PillarNet: Real-Time and High-Performance Pillar-based 3D Object Detection

Guangsheng Shi[1], Ruifeng Li[1⋆], and Chao Ma[2⋆]

[1] State Key Laboratory of Robotics and System, Harbin Institute of Technology
[2] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
sgsadvance@163.com, lrf100@hit.edu.cn, chaoma@sjtu.edu.cn

Abstract. Real-time and high-performance 3D object detection is of critical importance for autonomous driving. Recent top-performing 3D object detectors mainly rely on point-based or 3D voxel-based convolutions, which are both computationally inefficient for onboard deployment. In contrast, pillar-based methods use solely 2D convolutions, which consume less computation resources, but they lag far behind their voxel-based counterparts in detection accuracy. In this paper, by examining the primary performance gap between pillar- and voxel-based detectors, we develop a real-time and high-performance pillar-based detector, dubbed PillarNet. The proposed PillarNet consists of a powerful encoder network for effective pillar feature learning, a neck network for spatial-semantic feature fusion and the commonly used detect head. Using only 2D convolutions, PillarNet is flexible to an optional pillar size and compatible with classical 2D CNN backbones, such as VGGNet and ResNet. Additionally, PillarNet benefits from our designed orientation-decoupled IoU regression loss along with the IoU-aware prediction branch. Extensive experimental results on the large-scale nuScenes Dataset and Waymo Open Dataset demonstrate that the proposed PillarNet performs well over state-of-the-art 3D detectors in terms of effectiveness and efficiency. Code is available at https://github.com/agent-sgs/PillarNet.

Keywords: 3D object detection, point cloud, autonomous driving

1 Introduction

With the success in point cloud representation learning using deep neural networks, LiDAR-based 3D object detection has made remarkable progress recently. However, the top-performing point cloud 3D object detectors on the large-scale benchmark datasets, such as nuScenes Dataset [2] and Waymo Open Dataset [37], entail heavy computational load and large memory storage. Hence, it is desirable to develop a top-performing 3D detector with real-time speed for the onboard deployment on autonomous vehicles.

[⋆] Corresponding authors. Work done while G. Shi visits the Vision and Learning Group at Shanghai Jiao Tong University.

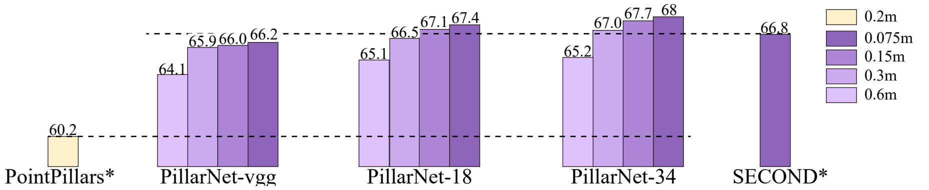


Fig. 1. Comparison between PillarNet variants along with different pillar sizes and two baselines on the nuScenes *val* set in nuScenes detection score (NDS). The reported results of these two baselines are from the latest CenterPoint [46]. All of our PillarNet variants use the same training schedules with CenterPoint-SECOND [46]. * denotes the reproduced two baselines using the center-based head from CenterPoint [46].

Existing point cloud 3D object detectors mainly use the grid-based representation over point cloud and can be broadly categorized into two groups, *i.e.*, 3D voxel-based and 2D pillar-based methods. Both of these two groups take the classical "encoder-neck-head" detection architecture [11,14,16,27,39,40,44,48,51]. Voxel-based methods [11,14,40,48,51] typically divide the input point cloud into regular 3D voxel grid. An encoder with sparse 3D convolutions [10] is then used to learn geometric representation across multiple levels. Following the encoder, a neck module with standard 2D CNNs fuses multi-scale features before feeding to the detection head. In contrast, pillar-based methods [16,39,44,27] project 3D point clouds into a 2D pseudo-image on the BEV plane, and then directly build the neck network upon the 2D CNN-based feature pyramid network (FPN) to fuse multi-scale features. For voxel-based methods, the effective voxel-wise feature learning powered by sparse 3D CNN delivers favorable detection performance. However, due to the 3D sparse convolution within the encoder, it is hard to aggregate multi-scale features with different resolutions on the BEV space. For pillar-based methods, a light encoder for pillar feature learning yields unsatisfied performance compared with their voxel-based counterparts. Moreover, the small sized pseudo-image and the large initial pillar further limit the detection performance. It is because a finer pillar leads to larger pseudo-image and more favorable performance but heavier computational load. Interestingly, both voxel- and pillar-based methods perform 3D detection using the aggregated multi-scale features on the BEV space (see Sec. 3.1)

We observe that previous pillar-based methods do not have powerful pillar feature encoding, which is the main cause of the unsatisfied performance. In addition, progressively downsampling pillar scales can help to decouple the output feature map size and the initial pseudo-image projection scale. As such, we design a real-time and high-performance pillar-based 3D detection method, dubbed PillarNet, that consists of an encoder for hierarchical deep pillar feature extraction, a neck module for multi-scale feature fusion, and the commonly-used center-based detect head. In our PillarNet, the powerful encoder network involves 5 stages. Stage 1 to 4 follow the same setting as the conventional 2D detection networks such as VGG [36] and ResNet [12] but substituted 2D convolutions with its sparse counterparts for resource savings. Stage 5 with standard 2D convolutions possesses a larger receptive field and feeds semantic features to the following neck network. The neck network exchanges sufficient information through stacked convolution layers between the further enriched high-level semantic feature from the encoder stage 5 and the low-level spatial feature from the encoder stage 4. For tuning the hard-balanced pillar size in previous pillar-based methods, PillarNet offers an effective solution by skillfully detaching the corresponding encoder stages for the chosen pillar scale. For example, to accommodate the input with 8 times pillar size ($0.075 * 8$m in nuScenes Dataset), we can simply remove the 1x, 2x, and 4x downsampled encoder stages.

As shown in Fig. 1, our PillarNet with variant configurations, *i.e.*, PillarNet-vgg/18/34, offer the scalability and flexibility for point cloud-based 3D object detection by using merely 2D convolutions. Our PillarNet significantly advances pillar-based 3D detectors and sheds new light on further research on point cloud object detection. Despite its simplicity, the proposed PillarNet achieves the state-of-the-art performance on two large-scale autonomous driving benchmarks [2,37] and runs in real-time (see Sec. 4).

2 Related Works

2.1 Point Cloud 3D Object Detection

3D object detection with point cloud alone can mainly be summarized into two categories: point-based and grid-based methods.

Point-based 3D object detectors. Powered by the pioneering PointNet [28,30], point-based methods directly process irregular point clouds and predict 3D bounding boxes. PointRCNN [33] proposes a point-based proposal generation paradigm directly from raw point clouds and then refines each proposal by devising an RoI pooling operation. STD [42] transforms point features inside of each proposal into compact voxel representation for RoI feature extraction. 3DSSD [23], as a one-stage 3D object detector, introduces F-FPS as a complement of existing D-FPS with set abstraction operation to benefit both regression and classification. These point-based methods naturally preserve accurate point location and enable flexible receptive fields with radius-based local feature aggregation. These methods, however, as summarized in [25], spend 90% of their runtime on organizing irregular point data rather than extracting features, and are not suitable for handling large-scale point clouds.

Grid-based 3D object detectors. Most existing methods discrete the sparse and irregular point clouds into regular grids including 3D voxels and 2D pillars, and then capitalize on 2D/3D CNN to perform 3D object detection. The pioneering VoxelNet [51] divides point cloud into 3D voxels, and encodes scene feature using 3D convolutions. To tackle the empty voxels typically for the large outdoor space, SECOND [40] introduces 3D sparse convolution to accelerate VoxelNet [51] and improves the detection accuracy. Until now, 3D voxel-based methods dominate the majority of 3D detection benchmarks. For a long time, even with sparse 3D convolution, it was hard to balance between the fine resolution of 3D voxels and associated resource costs.

PointPillars [16] uses 2D voxelization on the ground plane with a PointNet [28] based per-pillar feature extractor. It can utilize 2D convolutions for deployment on embedded systems with limited costs. MVF [6] utilizes multi-view features to augment point-wise information before projecting raw points into 2D pseudo-image. HVNet [44] fuses different scales of pillar features at the point-wise level to achieve good accuracy and high inference speed. HVPR[27] cleverly keeps the efficiency of pillar-based detection while implicitly leveraging the voxel-based feature learning regime for better performance. Current Pillar-based advancements, however, focus on the sophisticated pillar feature projection or multi-scale aggregation strategies, to narrow the huge performance gap relative to their voxel-based counterparts. In contrast, we resort to a powerful backbone network to address the above issues and boost 3D detection performance.

2.2 Multi-sensor based 3D Object Detection

Most approaches expect the complementary information from multiple sensors, such as camera image and LiDAR, to achieve high performance 3D object detection. MV3D [6] designs 3D object anchors and generates proposals from BEV representations and refine them using features from LiDAR and camera. AVOD [15] instead fuses these features at the proposal generation stage and provides better detection results. ContFuse [19] learns to fuse image features with point cloud features onto BEV space. MMF [18] struggles for LiDAR-Camera feature fusion via proxy tasks including depth completion on RGB images and ground estimation from point clouds. 3D-CVF [47] tackles the multi-sensor registration issue for cross-view spatial feature fusion in the BEV domain. Almost all of these multi-modality frameworks rely on the intermediate BEV representation to perform 3D object detection. Our method extracts point cloud features on BEV space and may be promising to seamlessly integrate into existing multi-modality frameworks for advanced performance.

3 PillarNet for 3D Object Detection

3.1 Preliminaries

The grid-based detectors perform 3D detection on BEV space, including 3D voxel-based detectors and 2D pillar-based detectors. Recent voxel-based detectors follow the SECOND [40] architecture with improved sparse 3D CNN for effective voxel feature encoding over the pioneering VoxelNet [51]. Pillar-based detectors generally follow the pioneering PointPillars [16] architecture with only 2D CNN for multi-scale feature fusion. We first revisit these two representative point cloud detection architectures, which motivate us to construct the proposed PillarNet method.

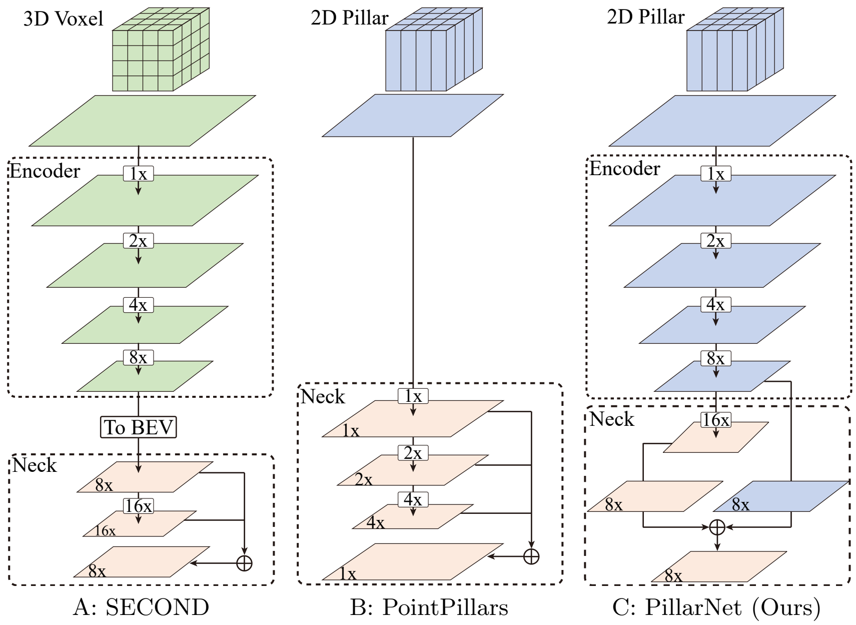


Fig. 2. Comparison of three types of architectures. The encoder uses sparse 3D CNN in SECOND [40] while sparse 2D CNN for PillarNet. The neck in all the three methods uses standard 2D CNN. On the nuScenes Dataset, the 3D voxel size in SECOND [40] is (0.075m, 0.075m, 0.2m), and the 2D pillar size in PointPillars and our proposed PillarNet is (0.2m, 0.2m) and (0.075m, 0.075m) respectively.

SECOND. SECOND [40] is a typical voxel-based one-stage object detector, which lays the groundwork for succeeding voxel-based detectors with specialized sparse 3D convolutions [9,10]. It divides the unordered point cloud into regular 3D voxels and performs box prediction on BEV space. The entire 3D detection architecture contains three basic parts: (1) An encoder hierarchically encodes the input non-empty voxel features into 3D feature volumes with the 1×, 2×, 4× and 8× downsampled sizes. (2) A neck module further abstracts the flattened encoder output on the BEV space into multiple scales in a top-down manner. (3) A detect head performs box classification and regression using the fused multi-scale BEV features.

PointPillars. PointPillars [16] projects raw point cloud on the X-Y plane via a tiny PointNet [28], yielding a sparse 2D pseudo-image. PointPillars uses a 2D CNN-based top-down network to process the pseudo-image with stride 1x, 2x, and 4x convolution blocks and then concatenates the multi-scale features for the detect head.

Analysis. Despite the favorable runtime and memory efficiency, PointPillar [16] still lags far behind SECOND [40] on performance. Under the premise that sparse 3D convolutions possess the superior representation ability for point cloud learning, recent advanced pillar-based methods mainly focus on exploring attentive pillar feature extraction [24,44] from raw points or sophisticated multi-scale strategies [39,44,27]. These methods, on the other hand, suffer from unfavourable latency and still under-performs their 3D voxel-based counterparts by a large margin.

Alternatively, we take a different view by considering grid-based detectors as BEV-based detectors and revisit the entire point cloud learning architecture. We identify that the performance bottleneck of pillar-based methods mainly lies in the sparse encoder network for spatial feature learning and effective neck module for sufficient spatial-semantic features fusion. Specifically, PointPillars directly applies the feature pyramid network to fuse multi-scale features on the projected dense 2D pseudo-image, lacking the sparse encoder network for effective pillar feature encoding as in SECOND. On the other hand, PointPillars couples the size of the final output feature maps with the initial projected pillar scale, increasing the entire calculation and memory cost sharply as the pillar scale gets finer.

To resolve the above issues, we stand by the "encoder-neck-head" detection architecture on BEV space to improve the performance of pillar-based methods. Specifically, we explore the significant difference and respective function for the encoder and neck networks:

- We redesign the encoder in SECOND by substituting sparse 3D convolutions by its sparse 2D convolutions counterpart on loss-less pillar features from raw point clouds. It has been validated in the 3^{rd} row of Table 4 that the sparse encoder process enhances 3D detection performance significantly.
- We formulate the neck module as the spatial-semantic feature fusion by inheriting the sparse spatial features from the sparse encoder output and further high-level semantic feature abstraction in low-resolution feature maps, as shown in the 6^{th} row of Table 4, which is efficient and effective.

Finally, we build our PillarNet using the relatively heavyweight sparse encoder network for hierarchical pillar feature learning and the lightweight neck module for sufficient spatial-semantic feature fusion.

3.2 PillarNet Design for 3D Object Detection

In this subsection, we present the detailed structure of our PillarNet design. The overall architecture in Fig. 3 consists of three components: the encoder for deep pillar feature extraction, the neck module for spatial-semantic feature aggregation, and the 3D detect head. With the commonly used center-based detect head [46], we present the flexibility and scalability of our PillarNet.

Encoder design. The encoder network aims to extract deep sparse pillar features hierarchically from the projected sparse 2D pillar features, where the detachable stages from 1 to 4 progressively down-sample sparse pillar features using sparse 2D CNN. Compared to PointPillars[16], our designed encoder have two advantages:

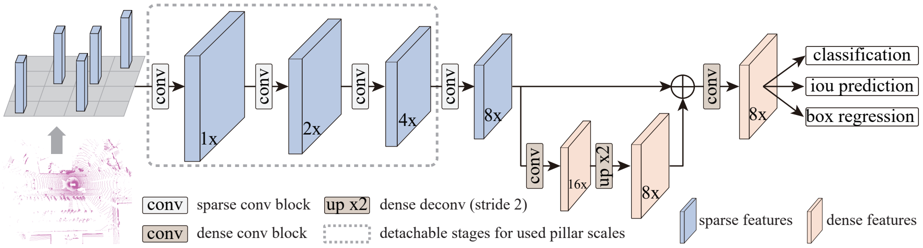


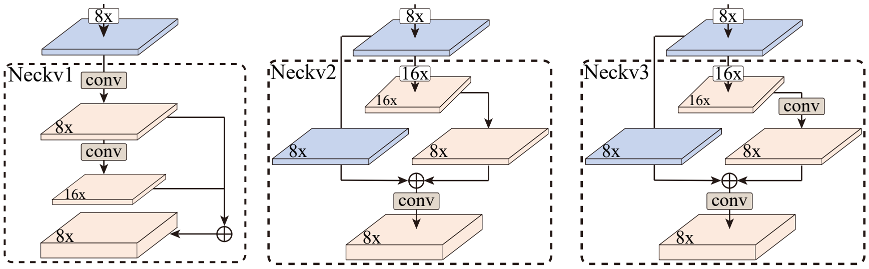
Fig. 3. The overall architecture of our proposed PillarNet. The input point clouds are first quantified into pillars to feed into the 2D sparse convolution-based encoder to learn multi-scale spatial features. Then the densified semantic feature is fused with the spatial feature in the neck module for the final 3D box regression, classification, and IoU prediction.

(1) The sparse encoder network can take the progress on image-based 2D object detection, such as VGGNet [35] and ResNet [12]. The simple encoder for pillar feature learning can largely improve 3D detection performance.
(2) The hierarchically downsampling structure allows PillarNet to skillfully operate the sparse pillar features with different pillar sizes, which alleviates the limitation of coupling pillar size in previous pillar-based methods.

Our constructed PillarNet with variant backbones, PilllarNet-vgg/18/34, with the similar complexities of VGGNet/ResNet-18/ResNet-34. The detailed network configurations can be found in the supplementary material.

Neck design. The neck module, as in FPN [21], aims to fuse high-level abstract semantic features and low-level fine-grained spatial features for mainstream detect head (*i.e.*, anchor boxes or anchor points). The additional 16X downsampled dense feature maps further abstracts high-level semantic feature using a group of dense 2D CNNs, to enrich receptive field for large objects and populate object center-positioned features for center-based detect head. Equipped with spatial features from sparse encoder network, there are two alternative neck designs for the spatial-semantic feature fusion from the starting design in SECOND [40]:

(1) The naive design neckv1 (Fig. 4(A)) from SECOND [40] applies a top-down network to generate multi-scale features and concatenate multi-scale dense feature maps as the final output.
(2) The aggressive design neckv2 (Fig. 4(B)) considers sufficient information exchange between high-level semantic feature from additional 16X downsampled dense feature maps and low-level spatial feature from sparse encoder network using a group of convolution layers.
(3) The design neckv3 (Fig. 4(C)) further enriches the high-level semantic features on 16X downsampled dense feature maps through a group of convolution layers and fuses the spatial-semantic features with the other group of convolution layers for robust feature extraction.



A: A naive neck design

B: An aggressive neck design using the 16x features

C: A enriched neck design using the 16x features

Fig. 4. Detailed structure of different neck designs. The neck A inherits directly from SECOND [40], while two alternative neck designs B and C introduce the spatial features from sparse encoder network and semantic features from the 16× dense feature maps.

3.3 Orientation-Decoupled IoU Regression Loss

In general, the IoU metric highly correlates with the localization quality and classification accuracy of the predicted 3D boxes. Previous methods [20] show that using the 3D IoU quality to re-weight the classification and supervise the box regression can achieve better localization accuracy.

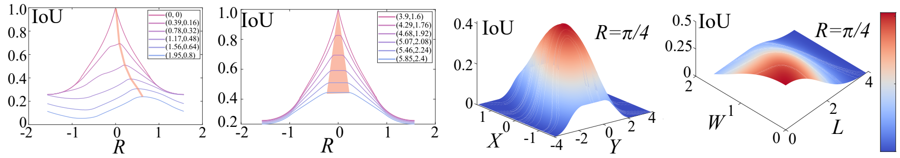
For the classification branch, we follow previous methods [14,48] and use the IoU-rectification scheme to incorporate the IoU information into the confidence scores. The IoU-Aware rectification function [14] at the post-processing stage can be formulated as:

$$\hat{S} = S^{1-\beta} * W_{\text{IoU}}^{\beta} \tag{1}$$

where S indicates the classification score and W_{IoU} is the IoU score. β is a hyperparameter. For predicting the IoU score, we use L1 loss $\mathcal{L}_{iou}$ to supervise the IoU regression, where the target 3D IoU score W between the predicted 3D box and the ground truth box is encoded by $2 * (W - 0.5) \in [-1, 1]$.

For the regression branch, recent methods [20,49] extend the GIoU [31] loss or DIoU [50] loss from 2D detection to the 3D domain. However, the non-trivial 3D IoU computation slows down the training process. Furthermore, the coupled orientation for the IoU-related regression may negatively affect the training process. Fig. 5 shows such an example. Given a typical 2D bounding box $[x, y, l, w, \theta] = [0, 0, 3.9, 1.6, 0]$, there exist cross-effects of orientation with center bias for x and y positions or of scale for width and length sizes during the optimization for IoU metric between biased box with ground-truth box as follows:

- The effect of center deviation on orientation regression. The training phase easily settles into a local optimum, if the box center deviates far. See the red curve in Fig. 5(A).
- The effect of size variation on orientation regression. The training phase settles into the notorious optimization plateau, if box sizes change largely. See the red region in Fig. 5(B).



A: IoU-R curve for center deviation

B: IoU-R curve for size variation

C: IoU-XY surface when $R = \pi/4$

D: IoU-WL surface when $R = \pi/4$

Fig. 5. The IoU metric-based interplay of orientation (R) with center or size for a 2D rotated box [0, 0, 3.9, 1.6, 0]. A and B depict the effect of center variation and size oscillation on orientation regression separately. Red curve in A indicates the local optimum while red region in B for optimization plateau. C and D depict the effect of orientation bias $R = \pi/4$ on center and size regression separately.

- The effect of orientation bias on center and size regression. The optimization direction remains consistent even if the orientation deviation is significant.

As a result, we present an alternative Orientation-Decoupled IoU-related regression loss by decoupling the orientation θ from the mutually-coupled seven parameters $(x, y, z, w, l, h, \theta)$. Specifically, we extend the IoU regression loss $\mathcal{L}_{od-iou}$ (OD-IoU/OD-GIoU/OD-DIoU) from the IoU loss [31], GIoU loss [31] and DIoU loss [50], respectively.

3.4 Overall Loss Function

Following [46], we apply the focal loss [22] for the heatmap classification $\mathcal{L}_{cls}$, and the L1 loss for localization offset $\mathcal{L}_{off}$, the z-axis location $\mathcal{L}_z$, 3D object size $\mathcal{L}_{size}$ and orientation $\mathcal{L}_{ori}$. The overall loss $\mathcal{L}_{total}$ is jointly optimized as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \mathcal{L}_{iou} + \lambda(\mathcal{L}_{od-iou} + \mathcal{L}_{off} + \mathcal{L}_z + \mathcal{L}_{size} + \mathcal{L}_{ori}) \tag{2}$$

where the loss weight λ is empirically set parameter as in [46].

4 Experiments

nuScenes Dataset. nuScenes [2] contains 1000 driving sequences, with 700, 150, 150 sequences for training, validation, and testing, respectively. Each sequence is approximately 20-second long, with a LiDAR frequency of 20 FPS. nuScenes uses a 32 lanes LiDAR, which produces approximately 30k points per frame. The annotations include 10 classes with a long-tail distribution. The official evaluation metrics are mean Average Precision (mAP) and nuScenes detection score (NDS). We follow the convention to accumulate 10 LiDAR sweeps to densify the point clouds and report results by using the official evaluation protocol.

Waymo Open Dataset. Waymo Open Dataset [37] is currently the largest dataset with LiDAR point clouds for autonomous driving. There are total 798 training sequences with around 160k LiDAR samples, and 202 validation sequences with 40k LiDAR samples. It annotated the objects in the full 360° field. The evaluation metrics are calculated by the official evaluation tools, where the mean average precision (mAP) and the mean average precision weighted by heading (mAPH) are used for evaluation. The 3D IoU threshold is set as 0.7 for vehicle detection and 0.5 for pedestrian/cyclist detection.

Training and Inference details. We use the same training schedules as prior CenterPoint-SECOND [46], where Adam optimizer is used with one-cycle learning rate policy, weight decay 0.01, and momentum 0.85 to 0.95 on 4 Tesla V100 GPUs. We make runtime comparison with two baselines (*i.e., CenterPoint-SECOND and CenterPoint-PointPillars*) on desktop equipped with an i9 CPU and RTX 3090 GPU. To project raw point clouds into the pillar feature, we apply one-layer MLP-based PointNet associated with *atomic max*-based pooling on augmented point-wise feature of all inside points per pillar. We adopt the widely used data augmentation strategies as [46] during training, including the random scene flipping along, random rotation, random scene scaling, and random translation.

For nuScenes Dataset, we set the detection range to $[-54m, 54m]$ for the X and Y axis, and $[-5m, 3m]$ for the Z axis. We use $(0.075m, 0.075m)$ as the basic pillar size for experiments. We train the PillarNet from scratch with batch size 16, max learning rate 1e-3 for 20 epochs. For the post-processing process during inference, following [46], we use class-agnostic NMS with the score threshold set to 0.1 and rectification factor β to 0.5 for all 10 classes. To compare on the nuScenes test set, we do not use any model ensembling except double-flip test-time augmentation as CenterPoint [46].

For Waymo Open Dataset, we set the detection range to $[-75.2m, 75.2m]$ for X and Y axes, and $[-2m, 4m]$ for Z axis. W use $(0.1m, 0.1m)$ as the basic pillar size for experiments and train the PillarNet from scratch with batch size 16, max learning rate 3e-3 for 36 epochs. During inference, we simply follow [14] by using class-specific NMS with the IoU thresholds (0.8, 0.55, 0.55) and rectification factor β to (0.68, 0.71, 0.65) for vehicle, pedestrian and cyclist respectively.

4.1 Overall Results

Evaluation on nuScenes *test* set. We also compare our PillarNet variants with previous LiDAR-only non-ensemble methods on the nuScenes *test* set. As shown in Table 1, all our PillarNet-vgg/18/34 go beyond the stage-of-the-art methods by a large margin while running at a real-time speed of 14, 13 and 12 FPS, respectively. In addition, the promising results of PillarNet variants validate the good scalability of our PillarNet, where the performance behaves more favorably as the computational complexity rises. Typically, PillarNet-18 surprisingly surpasses the most advanced AFDetV2 by +2.3% NDS or +2.6% mAP. To the best of our knowledge, PillarNet-vgg/18/34 surpasses all the published LiDAR-only non-ensemble methods on the nuScenes Detection leaderboard on

Table 1. The LiDAR-only non-ensemble 3D detection performance comparison on the nuScenes *test* set. The table is mainly sorted by nuScenes detection score (NDS) which is the official ranking metric.

Methods	Stages	NDS	mAP	Car	Truck	Bus	Trailer	Cons.Veh.	Ped.	Motor.	Bicycle	Tr.Cone	Barrier
WYSIWYG [13]	One	41.9	35.0	79.1	30.4	46.6	40.1	7.1	65.0	18.2	0.1	28.8	34.7
PointPillars [16]	One	45.3	30.5	68.4	23.0	28.2	23.4	4.1	59.7	27.4	1.1	30.8	38.9
3DVID [45]	One	53.1	45.4	79.7	33.6	47.1	43.1	18.1	76.5	40.7	7.9	58.8	48.8
3DSSD [41]	One	56.4	42.6	81.2	47.2	61.4	30.5	12.6	70.2	36.0	8.6	31.1	47.9
Cylinder3D [53]	One	61.6	50.6	-	-	-	-	-	-	-	-	-	-
CGBS [52]	One	63.3	52.8	81.1	48.5	54.9	42.9	10.5	80.1	51.5	22.3	70.9	65.7
CVCNet [3]	One	64.2	55.8	82.6	49.5	59.4	51.1	16.2	83.0	61.8	38.8	69.7	69.7
CenterPoint [46]	Two	65.5	58.0	84.6	51.0	60.2	53.2	17.5	83.4	53.7	28.7	76.7	70.9
HotSpotNet [4]	One	66.0	59.3	83.1	50.9	56.4	53.3	23.0	81.3	63.5	36.6	73.0	71.6
AFDetV2 [14]	One	68.5	62.4	86.3	54.2	62.5	58.9	26.7	85.8	63.8	34.3	80.1	71.0
PillarNet-vgg	One	69.6	63.3	86.9	56.0	62.2	62.0	28.6	86.3	62.6	33.5	79.6	75.6
PillarNet-18	One	70.8	65.0	87.4	56.7	60.9	61.8	**30.4**	87.2	67.4	40.3	82.1	76.0
PillarNet-34	One	**71.4**	**66.0**	**87.6**	**57.5**	**63.6**	**63.1**	27.9	**87.3**	**70.1**	**42.3**	**83.3**	**77.2**

Table 2. Single- (upper group) and multi-frame (lower group) LiDAR-only non-ensemble performance comparison on the Waymo Open Dataset *test* set. "L" and "LT" mean "all LiDARs" and "top-LiDAR only", respectively. † denotes the reported results from RSN [38].

Methods	Stages	Sensors	Frames	Vehicle (L1)		Vehicle (L2)		Ped. (L1)		Ped. (L2)		Cyc. (L1)		Cyc. (L2)	
				mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
† PointPillars [16]	One	LT	1	68.60	68.10	60.50	60.10	68.00	55.50	61.40	50.10	-	-	-	-
RCD [1]	Two	-	1	71.97	71.59	65.06	64.70	-	-	-	-	-	-	-	-
CenterPoint [46]	Two	LT	1	80.20	79.70	72.20	71.80	78.30	72.10	72.20	66.40	-	-	-	-
AFDetV2 [14]	One	LT	1	80.49	80.43	72.98	72.55	79.76	**74.35**	73.71	**68.61**	**72.43**	**71.23**	**69.84**	**68.67**
PillarNet-vgg	One	LT	1	81.16	80.68	73.64	73.20	78.30	70.28	72.23	64.68	67.26	66.07	64.79	63.65
PillarNet-18	One	LT	1	81.85	81.40	74.46	74.03	79.97	72.68	73.95	67.09	67.98	66.80	65.50	64.36
PillarNet-34	One	LT	1	**82.47**	**82.03**	**75.07**	**74.65**	**80.82**	74.13	**74.83**	**68.54**	69.08	67.91	66.60	65.47
3D-MAN [43]	Multi	L	15	78.71	78.28	70.37	69.98	69.97	65.98	63.98	60.26	-	-	-	-
RSN [38]	Two	LT	3	80.70	80.30	71.90	71.60	78.90	75.60	70.70	67.80	-	-	-	-
CenterPoint [46]	Two	L	2	81.05	80.59	73.42	72.99	80.47	77.28	74.56	71.52	74.60	73.68	72.17	71.28
Pyramid R-CNN [26]	Two	L	2	81.77	81.32	74.87	74.43	-	-	-	-	-	-	-	-
AFDetV2 [14]	One	LT	2	81.65	81.22	74.30	73.89	81.26	78.05	75.47	72.41	**76.41**	**75.37**	**74.05**	**73.04**
PillarNet-vgg	One	LT	2	82.18	81.73	74.93	74.49	80.41	76.86	74.52	71.14	68.75	67.89	66.52	65.68
PillarNet-18	One	LT	2	82.68	82.25	75.53	75.12	81.71	78.29	75.91	**72.66**	70.19	69.30	68.01	67.15
PillarNet-34	One	LT	2	**83.23**	**82.80**	**76.09**	**75.69**	**82.38**	**79.02**	**76.66**	**73.46**	71.44	70.51	69.20	68.29

Mar 7, 2022. From this point on, PillarNet achieves new state-of-the-art performance using only 2D convolutions.

Evaluation on Waymo Open Dataset *test* set. We compare our PillarNet variants with previous methods on the Waymo Open Dataset *test* set. Table 2 contains two groups, where the upper group is single-frame LiDAR-only non-ensemble methods and the bottom group is multi-frame LiDAR-only non-ensemble methods. Our PillarNet-34 outperforms all the previous single-frame and multi-frame LiDAR-only models for the vehicle and pedestrian categories while running at a speed of 19 FPS separately. Our lightweight PillarNet-vgg still achieves the comparable performance for the vehicle while running at a faster speed of 24 FPS. Using merely 2D convolutions, our real-time PillarNet variants are suitable for onboard deployment.

Table 3. The single-frame LiDAR-only non-ensemble 3D AP/APH performance comparison on the Waymo Open Dataset *val* set. †: reported by [17].

Methods	Stages	Vehicle (L1)		Vehicle (L2)		Ped. (L1)		Ped. (L2)		Cyc. (L1)		Cyc. (L2)	
		mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
MVF [6]	One	62.93	-	-	-	65.33	-	-	-	-	-	-	
3D-MAN [43]	Multi	69.03	68.52	60.16	59.71	71.71	67.74	62.58	59.04	-	-	-	-
RCD [1]	Two	69.59	69.16	-	-	-	-	-	-	-	-	-	-
†SECOND [40]	One	72.27	71.69	63.85	63.33	68.70	58.18	60.72	51.31	60.62	59.28	58.34	57.05
†PointPillar [16]	One	56.62	-	-	-	59.25	-	-	-	-	-	-	-
LiDAR R-CNN [17]	Two	73.50	73.00	64.70	64.20	71.20	58.70	63.10	51.70	68.60	66.90	66.10	64.40
RangeDet [8]	One	72.85	-	-	-	75.94	-	-	-	65.80	-	-	-
MVF++ [29]	One	74.64	-	-	-	78.01	-	-	-	-	-	-	-
RSN [38]	Two	75.10	74.60	66.00	65.50	77.80	72.70	68.30	63.70	-	-	-	-
Voxel R-CNN [7]	Two	75.59	-	66.59	-	-	-	-	-	-	-	-	-
CenterPoint [46]	Two	76.70	76.20	68.80	68.30	79.00	72.90	71.00	65.30	-	-	-	-
Part-A^2 [34]	Two	77.05	76.51	68.47	67.97	75.24	66.87	66.18	58.62	68.60	67.36	66.13	64.93
PV-RCNN [32]	Two	77.51	76.89	68.98	68.41	75.01	65.65	66.04	57.61	67.81	66.35	65.39	63.98
AFDetV2 [14]	One	77.64	77.14	69.68	69.22	80.19	**74.62**	72.16	**66.95**	**73.72**	**72.74**	**71.06**	**70.12**
PillarNet-vgg	One	77.41	76.86	69.46	68.96	78.30	70.32	70.00	62.62	69.48	68.35	66.87	65.78
PillarNet-18	One	78.24	77.73	70.40	69.92	79.80	72.59	71.57	64.90	70.40	69.29	67.75	66.68
PillarNet-34	One	**79.09**	**78.59**	**70.92**	**70.46**	**80.59**	74.01	**72.28**	66.17	72.29	71.21	69.72	68.67
Two-frame 3D detection results of PillarNet variants for reference.													
PillarNet-vgg	One	78.26	77.73	70.56	70.07	80.88	77.53	72.73	69.58	67.72	66.88	65.54	64.72
PillarNet-18	One	79.59	79.06	71.56	71.08	82.11	78.82	74.49	71.35	70.41	69.57	68.27	67.46
PillarNet-34	One	79.98	79.47	72.00	71.53	82.52	79.33	75.00	71.95	70.51	69.69	68.38	67.58

Evaluation on Waymo Open Dataset *val* set. We compare our PillarNet variants with all published single-frame LiDAR-only non-ensemble methods on Waymo *val* set in Table 3. We also present the performance of PillarNet variants using two-frame-merged LiDAR points for reference. Typically, PillarNet-18 achieves the state-of-the-art performance on the vehicle category, making it a viable replacement for previous state-of-the-art 3D voxel-based methods. Our PillarNet-34 outperforms previous state-of-the-art works with remarkable performance gains (+1.24 for the vehicle in terms of mAPH of LEVEL_2 difficulty). Excluding the latest voxel-based detector AFDetV2 with self-calibrated module and channel-wise and spatial-wise attention, PillarNet-34 outperforms the previous one-stage and two-stage 3D detectors for the vehicle and pedestrian detection while operating at super real-time speed. With the two-frame input, PillarNet on variant backbones consistently show the superior performance compared with their single-frame counterparts. However, for the cyclist detection, two-frame results are not the best. The reason may be the unbalanced sample distribution of three categories. The number of vehicles, pedestrians and cyclists scattered in the Waymo train set are 4352210, 2037627 and 49518 respectively. The training process using two frames aggravates the adverse effect, and this issue may be alleviated by addressing the unbalanced sample distribution.

4.2 Ablation Studies

In this section, we investigate the individual components of the proposed PillarNet with extensive ablation experiments on the *val* set of nuScenes Dataset.

Table 4. The analysis of each component of PillarNet with the same training schedules as SECOND and also comparison with the two baselines (*i.e.*, PointPillars and SECOND) on nuScenes *val* dataset. †: reported by used codebase.

Methods	FPS	mAP	NDS	mATE	mASE	mAOE	mAVE	mAAE
†CenterPoint-PointPillars [16]	31	50.26	60.22	31.32	25.94	39.50	32.54	19.79
†CenterPoint-SECOND [40]	8	59.56	66.76	29.22	25.51	30.24	25.91	19.34
PointPillars (0.075m)[1]	9	48.63	59.51	30.70	26.35	35.81	35.52	19.70
PillarNet-18(neckv1)	17	57.87	66.16	29.88	26.05	27.86	25.78	18.14
PillarNet-18(neckv2)	16	58.53	66.41	29.82	26.05	29.56	24.53	18.65
PillarNet-18(neckv2-D)	16	59.40	66.96	29.07	25.86	29.61	24.69	18.17
PillarNet-18(neckv3)	16	59.48	67.15	29.03	25.83	27.54	24.54	18.90
PillarNet-18(OD-IoU)	16	59.51	67.09	28.51	25.57	29.31	24.63	18.64
PillarNet-18(OD-GIoU)	16	59.69	67.35	28.50	25.78	27.57	24.74	18.37
PillarNet-18(OD-DIoU)	16	59.72	67.39	28.40	25.81	27.38	24.67	18.41
PillarNet-18(IoU)	16	59.82	67.16	28.92	25.53	28.37	25.63	19.07
PillarNet-18	16	59.90	67.39	27.72	25.20	28.93	24.67	19.11

Analysis of PillarNet improvements. The key contribution can be summarized into two parts: the designed PillarNet architecture (*i.e.*, encoder and neck networks) and the IoU-related modules (*i.e.*, Orientation-Decoupled IoU (OD-IoU) regression loss and IoU-Aware rectification). To analyze how our designed encoder and neck networks improve the 3D detection performance, we use the same hyper-parameters settings as CenterPoint-SECOND.

Encoder network. Compared with CenterPoint-PointPillars [16] (1^{st} row of Table 4), our newly introduced encoder network can significantly improve the detection performance by about +7.61% mAP and +5.94% NDS. Using the heavy encoder with extra stage 5 in 3^{rd} to 6^{th} rows can boost the 3D detection performance by a large margin. Therein, the enriched semantic features from encoder stage 5 in 4^{th} to 5^{th} rows perform better than the aggressive fusion strategy in 6^{th} row.

Neck network. Compared with the naive neck module neckv1, as shown in Table 4, our fusion design from 4^{th} to 6^{th} rows with a group of convolution layers can improve detection performance by a large margin. The performance difference between neckv2 and neckv2-D shows that the dense convolutions enable stronger semantic abstraction at the object center than its sparse counterparts, due to LiDAR points sparsely scattering on the surface of the objects.

OD-IoU regression loss. All three types of losses (*i.e.*, OD-IoU, OD-GIoU and OD-DIoU) play a role in the critical positioning accuracy, while the OD-DIoU loss brings a maximum boost with +0.24% mAP or +0.24% NDS.

IoU-Aware rectification. The IoU-Aware rectification alleviates the misalignment between localization confidence and classification score. Adding IoU-Aware rectification benefits the IoU-based mAP with +0.34% increase.

Analysis of model variants. We investigate PillarNet-vgg/18/34 with different model complexity by detaching IoU-related modules for a clean comparison.

Table 5. The effect of different PillarNet variants by detaching two IoU-related modules.

Models	FPS	mAP	NDS
PillarNet-vgg	16	57.67	65.71
PillarNet-18	16	59.41	67.09
PillarNet-34	14	59.98	67.50

Table 6. The effect of different pillar sizes and its associated stages in PillarNet-18 encoder.

Pillar size	FPS	encoder stages	mAP	NDS
0.075m	16	(1x 2x 4x 8x 16x)	59.48	67.15
0.075*2m	16	(2x 4x 8x 16x)	58.70	66.56
0.075*4m	16	(4x 8x 16x)	57.87	66.05
0.075*8m	14	(8x 16x)	55.37	64.20

Table 5 shows that our PillarNet architecture can benefit from increasing model capacity with slightly more FLOPs and inference time. The good scalability can lead the pathway for deployment according to practical needs.

Analysis of pillar sizes. We investigate pillar size on detection performance by detaching IoU-related modules for a clean comparison. Specifically, we castrate the associated encoder stages to suit particular pillar scale, where a larger pillar size requires fewer encoder stages. From Table 6 and Fig 1, we can see that PillarNet benefit more from finer pillar scale and deeper pillar feature encoding. The much higher performance of PillarNet with 0.3m and 0.6m over PointPillars with 0.2m manifests the effectiveness of our architectural design. Moreover, PointPillars [16] with 0.075m in 3^{rd} row of Table 4 performs slightly worse than that of 0.2m. That is because the used lightweight encoder network hinders the gain from small pillar size. This also implies the importance of hierarchical pillar feature encoding of PillarNet for better performance with limited resource costs.

Runtime analysis. We analyze inference runtime by fairly comparing with two baseline counterparts. our PillarNet-18 achieves a good speed-accuracy trade-off with 16 FPS than CenterPoint-SECOND of 8 FPS on nuScenes Dataset, and 21 FPS than CenterPoint-PointPillars of 19 FPS on Waymo Open Dataset. The slow inference speed for PillarNet with coarser pillar size and reduced encoder stages in Table 6 may be due to the fact that cuda *atomic max* operation struggles to handle more inside points per pillar based on global memory. This issue can be alleviated by the input point cloud sub-sampling or other efficient operation (*e.g.,* streaming pillarization as [5]).

5 Conclusions

In this work, we propose a real-time and high-performance one-stage 3D object detector. From the perspective of "encoder-neck-head" architecture design, PillarNet achieves the scalability and flexibility for the hard-balanced pillar size and model complexities. We expect that our findings will stimulate further research into pillar-based point cloud representation learning.

Acknowledgements. This work was supported in part by NSFC (61906119), and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102). We thank Huawei Noah's Ark Lab gratefully for sponsoring large-scale GPUs.

References

1. Bewley, A., Sun, P., Mensink, T., Anguelov, D., Sminchisescu, C.: Range conditioned dilated convolutions for scale invariant 3d object detection. In: Conference on Robot Learning (CoRL) (2020)
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
3. Chen, Q., Sun, L., Cheung, E., Yuille, A.L.: Every view counts: Cross-view consistency in 3d object detection with hybrid-cylindrical-spherical voxelization. Advances in Neural Information Processing Systems **33**, 21224–21235 (2020)
4. Chen, Q., Sun, L., Wang, Z., Jia, K., Yuille, A.: Object as hotspots: An anchor-free 3d object detection approach via firing of hotspots. In: European conference on computer vision. pp. 68–84. Springer (2020)
5. Chen, Q., Vora, S., Beijbom, O.: Polarstream: Streaming lidar object detection and segmentation with polar pillars. arXiv preprint arXiv:2106.07545 (2021)
6. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 1907–1915 (2017)
7. 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 (2021)
8. Fan, L., Xiong, X., Wang, F., Wang, N., Zhang, Z.: Rangedet: In defense of range view for lidar-based 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2918–2927 (2021)
9. Graham, B., Engelcke, M., Van Der Maaten, L.: 3d semantic segmentation with submanifold sparse convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9224–9232 (2018)
10. Graham, B., van der Maaten, L.: Submanifold sparse convolutional networks. arXiv preprint arXiv:1706.01307 (2017)
11. 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)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
13. Hu, P., Ziglar, J., Held, D., Ramanan, D.: What you see is what you get: Exploiting visibility for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11001–11009 (2020)
14. Hu, Y., Ding, Z., Ge, R., Shao, W., Huang, L., Li, K., Liu, Q.: Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds (2021)
15. Ku, J., Mozifian, M., Lee, J., Harakeh, A., Waslander, S.L.: Joint 3d proposal generation and object detection from view aggregation. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1–8. IEEE (2018)
16. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12697–12705 (2019)

17. 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)
18. Liang, M., Yang, B., Chen, Y., Hu, R., Urtasun, R.: Multi-task multi-sensor fusion for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7345–7353 (2019)
19. Liang, M., Yang, B., Wang, S., Urtasun, R.: Deep continuous fusion for multi-sensor 3d object detection. In: Proceedings of the European conference on computer vision (ECCV). pp. 641–656 (2018)
20. Liang, Z., Zhang, Z., Zhang, M., Zhao, X., Pu, S.: Rangeioudet: Range image based real-time 3d object detector optimized by intersection over union. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7140–7149 (2021)
21. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
22. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
23. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
24. Liu, Z., Zhao, X., Huang, T., Hu, R., Zhou, Y., Bai, X.: Tanet: Robust 3d object detection from point clouds with triple attention. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 11677–11684 (2020)
25. Liu, Z., Tang, H., Lin, Y., Han, S.: Point-voxel cnn for efficient 3d deep learning. Advances in Neural Information Processing Systems **32** (2019)
26. 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)
27. Noh, J., Lee, S., Ham, B.: Hvpr: Hybrid voxel-point representation for single-stage 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14605–14614 (2021)
28. 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)
29. Qi, C.R., Zhou, Y., Najibi, M., Sun, P., Vo, K., Deng, B., Anguelov, D.: Offboard 3d object detection from point cloud sequences. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6134–6144 (2021)
30. 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)
31. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 658–666 (2019)
32. 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)

33. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 770–779 (2019)
34. Shi, S., Wang, Z., Shi, J., Wang, X., Li, H.: 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 **43**(8), 2647–2664 (2020)
35. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. Advances in neural information processing systems **27** (2014)
36. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
37. 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)
38. Sun, P., Wang, W., Chai, Y., Elsayed, G., Bewley, A., Zhang, X., Sminchisescu, C., Anguelov, D.: Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5725–5734 (2021)
39. Wang, B., An, J., Cao, J.: Voxel-fpn: multi-scale voxel feature aggregation in 3d object detection from point clouds. arXiv preprint arXiv:1907.05286 (2019)
40. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
41. Yang, Z., Sun, Y., Liu, S., Jia, J.: 3dssd: Point-based 3d single stage object detector. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11040–11048 (2020)
42. 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)
43. Yang, Z., Zhou, Y., Chen, Z., Ngiam, J.: 3d-man: 3d multi-frame attention network for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1863–1872 (2021)
44. 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)
45. Yin, J., Shen, J., Guan, C., Zhou, D., Yang, R.: Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11495–11504 (2020)
46. 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)
47. Yoo, J.H., Kim, Y., Kim, J., Choi, J.W.: 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In: European Conference on Computer Vision. pp. 720–736. Springer (2020)
48. Zheng, W., Tang, W., Chen, S., Jiang, L., Fu, C.W.: Cia-ssd: Confident iou-aware single-stage object detector from point cloud. In: AAAI (2021)
49. Zheng, W., Tang, W., Jiang, L., Fu, C.W.: Se-ssd: Self-ensembling single-stage object detector from point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14494–14503 (2021)

50. Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D.: Distance-iou loss: Faster and better learning for bounding box regression. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 12993–13000 (2020)
51. 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)
52. Zhu, B., Jiang, Z., Zhou, X., Li, Z., Yu, G.: Class-balanced grouping and sampling for point cloud 3d object detection. arXiv preprint arXiv:1908.09492 (2019)
53. Zhu, X., Zhou, H., Wang, T., Hong, F., Li, W., Ma, Y., Li, H., Yang, R., Lin, D.: Cylindrical and asymmetrical 3d convolution networks for lidar-based perception. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)