HT-SSPG:Hierarchical Transformers for Semantic Surface Point Generation in 3D Object Detection

Wenhao Kong[1][0009−0003−1346−615X] and Xiaowei Zhang(✉)2[0000−0003−4854−3736]

Qingdao University, Qingdao, China
kongwenhao@qdu.edu.cn,by1306114@buaa.edu.cn

Abstract. Currently, the incomplete point cloud structure in LiDAR point clouds has become the primary challenge for improving detector performance. Point cloud completion methods address this issue by adding more points to regions of interest, however, due to imprecise proposals and coarse feature extraction methods, these approaches often generate numerous low-quality points, which limits detection performance. To tackle this issue, we propose a hierarchical transformers for semantic surface point generation in 3D object detection (HT-SSPG), leveraging a voxel supervised network (VSN) and a hierarchical attention refinement (HAR) network to generate high-quality proposals and complete semantic surface points for precise detection. Specifically, the VSN enhances the backbone network's perception of spatial structures using 3D heatmaps, capturing complete structural and positional information of missing objects. The HAR module effectively integrates voxel and point cloud features using cross-attention transformers to accurately estimate the complete shape and position of objects, thus generating high-quality semantic surface points for precise detection. Extensive experiments demonstrate that our HT-SSPG achieves leading performance on the KITTI dataset. Compared to PG-RCNN, our method significantly improves detection accuracy for small objects such as pedestrians and cyclists. Specifically, it outperforms in pedestrian detection by 8.46% AP and 8.08% AP at moderate and hard levels, respectively.

Keywords: 3-D object detection · Semantic point generation · Hierarchical Transformers

1 Introduction

As advancements in artificial intelligence continue, 3D object detection is becoming increasingly crucial. However, due to the complex diversity of three-dimensional objects, achieving efficient and precise 3D object detection still faces significant challenges.

In the field of 3D object detection, LiDAR-based detectors dominate. Based on the process of bounding box generation, these detectors can be categorised

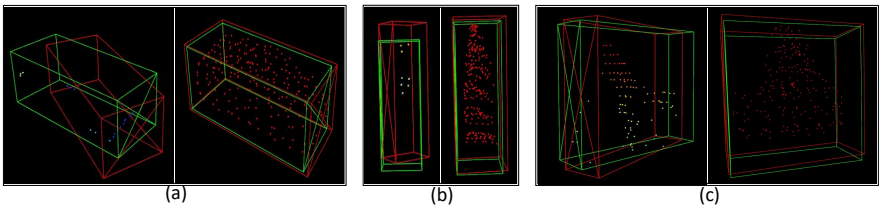

Fig. 1: (a), (b), and (c) show point cloud generation for cars, cyclists, and pedestrians. The left shows the original incomplete point cloud, while the right shows the generated point cloud. The generated semantic foreground is in red, the predicted bounding box is in red, and the ground truth box is in green.

into single-stage and two-stage detectors. A single-stage detector quickly determines the position and outline of the object but with lower accuracy. The two-stage detector first generates proposals and then refines them to achieve high-precision object detection. In recent years, with the application of Transformer [30], the capture of global features allows the detector to better understand contextual dependencies, which significantly improves the performance of point cloud-based detectors.

The sparsity and occlusion issues of 3D point clouds significantly impair the performance of LiDAR-based point cloud detectors. To address this challenge, researchers have attempted to generate new points to complete the point cloud structure. PG-RCNN [12] enhances precise detection capabilities by augmenting the regions of interest (ROI) with additional points to restore their integrity. However, the accuracy of proposals during the point generation process cannot be guaranteed. Furthermore, the generation process only considers spatial context collected from coarse voxelization within the ROI, where voxels suffer from information and geometric detail loss during quantization. Therefore, it cannot be guaranteed that the generated points match the actual situation, which limits the detection accuracy.

In order to generate high-quality semantic surface points that match the real-world situation, it is especially crucial to extract accurate and semantically rich ROI point features. Based on this concept, we propose a hierarchical transformers for semantic surface point generation in 3D object detection (HT-SSPG), a two-stage based point cloud generation network. This method is capable of fusing global and local features at different granularities through a cross layer attention mechanism to generate semantic surface points with high quality. As depicted in the Fig.1, the red points represent the generated foreground semantic points. Specifically, our approach mainly consists of a voxel supervision network (VSN) and a hierarchical attention refinement (HAR) module. The VSN is based on a 3D heatmap, enhancing the feature representation of the backbone network through supervised loss backpropagation. This helps the model capture complete shape and positional information of incomplete object and generate precise proposals. Additionally, the network does not directly participate in the inference

stage, making it more flexible. The HAR effectively integrates convolutional voxel features with original point cloud features. Unlike fusion method of PV-RCNN [25], our approach does not require strict alignment of voxel and point cloud features. Specifically, we employ two branches for extracting voxels and original point cloud features alternately. The point cloud branch can partition the input space freely without needing alignment with voxel features. Subsequently, our hierarchical cross-attention mechanism uses queries from one branch to match structures in the other, aggregating features of different granularities. Subsequently, by utilizing the aggregated ROI features, the shape and position of the object are accurately estimated, generating semantic surface points with high-quality foreground probabilities for object detection. Moreover, the module utilizes the previous stage's output to refine the proposal in the subsequent stage, thereby enhancing inter-stage correlation. By leveraging multi-level fusion features, our module produces more realistic semantic surface points.

Extensive experiments on the KITTI [7] dataset and Waymo Open Dataset [29] demonstrate the effectiveness of HT-SSPG. Our contributions are mainly:

- We proposed a flexible voxel supervised network based on 3D heatmaps, significantly enhancing the model's perceptual capabilities, aiding in precise proposal generation, and incurring no additional computational costs during inference.
- We propose extracting features sequentially from convolutional voxel layers and raw point clouds, utilizing hierarchical cross-attention transformers to aggregate features of varying granularities. These features are employed to generate high-quality semantic surface points with foreground probability, facilitating precise 3D object detection.
- HT-SSPG achieved competitive performance on KITTI [7] test set, with a bird's-eye view accuracy of 95.57% AP for easy car.

2 Related work

2.1 3D Object Detection based on LiDAR

In recent years, 3D object detection based on LiDAR point clouds has developed rapidly. According to different representations of point cloud learning, these methods can be divided into two categories: point-based and voxel-based.

Point-based methods [40,15,21,27] directly sample the raw point cloud and utilize symmetric operators to extract features for detection. Most point-based methods employ PointNet [22] and its variants as the backbone network. PointRCNN [47] proposed a 3D region proposal network that optimizes the proposals generated in the previous stage through precise regression branches, ultimately producing high-quality detection boxes. CenterPoint [39] innovatively discards proposal generation and directly regresses detection boxes by predicting foreground points. Although these methods retain as much of the original point cloud geometry as possible, the large number of point clouds limits their real-time performance. Voxel-based methods [4,37,38,41] convert irregular point clouds into

compact 3D voxels and then process the features using 3D CNNs. VoxelNet [48] is a pioneering work of this approach, introducing the voxel feature encoding method. Subsequently, SECOND [35] improved computational efficiency through enhanced sparse convolutions. Voxel R-CNN [3] is also a classic work in this category, optimizing the proposals from the previous stage through voxel ROI pooling in the second stage. While these methods are effective at generating accurate 3D proposals, the information loss caused by data quantization remains a significant challenge. Therefore, voxel-based methods are often combined with point-based methods [1,9,10]. PV-RCNN [25] aggregates voxel features on sampled key points to generate more discriminative features and uses these features to optimize proposals, efficiently combining the effectiveness of 3D sparse convolution with the flexible receptive field of PointNet-based methods. Although this method has achieved outstanding results, the sparse point clouds sampled from distant objects and occlusions remain significant challenges.

2.2 Point Generation for 3D Object Detection

Some recent work aims to optimise network performance by increasing the number of point clouds in a lidar scene to overcome the challenges posed by point cloud sparsity and incompleteness. Among the multimodal approaches, methods such as MSMDFusion [11] and MVP [40] exploit the rich semantic information in RGB images to generate a large number of pseudo-point clouds to compensate for the sparsity of the original point cloud. Among them, MSMDFusion [11] generates high-quality virtual points by extracting multiscale features from both modalities and performing LiDAR camera interactions in multiscale voxel space to combine the multi-granularity information in multimodality. In addition, MVP [40] first semantically segments the RGB image and then projects the point cloud into the image coordinates to generate virtual points based on the point cloud located in the foreground solid regions. In terms of unimodality, methods such as SIENet [17] utilise an additional point cloud generation network to generate high quality points that form plausible target shapes. Specifically, SIENet [17] inputs the original point cloud coordinates from the proposal into the network through the PCN [42] as a backbone network, and utilises the point set transform in Part - A^2 [26] Net and feature encoding in PointNet [22] to output a set of point coordinates to enhance the representation of spatial information.

3 Methods

Our proposed HT-SSPG is a two-stage detector, as illustrated in Fig.2. In the first stage, we employed the network structure of Voxel R-CNN [3] to process voxel inputs through sparse convolution. After 1×, 2×, 4× and 8× downsampling, voxel feature maps $\boldsymbol{f}_{(1)}$, $\boldsymbol{f}_{(2)}$, $\boldsymbol{f}_{(3)}$ and $\boldsymbol{f}_{(4)}$ are generated. Subsequently, we compress $\boldsymbol{f}_{(4)}$ along the height dimension to generate bird's eye view (BEV) feature maps, which was then fed into a 2D backbone network to generate proposals. Based on this, we design a Voxel Supervised Network (VSN) that aims

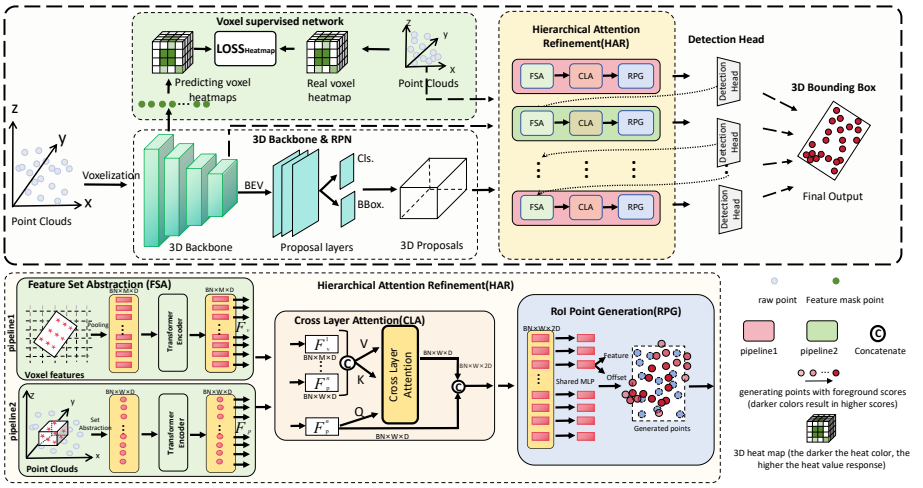

Fig. 2: Overall framework of HT-SSPG. Firstly, RPN generates 3D proposals, and a voxel supervision network enhances the backbone's perception. These proposals are then refined through HAR, with CLA fusing multi-scale features and RPG generating semantic surface points, leading to detection results.

to improve the spatial structure perception of the 3D backbone network. Specifically, during the training phase, the branch maps the feature maps generated by the backbone network onto the original point cloud to generate spatial-aware 3D heat maps. Subsequently, the supervised loss is computed by comparing with the 3D real heat map generated from the original point cloud. Finally, utilizing the backpropagation gradients from the supervised loss enhanced the representation capability of the backbone network, facilitating precise proposal generation.

In the second stage, we have designed a Hierarchical Attention Refinement (HAR) for proposal refinement, which consists of three parts: the Feature Set Abstraction module (FSA), the Cross Layer Attention module (CLA), and the Roi Point Generation module (RPG). In the FSA, we design two types of branches to independently aggregate voxel features $\boldsymbol{F}_v$ and raw point cloud features $\boldsymbol{F}_p$ at ROI grid points, and self-attention augmentation is applied to the ROI features to enhance the global dependency capability. Subsequently, the CLA part utilises the cross layer attention mechanism to fuse different granularity features from voxels and point clouds. Finally, the RPG part applies multilayer perceptron (MLP) to the points of the converged features to estimate the offset and semantic information of the points, respectively. Finally, the detection head utilizes the generated semantic surface points to produce the final detection boxes.

3.1 Voxel Supervised Network

The voxel supervised network is based on 3D heatmaps, aiming to enhance the perception ability of the backbone network for the complete structure and po-

sition of missing objects, and promote the generation of accurate proposals. Specifically, we use the heat value response $\boldsymbol{H}$ to respond to the distance of a point to the object centre, and specify that the heat value response is prescribed between [0,1]. As the value increases, it indicates that the point is closer to the centroid of the object. When the calorific value is '1', it indicates that the point is located at the centre of the object, on the contrary, when it is '0', it indicates that the point is a background point. Therefore, our designed heat value response effectively reflects the complete shape of objects, providing new ideas for enhancing the effectiveness of detecting incomplete objects. The branch consists of two parts: a predictive heat map generator and a real heat map generator.

Predictive Heat Map Generator. In this module, we predict that the heat map generator is embedded directly into the voxel feature maps generated by the 3D backbone network. The features in$\boldsymbol{f}_{(2)}$,$\boldsymbol{f}_{(3)}$,$\boldsymbol{f}_{(4)}$ are mapped to the original voxel space coordinates through the reverse computational data quantisation process, and the feature representation $\boldsymbol{F}_i^2 = \{\boldsymbol{P}_{\mathrm{i}}, \boldsymbol{f}_i^2\}, \boldsymbol{F}_i^3 = \{\boldsymbol{P}_{\mathrm{i}}, \boldsymbol{f}_i^3\}, \boldsymbol{F}_i^4 = \{\boldsymbol{P}_{\mathrm{i}}, \boldsymbol{f}_i^4\}$, where $\boldsymbol{F}_i^2$ denotes the i-th feature generated by mapping $\boldsymbol{f}_{(2)}$ to the original voxel space, $\boldsymbol{P}_{\mathrm{i}}$ denotes the position of the i-th feature in the original voxel coordinates, and $\boldsymbol{f}_i^2$ denotes the i-th feature's semantic feature. Subsequently, we concatenate these features to generate $\boldsymbol{F}_i = \{\boldsymbol{P}_i, \boldsymbol{f}_i^2 : \boldsymbol{f}_i^3 : \boldsymbol{f}_i^4\}$, and ultimately apply MLP to each feature to predict the calorific response $\hat{\boldsymbol{H}}$ for each voxel point.

Real Heat Map Generator. First, we voxelise the point cloud data in a voxel coordinate system with the lidar position as the origin, and we specify that this voxel block is represented using the position of the centroid of each voxel block. First, for a voxel in the background, we specify its thermal response to be '0'. Next, for any voxel $\boldsymbol{V}$ in an object, assuming its position is (V_x, V_y, V_z), the coordinates of the object centre are $\boldsymbol{Q}$=(Q_x, Q_y, Q_z), and the length, width and height of the object are (W, L, H), the calorific response of this voxel is defined as:

$$\boldsymbol{H} = \exp[-\frac{1}{2}(\boldsymbol{V}-\boldsymbol{Q})^T \sum{}^{-1}(\boldsymbol{V}-\boldsymbol{Q})], \tag{1}$$

where $(\boldsymbol{V}-\boldsymbol{Q}) = (V_x - Q_{\mathrm{x}}, V_y - Q_y, V_z - Q_z)$, The specific formula for the covariance matrix is:

$$\Sigma = \begin{bmatrix} \frac{W^2+L^2}{4} & & \\ & \frac{W^2+H^2}{4} & \\ & & \frac{H^2+L^2}{4} \end{bmatrix}. \tag{2}$$

After calculating the heat map response $\hat{\boldsymbol{H}}$ of the predicted heat map and the heat map response $\boldsymbol{H}$ of the real heat map, we use the smooth L1 loss function to calculate the loss value between them. The backpropagation of the loss values can significantly improve the ability of the 3D backbone network to perceive the overall structure of the object, thus effectively enhancing the model's ability to detect missing objects.

3.2 Hierarchical Attention Refinement

In a two-stage detector, the refinement stage typically extracts voxel features in the ROI. Subsequently, the refinement of the proposal is achieved through the detection head. It should be noted that although these methods can effectively extract geometric information, due to the quantization loss of voxels, many details in the original point cloud will be lost. Our research shows that these detailed information are crucial for generating high-quality points. Based on this idea, in order to capture more fine-grained semantic features, we design the Feature Set Abstraction (FSA) module using two independent branches to extract lower resolution voxel features and fine-grained raw point cloud features. Cross Layer Attention (CLA) further utilises cross-attention to integrate features from different granularities. Subsequently, semantic surface points with foreground probabilities are generated by the RoI Point Generation (RPG) module, and finally these points are directly fed into the feed-forward network (FFNs) for processing to output the object detection boxe. However, we found that it is difficult to achieve complete shape complementation after only one refinement. Therefore, we introduced the concept of cascading, where the detection results from each stage are used as proposal for the next stage to guide the refinement, and are simultaneously used to select the final high-quality detection boxes.

Feature Set Abstraction. The module is designed with voxel branching and point cloud branching for capturing features. For voxel branching, the proposal is first projected onto the voxel feature map, and then the proposal region is divided into a regular $G \times G \times G$ grid with a provision to use the centre point of the grid to represent the grid. Given a grid point $\boldsymbol{g}_i$, a set of neighbouring voxels with a quantity of k $\boldsymbol{T}^i = \{\boldsymbol{v}_1^i, \boldsymbol{v}_2^i, \boldsymbol{v}_3^i ... \boldsymbol{v}_k^i\}$ is obtained using voxel query, and subsequently the voxel features are aggregated to grid point $\boldsymbol{g}_i$ using PointNet [22]. The formula is $\boldsymbol{F}_{\mathrm{v}}^i = \{\max\{\psi([\boldsymbol{v}_x^i - \boldsymbol{g}_i; \boldsymbol{\theta}_{\mathrm{x}}^i])\}\}_{x=1}^k$, where $\psi(.)$ denotes the multilayer perceptron (MLP), $\boldsymbol{v}_x^i - \boldsymbol{g}_i$ denotes the relative coordinates of the grid point $\boldsymbol{g}_i$ to the x-th neighbouring voxel, and $\boldsymbol{\theta}_{\mathrm{x}}^i$ denotes the features of the x-th voxel. It is worth noting that we use the same operation to extract features on $\boldsymbol{f}_{(2)}$, $\boldsymbol{f}_{(3)}$, $\boldsymbol{f}_{(4)}$ respectively, which ultimately connects the aggregated features from different scales to obtain the ROI features.

Point cloud branching is similar to voxel branching, but we first project the proposal back into the original point cloud, and then use farthest point sampling (FPS) to select J keypoints from the original point cloud within the proposal. Subsequently, we query a set of point cloud $\boldsymbol{P}^j = \{\boldsymbol{p}_1^j, \boldsymbol{p}_2^j, \boldsymbol{p}_3^j ... \boldsymbol{p}_n^j\}$ in a spherical region of radius r, using the j-th keypoint as the center of the sphere, and n is the number of point clouds. Finally, local features are encoded as $\boldsymbol{F}_p^j$ at the keypoints using PointNet [22]. After extracting $\boldsymbol{F}_v^i$ and $\boldsymbol{F}_p^j$ respectively, we further process the features using a self-attention encoder in order to capture the context dependent information of ROIs. The refined features can be represented as (in terms of voxel branching):

$$\boldsymbol{F}_{\mathrm{v}} = \mathit{Attention}(\hat{\boldsymbol{Q}}, \hat{\boldsymbol{K}}, \hat{\boldsymbol{V}}), \tag{3}$$

where $\hat{\boldsymbol{Q}} = \boldsymbol{W}_q \boldsymbol{F}_v^i$, $\hat{\boldsymbol{K}} = \boldsymbol{W}_k \boldsymbol{F}_v^i$, $\hat{\boldsymbol{V}} = \boldsymbol{W}_v \boldsymbol{F}_v^i$, and $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, $\boldsymbol{W}_v$ are learnable parameters.

Cross Layer Attention. In order to make more effective use of voxel features and point cloud features, we discard the method of directly connecting the two features in the PV-RCNN [25], and design a cross layer attention transformer to achieve the aggregation of local and global features. Specifically, in the voxel branch, we use the voxel features extracted in that stage as queries and the point cloud features captured in the previous stage as keys and values. The voxel features are used to query the related point cloud features and finally generate the fusion features determined by the voxel features. On the contrary, in the point cloud branch, we use the point cloud features extracted in that stage as a query and the voxel features extracted in the previous stage as keys and values to output the fused features. We give the specific formulae using voxel branching as a column:

$$\begin{aligned} \boldsymbol{F} &= \boldsymbol{F}_v + MHCA(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) \\ &= \boldsymbol{F}_v + [\boldsymbol{Head}^1, ..., \boldsymbol{Head}^d]\boldsymbol{W}_O, \end{aligned} \tag{4}$$

where $\boldsymbol{Q} = \boldsymbol{W}_Q \boldsymbol{F}_v$ is used as a query and $\boldsymbol{K} = \boldsymbol{W}_K \boldsymbol{F}_p, \boldsymbol{V} = \boldsymbol{W}_V \boldsymbol{F}_p$ are keys and values respectively. $\boldsymbol{Head}^i$ denotes the cross attention module, $\boldsymbol{W}_{\{Q,K,V,O\}}$ is the learnable parameter.

RoI Point Generation. This module uses the extracted features to generate high-quality semantic surface points with foreground probabilities. Specifically, we apply the two-layer $\psi(.)$(MLP) to the features $\boldsymbol{F}$, resulting in the generated feature points with positional offsets $\boldsymbol{O}_i$ and semantic features $\boldsymbol{Y}_i$. The coordinates of the points generated based on this can then be expressed in terms of the feature point coordinates add the positional offset. The foreground probability score $\boldsymbol{S}_i$ for each generated point is obtained by applying the linear projection $L(.)$ and the sigmoid function $\alpha(.)$ on the semantic features, i.e.$\boldsymbol{S}_i = \alpha(L(\boldsymbol{Y}_i))$.

3.3 Detection Header

In this section, we referred on the Point-RCNN [47] to extract semantic features from the generated points using the PointNet++ [23] encoder for the detection task. Specifically, we first map the ROI to the generated point cloud, and then perform the following operations to obtain local semantic features $\boldsymbol{X}_i = \psi([x_i, y_i, z_i, b_i, S_i])$. Where $\psi(.)$ denotes MLP, $[x_i, y_i, z_i]$ denotes the coordinates of the generated point under the proposal and $b_i = \sqrt{{x_i}^2 + {y_i}^2 + {z_i}^2}$ is used as additional information to enhance the depth information of the point. Subsequently, we connect the local semantic feature $\boldsymbol{X}_i$ and the generating point feature $\boldsymbol{Y}_i$ in one piece and input it into the PointNet++ [23] network to obtain ROI features.The specific formula can be expressed as:

$$\boldsymbol{F}_{roi} = \{\max\{\psi([\boldsymbol{Y}_i : \boldsymbol{X}_i])\}\}_{i=1}^{M}, \tag{5}$$

where M denotes the number of points in the proposal. Ultimately, we apply FFN on the features to obtain the confidence classification of this predicted boxes and the discrimination of the box refinement to generate the detection results. It is worth noting that in order to strengthen the correlation between the layers, we input the detection results from each layer to the next layer as proposal and collect the detection boxes from all the layers, and then apply non-maximum suppression (NMS) to eliminate the redundant bounding boxes to obtain more reliable final detection results.

3.4 Training Losses

We design the HT-SSPG sampling end-to-end training strategy with a total loss $\mathcal{L}$ consisting of four components, the region proposal loss $\mathcal{L}_{RPN}$, the proposal refinement loss $\mathcal{L}_{Head}$, the voxel supervised loss $\mathcal{L}_{Heatmap}$, and the point generation loss $\mathcal{L}_{P\text{oint}}$, represented as:

$$\mathcal{L} = \mathcal{L}_{RPN} + \mathcal{L}_{Head} + \mathcal{L}_{Heatmap} + \mathcal{L}_{P\text{oint}}\,. \tag{6}$$

Both $\mathcal{L}_{RPN}$ and $\mathcal{L}_{Head}$ consist of a classification term and a regression term. The classification branch of $\mathcal{L}_{RPN}$ uses focus loss to measure the gap between proposals and ground truth bounding boxes. Similarly, $\mathcal{L}_{Head}$ uses binary cross entropy loss. For the regression branch, both losses use smooth L1 loss to measure the gap between the predicted and ground truth values.

The $\mathcal{L}_{Heatmap}$ is computed from the voxel supervised network to measure the gap between the thermal response $\hat{\boldsymbol{H}}$ output by the predictive heat map generator and the thermal response $\boldsymbol{H}$ produced by the real heat map generator. In this calculation, we use the smooth L1 function with the following formula:

$$\mathcal{L}_{Heatmap} = \frac{1}{N}\sum_{i=1}^{N}\begin{cases} \frac{1}{2}\left|\hat{\boldsymbol{H}}_i - \boldsymbol{H}_i\right|^2 & \left|\hat{\boldsymbol{H}}_i - \boldsymbol{H}_i\right| < 1 \\ \left|\hat{\boldsymbol{H}}_i - \boldsymbol{H}_i\right| - \frac{1}{2} & \left|\hat{\boldsymbol{H}}_i - \boldsymbol{H}_i\right| \geq 1 \end{cases}, \tag{7}$$

where N is the number of voxels.

The $\mathcal{L}_{P\text{oint}}$ consists of two parts, $\mathcal{L}_{score}$ and $\mathcal{L}_{shape}$, which are used to assess the quality of the generated points. Where $\mathcal{L}_{score}$ is used to ensure that the generated points lie within the ground truth bounding box. Specifically, in order to reduce the computational cost, we calculate the focal loss by selecting N_p points from the many generated points via FPS as follows:

$$\mathcal{L}_{score} = -\frac{1}{N_p}\sum_{j=1}^{N_p}(1 - S_j)^{\gamma}\log S_j, \tag{8}$$

where S_j denotes the foreground probability score of the sampled points. Although $\mathcal{L}_{score}$ ensures that the generated points are within the true bounding box, considering only the points within the box does not guarantee that the distribution of generated points conforms to the actual shape of the object. For

this, we have designed an additional $\mathcal{L}_{shape}$. Specifically, we use other point cloud structures in the dataset to approximate the complete structure of the desired target point cloud. Then, using the generated complete point cloud structure as the target, the recommended foreground for using chamfer distance is as follows:

$$\begin{aligned}\mathcal{L}_{shape} = \frac{1}{N_{fp}} \sum_{r=1}^{N_{fp}} \Big(\frac{1}{|\boldsymbol{P}_r|} \sum_{x \in \boldsymbol{P}_r} \min_{y \in \boldsymbol{P}_r^*} \|x - y\|_2^2 \\ + \frac{1}{|\boldsymbol{P}_r^*|} \sum_{y \in \boldsymbol{P}_r^*} \min_{x \in \boldsymbol{P}_r} \|y - x\|_2^2 \Big),\end{aligned} \tag{9}$$

where N_{fp} is number of initial proposals, $\boldsymbol{P}_r$ denotes the generated point cloud, and $\boldsymbol{P}_r^*$ denotes the real target point cloud.

Table 1: Our detection results were compared with several state-of-the-art methods on the KITTI validation set, using IoU thresholds of 0.7 for cars and 0.5 for pedestrians and bicycles. The best performance is represented by red, the second place is represented by blue, and the third place is represented by green.

Method	Reference	Car 3D AP_{R40}(%)			Ped. 3D AP_{R40}(%)			Cyc. 3D AP_{R40}(%)		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
PV-RCNN [25]	CVPR'2020	92.57	84.83	82.69	64.26	56.67	51.91	88.88	71.95	66.78
Voxel R-CNN [3]	AAAI'2021	92.38	85.29	82.86	-	-	-	-	-	-
CT3D [24]	ICCV'2021	92.85	85.82	83.46	61.05	55.57	51.10	89.01	71.88	67.91
SIENet [17]	PR'2022	92.49	85.43	83.05	-	-	-	-	-	-
BtcDet [34]	AAAI'2022	**93.15**	**86.28**	**83.86**	69.39	61.19	55.86	91.45	74.70	70.08
Casa+PV [32]	TGRS'2022	92.73	85.89	**83.57**	68.90	59.86	53.66	90.95	70.69	65.82
PDV [10]	CVPR2022	92.56	85.29	83.05	66.90	60.80	55.85	92.72	74.23	69.60
LoGoNet [14]	CVPR'2023	92.04	85.04	**84.31**	**70.20**	**63.72**	**59.46**	91.74	**75.35**	**72.42**
PG-RCNN [12]	ICCV'2023	92.73	85.26	82.83	68.44	60.63	55.36	**93.84**	74.85	70.15
GraVos(*Part-*A^2) [28]	CVPR'2023	91.68	82.58	81.67	65.82	59.58	54.55	90.64	74.03	69.64
BSAODet [33]	TCSVT'2023	92.27	85.06	82.75	**71.98**	**66.00**	**60.49**	**93.23**	**76.07**	**72.31**
HCPVF [5]	TCSVT'2023	**93.40**	**86.79**	**84.31**	69.52	61.57	56.16	91.56	74.87	70.16
MsSVT++ [13]	TPAMI'2024	89.24	84.93	78.9	66.37	59.58	53.92	92.63	73.98	69.31
HT-SSPG	Ours	**93.16**	**86.06**	83.51	**75.95**	**69.09**	**63.44**	**95.86**	**76.04**	**71.32**

4 Experiment

4.1 Dataset

KITTI Dataset. The KITTI dataset [7] comprises 7481 annotated training frames and 7518 unannotated test frames. Following recent studies, we split the 7481 training frames into 3712 frames for training and 3769 frames for validation. We used the 3D Average Precision (AP) under 40 recall positions (R40) as the evaluation metric, targeting the detection of cars, cyclists, and pedestrians. The IoU thresholds for cars, cyclists, and pedestrians are set to 0.7, 0.5, and 0.5, respectively.

Waymo Open Dataset. The Waymo Open Dataset [29] is a vast autonomous driving dataset comprising 1000 sequences, with 798 sequences allocated for training and the remaining 202 for validation. This dataset employs mean Average Precision (mAP) and 3D mAP weighted by Heading (mAPH) as the evaluation metric for 3D object detection. For detecting cars, cyclists, and pedestrians, the dataset specifies IoU thresholds of 0.7, 0.5, and 0.5 respectively.

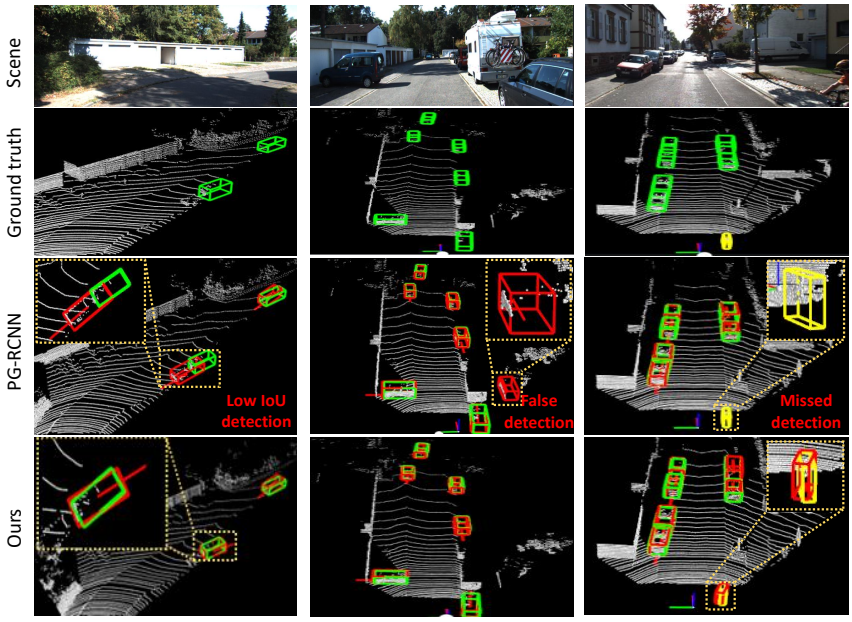

Fig. 3: Visualization of detection results on the KITTI validation set. The ground truth of the car and the cyclist is represented in green and yellow respectively, and the predicted bounding box is represented in red.

4.2 Implementation Details

Network Details. For the KITTI dataset [7], the detection ranges are defined as follows: X-axis [0m, 70.4m], Y-axis [-40m, 40m], Z-axis [-3m, 1m], with voxel dimensions of (0.05m, 0.05m, 0.1m) respectively. For the Waymo Open Dataset [29], the X and Y axes are set within [-75.2m, 75.2m], the Z-axis within [-2, 4], and voxel dimensions of (0.1m, 0.1m, 0.15m). The feature dimensions output by the 3D backbone network are (16, 32, 64, 64). In the HAR module, taking the voxel branch as an example, after the ROI pooling operation, the feature dimension $\boldsymbol{F}_v^i$ aggregated at each grid point $\boldsymbol{g}_i$ is 96, with G set to 6. After processing through the cross-attention transformer, The fusion feature $\boldsymbol{F}$ dimension is 192.

Table 2: Our detection results were compared with several state-of-the-art methods on the KITTI test set. The threshold was set at 0.7 IoU for cars and 0.5 IoU for cyclists. Data visualization follows the color scheme in Tab.1.

Method	Reference	Car 3D AP_{R40} (%)		Car BEV AP_{R40} (%)		Cyc. 3D AP_{R40} (%)		Cyc. BEV AP_{R40} (%)	
		Easy	Mod.	Easy	Mod.	Easy	Mod.	Easy	Mod.
PV-RCNN [25]	CVPR'2020	90.25	81.43	**94.98**	**90.65**	78.60	63.71	82.49	68.89
CT3D [24]	ICCV'2021	87.83	81.77	92.36	88.83	-	-	-	-
SIENet [17]	PR'2022	88.22	81.71	92.38	88.65	**83.00**	67.61	**84.64**	**71.21**
BtcDet [34]	AAAI'2022	90.64	**82.86**	92.81	89.34	82.81	**68.68**	84.48	**71.76**
Ada3D-B [44]	ICCV'2023	87.46	79.41	-	-	76.09	61.04	-	-
FDUE-Net [45]	ICME'2023	88.90	79.93	-	-	-	-	-	-
PG-RCNN [12]	ICCV'2023	89.38	82.13	93.39	89.46	82.77	67.82	**84.94**	70.65
OcTr [46]	CVPR'2023	**90.88**	82.64	-	-	-	-	-	-
PVT-SSD [36]	CVPR'2023	**90.65**	82.29	**95.23**	**91.63**	-	-	-	-
BSAODet [33]	TCSVT'2023	88.66	81.95	-	-	**83.17**	**70.48**	-	-
HCPVF [5]	TCSVT'2023	89.34	82.63	-	-	-	-	-	-
MsSVT++ [13]	TPAMI'2024	90.22	**82.87**	-	-	-	-	-	-
HT-SSPG	Ours	**90.82**	**82.72**	**95.57**	**89.53**	**83.03**	**68.75**	**86.38**	**71.65**

Finally, the semantic feature vector $\boldsymbol{Y}_i$ and the local spatial feature vector $\boldsymbol{X}_i$ of the generated points have dimensions of 32 and 64, respectively.

Training Details. Our designed HT-SSPG model is trained end-to-end using the Adam optimizer. The initial learning rate is set to 0.01, and the training is conducted for 100 epochs. During the training phase, we randomly sample 512 ROIs as samples for training. In the testing phase, we keep the top 100 refined proposals based on their scores and apply Non-Maximum Suppression (NMS) with an IoU threshold of 0.1 to filter out redundant bounding boxes, generating the final detection results.

Table 3: Comparison of our method with advanced methods in vehicle detection performance on the Waymo open dataset validation set, the evaluation metrics are 3D mAP and mAPH. Data visualization follows the color scheme in Tab.1.

Method	Reference	Vehicle(LEVEL1)		Vehicle(LEVEL2)	
		mAP(%)	mAPH(%)	mAP(%)	mAPH(%)
PV-RCNN [25]	CVPR'2020	70.30	65.36	**69.69**	64.79
VoTr-TSD [20]	ICCV'2021	74.95	74.25	65.91	65.29
Pyramid-PV [18]	ICCV'2021	76.30	75.68	67.23	66.68
LiDAR R-CNN [15]	CVPR'2021	73.50	73.00	64.70	64.20
SST-TS [6]	CVPR'2022	76.22	75.79	68.04	67.64
IA-SSD [43]	CVPR'2022	70.53	69.67	61.55	60.80
PDV [10]	CVPR'2022	**76.85**	**76.33**	**69.30**	**68.81**
itKD [2]	CVPR'2023	67.43	66.72	59.44	58.81
ConQueR [49]	CVPR'2023	76.10	75.60	68.70	**68.20**
PIPC-3Ddet [41]	TCSVT'2023	**76.69**	**76.19**	68.17	67.71
CluB$_{light}$ [31]	NeurIPS'2023	76.50	76.00	68.40	68.00
HT-SSPG	Ours	**77.80**	**77.09**	**69.13**	**68.60**

Table 4: Ablation experiments on the effectiveness of voxel supervised networks. "N-VSN" and "VSN" represent unsupervised and added voxel supervised networks, respectively.

Method	N-VSN	VSN	3D AP_{R40} (%) (Mod.)		
			Car	Ped.	Cyc.
(a)	✓		**86.13**	66.60	72.38
(b)		✓	86.06	**69.09**	**76.04**

Table 5: The ablation experiment evaluated the effectiveness of the HAR method. "N-HAR", "V-HAR", and "HAR" represent the version without multi-layer optimization, the version using only voxel extraction features, and the version using cross attention fusion voxel and point cloud features, respectively.

Method	N-HAR	V-HAR	HAR	3D AP_{R40} (%) (Mod.)		
				Car	Ped.	Cyc.
(a)	✓			85.83	63.66	73.57
(b)		✓		85.61	66.74	75.77
(c)			✓	**86.06**	**69.09**	**76.04**

4.3 Detection Results on the KITTI

Tab.1 and Tab.2 present a comparison of our HT-SSPG with some state-of-the-art (SOTA) methods on the KITTI [7] validation and test datasets. On the validation set, our method surpasses PG-RCNN [12] in car detection accuracy at 40 recall positions and shows significant improvements in detecting small objects such as cyclists and pedestrians. Specifically, for pedestrian detection at moderate and hard levels, our method achieves improvements of 8.46% AP and 8.08% AP respectively compared to PG-RCNN [12]. On the test set, our method also achieves competitive performance in detecting small objects like cyclists. Fig.3 illustrates the detection results of our method on the KITTI dataset [7]. Overall, our HT-SSPG demonstrates outstanding detection performance.

4.4 Detection Results on the Waymo

Tab.3 presents the comparison of vehicle detection performance between HT-SSPG and several SOTA methods on the Waymo open dataset [29] validation set. At an IoU threshold of 0.7, HT-SSPG achieves mAP and mAPH accuracies of 77.80 % and 77.09 % for LEVEL 1, and achieved competitive performance in LEVEL 2. Overall, our HT-SSPG demonstrates excellent detection performance across various scenarios. This demonstrates that HT-SSPG can efficiently perform point cloud completion across various scenarios.

4.5 Ablation Studies

In this section, we conducted comprehensive ablation experiments on HT-SSPG to validate the effects of each module. All our experiments were performed on the

KITTI [7] validation set, using 3D average precision (AP) at 40 recall positions (R40) as the evaluation metric for car, cyclist, and pedestrian detection, with IoU thresholds of 0.7, 0.5, and 0.5 respectively.

The effects of voxel supervised network. In this section, we will validate the effectiveness of the voxel supervision network. As shown in Tab.4. Method (a) in the table represents the case without the voxel supervision network, while method (b) represents the detection performance after integrating our designed voxel supervision network. It can be observed that with the addition of voxel supervised networks, HT-SSPG improves the moderate level of cyclists and pedestrians by 2.49% AP and 3.66% AP, respectively. This enhancement is attributed to the effective enhancement of the network's perception of complete structures of missing objects, facilitated by our designed voxel supervision network, thereby aiding in the generation of precise proposals and high-quality surface points.

The effects of hierarchical attention refinement. In this section, we will verify the effectiveness of HAR. As shown in Tab.5. Method (a) in the table serves as our baseline, representing the scenario without undergoing multi-level ROI optimization. Method (b) denotes feature extraction solely within voxels using cross layer attention. Method (c), which we employ in this paper, involves sequential feature extraction from both voxels and point clouds, followed by fusion using cross-level attention. It was observed that method (c) improved by 0.23%, 5.43%, and 2.47% AP respectively in the moderate level detection of cars, cyclists, and pedestrians compared to method (a). This is attributed to our effective integration of multi-granular features from different sources, greatly enriching spatial structural information within the features.

5 Conclusion

In this article, we propose a two-stage detector called HT-SSPG, aimed at overcoming the detection difficulties of incomplete and small objects based on LiDAR point cloud detectors. We achieve this goal by enhancing the network's perception of the complete structure of incomplete objects and generating semantic surface points of missing objects. Specifically, we designed a VSN that implicitly enhances the spatial perception ability of the backbone network by calculating the difference between predicted and true values of heat map. In addition, our HAR effectively integrates voxels and point cloud features from different granularities using cross attention, generating high-quality semantic surface points for detection. Finally, we will summarize the detection results generated in each stage and select the final detection box. A large number of experiments have proven the effectiveness of our method. However, our method has certain shortcomings in real-time detection. In the future, we will focus on improving the computational efficiency of detection networks.

References

1. Chen, Y., Liu, S., Shen, X., Jia, J.: Fast point r-cnn. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9775–9784 (2019)
2. Cho, H., Choi, J., Baek, G., Hwang, W.: itkd: Interchange transfer-based knowledge distillation for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13540–13549 (2023)
3. Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., Li, H.: Voxel r-cnn: Towards high performance voxel-based 3d object detection. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 1201–1209 (2021)
4. Deng, J., Zhou, W., Zhang, Y., Li, H.: From multi-view to hollow-3d: Hallucinated hollow-3d r-cnn for 3d object detection. IEEE Transactions on Circuits and Systems for Video Technology **31**(12), 4722–4734 (2021)
5. Fan, B., Zhang, K., Tian, J.: Hcpvf: Hierarchical cascaded point-voxel fusion for 3d object detection. IEEE Transactions on Circuits and Systems for Video Technology (2023)
6. Fan, L., Pang, Z., Zhang, T., Wang, Y.X., Zhao, H., Wang, F., Wang, N., Zhang, Z.: Embracing single stride 3d object detector with sparse transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8458–8468 (2022)
7. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
8. He, C., Li, R., Li, S., Zhang, L.: 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. pp. 8417–8427 (2022)
9. He, C., Zeng, H., Huang, J., Hua, X.S., Zhang, L.: Structure aware single-stage 3d object detection from point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11873–11882 (2020)
10. Hu, J.S., Kuai, T., Waslander, S.L.: Point density-aware voxels for lidar 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8469–8478 (2022)
11. Jiao, Y., Jie, Z., Chen, S., Chen, J., Ma, L., Jiang, Y.G.: Msmdfusion: Fusing lidar and camera at multiple scales with multi-depth seeds for 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21643–21652 (2023)
12. Koo, I., Lee, I., Kim, S.H., Kim, H.S., Jeon, W.j., Kim, C.: Pg-rcnn: Semantic surface point generation for 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18142–18151 (2023)
13. Li, J., Dong, S., Ding, L., Xu, T.: Mssvt++: Mixed-scale sparse voxel transformer with center voting for 3d object detection. IEEE transactions on pattern analysis and machine intelligence (2023)
14. Li, X., Ma, T., Hou, Y., Shi, B., Yang, Y., Liu, Y., Wu, X., Chen, Q., Li, Y., Qiao, Y., et al.: Logonet: Towards accurate 3d object detection with local-to-global cross-modal fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17524–17534 (2023)
15. Li, Z., Wang, F., Wang, N.: Lidar r-cnn: An efficient and universal 3d object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7546–7555 (2021)
16. Li, Z., Yao, Y., Quan, Z., Xie, J., Yang, W.: Spatial information enhancement network for 3d object detection from point cloud. Pattern Recognition **128**, 108684 (2022)

17. Li, Z., Yao, Y., Quan, Z., Yang, W., Xie, J.: Sienet: Spatial information enhancement network for 3d object detection from point cloud. Pattern Recognition **128**, 108684 (2022)
18. Mao, J., Niu, M., Bai, H., Liang, X., Xu, H., Xu, C.: Pyramid r-cnn: Towards better performance and adaptability for 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2723–2732 (2021)
19. Mao, J., Xue, Y., Niu, M., Bai, H., Feng, J., Liang, X., Xu, H., Xu, C.: Voxel transformer for 3d object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3164–3173 (2021)
20. Mao, J., Xue, Y., Niu, M., Bai, H., Feng, J., Liang, X., Xu, H., Xu, C.: Voxel transformer for 3d object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3164–3173 (2021)
21. Pan, X., Xia, Z., Song, S., Li, L.E., Huang, G.: 3d object detection with pointformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7463–7472 (2021)
22. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
23. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30** (2017)
24. Sheng, H., Cai, S., Liu, Y., Deng, B., Huang, J., Hua, X.S., Zhao, M.J.: Improving 3d object detection with channel-wise transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2743–2752 (2021)
25. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10529–10538 (2020)
26. Shi, S., Wang, Z., Wang, X., Li, H.: Part-aˆ 2 net: 3d part-aware and aggregation neural network for object detection from point cloud. arXiv preprint arXiv:1907.03670 **2**(3) (2019)
27. Shi, W., Rajkumar, R.: Point-gnn: Graph neural network for 3d object detection in a point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1711–1719 (2020)
28. Shrout, O., Ben-Shabat, Y., Tal, A.: Gravos: Voxel selection for 3d point-cloud detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21684–21693 (2023)
29. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2446–2454 (2020)
30. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
31. Wang, Y., Deng, J., Hou, Y., Li, Y., Zhang, Y., Ji, J., Ouyang, W., Zhang, Y.: Club: Cluster meets bev for lidar-based 3d object detection. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 40438–40449. Curran Associates, Inc. (2023)
32. Wu, H., Deng, J., Wen, C., Li, X., Wang, C., Li, J.: Casa: A cascade attention network for 3-d object detection from lidar point clouds. IEEE Transactions on Geoscience and Remote Sensing **60**, 1–11 (2022)

33. Xiao, W., Peng, Y., Liu, C., Gao, J., Wu, Y., Li, X.: Balanced sample assignment and objective for single-model multi-class 3d object detection. IEEE Transactions on Circuits and Systems for Video Technology (2023)
34. Xu, Q., Zhong, Y., Neumann, U.: Behind the curtain: Learning occluded shapes for 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 2893–2901 (2022)
35. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
36. Yang, H., Wang, W., Chen, M., Lin, B., He, T., Chen, H., He, X., Ouyang, W.: Pvt-ssd: Single-stage 3d object detector with point-voxel transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13476–13487 (2023)
37. Yang, Z., Sun, Y., Liu, S., Shen, X., Jia, J.: Std: Sparse-to-dense 3d object detector for point cloud. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1951–1960 (2019)
38. Ye, M., Xu, S., Cao, T.: Hvnet: Hybrid voxel network for lidar based 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1631–1640 (2020)
39. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11784–11793 (2021)
40. Yin, T., Zhou, X., Krähenbühl, P.: Multimodal virtual point 3d detection. Advances in Neural Information Processing Systems **34**, 16494–16507 (2021)
41. Yu, C., Peng, B., Huang, Q., Lei, J.: Pipc-3ddet: Harnessing perspective information and proposal correlation for 3d point cloud object detection. IEEE Transactions on Circuits and Systems for Video Technology (2023)
42. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: Pcn: Point completion network. In: 2018 international conference on 3D vision (3DV). pp. 728–737. IEEE (2018)
43. Zhang, Y., Hu, Q., Xu, G., Ma, Y., Wan, J., Guo, Y.: Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18953–18962 (2022)
44. Zhao, T., Ning, X., Hong, K., Qiu, Z., Lu, P., Zhao, Y., Zhang, L., Zhou, L., Dai, G., Yang, H., et al.: Ada3d: Exploiting the spatial redundancy with adaptive inference for efficient 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17728–17738 (2023)
45. Zhi, P., Zhou, K., Li, Y., Wang, S.: Feature decoupling and uncertainty estimation for 3d object detection. In: 2023 IEEE International Conference on Multimedia and Expo (ICME). pp. 1133–1138. IEEE (2023)
46. Zhou, C., Zhang, Y., Chen, J., Huang, D.: Octr: Octree-based transformer for 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5166–5175 (2023)
47. Zhou, Q., Yu, C.: Point rcnn: An angle-free framework for rotated object detection. Remote Sensing **14**(11), 2605 (2022)
48. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4490–4499 (2018)
49. Zhu, B., Wang, Z., Shi, S., Xu, H., Hong, L., Li, H.: Conquer: Query contrast voxel-detr for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9296–9305 (2023)