | - | - |
| DVF [37] | 67.62 | 67.09 | 62.66 | 62.17 | - | - | - | - | - | - | - | - |
| Part-A2† [38] | 74.82 | 74.32 | 65.88 | 65.42 | 71.76 | 63.64 | 62.53 | 55.3 | 67.35 | 66.15 | 65.05 | 63.89 |
| Voxel-RCNN [6] | 75.59 | - | 66.59 | - | - | - | - | - | - | - | - | - |
| PV-RCNN† [2] | 78.00 | 77.50 | 69.43 | 68.98 | 79.21 | 73.03 | 70.42 | 64.72 | 71.46 | 70.27 | 68.95 | 67.79 |
| PV-RCNN++† [36] | **79.10** | **78.63** | 70.34 | 69.91 | 80.62 | 74.62 | 71.86 | 66.30 | 73.49 | 72.38 | 70.70 | 69.62 |
| PDV [39] | 76.85 | 76.33 | 69.30 | 68.81 | 74.19 | 65.96 | 65.85 | 58.28 | 68.71 | 67.55 | 66.49 | 65.36 |
| VoxSet [12] | 77.82 | - | 70.21 | - | - | - | - | - | - | - | - | - |
| AGO-Net [26] | 69.20 | 68.70 | 60.60 | 60.10 | 59.30 | 48.70 | 51.8 | 42.4 | 55.3 | 54.2 | 53.5 | 52.5 |
| BtcDet [13] | 78.58 | 78.06 | 70.10 | 69.61 | - | - | - | - | - | - | - | - |
| VoTr-TSD* [7] | 74.95 | 74.25 | 65.91 | 65.29 | - | - | - | - | - | - | - | - |
| Pyramid-PV* [8] | 76.3 | 75.68 | 67.23 | 66.68 | - | - | - | - | - | - | - | - |
| PA-RCNN* | 77.91 | 77.44 | 69.54 | 69.11 | 80.50 | 74.94 | 72.09 | 66.88 | 74.05 | 72.96 | 71.37 | 70.32 |
| PA-RCNN | 78.75 | 78.29 | **70.43** | **70.00** | **81.73** | **76.55** | **73.46** | **68.59** | **74.23** | **73.18** | **71.54** | **70.52** |

TABLE II
PERFORMANCE COMPARISON ON THE WAYMO OPEN DATASET WITH 202 VALIDATION SEQUENCES FOR VEHICLE CLASS BY RANGE. †: RESULTS FROM [36]

| Methods | Vehicle LEVEL_1 mAP/mAPH | | |
|---|---|---|---|
| | 0-30m | 30-50m | 50m-inf |
| SECOND† [19] | 90.66/- | 70.03/- | 47.55/- |
| PointPillars [20] | 81.01/- | 51.75/- | 27.94/- |
| Part-A2† [38] | 92.35/- | 75.91/- | 54.06/- |
| Voxel-RCNN [6] | 92.49/- | 74.09/- | 53.15/- |
| CT3D [24] | 92.51/- | 75.07/- | 55.36/- |
| PV-RCNN† [2] | 92.96/- | 76.47/- | 55.96/- |
| PV-RCNN++† [36] | 93.34/- | **78.08**/- | 57.19/- |
| VoxSet [12] | 92.5/- | 70.10/- | 43.20/- |
| PDV [39] | 93.13/92.71 | 75.49/74.91 | 54.75/53.90 |
| BtcDet [13] | **96.11**/- | 77.64/- | 54.45/- |
| PA-RCNN | 92.88/92.48 | 77.60/77.10 | **57.71/56.99** |

| Methods | Vehicle LEVEL_2 mAP/mAPH | | |
|---|---|---|---|
| | 0-30m | 30-50m | 50m-inf |
| Voxel-RCNN [6] | 91.74/- | 67.89/- | 40.80/- |
| CT3D [24] | 91.76/- | 68.93/- | 42.60/- |
| PDV [39] | 92.41/**91.99** | 69.36/68.81 | 42.16/41.48 |
| BtcDet [13] | **95.99**/- | 70.56/- | 43.87/- |
| PA-RCNN | 91.69/90.91 | **71.20/70.73** | **45.30/44.71** |

TABLE III
PERFORMANCE COMPARISON ON THE WAYMO OPEN DATASET WITH 202 VALIDATION SEQUENCES FOR PEDESTRIAN CLASS BY RANGE. †: RESULTS FROM [36]

| Methods | Pedestrian LEVEL_1 mAP/mAPH | | |
|---|---|---|---|
| | 0-30m | 30-50m | 50m-inf |
| SECOND† [19] | 74.39/- | 67.24/- | 56.71/- |
| PointPillars [20] | 67.99/- | 57.01/- | 41.29/- |
| Part-A2† [38] | 81.87/- | 73.65/- | 62.34/- |
| PV-RCNN† [2] | 83.33/- | 78.53/- | 69.36/- |
| PV-RCNN++† [36] | 84.88/- | 79.65/- | 70.64/- |
| PDV [39] | 80.32/73.60 | 72.97/63.28 | 61.69/50.07 |
| PA-RCNN | **86.13/82.17** | **80.73/74.89** | **73.14/64.57** |

| Methods | Pedestrian LEVEL_2 mAP/mAPH | | |
|---|---|---|---|
| | 0-30m | 30-50m | 50m-inf |
| PDV [39] | 75.26/68.82 | 65.78/56.85 | 47.46/38.30 |
| PA-RCNN | **81.41/77.55** | **73.62/68.11** | **59.04/51.70** |

TABLE IV
PERFORMANCE COMPARISON ON THE WAYMO OPEN DATASET WITH 202 VALIDATION SEQUENCES FOR CYCLIST CLASS BY RANGE. †: RESULTS FROM [36]

| Methods | Cyclist LEVEL_1 mAP/mAPH | | |
|---|---|---|---|
| | 0-30m | 30-50m | 50m-inf |
| SECOND† [19] | 73.33/- | 55.51/- | 41.98/- |
| Part-A2† [38] | 80.87/- | 62.57/- | 45.04/- |
| PV-RCNN† [2] | 81.10/- | 65.65/- | 52.58/- |
| PV-RCNN++† [36] | **83.65**/- | 68.90/- | 51.41/- |
| PDV [39] | 80.86/79.83 | 62.61/61.45 | 46.23/44.12 |
| PA-RCNN | 83.56/**82.47** | **70.84/69.68** | **54.40/52.91** |

| Methods | Cyclist LEVEL_2 mAP/mAPH | | |
|---|---|---|---|
| | 0-30m | 30-50m | 50m-inf |
| PDV [39] | 80.42/79.40 | 58.95/57.87 | 43.05/41.09 |
| PA-RCNN | **82.96/81.88** | **67.03/65.93** | **50.71/49.30** |

### C. *Waymo Results*

The main results on Waymo Open dataset are shown in Table I, where the comparison is made between the proposed method and the recent studies across two difficulty levels and three object categories. PA-RCNN achieves state-of-the-art performance on all columns except for Vehicle LEVEL_1, on which competitive accuracy is also obtained. Note that our method outperforms BtcDet [13], which consists of generative modules, on mAPH by 0.23% and 0.39% on both levels of the Vehicle category. It is also worth mentioning that higher improvements can be seen from vulnerable targets, which are more difficult to detect. The proposed method raises the best mAPH by 1.93% and 2.29% on Pedestrian LEVEL_1 and LEVEL_2 respectively. There is also an increase of 0.8% and 0.9% in mAPH on both levels of Cyclist. The results of the model trained with 20% of the training split are also presented to compare with VoTr-TSD [7] and Pyramid-PV [8].

Tables II, III and IV show the comparisons pertaining to the detection range. The close-range targets are easier to identify as the point density is higher, while the distant targets are more difficult to recognise as the point density decreases with the increasing distance. While achieving competitive results for close-range vehicles, a larger advantage of the proposed method can be observed on distant objects. Although BtcDet [13] with generative modules dominates 0-30$m$ in vehicle detection, PA-RCNN obtains better results for vehicles at

TABLE V

RESULTS ON KITTI *val* SET FOR CAR, PEDESTRIAN AND CYCLIST CLASSES, WITH AN AVERAGE PRECISION OF 40 RECALL POINTS (R40). *: BEST AMONG LIDAR-ONLY NON-GENERATIVE METHODS.

| Methods | Car | | | Pedestrian | | | Cyclist | | |
|---|---|---|---|---|---|---|---|---|---|
| | Easy | Mod. | Hard | Easy | Mod. | Hard | Easy | Mod. | Hard |
| RGB+LiDAR | | | | | | | | | |
| MV3D [41] | 71.29 | 62.68 | 56.56 | - | - | - | - | - | - |
| F-PointNet [17] | 83.76 | 70.92 | 63.65 | 70.00 | 61.32 | 53.59 | 77.15 | 56.49 | 53.37 |
| DVF [37] | 93.07 | 85.84 | 83.13 | - | - | - | - | - | - |
| CAT-Det [42] | 90.12 | 81.46 | 79.15 | 74.08 | 66.35 | 58.92 | 87.64 | 72.82 | 68.20 |
| LiDAR | | | | | | | | | |
| PointRCNN [5] | 88.72 | 78.61 | 77.82 | 62.72 | 53.85 | 50.24 | 86.64 | 71.62 | 65.59 |
| PointPillars [20] | 87.75 | 78.39 | 75.18 | 57.30 | 51.41 | 46.87 | 81.57 | 62.94 | 58.98 |
| SA-SSD [43] | 92.23 | 84.30 | 81.36 | - | - | - | - | - | - |
| Voxel-RCNN [6] | 92.38 | 85.29 | 82.86 | - | - | - | - | - | - |
| PV-RCNN [2] | 92.57 | 84.83 | 82.69 | 64.26 | 56.67 | 51.91 | 88.88 | 71.95 | 66.78 |
| SE-SSD [16] | 93.19 | 86.12 | 83.31 | - | - | - | - | - | - |
| PDV [39] | 92.56 | 85.29 | 83.05 | 66.90 | 60.80 | 55.85 | 92.72 | 74.23 | 69.60 |
| Generative Model | | | | | | | | | |
| PC-RGNN [25] | 90.94 | 81.43 | 80.45 | - | - | - | - | - | - |
| AGO-Net [26] | - | - | - | 60.39 | 54.81 | 50.59 | 87.57 | 69.24 | 64.74 |
| SIENet [27] | 92.49 | 85.43 | 83.05 | - | - | - | - | - | - |
| BtcDet [13] | 93.15 | 86.28 | 83.86 | 69.39 | 61.19 | 55.86 | 91.45 | 74.70 | 70.08 |
| SFD† [15] | 95.47 | 88.56 | 85.74 | - | - | - | - | - | - |
| SPG [14] | 92.53 | 85.31 | 82.82 | - | - | - | - | - | - |
| PA-RCNN(P) | 90.21 | 81.39 | 79.94 | 64.50 | 55.82 | 52.89 | 87.36 | 72.02 | 66.07 |
| PA-RCNN | 92.77* | 85.77* | 83.31* | 70.29* | 62.98* | 57.61* | 94.18* | 73.13 | 69.19 |

TABLE VI

RESULTS ON KITTI *test* SET FOR CAR, PEDESTRIAN AND CYCLIST CLASSES, WITH AN AVERAGE PRECISION OF 40 RECALL POINTS (R40). *: BEST AMONG LIDAR-ONLY NON-GENERATIVE METHODS.

| Methods | Car | | | Pedestrian | | | Cyclist | | |
|---|---|---|---|---|---|---|---|---|---|
| | Easy | Mod. | Hard | Easy | Mod. | Hard | Easy | Mod. | Hard |
| RGB+LiDAR | | | | | | | | | |
| MV3D [41] | 74.97 | 63.63 | 54.00 | - | - | - | - | - | - |
| F-PointNet [17] | 82.19 | 69.79 | 60.59 | 51.21 | 44.89 | 40.23 | 72.27 | 56.12 | 49.01 |
| DVF [37] | 90.99 | 82.40 | 77.37 | - | - | - | - | - | - |
| CAT-Det [42] | 89.87 | 81.32 | 76.68 | 54.26 | 45.44 | 41.94 | 83.68 | 68.81 | 61.45 |
| LiDAR | | | | | | | | | |
| SECOND [19] | 83.34 | 72.55 | 65.82 | 48.73 | 40.57 | 37.77 | 71.33 | 52.08 | 45.83 |
| PointPillars [20] | 82.58 | 74.31 | 68.99 | 51.45 | 41.92 | 38.89 | 77.10 | 58.65 | 51.92 |
| Part-A2 [38] | 87.81 | 78.49 | 73.51 | 53.10 | 43.35 | 40.06 | 79.17 | 63.52 | 56.93 |
| Point-RCNN [5] | 86.96 | 75.64 | 70.70 | 47.98 | 39.37 | 36.01 | 74.96 | 58.12 | 49.01 |
| 3DSSD [3] | 88.36 | 79.57 | 74.55 | 50.64 | 43.09 | 39.65 | 82.48 | 64.10 | 56.90 |
| SA-SSD [43] | 88.75 | 79.79 | 74.16 | - | - | - | | | |
| Voxel-RCNN [6] | 90.90 | 81.62 | 77.06 | - | - | - | - | - | - |
| PV-RCNN [2] | 90.25 | 81.43 | 76.82 | 52.17 | 43.29 | 40.29 | 78.60 | 63.71 | 57.65 |
| PV-RCNN++ [36] | 90.14 | 81.88 | 77.15 | 54.29 | 47.19 | 43.49 | 82.22 | 67.44 | 60.04 |
| PDV [39] | 90.43 | 81.86 | 77.36 | - | - | - | 83.04 | 67.81 | 60.46 |
| CIA-SSD [28] | 89.59 | 80.28 | 72.87 | - | - | - | - | - | - |
| VoxSet [12] | 88.53 | 82.06 | 77.46 | - | - | - | - | - | - |
| SE-SSD [16] | 91.94 | 82.54 | 77.15 | - | - | - | - | - | - |
| VoTr-TSD [7] | 89.90 | 82.09 | 79.14 | - | - | - | - | - | - |
| Pyramid-PV [8] | 88.39 | 82.08 | 77.49 | - | - | - | - | - | - |
| SASA [4] | 88.76 | 82.16 | 77.16 | - | - | - | - | - | - |
| Generative Model | | | | | | | | | |
| PC-RGNN [25] | 89.13 | 79.90 | 75.54 | - | - | - | - | - | - |
| AGO-Net [26] | 91.53 | 80.77 | 75.23 | 45.18 | 37.22 | 34.62 | 72.82 | 57.60 | 51.53 |
| SIENet [27] | 88.22 | 81.71 | 77.22 | - | - | - | 83.00 | 67.61 | 60.09 |
| BtcDet [13] | 90.64 | 82.86 | 78.09 | 47.80 | 41.63 | 39.30 | 82.81 | 68.68 | 61.81 |
| SFD† [15] | 91.73 | 84.76 | 85.74 | - | - | - | - | - | - |
| SPG [14] | 90.50 | 82.13 | 78.90 | - | - | - | - | - | - |
| PA-RCNN | 90.94 | 82.44 | 77.69 | 51.25 | 43.57 | 40.35 | 83.32* | 68.04* | 59.88 |

further than 30$m$. This may be explained by the greater effectiveness of point augmentation and object-guided sampling on instances with fewer points. The improvement in close-range performance is limited, as the objects in this range are usually comparatively more visible and comprise denser point sets. Moreover, the proposed methods show superior performance on Pedestrian and Cyclist across all ranges and difficulties except for Cyclist LEVEL_1 in 0-30$m$. A bigger margin is also observed from the most difficult Pedestrian class.

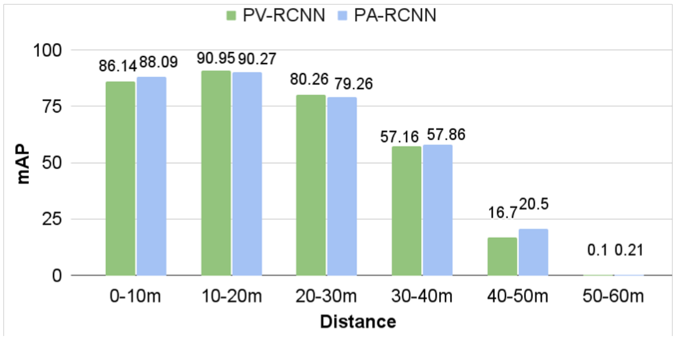


Fig. 4. Car mAP by distance on KITTI *val* set.

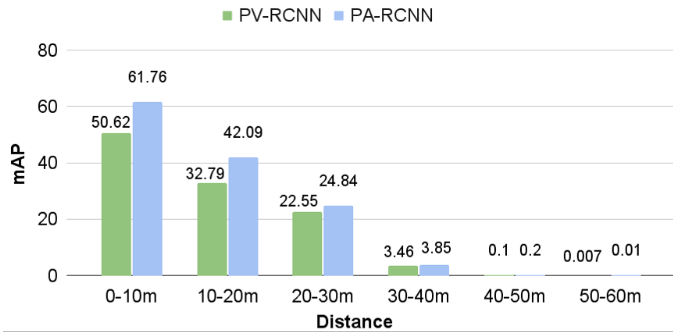


Fig. 5. Pedestrian mAP by distance on KITTI *val* set.

### D. KITTI Results

Table V illustrates the results of the KITTI *val* split. Our method improves Car mAP by +0.94% and +0.48% as compared with those of its baseline (PV-RCNN [2]) and the second best model (PDV [39]) with respect to the moderate difficulty. In both the Pedestrian and Cyclist classes, the proposed method also shows competitive performance to some of the multi-modality methods. We also include the performance of PA-RCNN(P), where the only Point Augmentation module is adapted to the PointRCNN [5] framework. Since PointRCNN is a point-only detector, we can observe the improvement solely induced by PA. However, the improvement on vulnerable objects is limited as a result of less accurate proposal bounding boxes. Following the KITTI guideline, we submit our best model to the test server.

Table VI presents the results on the KITTI testing server. While achieving an improvement on the cyclist class over the multi-class detectors (PV-RCNN [2], PV-RCNN++ [36] and PDV [39]), the proposed model obtains the best overall detection on cyclists among methods without generative modules. However, the performance enhancement in the Car category is limited compared to the state-of-the-art. This can be explained by the smaller voxel size on the KITTI dataset. Note that WOD has a voxel size of $(0.1, 0.1, 0.15)\mathrm{m}$, while KITTI has a voxel size of $(0.05, 0.05, 0.1)\mathrm{m}$. Compared to WOD, the finer-grained voxelisation on the KITTI dataset permitted by the smaller detection range allows the baseline detectors to extract information from denser feature maps. This limits the improvement provided by the proposed point augmentation and the GVPFA modules, which aim to compensate for the information loss due to the partition effects. The degradation in the Pedestrian category can also be explained by the finer input of KITTI. Furthermore, by examining the prediction results on the *val* set, some false positives regarding the Pedestrian class are actual target pedestrians visually observable from RGB images. A sample is shown on the left in Figure 7a, where the unlabelled pedestrians are correctly detected. It can be hypothesised that a similar case would be observed on the more difficult *test* set. Such results are less frequent on WOD.

Figure 4, 5 and 6 show the mAP by distance on KITTI *val* set. It is noticeable that PA-RCNN outperforms the baseline in all ranges, except for the cars from between 10 and 30 meters.

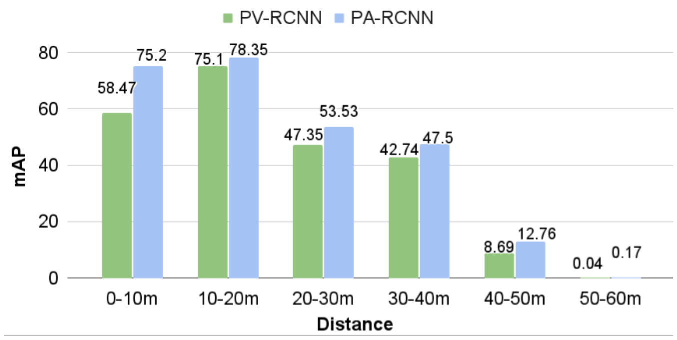


Fig. 6. Cyclist mAP by distance on KITTI *val* set.

### E. Ablation Study

This section compares the effectiveness of each component and the variation of the network.

**Effect of network components**. Table VII shows the quantitative improvement as LEVEL_2 mAPH contributed by each component on a 10% training set where Config. 1 is the baseline PVRCNN [2] network re-implemented in the OpenPCDet codebase [40]. By introducing the attention-based semantic mining module to the BEV feature map (Config. 2), a 0.68%, 1.14% and 0.46% increases are observed in the three classes respectively. The object-guided sampling method (Config. 3) gives a boost of 0.5%, 0.96% and 0.61% to the final results. By incorporating local grid feature aggregation to the refinement stage (Config. 4), additional mAPH scores of 0.68%, 0.56% and 0.67% are gained for each of the classes. The enriched point clouds estimated with the point augmentation module (Config. 5) give a significant improvement of 1.89% and 1.15% to both vulnerable categories.

**Effect of point augmentation scheme**. The point augmentation is implemented with three schemes: no transformation, rotation and mirroring as shown in Figure 1. The augmented points provide additional information on the unseen prospects of the object. Table VIII shows a comparison of different schemes. By just including all raw points around the proposal boxes (Config. 4-a), the largest improvement of 0.98% is seen on the Pedestrian LEVEL_2 mAPH. Minor improvements are observed for rotating the points around the box centres and mirroring the points about the transversal centre (yz) plane (Config. 4-b and 4-c). It is hypothesised that the orientation information can be corrupted through the transformations,

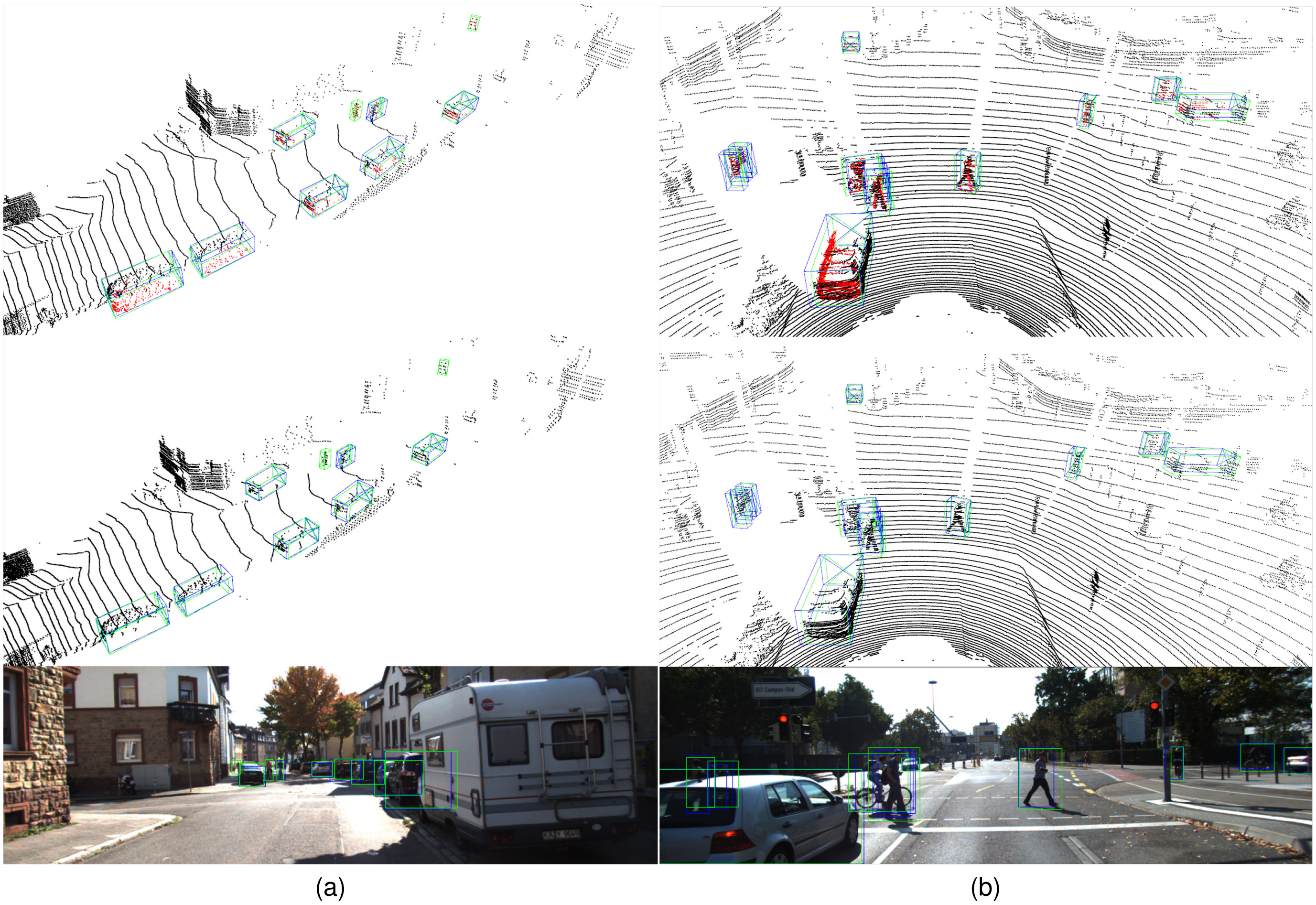


Fig. 7. Visualisations of detection results on KITTI *val* set. From top to bottom, it includes the point clouds with augmented points in red, the original point clouds and the RGB images Blue and green boxes indicate ground truths and predictions by PA-RCNN respectively.

TABLE VII
COMPONENT ANALYSIS ON 10% WAYMO OPEN DATASET. SM, OG, GB AND PA REPRESENT THE SEMANTIC MINING BEV ENCODER, OBJECT-GUIDED POINT SAMPLING, GRID-BASED VOXEL-TO-POINT FEATURE AGGREGATION AND POINT AUGMENTATION RESPECTIVELY.

| Config. | SM | OG | GB | PA | LEVEL_2 mAPH | | |
|---|---|---|---|---|---|---|---|
| | | | | | Vehicle | Pedestrian | Cyclist |
| 1 | | | | | 66.53 | 61.71 | 66.79 |
| 2 | ✓ | | | | 67.21 | 62.85 | 67.25 |
| 3 | ✓ | ✓ | | | 67.71 | 63.81 | 67.86 |
| 4 | ✓ | ✓ | ✓ | | 68.39 | 64.37 | 68.53 |
| 5 | ✓ | ✓ | ✓ | ✓ | **68.84** | **66.28** | **69.68** |

TABLE VIII
COMPARISON OF DIFFERENT IMPLEMENTATIONS OF POINT AUGMENTATION ON WAYMO OPEN DATASET.

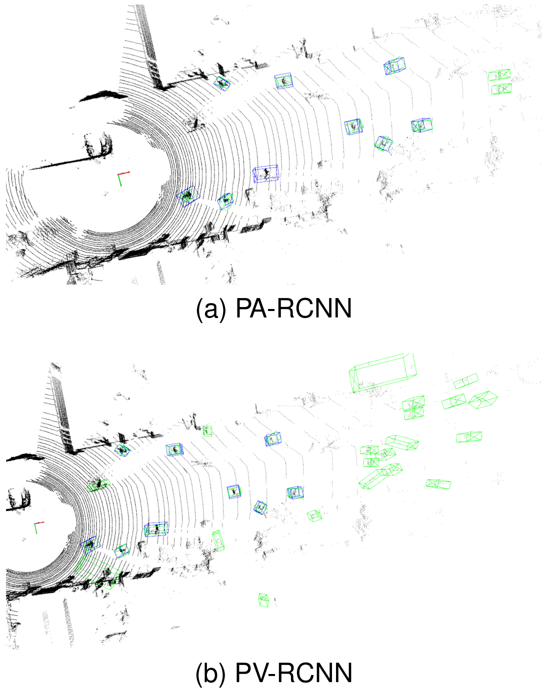
| Config. | Point augmentation scheme | LEVEL_2 mAPH | | |
|---|---|---|---|---|
| | | Vehicle | Pedestrian | Cyclist |
| 4 | Baseline | 68.39 | 64.37 | 68.53 |
| 4-a | No tranformation | 68.42 | 65.35 | 68.81 |
| 4-b | Rotation | 68.48 | 65.52 | 69.02 |
| 4-c | Mirroring (yz-plane) | 68.44 | 65.42 | 69.12 |
| 5 | Mirroring (xz-plane) | **68.84** | **66.28** | **69.68** |

TABLE IX
COMPARISON OF THE NUMBERS OF FALSE POSITIVES ON THE MODERATE DIFFICULTY OF KITTI DATASET

| Categories | #FP KITTI moderate | |
|---|---|---|
| | PV-RCNN | PA-RCNN |
| Car | 6606 | 1878 |
| Pedestrian | 7142 | 5589 |
| Cyclist | 964 | 847 |

leading to sub-optimal results on the heading weighted performances. However, the improvement provided by the increased point density has overpowered the deficiency of sub-optimal transformation. The more accurate localisation of bounding boxes compensates for the corrupted heading estimation. This can be explained by the visualisations in Figure 1, where all three transformations provide rich geometry information for locating the target accurately and directional information is degraded when the object is rotated or mirrored longitudinally.

By mirroring the point about the longitudinal centre (xz) plane (Config. 5), the symmetric characteristics can be fully utilised. Noticeable gains are achieved in all classes. Particularly, a further boost of 0.93% and 0.87% is observed in the vulnerable pedestrian and cyclist classes. Visualisations of the point augmentation are shown in Figure 7. It can be observed that the fictional points generated by the point augmentation module provide extra information on distant


(a) PA-RCNN

(b) PV-RCNN

Fig. 8. Visualizations of results from (a) PA-RCNN and (b) PV-RCNN on KITTI *val* set. Green and blue boxes indicate predictions and ground truths respectively. Significantly less false positives can be observed on the proposed PA-RCNN.

objects by estimating the complete shapes. The RGB image in Figure 7a shows that PA-RCNN can successfully detect distant pedestrians and cyclists, which are visually visible but without groundtruth labels.

### F. Qualitative Results

Figure 7 includes two samples from the KITTI *val* set. A dense point set can be seen with the augmented points generated by the PA module, achieving accurate detection on distance and vulnerable targets.

Figure 8 shows a visualisation of the detection results on a sample instance from the KITTI *val* set. Table IX summarises the numbers of false positives on KITTI dataset. With more distinctive details added by the point augmentation module, it is noticeable that the amount the false positives is drastically reduced. In addition, the inclusion of point density information also provides another confidence measure to help reduce false positives, especially on vulnerable targets.

### G. Comparison with Other Augmentation Methods

The point augmentation module in the proposed method aims to estimate more complete shapes of the targets. The same task can be achieved by a point completion (PC) module in PC-RGNN [25], where a multi-resolution graph encoder and a point pyramid decoder are used. The PC module is applied to the 3D proposals and trained with the completion loss and adversarial loss, with the discriminator aiming to distinguish fictitious points from the real point cloud. While more trainable parameters need to be considered in PC-RGNN, the proposed PA-RCNN requires no additional optimisation targets for shape estimation. SIENet [27] builds a Spatial Information Enhancement (SIE) module based on PV-RCNN. The SIE module is tasked to complete the shape of proposals from the first stage RPN. The spatial shape prediction module in SIENet consists of a PointNet-based encoder-decoder, which maps an $N \times 3$ incomplete shape to a dense and complete shape with 1024 points. The SIE module is pre-trained with samples from the external ShapeNet [44] dataset for the Car category. For the Pedestrian and Cyclist categories, training samples are taken from KITTI dataset, due to the lack of corresponding external data sources. The semantic point generation (SPG) module [14] generates augmented points based on the voxel features in the proposal regions. The point generation module is trained to map the voxel or pillar features to voxel centroids and mean point features. Similar to SIENet, BtcDet [13] estimates the complete shape of objects by leveraging the more complete objects in KITTI dataset. While the more complete objects are used as training targets in SIENet, BtcDet finds the best match from a collection of labelled objects according to a heuristic function. The points of the best match are then added to the proposal bounding box. Note that an extra database of the labelled object with complete object points is required before the training. Moreover, due to the database being only generated from KITTI dataset, the performance is limited when the point distribution is different in an unseen dataset. While most of the above methods require the design of additional training objectives, the proposed PA-RCNN model is a pure end-to-end network, where no extra trainable parameters are added for shape completion. SFD [15] augments the detection workflow by performing depth completion on RGB images. The generated pseudo clouds with RGB information realise the depth-based data augmentation. Despite that SFD achieved remarkable performance on single-class Car detection on the KITTI *test* set, PA-RCNN explores the improvement with only point cloud inputs and provides competitive multi-class detection results.

Experiments have also been conducted for feature-level augmentation. However, it requires additional memory for operation and provides a limited improvement on the final results in our investigations. Based on the study of related works, some existing methods have explored feature-level augmentation, such as AGO-Net [26], BtcDet [13] and SFD [15]. AGO-Net [26] uses the conceptual-perceptual approach and is trained in a self-contained manner. The conceptual network is trained with a fully augmented dataset, where the incomplete objects are replaced by their closest pairs with appropriate transformation. The perceptual network is trained with the original dataset, with an additional loss for feature adaptation. The parameters of the perceptual network are then adjusted in accordance with the conceptual features, which are generated by the conceptual network from the same training samples with full augmentation. BtcDet [13] creates a database for the occluded regions, which is used to train the model for estimating occupancy probability. The occupancy probability map is used as additional features to boost the main detection performance. SFD [15] performs feature level augmentation by generating pseudo point clouds with additional image inputs, which are encoded by the Colour Point Convolution

(CPConv) [15] to obtain the pseudo RoI features. Feature level augmentation provides significant improvements on detection accuracy with the help of additional inputs or hand-crafted optimisation targets. However, neither model pre-training nor an extra database is required to train PA-RCNN end-to-end.

## V. Conclusion

This article presents a two-stage detection network for intelligent vehicles incorporating the enhanced feature encoding and aggregation scheme to focus on detecting pedestrians and cyclists. A shape estimation module with no trainable parameters is introduced to remedy the point sparsity and signal loss on vulnerable road users. Promising results on KITTI and Waymo Open datasets show the effectiveness of each component of the architecture. The compatibility of the current model allows us to adapt the proposed method to more 3D detection backbones in the future. However, the improvement is limited when the voxelisation is more sophisticated. In the context of smaller voxels, 1) the error between voxel centres and the actual point locations is smaller; and 2) there is less discrepancy in point density in each voxel. Furthermore, while the proposed method induces considerably less computation burden compared to generative models, there is still a future plan to optimise the point augmentation module for better memory usage and inference time. The extra reduction in computational resources used by the new modules can facilitate the deployment of the algorithm to onboard computers of intelligent vehicles. The scalability of the framework should also be investigated to incorporate a larger backbone for more complex scenes.

## References

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[2] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "PV-RCNN: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 526–10 535.

[3] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-based 3d single stage object detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 040–11 048.

[4] C. Chen, Z. Chen, J. Zhang, and D. Tao, "SASA: Semantics-augmented set abstraction for point-based 3d object detection," in *AAAI Conference on Artificial Intelligence*, vol. 1, 2022.

[5] S. Shi, X. Wang, and H. Li, "PointRCNN: 3d object proposal generation and detection from point cloud," in *in Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 770–779.

[6] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel R-CNN: Towards high performance voxel-based 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1201–1209.

[7] J. Mao, Y. Xue, M. Niu, H. Bai, J. Feng, X. Liang, H. Xu, and C. Xu, "Voxel transformer for 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3164–3173.

[8] J. Mao, M. Niu, H. Bai, X. Liang, H. Xu, and C. Xu, "Pyramid R-CNN: Towards better performance and adaptability for 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2723–2732.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[10] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[11] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[12] C. He, R. Li, S. Li, and L. Zhang, "Voxel set transformer: A set-to-set approach to 3d object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8417–8427.

[13] Q. Xu, Y. Zhong, and U. Neumann, "Behind the curtain: Learning occluded shapes for 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, 2022, pp. 2893–2901.

[14] Q. Xu, Y. Zhou, W. Wang, C. R. Qi, and D. Anguelov, "SPG: Unsupervised domain adaptation for 3d object detection via semantic point generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 446–15 456.

[15] X. Wu, L. Peng, H. Yang, L. Xie, C. Huang, C. Deng, H. Liu, and D. Cai, "Sparse fuse dense: Towards high quality 3d detection with depth completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5418–5427.

[16] W. Zheng, W. Tang, L. Jiang, and C.-W. Fu, "SE-SSD: Self-ensembling single-stage object detector from point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 494–14 503.

[17] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 918–927.

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

[19] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/10/3337

[20] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.

[21] J. Li, H. Dai, L. Shao, and Y. Ding, "From voxel to point: Iou-guided 3d object detection for point cloud with voxel-to-point decoder," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4622–4631.

[22] Y. He, G. Xia, Y. Luo, L. Su, Z. Zhang, W. Li, and P. Wang, "DVFENet: Dual-branch voxel feature extraction network for 3d object detection," *Neurocomputing*, vol. 459, p. 201–211, 2021.

[23] L. Wang, C. Wang, X. Zhang, T. Lan, and J. Li, "S-AT GCN: spatial-attention graph convolution network based feature enhancement for 3d object detection," *arXiv preprint arXiv:2103.08439*, 2021.

[24] H. Sheng, S. Cai, Y. Liu, B. Deng, J. Huang, X.-S. Hua, and M.-J. Zhao, "Improving 3d object detection with channel-wise transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2743–2752.

[25] Y. Zhang, D. Huang, and Y. Wang, "PC-RGNN: Point cloud completion and graph neural network for 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, 2021, pp. 3430–3437.

[26] L. Du, X. Ye, X. Tan, E. Johns, B. Chen, E. Ding, X. Xue, and J. Feng, "AGO-Net: Association-guided 3d point cloud object detection network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8097–8109, 2021.

[27] Z. Li, Y. Yao, Z. Quan, J. Xie, and W. Yang, "Spatial information enhancement network for 3d object detection from point cloud," *Pattern Recognition*, vol. 128, p. 108684, 2022.

[28] W. Zheng, W. Tang, S. Chen, L. Jiang, and C.-W. Fu, "CIA-SSD: Confident iou-aware single-stage object detector from point cloud," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, 2021, pp. 3555–3562.

[29] W. Zheng, L. Jiang, F. Lu, Y. Ye, and C.-W. Fu, "Boosting single-frame 3d object detection by simulating multi-frame point clouds," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 4848–4856.

[30] W. Lu, D. Zhao, C. Premebida, W.-H. Chen, and D. Tian, "Semantic feature mining for 3d object classification and segmentation," in

*IEEE International Conference on Robotics and Automation*, 2021, pp. 13 539–13 545.

[31] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[32] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 510–519.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[34] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

[35] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020.

[36] S. Shi, L. Jiang, J. Deng, Z. Wang, C. Guo, J. Shi, X. Wang, and H. Li, "PV-RCNN++: Point-voxel feature set abstraction with local vector representation for 3d object detection," *arXiv preprint arXiv:2102.00463v2*, 2021.

[37] A. Mahmoud, J. S. Hu, and S. L. Waslander, "Dense voxel fusion for 3d object detection," *arXiv preprint arXiv:2203.00871*, 2022.

[38] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 8, pp. 2647–2664, 2020.

[39] J. S. Hu, T. Kuai, and S. L. Waslander, "Point density-aware voxels for lidar 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8469–8478.

[40] OpenPCDet Development Team, "OpenPCDet: An open-source toolbox for 3d object detection from point clouds, Ver. 0.5.2," 2022. [Online]. Available: https://github.com/open-mmlab/OpenPCDet

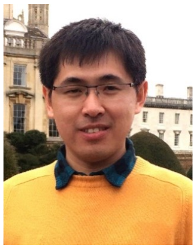
[41] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.

[42] Y. Zhang, J. Chen, and D. Huang, "CAT-Det: Contrastively augmented transformer for multi-modal 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 908–917.

[43] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, "Structure aware single-stage 3d object detection from point cloud," in *Proceedings of the IIEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 873–11 882.

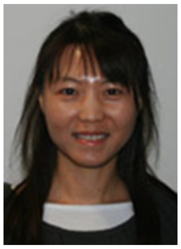
[44] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Stanford University — Princeton University — Toyota Technological Institute at Chicago, Tech. Rep. arXiv:1512.03012 [cs.GR], 2015.

**Weihao Lu** received the M.Eng. degree in Aeronautical Engineering from Imperial College London, UK, in 2018. He is currently pursuing a Ph.D. degree with the Department of Autonomous Systems & Connectivity, University of Glasgow, Glasgow, UK. His research interests include autonomous driving, 3D point cloud processing and 3D object detection.



**Dezong Zhao (Senior Member, IEEE)** received the B.Eng. and M.S. degrees from Shandong University in 2003 and 2006, respectively, and the Ph.D. degree from Tsinghua University in 2010, all in Control Engineering. He is currently a Senior Lecturer in Autonomous Systems with the University of Glasgow. His research interests include Connected and Automated Vehicles, Robotics, Machine Learning and Control Engineering. He has been an EPSRC Innovation Fellow since 2018 and a Royal Society-Newton Advanced Fellow since 2020.

**Cristiano Premebida** is Assistant Professor in the Department of Electrical and Computer Engineering (DEEC) at the University of Coimbra (UC), Portugal, where he is a member of the Institute of Systems and Robotics (ISR-UC). C. Premebida is member of the IEEE ITS and RAS societies. His main research interests are autonomous systems, intelligent vehicles, robotic perception, machine learning, and sensor fusion.



**Li Zhang (Senior Member, IEEE)** is a Reader in Department of Computer Science, Royal Holloway, University of London, UK. She received a PhD degree from the University of Birmingham, UK. She holds expertise in machine learning, deep learning, computer vision, and intelligent robotics.

**Wenjing Zhao** received Ph.D. degree in traffic engineering with Central South University in 2022. She is currently a Postdoctoral Fellow at the Department of Civil and Environmental Engineering of Hong Kong Polytechnic University. Her research interests include traffic safety, connected and autonomous vehicles, and human factors.

**Daxin Tian (Senior Member, IEEE)** is currently a Professor with the School of Transportation Science and Engineering, Beihang University, Beijing, China. His current research interests include mobile computing, intelligent transportation systems, vehicular ad hoc networks, and swarm intelligence. He is an IEEE Intelligent Transportation Systems Society Member and an IEEE Vehicular Technology Society Member.

# Appendix

## More Qualitative Results

In this section, more visualisations are provided. Figure 9a shows that the PA-RCNN has the capability of detecting distant objects. Effective pedestrian detection is seen from Figure 9b. More complicated scenarios can be observed in Figure 10 for the Waymo Open Dataset. It is noticeable that PA-RCNN is able to accurately locate distant pedestrians and vehicles with occlusions and incomplete shapes in crowded urban scenes. Figure 11 depicts the point augmentation on the pedestrian class.

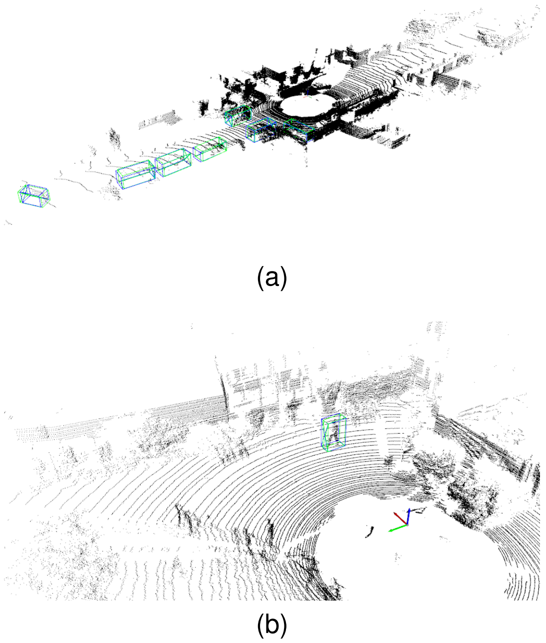
Figure 14 shows the visualisations of proposals generated by PA-RCNN for a complex sample with a number of occlusions and overlaps in the KITTI *val* split. Figure 13 shows the visualisation of proposals with respective three transformation schemes for the above example. For clarity, only 50 proposals are displayed in green. Figure 14 shows that the sub-optimal transformation scheme leads to the false positive detection on several ambiguous targets (*i.e.* humans sitting in the background highlighted in orange) in a complicated scene. Figure 12 shows the comparisons of final detection results. It can be seen that the optimal transformation scheme Mirror (yz-plane) is effective in reducing false positives, especially in distance.

## Network Efficiency

Table X shows the comparisons of inference speeds and numbers of parameters on the KITTI dataset. The inference speeds are measured with a single GTX 1080 GPU. With only 26 ms added to the baseline, the proposed method maintains efficiency, while achieving better detection accuracies. Note that the inference speed can be largely improved by using a more powerful GPU.

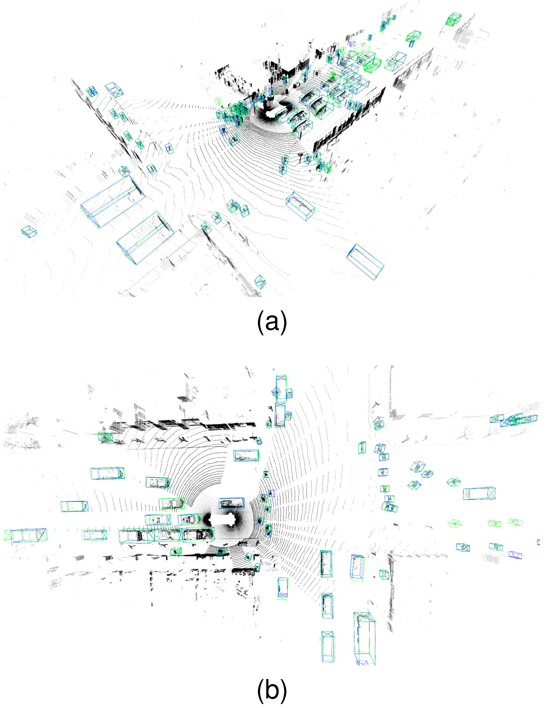
TABLE X
Comparison of inference speeds and numbers of model parameters on KITTI dataset.

| Methods | Inference Speed (ms) | #Parameters |
|---|---|---|
| PV-RCNN | 182 | 13.1M |
| PA-RCNN | 208 | 16.2M |


(a)

(b)

Fig. 9. Visualisations of detection results on KITTI *val* set on (a) Car and (b) Pedestrian category. Blue and green boxes indicate ground truths and predictions by PA-RCNN respectively.


(a)

(b)

Fig. 10. Visualisations of two detection samples on Waymo Open Dataset. Both samples are able to show the performance of PA-RCNN in complicated urban scenarios. Blue and Green boxes indicate ground truths and predictions by PA-RCNNs respectively.

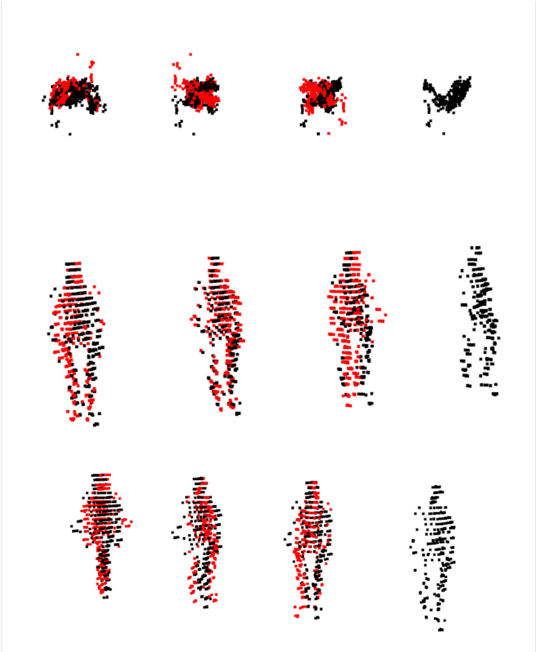
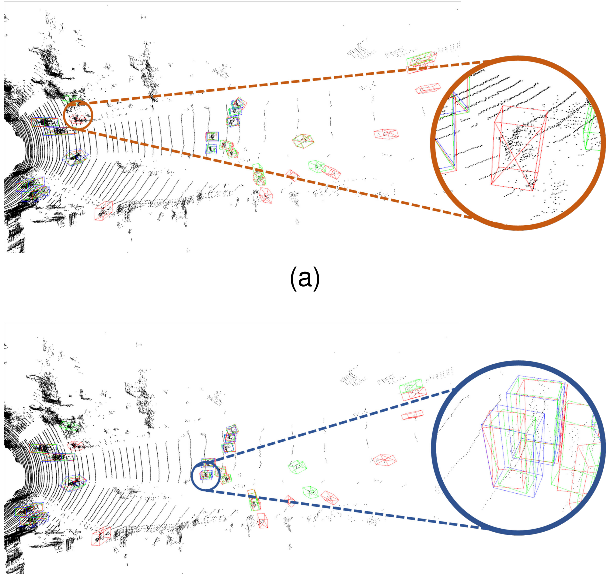


Fig. 11. Shape completion on a pedestrian with Point Augmentation module utilising symmetry. From top to bottom, there are one top view and two frontal views. From right to left, it includes Original, Mirroring (xz-plane), Mirroring (yz-plane), and Rotation displays. Original and augmented points are in black and red respectively.



(a)

(b)

Fig. 12. Visualisations of detection of different transformation schemes on KITTI *val* split. The two images show the comparison of the ground truths and Mirror (yz-plane) with Rotation in (a) and Mirror (xz-plane) in (b), respectively. Ground truth bounding boxes are in blue and the predicted bounding boxes with Mirror (yz-plane) scheme are in green. The bounding boxes in red are predicted with Rotation and Mirror (xz-plane) schemes in (a) and (b), respectively. Circles in orange and blue highlight the sub-optimal detection examples. The front-facing direction is shown by a cross on the bounding box surface.

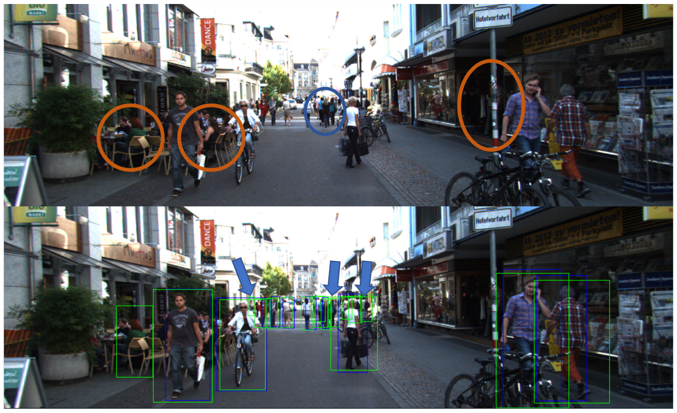
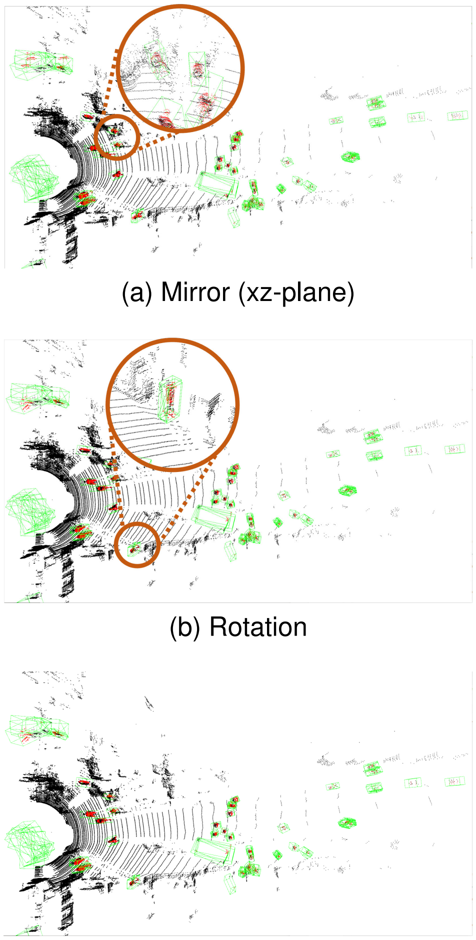


Fig. 13. Visualisations of detection generated by PA-RCNN on KITTI *val* split split for an example image with occlusions and overlaps. Top: the origin image. Bottom: image with the annotations of ground truth bounding boxes in blue and predicted bounding boxes in green. Arrows indicate the correctly detected targets, which are not labelled in the dataset. Orange circles highlight the ambiguous targets associated with the wrong detection shown in Figure 14. The blue circle indicates the target, whose front-facing direction is incorrectly predicted with a Mirror (xz-plane) transformation shown in Figure 12b.



(a) Mirror (xz-plane)

(b) Rotation

(c) Mirror (yz-plane)

Fig. 14. Visualisations of proposals with augmented points on KITTI *val* split. Proposal bounding boxes are in green and the augmented points are in red. Orange circles depict the details of the wrong detection due to the sub-optimal transformation scheme.