Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3481517, 9 pages
https://doi.org/10.1155/2022/3481517

Research Article

An Anchor-Free 3D Object Detection Approach Based on Hierarchical Pillars

Xudie Ren [1] **and Shenghong Li** [1,2]

[1]*School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China*
[2]*MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China*

Correspondence should be addressed to Xudie Ren; renxudie@sjtu.edu.cn and Shenghong Li; shli@sjtu.edu.cn

Received 17 June 2022; Revised 18 July 2022; Accepted 29 July 2022; Published 18 August 2022

Academic Editor: Kuruva Lakshmanna

Object detection in the 2D domain is well developed owing to the wide application of CMOS image sensors and the great success of deep learning technologies in recent years. However, under circumstances such as autonomous driving, the variation of weather conditions and light conditions makes it impossible to perform reliable detection using regular 2D image sensors. 3D data generated by a Lidar or Radar is more robust to such environments, hence serving as an essential complement to 2D data in such scenarios. Well-established anchor-based detectors in the 2D domain suffer from time-consuming anchor configuration and cannot be exploited directly to process 3D data. This paper proposes an anchor-free network that encodes the raw point cloud into a hierarchical pillar representation to locate objects. Without predefined anchors and NMS postprocessing, our method directly predicts the center points and box properties to accomplish the detection task efficiently. In addition, a PCA-based initialization for the convolutional kernel is proposed to accelerate the training process. Experiments are implemented on the KITTI benchmark, and our method can achieve competitive performance with other anchor-based methods. Comprehensive ablation studies further verify the validity and rationality of each part of the proposed method.

1. Introduction

Object detection is one of the most important tasks in the field of computer vision, which has a wide range of applications in individual recognition, content understanding, autonomous driving, etc. In general, the task of object detection is to mark locations and determine categories of key targets with bounding boxes. In the past decades, huge 2D data have been collected from widely applied commercial 2D image sensors. Taking advantage of the cutting-edge deep learning technology, many convolutional neural network-(CNN-) based algorithms [1–9] have been designed for 2D object detection and have shown their superiority and effectiveness. Because of the 3D nature of many real-world problems, 3D object detection attracts more and more attention.

3D data, usually represented as point clouds, can effectively depict the real world with accurate geometry information, which is robust to changing light conditions, different object textural, and color variation. With the increasing availability, 3D data has been serving as an essential complement to general 2D sensors in many scenarios. However, the sparse and unordered structure of point clouds could not be directly processed by a conventional CNN, which urges novel network structures to encode the point clouds.

Considering point clouds' irregular and sparse properties, most existing data-driven 3D object detection approaches can be categorized into point-based and voxel-based ones. Inspired by the pioneering work PointNet [10] and PointNet++ [11], point-based methods can take the raw point clouds as input to extract features without any data transformation or information loss. However, the real-time performance and effectiveness of point-based methods are not satisfactory due to the time-consuming point sampling procedure and the poor encoder perceptual ability. On the other hand, voxel-based methods [12, 13] transform the point clouds into some regular data representations which can be processed by CNN. Furthermore, the introduction of sparse convolution dramatically improves

the performance and speed of voxel-based methods. Nevertheless, these methods are sensitive to the parameters of voxel partition and cause local information loss in raw point clouds inevitably. Recently, PointPillars [14] utilizes the vertical columns called pillars to organize point clouds and avoid complicated 3D convolutional operations. It alleviates the parameter configuration problem in preprocessing and shows considerable accuracy and speed. Pillar-based methods organize the raw point cloud into two-dimensional regular grids so that traditional 2D convolution operations can be applied. It is significantly effective for Lidar-based sparse point clouds; however, it is prone to missing small or far-away objects. Inspired by this, this paper focuses on how to solve the local information loss problem based on a pillar-based object detection model.

Current popular voxel-based (including pillar-based) methods often leverage anchors, which are some manually designed bounding boxes, to accomplish detecting and classifying. Although anchors provide some useful priors and enable the methods to predict offsets directly, applying them in the 3D objection detection is difficult. First, hyperparameters including aspect ratios, orientations, and anchor numbers needed to be predetermined and adjusted accordingly to diverse datasets. Manual hyperparameter tuning is time-consuming and inaccurate, which limits the applicability. Second, a great number of anchor boxes are generated during training and inference so that all possible locations of the ground truth bounding boxes can be covered. This introduces huge memory consumption and a serious class imbalance between the positive and negative anchors. Third, the necessary Non-Maximum Suppression (NMS) for determining the final detection results can lead to an extra computational burden.

One solution is to employ an anchor-free detector. Recently, anchor-free detectors have obtained continuous developments and breakthroughs in the field of 2D object detection [15, 16]. They directly estimate the key points and sizes of the objects without hyperparameter configuration and the generation of anchors. Later, some anchor-free 3D object detection approaches [17] have been proposed and outperform classical anchor-based ones.

All in all, the pillar-based approach improves voxel-based approaches by encoding the raw data into a lower dimensional representation. It has a faster inference speed but at the cost of local information loss. The anchor-free detector can learn from the data rather than relying on predefined anchors and boxes. It can regress more accurate bounding boxes and has achieved great success in 2D object detection, it deserves more investigation in the 3D domain.

Motivated by these facts, this paper proposes a hierarchical pillar-based anchor-free 3D object detection model. Compare with other pillar-based approaches, we further partition the pillars into subpillars and learn the hierarchical features of local regions. Then, the proposed method aggregates multilevel features to generate high-quality spatial representations with the CNN backbone. In addition to existing anchor-free approaches, we introduce an improved center point allocation strategy to further improve the accuracy and alleviate the positive-negative imbalance problem. At the training stage, we exploit a principal component analysis- (PCA-) based method to initialize the convolutional kernels. At the inference stage, our model can generate the center location directly and avoid the NMS for postprocessing. Experiments and ablation studies are carried out on a well-known benchmark KITTI [18] to evaluate the performance of the proposed method.

The contributions of this paper can be summarized as follows:

(1) An anchor-free detector for point cloud 3D object detection without NMS is proposed. It can be end-to-end joint optimized and achieve competitive performance with other anchor-based methods

(2) The point cloud is encoded into a hierarchical pillar-based feature representation, which can capture the local structure and mitigate the information loss in preprocessing. Subsequent multilevel feature aggregation in the CNN backbone can extract robust features and enhance detection accuracy

(3) Our proposed PCA-based initialization [19] is incorporated into the CNN backbone for 3D objection detection. The convolutional kernels can be initialized with more informative values, which accelerates the training process of CNN and reduces the effect of gradient diffusion caused by random parameters

(4) A novel center point allocation strategy is designed to train the model. Experimental results demonstrate its effectiveness in the 3D object detection problem

The rest of the paper is organized as follows: Section 2 describes the related works. Section 3 provides an introduction to the proposed method for 3D object detection. Experimental results on the dataset are presented and discussed in Section 4. Section 5 concludes the paper.

2. Related Works

2.1. 3D Object Detection with Point Clouds. A point cloud is a set of points with sparse distribution and irregularity. PointNet [10] is the pioneer to take raw point clouds as input and extract 3D features by shared multilayer perceptrons. PointNet++ [11] further proposes the set abstraction levels to capture local patterns among the point clouds. Successive object detection works [20–22] based on the above-mentioned PointNet or PointNet++ model to process original points directly are called point-based approaches. [20] detects the 2D proposals from the RGB images and implements the 3D frustum projection on them. Then, a PointNet is applied to extract ROI features of the points in frustums and refine the 3D bounding box. [21] directly proposes 3D proposals from the point cloud and combines the local spatial features learned on canonical coordinates with global semantic features to obtain better locations. [22] produces some initial predictions with voxel representation as input and generates the fused features of interior points for further refinement. [23] proposed a two-stage 3D object detection

approach from sparse-to-dense. In the first stage, it makes proposals at all the foreground points; then, in the second stage, it incorporates the point cloud feature and the semantic feature to refine the bounding box. [24] improves the set abstraction layer in PointNet++ and designs a novel sampling strategy called F-FPS. Then, it uses an anchor-free detector to regress the object's position. These point-based methods prefer to design a second fine-tuning stage to regress a more accurate box position locating. Although they show impressive performance, they trade off efficiency for accuracy and are not suitable for real-time applications.

Another category falls into voxel-based approaches, which preprocesses the raw point clouds into some compact representations. VoxelNet [12] organizes the points into voxels and then extracts 3D dense features through the voxel feature encoding (VFE) layer and 3D convolution. SECOND [13] utilizes a sparse convolution network to accelerate the convolution operations in training and inference. Recently, PointPillars [14] generates 2D pseudoimages by encoding point clouds on vertical columns (pillars) and eliminates the time-consuming 3D convolution. Most of the voxel-based methods are one-stage detectors with high computational efficiency but suffer from the information loss problem due to voxelization. [25] proposed a two-stage pillar-based approach to address the imbalance issue caused by anchors; it incorporates the concept of the pillars and multi-view feature learning; then, a pillar-to-point projection is employed to refine the result. Our method is aimed at preserving more local information using a hierarchical pillar representation at a minimal cost of speed.

2.2. Anchor-Free Object Detection. Most of the existing object detection methods design a large number of predefined anchors for bounding box generation, which results in complex hyperparameter configuration and huge memory consumption. Anchor-free detectors directly predict the key points and sizes of the bounding boxes with high speed. The success of anchor-free methods in 2D object detection [15, 16] inspired researchers to investigate anchor-free 3D detectors. VoteNet [26] aggregates the votes of object centroids to obtain the object proposals directly from point clouds. But VoteNet is not a completely anchor-free model for the reason that it employs some anchor templates in the size prediction process. Later, [17] proposes an anchor-free detector and further simplifies the postprocessing to increase the detection efficiency. However, the performance of anchor-free approaches highly depends on the central point allocation strategy and 3D bounding box regression.

All the related works are summarized in Table 1.

3. Proposed Method

In this section, we introduce the proposed hierarchical pillar-based anchor-free 3D object detection model. As shown in Figure 1, the overall network is composed of the following parts: a point cloud encoder that transforms the unordered point clouds into 2D pseudoimages; a CNN backbone based on PCA initialization to further extract features, and anchor-free detectors. In the following, we describe each part of the proposed method in detail.

3.1. Point Cloud Encoder. Our proposed point cloud encoder is based on the high-efficiency PointPillars [14] but can further capture the local structure and mitigate the information loss in the point cloud encoding process. The input point cloud P is a set of points with irregular distribution in Euclidean space.

First, P is discretized into $W * H$ vertical columns with uniform grids in the x-y plane. Considering the sparsity of point cloud data, we apply zero-padding when one pillar contains too few points and K nonempty pillars are preserved. Second, several hierarchical feature extraction (HFE) levels are introduced to group the points into local patterns and aggregate the information. At the first HFE level, each pillar with enough points is divided into M amount of evenly height subpillars according to the resolution parameters in the vertical direction. Following [14], the points in each subpillar are augmented as a 9-dimensional representation and are used to calculate the average vertical coordinate $\bar{z}_i$. Random sampling is implemented on the subpillars with more than N amount of points. Then, the selected subpillars are applied with a linear layer followed by a batch normalization (BN) layer, a ReLU layer, and a max operation to produce an output subpillar feature.

In the subsequent HFE levels, we employ the FPS algorithm [11] to sample the input subpillars according to the average vertical coordinates. Then, we group the information of two adjacent subpillars in the vertical direction for each selected subpillar to generate fewer larger subpillars. By using the identical structure in the first HFE level, we can further obtain the corresponding output features in the current level i. Through these hierarchical groupings of the points and subpillars, our encoder can abstract local patterns of the points and retain the information of vertical direction in the final point cloud feature with the size (C, K). At last, a pseudoimage of size (C, H, W) is created by scattering the feature to the raw pillar locations.

3.2. CNN Backbone Based on PCA Initialization. Inspired by our previous work [17, 19], we design a CNN backbone based on PCA initialization to create the dense features for the following anchor-free detector. As depicted in Figure 2, the backbone is composed of the top-down part and the upsampling-concatenation part. The top-down part can be formulated as several blocks with 2D convolution and downsampling operations to extract features with high semantic and decreasing spatial size. Each block is applied with several convolution layers, a BN layer, and a ReLU layer sequentially. The convolutional kernels are all initialized by PCA as follows:

For a convolution layer with C_i input channels and C_o output channels, we firstly randomly divide the input feature maps into C_o groups. Then, we implement a fully covered sampling on each feature map group with the kernel size to obtain the patch sets. After mean normalization, the covariance matrix and eigenvector matrix for each patch set is calculated. And the eigenvector with the largest eigenvalue is selected to initialize the weights of convolution kernels for each group, the initialization value is along which the data has the maximum variance, i.e., maximum information entropy of the sampled patches.

Table 1: Summary of methods for 3D object detection.

Methods	Data representation	Detector	Key innovation
[12]	Voxel-based	Anchor-based	(1) Voxel feature encoding layer (2) 3D CNN
[13]	Voxel-based	Anchor-based	(1) 3D sparse convolution (2) Better data augmentation
[14]	Pillar-based	Anchor-based	(1) Voxel division without vertical direction (2) Avoid 3D convolution operation
[25]	Pillar-based	Anchor-free	(1) Aligned pillar-to-point projection (2) Multiview feature learning
[17]	Voxel-based	Anchor-free	Anchor-free detector
[26]	Point-based	Anchor-based	Deep Hough voting
[20]	Point-based	Anchor-based	(1) Introduction of RGB images (2) Frustum projection
[21]	Point-based	Anchor-based	(1) Rough 3D proposals based on raw point cloud (2) Fuse local spatial features and global semantic features on the second stage
[22]	Voxel-based	Anchor-based	(1) Voxel-based first-stage prediction (2) Combined with point features for refinement
[23]	Point-based	Anchor-based	(1) Spherical anchor (2) PointsPool
[24]	Point-based	Anchor-free	(3) Feature-based furthest point sampling (F-FPS) (4) 3D centerness assignment strategy

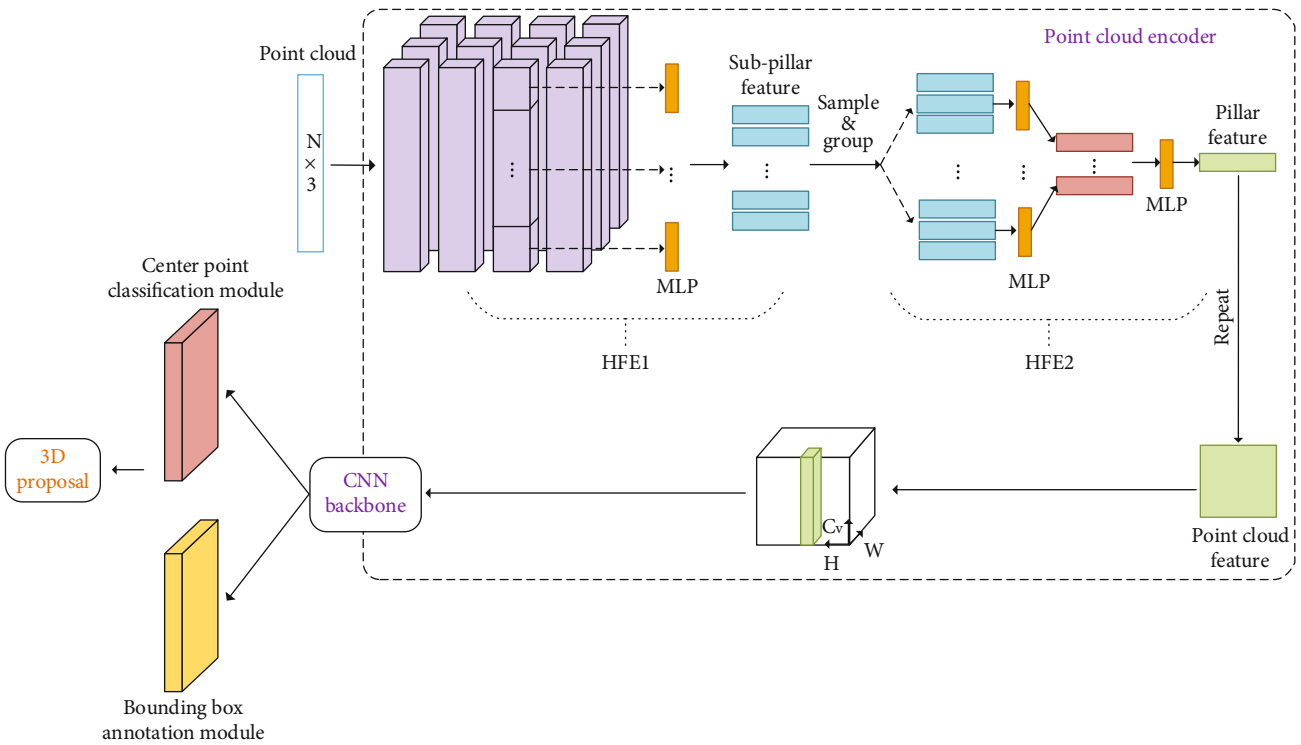

Figure 1: The network architecture of a hierarchical pillar-based anchor-free 3D detection model.

The features produced by each block are then upsampled to the same spatial size by applying transposed convolution followed by BN and ReLU. Finally, features from various blocks are concatenated to the final point cloud feature.

3.3. Anchor-Free Detector. The proposed anchor-free detector contains two modules to accomplish the proposal generating and classification: (1) a center point classification module that produces the keypoint heatmap in the x-y plane for each object category and (2) a bounding box annotation

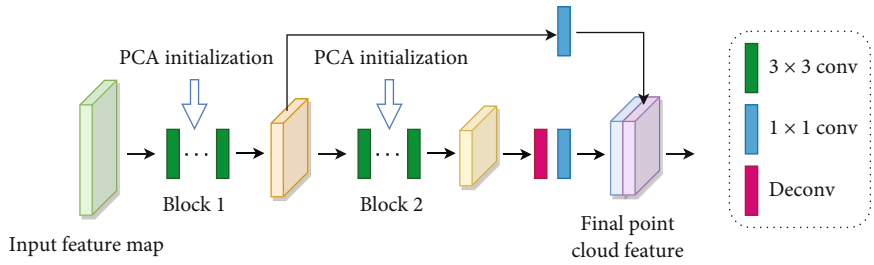

Figure 2: The CNN backbone structure of the proposed method.

module that regresses offset, 3D object size properties, and the orientation. All heads of the two modules share the common features from the backbone, and each of them consists of an independent $3 * 3$ convolution layer and a $1 * 1$ convolution layer.

3.3.1. Center Point Classification Module. The center point classification module generates the keypoint heatmap H, which describes the object centers in the x-y plane. The 3D ground truth bounding box parameters are converted to the center point location label in the discretized x-y coordinate system by applying origin location subtraction, pillar side length division, and floor operation.

In the traditional center point allocation, there is only one pixel selected to be the positive sample for each ground truth center. This results in a severe positive and negative sample imbalance in center point classification. To mitigate the problem, we calculate the heatmap label for each pixel of the pseudoimage as follows and propose an improved center point allocation strategy:

$$H_{x,y,c} = \begin{cases} 1, & \text{if } d = 0, \\ 1 - \dfrac{1}{r(x,y)}, & \text{if } d = 1, \\ \dfrac{1}{d}, & \text{else}, \end{cases} \tag{1}$$

where $r(x, y)$ is the diagonal length of the ground truth 2D bounding box and d is the maximum Euclidean distance between the pixel (x, y) of the pseudoimage and the ground truth 2D bounding box centroid along both x- and y-axes in BEV.

Because some pixels around the ground truth center location can create a bounding box with sufficient IoU with the ground truth box, we divide the pixels of the pseudoimage into positive set P and negative set N following the threshold values. All other pixels are ignored in the training stage. To further balance the gradient of positive and negative sets, we introduce the following focal loss [9] as the center classification loss to train the heatmap.

$$L_{\text{cls}} = -\frac{1}{N}\sum_{x,y,c}\begin{cases} \left(1 - \hat{H}_{x,y,c}\right)^{\alpha} \log\left(\hat{H}_{x,y,c}\right), & \text{if } H_{x,y,c} > t_{\text{pos}}, \\ \left(1 - H_{x,y,c}\right)^{\beta}\left(\hat{H}_{x,y,c}\right)^{\alpha} \log\left(1 - \hat{H}_{x,y,c}\right), & \text{if } H_{x,y,c} < t_{\text{neg}}, \\ 0, & \text{else}, \end{cases} \tag{2}$$

where N is the number of center points in the detection range and α, β are the hyperparameters set to 2 and 4 in the experiments.

3.3.2. Bounding Box Annotation Module. This module regresses the corresponding bounding box annotation for each positive center point, which includes a two-dimensional center point offset regression, a z-axis center coordination regression, a three-dimensional object size regression, and a two-dimensional orientation regression.

There exists a discretization error when transforming the float center point locations to 2D pillar coordinates in the previous center point classification module. Moreover, the increase in positive center point samples and some wrong predictions in the heatmap can lead to an inaccurate center point location in BEV. To recover the deviation caused by these reasons and obtain more precise object centers, the offset head generates the offset map $\hat{O}$ for the center points in the x-y plane which is shared by all object categories. A logistic function is applied to constrain the output values to fall between 0 and 1. We use the L1 loss [4] as the offset loss.

$$L_{\text{off}} = \frac{1}{N}\sum_{p_c}\sum_{i}\left|\varphi\left(\hat{O}_{\tilde{p}_c,i}\right) - O_{p_c,i}\right|. \tag{3}$$

To further obtain the center points in 3D Lidar coordinate system, the z-axis coordinate head regresses the center location in the z-axis. This head creates a z-value map Z for the center points which are shared by all object categories. Due to the unconstrained z-value regression range, the gradients of inliers and outliers are imbalanced in the traditional L1 loss, making it difficult to regress. Following [27], we use the balanced L1 loss to train the z-axis coordinate.

$$L_z = \frac{1}{N}\sum_{p_c} L_b\left(\left|\hat{Z}_{\tilde{p}_c} - Z_{p_c}\right|\right). \tag{4}$$

where

$$L_b(x) = \begin{cases} \dfrac{a}{b}\left(b|x| + 1\right) \ln\left(b|x| + 1\right) - a|x|, & \text{if } |x| < 1, \\ \gamma|x| + C, & \text{otherwise}, \end{cases} \tag{5}$$

where a, b and γ are the hyperparameters, which satisfy *al*

Input: $\widehat{P}_c(\widehat{X}, \widehat{Y})$: the set of detected center locations of category c in BEV; n_c: the number detected centers of category c; a: the pillar side length
Output: Detected bounding box set B
1: **for** $i = 1, 2, 3, \cdots, n_c$ **do**
2: Obtain the corresponding x-y offset, z coordinate, size, orientation and fine-tune offset:
$$(\widehat{o}_i^{(x)}, \widehat{o}_i^{(y)}), \widehat{z}_i, (\widehat{l}_i, \widehat{w}_i, \widehat{h}_i), (\widehat{\sin}_i, \widehat{\cos}_i), (\widehat{f}_i^{(z)}, \widehat{f}_i^{(h)})$$
3: $$B_i = (a(\widehat{x}_i + 0.5) + \widehat{o}_i^{(x)}, a(\widehat{y}_i + 0.5) + \widehat{o}_i^{(y)}, \widehat{z}_i + \widehat{f}_i^{(z)}, \widehat{l}_i, \widehat{w}_i, \widehat{h}_i + \widehat{f}_i^{(z)}, \text{atan2}(\widehat{\sin}_i, \widehat{\cos}_i))$$
4: **end for**

Algorithm 1: The detected 3D bounding box generating algorithm.

$n(b+1) = \gamma$ and are set to $a = 0.5$, $\gamma = 1.5$ in the experiments.

We also regress the length, width, and height properties for the bounding box in the object size head. Similar to z -values, the size loss is in a balanced L1 form.

$$L_{\text{size}} = \frac{1}{N} \sum_{p_c} L_b \left(\left| \widehat{S}_{\widehat{p}_c} - S_{p_c} \right| \right), \tag{6}$$

where $\widehat{S}$ denotes predicted object sizes and S is the ground truth values.

Finally, the yaw rotation around the z-axis is predicted in the orientation head. To avoid angle confusion, we regress two trigonometric functions ($\sin(\theta)$ and $\cos(\theta)$) for the rotation angle θ and decode it in the inference stage. We employ the L1 loss as the orientation loss to train the orientation regression.

$$L_{\text{ori}} = \frac{1}{N} \sum_{p_c} \sum_{i} \left| \varphi \left(\widehat{R}_{\widehat{p}_c, i} \right) - R_{p_c, i} \right|, \tag{7}$$

where $\widehat{R}$ represents the predicted orientation feature map and R is the ground truth values.

The overall loss for the first stage of the proposed network is defined as follows:

$$L_1 = \lambda_{\text{cls}} L_{\text{cls}} + \lambda_{\text{off}} L_{\text{off}} + \lambda_z L_z + \lambda_{\text{size}} L_{\text{size}} + \lambda_{\text{ori}} L_{\text{ori}}, \tag{8}$$

where λ denotes the weight for the center point classification loss and the regression losses.

3.4. Inference Stage. In this stage, we employ the max pooling operation to filter the peaks in the generated heatmap as the predicted centers, which is efficient and can avoid the time-consuming NMS. The inference algorithm for generating the detected 3D bounding boxes is shown in Algorithm 1.

4. Experiments and Result Analysis

In this section, we describe the dataset and summarize the implementation details first. Moreover, for verifying the validity and improvement of our method in the 3D object detection problems, we provide the ablation studies and compare the performance with other detection models on the dataset.

4.1. Dataset and Implementation Details. We employ the KITTI benchmark dataset [18] to evaluate the proposed method for the 3D object detection problem and we only use the Lidar point clouds. The dataset has a total number of 7,481 training samples with annotation and 7,518 testing samples without labeling. Following the standard convention, we split them into 3,712 samples for the training set and 3,769 samples for the validation set. Three classes of objects, i.e., cars, pedestrians, and cyclists, have been annotated. Since the car class has the most samples and diversity, as advocated by other researchers [17, 22], only the car category is taken into consideration during the evaluation in this paper. Following the official evaluation protocol, average precision (AP) with the IoU threshold of 0.7 is selected as the metric for the car class.

For KITTI car detection, we followed PointPillars [14] to use a detection range $[(0,70.4), (-40, 40), \text{and } (-3, 1)]$ along the X-, Y-, and Z-axes. The pillar side length is set to 0.16 m in the x-y plane. The max number of pillars and max amount of points in each pillar is set to 12,000 and 100, respectively. We arrange two HFE levels in the point cloud encoder. Two CNN blocks are applied to generate the pseudoimage, the number of convolutional layers is set to 7 and 8 for each block, and the number of feature map channels is set to 64 and 128 for each block, respectively. We utilize the Adam optimizer to train the network. The batch size is set to 4, the learning rate is set to 0.0001, and trained for 180 epochs. At inference time, the $3 * 3$ max pooling and AND operations are applied to obtain the center points.

4.2. Ablation Studies. In this section, we will pay attention to verifying the effectiveness and reliability of different parts in the proposed method for the 3D object detection problem. The ablation studies are implemented in four aspects and the baseline model is a simplified version of the proposed method, in which the normal pillar representation and the traditional center point allocation strategy are applied, similar to PointPillars.

Table 2: Experiment configuration evaluated.

Key innovations	Baseline	m2	m3	m4	HPAF
HFE	×	√	√	√	√
PCA initialization	×	×	√	√	√
Anchor-free (w/o size loss term)	×	×	×	√	×
Anchor-free (w/size loss term)	×	×	×	×	√
Inference time (ms)	23	45	42	27	28
mAP@IOU = 0.7 (%)	74.73	76.88	77.17	77.22	78.65

Table 3: Detection performance comparison among several methods on KITTI validation set (car class).

Methods	Modality	Anchor free	Stage	Speed (Hz)	3D detection AP (IoU = 0.7)			BEV detection AP (IoU = 0.7)		
					Easy	Moderate	Hard	Easy	Moderate	Hard
F-PointNet [20]	L+C	N	Two	5.9	83.76	70.92	63.65	88.16	84.02	76.44
PointRCNN [21]	L	N	Two	10	89.19	78.85	77.91	90.21	87.89	85.51
Fast-PointRCNN [22]	L	N	Two	15.4	89.12	79	77.48	90.12	88.1	86.24
VoxelNet [12]	L	N	One	4.4	81.97	65.46	62.85	89.6	84.81	78.57
SECOND [13]	L	N	One	20	87.43	76.48	69.1	89.96	87.07	79.66
PointPillars [14]	L	N	One	42	87.44	77.67	75.76	89.88	87.43	85.01
AFDet [17]	L	Y	One	35	85.68	75.57	69.31	89.42	85.45	80.56
Ours	L	Y	One	35.7	88.65	78.20	76.17	90.04	87.80	84.14

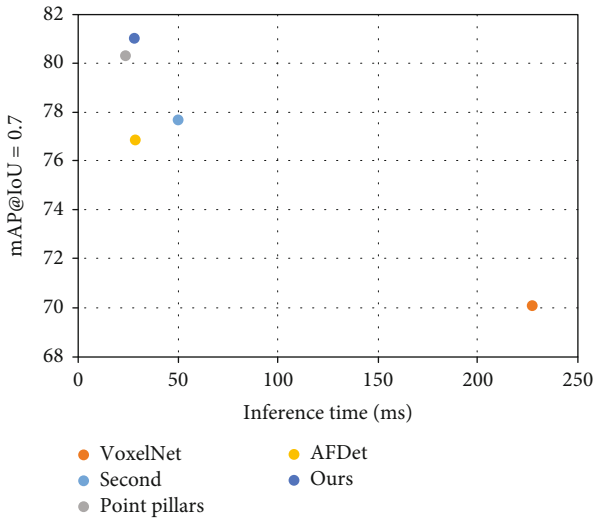

Figure 3: mAP versus inference time of one-stage methods.

The studies are carried out on a small subset of the KITTI validation dataset, and the results are summarized in Table 2.

To evaluate the effectiveness of the proposed point cloud encode network, we replace the traditional pillar-based encoder with it and denote it as method2 (m2). Compared with the baseline method, m2 takes 1.95x time and gets a 2.45% mAP gain. It proves that vertical division can improve the method further.

Method3 (m3) improves m2 with PCA initialization of the CNN kernels. As shown in Table 2, m3 slightly improves m2 on both time and accuracy.

Method4 (m4) employs the anchor-free detector but without the size loss L_{size}. Anchor-free detector reduces the inference time significantly owing to the avoidance of generating a large number of anchor boxes and also improves the mAP slightly.

Mehtod5 (HPAF) adds the size loss term to the total loss in addition to method4. With the loss term added, the final proposed HPAF achieves the best mAP among all the configurations evaluated and spends no more time than m4.

4.3. Comparison with Other Methods. To further test the effectiveness and robustness of the proposed model in 3D object detection, we compare it with the state-of-the-art ones including several one-stage methods and some two-stage detectors. The AP results for 3D detection and BEV detection on the KITTI test set are shown in Table 3. As it can be seen from Table 3, most of the two-stage methods outperform the one-stage ones. This indicates the introduction of the second stage can contribute to refining the object location in the first stage and enhance the detection performance. However, these two-stage methods are time-consuming and model-complicated. Among all the one-stage methods, the proposed method outperforms VoxelNet, SECOND, PointPillars, and AFDet by 15.57%, 4.29%, 0.89%, and 5.40% for 3D mAP (IoU = 0.7), respectively. On the other hand, the anchor-based methods need to predefine a large number of anchors and employ postprocessing to filter the predicted bounding boxes, which brings the computational burden and shows poor speed performance. Though the proposed anchor-free method is outperformed by most two-stage methods, it achieves competitive AP results compared to the SOTA one-stage methods and shows its superiority concerning detection speed, as illustrated in Figure 3.

5. Conclusion

In this paper, we proposed a hierarchical pillar-based anchor-free detector to address the 3D object detection task. It encodes the raw point cloud as a hierarchical pillar representation and predicts object center points directly without predefined anchors. Experiments are conducted on the KITTI dataset to examine the performance. Our method can achieve competitive performance with the anchor-based ones and speed up the model efficiency by introducing PCA-based initialization and avoiding NMS postprocessing.

However, like other pillar-based methods, organizing point cloud into global structures, such as voxels and pillars, will cause local information loss inevitably. As a result, the hyperparameters, such as voxel/pillar size, and the number of hierarchy levels should be carefully tuned to fit into the dataset so that one can get the best performance. Further work will be focused on solving the 3D object detection problems with the incorporation of RGB images and developing a more suitable loss function for object parameter regression to improve the accuracy.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no competing interests.

Authors' Contributions

X. Ren and S. Li were responsible for the conceptualization. X. Ren and S. Li were responsible for the methodology. X. Ren was responsible for the software. X. Ren was responsible for the validation. X. Ren was responsible for the formal analysis. X. Ren was responsible for the investigation. X. Ren was responsible for the resources. X. Ren was responsible for the data curation. X. Ren was responsible for the original draft preparation. X. Ren and S. Li were responsible for the review and editing of the paper. X. Ren was responsible for the visualization. S. Li was responsible for the supervision. S. Li was responsible for the project administration. S. Li was responsible for the funding acquisition. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This research work is funded by the National Nature Science Foundation of China under Grant 61971283 and Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102.

References

[1] C. L. Chowdhary, M. Alazab, A. Chaudhary, S. Hakak, and T. R. Gadekallu, *Computer Vision and Recognition Systems Using Machine and Deep Learning Approaches: Fundamentals, Technologies and Applications*, Institution of Engineering and Technology, 2021.

[2] G. Ross, D. Jeff, D. Trevor, and M. Jitendra, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2014.

[4] R. Girshick, "Fast r-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2015.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and A. C. Berg, "SSD: single shot multibox detector," in *Computer Vision – ECCV 2016. ECCV 2016*, vol. 9905 of Lecture Notes in Computer Science, Springer, Cham, 2016.

[8] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017.

[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2017.

[10] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: deep learning on point sets for 3D classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017.

[11] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[12] Y. Zhou and O. Tuzel, "Voxelnet: end-to-end learning for point cloud based 3D object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018.

[13] Y. Yan, Y. Mao, and B. Li, "SECOND: sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[14] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: fast encoders for object detection from point clouds," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019.

[15] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, https://arxiv.org/abs/1904.07850.

[16] H. Law and J. Deng, "Cornernet: detecting objects as paired keypoints," in *Computer Vision – ECCV 2018. ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11218 of Lecture Notes in Computer Science, pp. 765–781, Springer, Cham, 2018.

[17] R. Ge, Z. Ding, Y. Hu et al., "AFDet: anchor free one stage 3D object detection," 2020, https://arxiv.org/abs/2006.12671.

[18] 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*, Providence, RI, USA, 2012.

[19] X.-D. Ren, H.-N. Guo, G.-C. He, X. Xu, C. Di, and S.-H. Li, "Convolutional neural network based on principal component analysis initialization for image classification," *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)*, 2016, Changsha, China, 2016, 2016.

[20] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3D object detection from RGB-d data," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018.

[21] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019.

[22] Y. Chen, S. Liu, X. Shen, and J. Jia, "Fast point r-CNN," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019.

[23] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "STD: sparse-to-dense 3D object detector for point cloud," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1951–1960, Seoul, Korea (South), 2019.

[24] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: point-based 3D single stage object detector," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11040–11048, Seattle, WA, USA, 2020.

[25] Y. Wang, A. Fathi, A. Kundu et al., "Pillar-based object detection for autonomous driving," in *Computer Vision – ECCV 2020. ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J. M. Frahm, Eds., vol. 12367 of Lecture Notes in Computer Science, Springer, Cham, 2020.

[26] C. R. Qi, O. Litany, K. He, and L. Guibas, "Deep Hough voting for 3D object detection in point clouds," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South)., 2019.

[27] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra r-CNN: towards balanced learning for object detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019.