PillarNeXt: Rethinking Network Designs for 3D Object Detection in LiDAR Point Clouds

Jinyu Li Chenxu Luo Xiaodong Yang*
QCraft

Abstract

In order to deal with the sparse and unstructured raw point clouds, most LiDAR based 3D object detection research focuses on designing dedicated local point aggregators for fine-grained geometrical modeling. In this paper, we revisit the local point aggregators from the perspective of allocating computational resources. We find that the simplest pillar based models perform surprisingly well considering both accuracy and latency. Additionally, we show that minimal adaptions from the success of 2D object detection, such as enlarging receptive field, significantly boost the performance. Extensive experiments reveal that our pillar based networks with modernized designs in terms of architecture and training render the state-of-the-art performance on two popular benchmarks: Waymo Open Dataset and nuScenes. Our results challenge the common intuition that detailed geometry modeling is essential to achieve high performance for 3D object detection.

1. Introduction

3D object detection in LiDAR point clouds is an essential task in an autonomous driving system, as it provides crucial information for subsequent onboard modules, ranging from perception [24, 25], prediction [27, 42] to planning [2, 23]. There have been extensive research efforts on developing sophisticated networks that are specifically designed to cope with point clouds in this field [14, 32, 33, 43, 47].

Due to the sparse and irregular nature of point clouds, most existing works adopt the grid based methods, which convert point clouds into regular grids, such as pillar [14], voxel [47] and range view [26], such that regular operators can be applied. However, it is a common belief that the grid based methods (especially for pillar) inevitably introduce information loss, leading to inferior results, in particular for small objects (e.g., pedestrians). Recent research [33] proposes the hybrid design for fine-grained geometrical modeling to combine the point and gird based representations. We name all the above operators as local point aggregators because they aim to aggregate point features in a certain neighborhood. We observe that the current mainstream of 3D object detection is to develop more specific operators for point clouds, while leaving the network architectures almost unexplored. Most existing works are still built upon the very original architectures of SECOND [44] or PointPillars [14], which lacks of modernized designs.

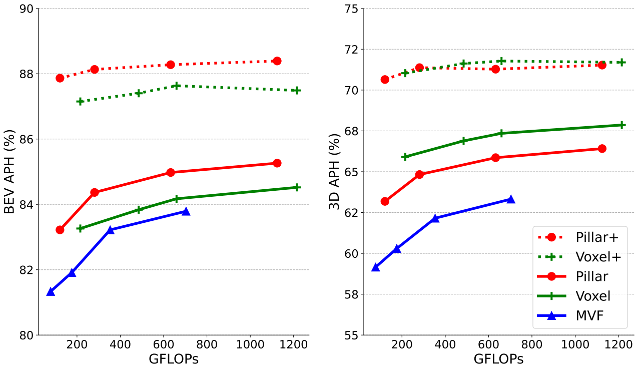

Figure 1. Overview of pillar, voxel and multi-view fusion (MVF) based 3D object detection networks under different GFLOPs. The Pillar/Voxel+ denote the corresponding models, which are trained with an enhanced strategy. We report the L2 BEV and 3D APH of vehicle on the validation set of Waymo Open Dataset.

Meanwhile, in a closely related area, 2D object detection in images has achieved remarkable progress in both accuracy and efficiency, which can be largely attributed to the advances in network architectures. Among them, the powerful backbones (e.g., ResNet [12], ViT [10] and Swin Transformers [20]) as well as the effective necks (e.g., FPN [17], BiFPN [40] and YOLOF [6]) are in particular notable. Therefore, the research focus of 2D object detection is largely different from that of 3D object detection.

In light of the aforementioned observations, we rethink what should be the focus for 3D object detection in LiDAR point clouds. Specifically, we revisit the two fundamental issues in designing a 3D object detection model: the local point aggregator and the network architecture.

First, we compare local point aggregators from a new perspective, i.e., the computational budget. A fine-grained aggregator usually demands more extensive computing resources than the coarse one. For instance, the voxel based aggregator employs 3D convolutions, which require more network parameters and run much slower than 2D convo-

*Corresponding author `xiaodong@qcraft.ai` lutions. This raises a question of how to effectively allocate the computational budget or network capacity. Should we spend the resources on fine-grained structures or assign them to coarse grids? Surprisingly, as shown in Figure 1, when training with an enhanced strategy and under a comparable budget, a simple pillar based model can achieve superior or on par performance with the voxel based model, even for small objects such as pedestrians, while significantly outperform the multi-view fusion based model. This challenges the actual performance gain and the necessity of fine-grained local 3D structures. Our finding is also consistent with the recent works [19, 31] in general point cloud analysis, demonstrating that different local point aggregators perform similarly under strong networks.

Second, for the network architecture, we do not aim to propose any domain specific designs for point clouds, instead, we make minimal adaptations from the success of 2D object detection and show that they already outperform most of the existing methods with specific designs for point clouds. One key finding is that enlarging receptive field properly brings significant improvement. Unlike previous works [32, 47] that rely on multi-scale feature fusion, we show that a single scale at the final stage with sufficient receptive field obtains better performance. Such promising results suggest that 3D object detection can inherit the successful practices well developed in 2D domain.

Levaraging on the findings above, we propose a pillar based network, dubbed as **PillarNeXt**, which leads to the state-of-the-art results on two popular benchmarks [3, 36]. Our approach is simple yet effective, and enjoys strong scalability and generalizability. We develop a series of networks with different trade-offs between accuracy and latency by tuning the number of network parameters, which can be used for both on-board [14] and off-board [30] applications in autonomous driving.

Our main contributions can be summarized as follows. (1) To our knowledge, this is the first work that compares different local point aggregators (pillar, voxel and multi-view fusion) from the perspective of computational budget allocation. Our findings challenge the common belief by showing that pillar can achieve comparable 3D mAP and better bird's eye view (BEV) mAP compared to voxel, and substantially outperform multi-view fusion in both 3D and BEV mAP. (2) Inspired by the success of 2D object detection, we find that enlarging receptive field is crucial for 3D object detection. With minimal adaptions, our detectors outperform existing methods with sophisticated designs for point clouds. (3) Our networks with appropriate training achieve superior results on two large-scale benchmarks. We hope our models and related findings can serve as a strong and scalable baseline for future research in this community. Our code and model will be made available at https://github.com/qcraftai/pillarnext.

2. Related Work

LiDAR based 3D Object Detection. Existing methods can be roughly categorized into point, grid and hybrid based representations, according to the local point aggregators. As a point based method, PointRCNN [34] generates proposals using [29] and then refines each proposal by RoI pooling. However, conducting neighboring point aggregation in such methods is extremely expensive, so they are not feasible to handle large-scale point clouds in autonomous driving. On the other hand, the grid based methods discretize point clouds into structured grids, where 2D or 3D convolutions can be applied. In [47] VoxelNet partitions a 3D space into voxels and aggregates point features inside each voxel, then dense 3D convolutions are used for context modeling. SECOND [44] improves the efficiency by introducing sparse 3D convolutions. PointPillars [14] organizes point clouds as vertical columns and adopts 2D convolutions. Another grid based representation is the range view [26] that can be also efficiently processed by 2D convolutions. The multi-view fusion methods [43, 46] take advantage of both pillar/voxel and range view based representations.

In spite of efficiency, it is commonly and intuitively believed that the grid based methods induce fine-grained information loss. Therefore, the hybrid methods are proposed to incorporate point features into grid representations [21, 33]. In this work, we instead focus on the basic network architecture and associated training, and show that the fine-grained local geometrical modeling is overestimated.

Feature Fusion and Receptive Field. The multi-scale feature fusion starts from feature pyramid network (FPN) [17] that aggregates hierarchical features in a top-down manner. It is widely used in 2D object detection to combine high-level semantics with low-level spatial cues. In [18], PANet further points out the bottom-up fusion is also important. Both of them perform fusion by adding up feature maps directly. BiFPN [40] shows that features from different scales contribute unequally, and adopts learnable weights to adjust the importance. As an interrelated factor of feature fusion, receptive field is also broadly studied and verified in 2D detection and segmentation. In [4], the atrous spatial pyramid pooling (ASPP) is proposed to sample features with multiple effective receptive fields. TridentNet [16] applies three convolutions with different dilation factors to make receptive field range to match with object scale range. A similar strategy is also introduced in YOLOF [6] that employs a dilated residual block for enlarging receptive field and meanwhile keeping original receptive field.

Although these techniques regarding feature fusion and receptive field have been extensively adopted in 2D domain, they are hardly discussed in 3D domain. Most previous networks in this field still follow the architecture of VoxelNet [47]. In this work, we aim to integrate the up-to-date designs into 3D object detection networks.

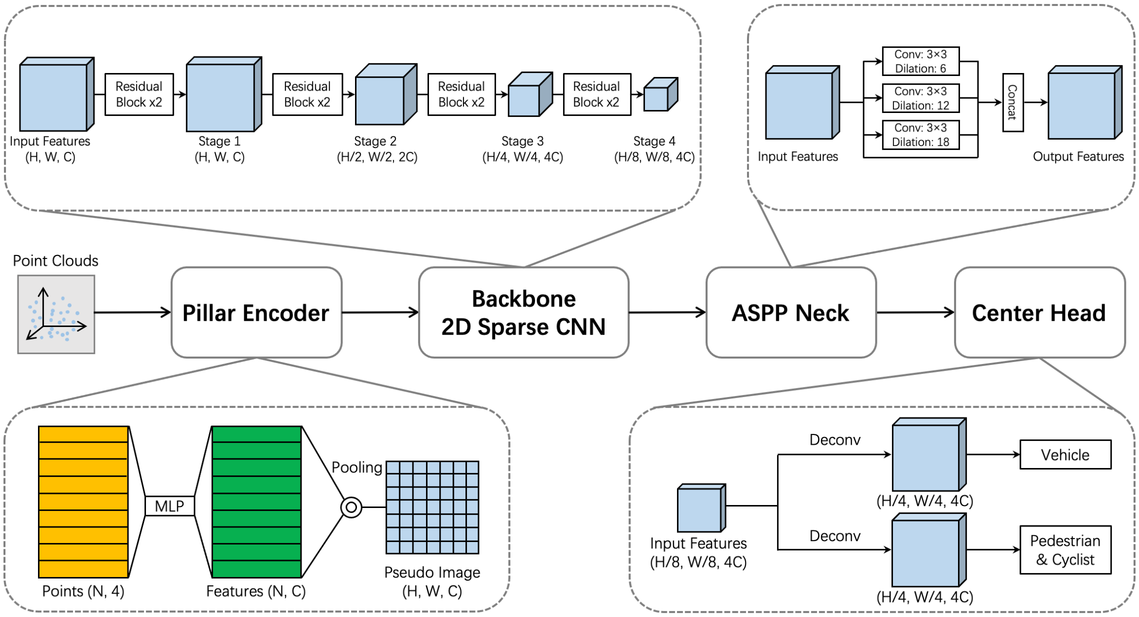

Figure 2. A schematic overview of the network architecture of the proposed PillarNeXt. Our model takes raw point clouds as input and relies on a simple pillar encoder, which consists of MLPs to convert point clouds into a pseudo image. We then apply ResNet-18 with sparse convolutions as the backbone, and adopt ASPP based neck to enlarge receptive field. After that, we upsample the feature maps to yield more detailed representations, and use the center based multi-group head to produce the detection output.

Model Scaling. This aspect has been well studied in image domain, including classification, detection and segmentation tasks. It is observed that jointly increasing the depth, width and resolution improves the accuracy. EfficientNet in [39] proposes a compound scaling rule for classification, and this rule is later on extended to object detection [9, 40]. It is a general consensus that the model capacity affects the model performance. Therefore, the comparison of different methods should be under the consideration of model capacity in order to receive sound conclusions.

There is only one work [41] that studies model scaling in 3D object detection, to the best of our knowledge. It scales the depth, width and resolution of SECOND [44] to find the importance of each component. It however only focuses on a single type of model, while we systematically compare different local point aggregators under similar computational budgets across a wide range of model scales.

3. Network Architecture Overview

We focus on the grid based models due to the runtime efficiency and proximity to 2D object detection. Typically, a grid based network is composed of (i) a grid encoder to convert raw point clouds into structured feature maps, (ii) a backbone for general feature extraction, (iii) a neck for multi-scale feature fusion, and (iv) a detection head for the task-specific output. Existing works often couple all these components together. In this section, we have them decoupled and review each part briefly.

3.1. Grid Encoder

A grid encoder is used to discretize point clouds into structured grids, and then convert and aggregate the points within each grid into the preliminary feature representation. In this work, we target at the following three grid encoders.

- **Pillar:** A pillar based grid encoder arranges points in vertical columns, and applies multilayer perceptrons (MLPs) followed by max pooling to extract pillar features, which are represented as a pseudo image [14].
- **Voxel:** Similar to pillar, the voxel based grid encoder organizes points in voxels and obtains corresponding features [47]. Compared with pillar, the voxel encoder preserves details along the height dimension.
- **Multi-View Fusion (MVF):** A MVF based grid encoder combines the pillar/voxel and range view based representations. Here we follow [43] to incorporate a pillar encoder with a cylindrical view based encoder that groups points in the cylindrical coordinates.

3.2. Backbone and Neck

A backbone performs further feature abstraction based on the preliminary features extracted by the grid encoder. For fair comparisons, we utilize ResNet-18 as the backbone, since it is commonly used in the previous works [32, 45, 49]. Specifically, we make use of sparse 2D convolutions in the backbone with the pillar or MVF based encoder, and sparse 3D convolutions in the backbone with the voxel based en-

Model	Channels	#Params (M)	FLOPs (G)	Latency (ms)	Vehicle		Pedestrian	
					3D	BEV	3D	BEV
Pillar-T	[32, 64, 128, 128]	1.65	70	52	**62.03**	**82.26**	**67.63**	**75.76**
MVF-T	[32, 64, 128, 128]	3.44	78	137	59.16	81.33	64.10	73.42
Pillar-S	[42, 84, 168, 168]	2.83	121	79	63.18	**83.22**	68.12	**76.37**
Voxel-S	[12, 24, 48, 96]	1.53	121	169	**64.67**	82.45	**69.10**	76.25
MVF-S	[44, 88, 176, 176]	6.38	148	186	61.06	82.51	65.15	74.24
Pillar-B	[64, 128, 256, 256]	6.53	281	103	64.83	**84.37**	69.04	76.96
Voxel-B	[16, 32, 64, 128]	2.71	215	186	**65.91**	83.26	**70.18**	**77.18**
MVF-B	[68, 136, 272, 272]	15.02	353	291	62.15	83.22	62.12	75.00
Pillar-L	[96, 192, 384, 384]	14.63	632	194	65.86	**84.98**	68.42	76.67
Voxel-L	[28, 56, 112, 224]	8.27	660	299	**67.35**	84.17	**70.47**	**77.44**
MVF-L	[96, 192, 384, 384]	29.67	704	390	63.32	83.79	66.87	75.34
Enhanced Versions								
Pillar-B+	[64, 128, 256, 256]	6.53	281	103	**71.37**	**88.13**	73.93	80.28
Voxel-B+	[16, 32, 64, 128]	2.71	215	186	71.03	87.15	**74.48**	**80.54**
Pillar-L+	[96, 192, 384, 384]	14.63	632	194	71.28	**88.28**	72.92	79.59
Voxel-L+	[28, 56, 112, 224]	8.27	660	299	**71.78**	87.63	**74.40**	**80.39**

Table 1. Comparison of the pillar, voxel and MVF based networks with model scales from tiny, small, base to large. We report the L2 3D and BEV APH on vehicle and pedestrian on the validation set of WOD. Groups 1 and 2 correspond to the regular and enhanced versions.

coder. A neck can be then utilized to aggregate features from the backbone for enlarging receptive field and fusing multi-scale context. However, it has not been well explored for object detection in point clouds compared with images. We aim to close this gap by integrating the advanced neck designs from 2D object detection, such as BiFPN [40] using improved multi-level feature fusion or ASPP [4] using convolutions with multiple dilated rates on a single feature level, into the model architectures of 3D object detection.

3.3. Detection Head

In the pioneering works of SECOND [44] and PointPillars [14], the anchor based detection head is employed to pre-define the axis-aligned anchors at each location on the input feature maps to head. CenterPoint [45] instead represents each object by its center point, and predicts a centerness heatmap where the regression of bounding box is realized at each center location. Due to its simplicity and superior performance, we adopt the center based detection head in all our networks. We show a set of simple modifications in head, such as feature upsampling, multi-grouping and IoU branch, improve the performance notably.

4. Experiments

In this section, we start from introducing the experimental setup including datasets and implementation details. We then perform comprehensive network design studies on each component in a 3D object detection model. In the end, we present extensive comparisons with the state-of-the-art methods on two popular benchmarks.

4.1. Experimental Setup

We conduct experiments on two large-scale autonomous driving benchmarks: Waymo Open Dataset (WOD) [36] and nuScenes [3]. **WOD** consists of 798 sequences (160K frames) for training and 202 sequences (40K frames) for validation, which are captured with 5 LiDARs at 10Hz. Following the official protocol, we use the average precision (AP) and average precision weighted by heading (APH) as the evaluation metrics. We break down the performance into two difficulty levels, L1 and L2, where the former evaluates objects with at least 5 points and the latter cover all objects with at least one point. We set the IoU thresholds for vehicle, pedestrian and cyclist to 0.7, 0.5 and 0.5. In addition to 3D AP/APH, we also report the results under BEV. **nuScenes** contains 1,000 scenes of roughly 20 seconds each, captured by a 32-beam LiDAR at 20Hz. Annotations are available on keyframes at 2Hz. We follow the official evaluation metrics by averaging over 10 classes under mean average precision (mAP) and nuScenes detection score (NDS), which is a weighted average of mAP and ATE, ASE, AOE, AVE and AAE measuring the translation, scale, orientation, velocity and attribute related errors.

We implement our networks in PyTorch [28]. All models are trained by using AdamW [22] as the optimizer and under the one-cycle [35] learning rate schedule. As for **WOD**, the detection range is set to [-76.8m, 76.8m] horizontally and [-2m, 4m] vertically. Our pillar size is 0.075m in x/y-axis (with 0.15m in z-axis for voxel based). For MVF based, we keep the same pillar size and use [1.8°, 0.2m] for yaw and z-axis in the cylindrical view. We train each model for

Method	Vehicle L1		Vehicle L2		Pedestrian L1		Pedestrian L2	
	AP	APH	AP	APH	AP	APH	AP	APH
Neck of PillarNet [32]	91.39	90.58	84.54	83.72	**87.90**	**83.02**	81.93	77.20
FPN [17]	92.17	91.35	85.96	85.13	87.88	82.91	82.05	77.23
BiFPN [40]	92.71	91.90	86.92	86.09	87.86	82.88	82.05	77.23
Plain	91.01	90.19	83.86	83.04	87.59	82.61	81.51	76.71
Dilated Block [6]	92.70	91.90	86.61	85.79	87.84	82.91	**82.09**	**77.29**
ASPP [4]	**92.77**	**91.94**	**86.99**	**86.14**	87.74	82.85	82.00	77.26

Table 2. Comparison of different neck modules integrated in our network. Groups 1 and 2 correspond to the multi-scale and sing-scale necks, respectively. We report the L1 and L2 BEV AP and APH for vehicle and pedestrian on the validation set of WOD.

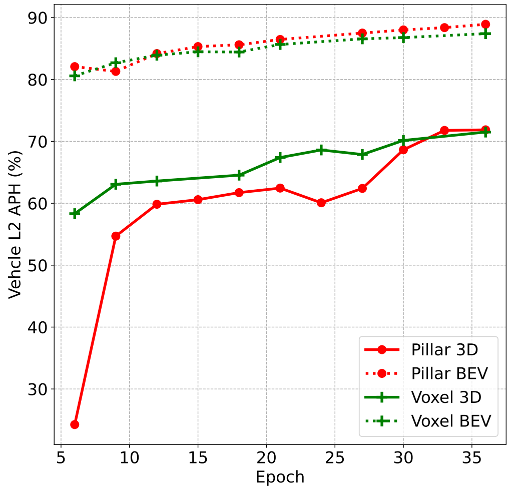

Figure 3. Comparison of the learning behaviors between the pillar and voxel based models. We report the L2 3D and BEV APH of vehicle on the validation set on WOD.

12 epochs and take 3 frames as input unless otherwise specified. For inference, we use the non-maximum suppression (NMS) thresholds of 0.7, 0.2 and 0.25 for vehicle, pedestrian and cyclist. As for **nuScenes**, we set detection range to [-50.4m, 50.4m] horizontally and take 10 frames as input. For inference, we use NMS threshold of 0.2 for all classes. Other settings are the same as WOD. We report inference latency on a single NVIDIA TITAN RTX GPU. More implementation details can be found in Appendix.

4.2. Network Design Study

We perform extensive studies to analyze and understand the contribution of each individual network design. We first evaluate the impact of grid encoder, and demonstrate the importance of neck module, then investigate the effect of resolution, finally summarize the components one by one to show the improvement roadmap.

4.2.1 Study of Grid Encoders

We begin with evaluating the three representative grid encoders as introduced in Section 3.1, i.e., pillar, voxel and MVF. Although the three encoders have been proposed for a long time, they have never been fairly compared under the same network architecture and grid resolution. Here, we experiment with a sparse ResNet-18 [32] as the backbone for its effectiveness and efficiency. We scale the width of the backbone and obtain a series of networks ranging from tiny, small, base to large, namely Pillar/Voxel/MVF-T/S/B/L. Table 1 lists the channel and parameter numbers, FLOPs, and latency of each model. Note there is no Voxel-T as FLOPs of voxel based models are much higher and the smallest one starts from a similar computational cost as Pillar-S.

In Size	Backbone ↓	Head ↑	Out Size	Veh	Ped	Latency
0.3	1	1	0.3	65.0	67.2	255
0.075	8	1	0.6	62.8	66.6	131
0.075	8	2	0.3	64.8	69.0	173

Table 3. Comparison of different resolutions. We adopt the pillar size (m) to represent the resolutions of input grids and output features (consumed by head). We evaluate the overall downsampling rate in backbone and the upsampling rate in head. We report the L2 3D APH and latency (ms) on the validation set of WOD.

For the first part in Table 1, we compare the three grid encoders with different model scales under the regular training schedule (i.e., 12 epochs), which is commonly adopted. As can be seen in the table, under BEV APH, the pillar encoder performs favorably on vehicle and is comparable on pedestrian, while with remarkably lower latency. This conclusion is further supported by the per-class comparison on nuScenes in Table 8. Note pedestrians usually take only a few pillars in the perspective of BEV, nevertheless, the pillar encoder is sufficient to achieve superior or on-par performance, including for small objects.

However, the pillar encoder still lacks behind under 3D APH. To further study the reasons for this gap, we enhance the models by first extending the training schedule to 36 epochs with an extra IoU regression loss [32], and then incorporating an IoU score branch [13] in the multi-group detection head (one for vehicle and the other for pedestrian and cyclist, as illustrated in Figure 2). We call the models trained under this enhanced strategy as Pillar/Voxel+. As compared in the second part of Table 1, the pillar models achieve comparable or even better results than

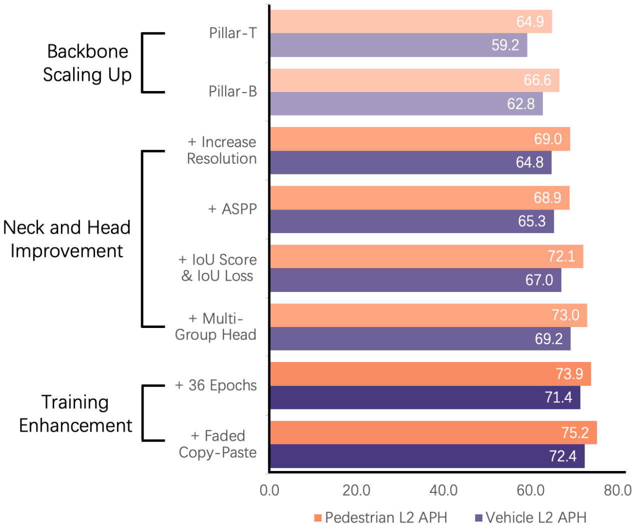

Figure 4. Improvement by each individual component to illustrate the performance boosting roadmap. We report L2 APH of vehicle and pedestrian on the validation set of WOD.

the voxel models on vehicle under 3D APH, while running much faster. We hypothesize that without explicit height modeling, the pillar based networks require refined designs such as longer training to be able to fully converge. This challenges the common belief that the pillar encoder loses height information, and suggests that the fine-grained local geometrical modeling may not be necessary.

This counter-intuitive result motivates us to rethink how to efficiently and effectively allocate the computational resources for a 3D object detection network. The pillar based models allocate resources only in the BEV space, while the voxel and MVF based methods also spend computations along the height dimension. When comparing these methods, previous works fail to take into account the computational budget. In our experiments, we show that under similar FLOPs, allocating computations to the height dimension is not beneficial. We conclude that investing all FLOPs in the BEV space is not only more efficient but also more effective, as shown in Figure 1 and Tables 1 and 8.

It also reveals that training matters. Most previous works usually employ the regular or short training schedule for comparison, which could result in different conclusions. We demonstrate the learning behaviors of pillar and voxel based networks in Figure 3. Interestingly, the pillar model is found to converge faster than the voxel model in BEV APH. While for 3D APH, the pillar model converges much slower than the voxel model. However, this gap diminishes when training continues for sufficient epochs. This indicates that the performance gap in 3D APH between pillar and voxel reported by the previous methods are partially caused by their different convergence rates, instead of the more fine-grained geometrical modeling in voxel.

4.2.2 Study of Neck Modules

In the previous study, we conclude that the different local point aggregators may not be essential to the final results. In the following, we show that a simple upgrade on the network architecture improves the performance greatly. In particular, we focus on the neck module design, which has not been well explored in 3D object detection.

Most current networks in the field rely on the multi-scale fusion as used in [32, 47], which upsample feature maps from different stages to the same resolution and then concatenate them. How to design a neck module to conduct more effective feature aggregation has been extensively researched in 2D object detection, but most advanced techniques have not been adopted in 3D object detection. We first integrate with the two popular designs, i.e., FPN [17] and BiFPN [40]. As shown in the first group of Table 2, we observe up to 2.38% improvement on vehicle over the neck developed in the most recent work PillarNet [32].

One main goal of using multi-scale features is to deal with large variations of object scales. However, 3D objects in the BEV space do not suffer from such a problem. This motivates us to rethink whether the multi-scale representation is required for 3D object detection. We therefore further investigate three single-scale neck modules. The baseline is a plain neck using a residual block without downsampling or upsampling, which gets inferior performance due to the limited receptive field. In YOLOF [6], it is argued that 2D object detector performs better when the receptive field matches with the object size. Inspired by this observation, we apply the dilated blocks as in YOLOF to enlarge the receptive field and yield better performance on vehicle. We also integrate with the ASPP block [4] and obtain up to 3.13% improvement on vehicle compared with the neck used in PillarNet. All the designs achieve comparable performance on pedestrian. These comparisons collectively imply that the multi-scale features may not be necessary, instead, enlarging receptive field plays the key role.

This study demonstrates that simply adapting the neck modules from 2D object detection brings non-trivial improvements to 3D object detection, which is encouraging to explore more successful practices in the image domain to upgrade the network designs for point clouds.

4.2.3 Study of Resolutions

Intuitively, a smaller grid size retains more fine-grained information but requires a higher computational cost. Downsampling can effectively reduce the cost but degrade the performance. We experiment with different grid and feature resolutions by changing grid sizes and feature sampling rates in backbone and head. As shown in Table 3, if the output feature resolution is fixed (0.3), using a large grid size (0.075 to 0.3) does not affect the performance of large ob-

Method	Frames	Vehicle L1		Vehicle L2		Pedestrian L1		Pedestrian L2		Cyclist L1		Cyclist L2	
		AP	APH	AP	APH	AP	APH	AP	APH	AP	APH	AP	APH
SST-TS* [11]	1	76.22	75.79	68.04	67.64	81.39	74.05	72.82	65.93	-	-	-	-
SWFormer [37]	1	77.8	77.3	69.2	68.8	80.9	72.7	72.5	64.9	-	-	-	-
PillarNet-18 [32]	1	78.24	77.73	70.40	69.92	79.80	72.59	71.57	64.90	70.40	69.29	67.75	66.68
AFDetV2 [13]	1	77.64	77.14	69.68	69.22	80.19	74.62	72.16	66.95	73.72	72.74	71.06	70.12
PV-RCNN++* [33]	1	79.25	78.78	70.61	70.18	81.83	76.28	73.17	68.00	73.72	72.66	71.21	70.19
PillarNeXt-B	1	78.40	77.90	70.27	69.81	82.53	77.14	74.90	69.80	73.21	72.20	70.58	69.62
PillarNet-18 [32]	2	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 [32]	2	79.98	79.47	72.00	71.53	82.52	79.33	75.00	71.95	70.51	69.69	68.38	67.58
PV-RCNN++* [33]	2	80.17	79.70	72.14	71.70	83.48	80.42	75.54	72.61	74.63	73.75	72.35	71.50
RSN* [38]	3	78.4	78.1	69.5	69.1	79.4	76.2	69.9	67.0	-	-	-	-
SST-TS* [11]	3	78.66	78.21	69.98	69.57	83.81	80.14	75.94	72.37				
SWFormer [37]	3	79.4	78.9	71.1	70.6	82.9	79.0	74.8	71.1	-	-	-	-
PillarNeXt-B	3	**80.58**	**80.08**	**72.89**	**72.42**	**85.04**	**82.11**	**78.04**	**75.19**	**78.92**	**77.93**	**76.71**	**75.74**
CenterFormer [48]	8	78.8	78.3	74.3	73.8	82.1	79.3	77.8	75.0	75.2	74.4	73.2	72.3
MPPNet [7]	16	82.74	82.28	75.41	74.96	84.69	82.25	77.43	75.06	77.28	76.66	75.13	74.52
3DAL† [30]	ALL	84.50	-	-	-	82.88	-	-	-	-	-	-	-

Table 4. Comparison of PillarNeXt-B and the state-of-the-art methods under the 3D metrics on the validation set of WOD. * denotes the two-stage methods and † indicates the offboard approach.

Method	Frames	Vehicle L1		Vehicle L2		Pedestrian L1		Pedestrian L2		Cyclist L1		Cyclist L2	
		AP	APH	AP	APH	AP	APH	AP	APH	AP	APH	AP	APH
PV-RCNN++* [33]	1	91.57	-	-	-	85.43	-	-	-	75.94	-	-	-
PillarNeXt-B	1	93.30	92.60	87.26	86.53	88.19	82.13	81.77	75.82	75.67	74.61	72.97	71.95
SWFormer [37]	3	92.60	-	-	-	87.50	-	-	-	-	-	-	-
PillarNeXt-B	3	**94.41**	93.74	89.38	88.68	**90.27**	87.01	84.73	81.42	80.36	79.34	78.15	77.15
3DAL† [30]	ALL	93.30	-	-	-	86.32	-	-	-	-	-	-	-

Table 5. Comparison of PillarNeXt-B and the state-of-the-art methods under the BEV metrics on the validation set of WOD. * denotes the two-stage methods and † indicates the offboard approach.

jects like vehicles, but deteriorates the accuracy of small objects such as pedestrians. Downsampling the output feature resolution (0.3 to 0.6) impairs the performance of both categories. However, if simply providing an upsampling layer in the detection head, we obtain significant improvement, especially for small objects. This suggests that the fine-grained information may have already been encoded in the downsampled feature maps, and a simple upsampling layer in head can effectively recover the details.

4.2.4 Summary

We provide the improvement of each component one by one to elucidate the boosting roadmap in Figure 4. As compared in this figure, one can see that the model scaling (e.g., tiny to base), the enhanced network neck and head (e.g., ASPP based neck and simple modifications in head), and the appropriate training (e.g., sufficient training epochs and data augmentation), produce tremendous improvements over the original baseline model. In the following experiments, we exploit Pillar-B with ASPP as the default setting of our proposed network PillarNeXt, which is extensively compared with the state-of-the-art methods that are specifically developed for point clouds. We illustrate the overall architecture of PillarNeXt in Figure 2.

4.3. Comparison with State-of-the-Art on WOD

We compare PillarNeXt-B with the published results on the validation set of WOD. As a common practice, we list the methods of using single and multiple frames separately. For completeness, we also compare to the methods with long-term temporal modeling. Our model is trained for 36 epochs with the faded copy-and-paste data augmentation.

As compared in Table 4, our single-stage model outperforms many two-stage methods. It is also worth noting that our pillar based approach without explicit temporal modeling even achieves better results for small objects such as pedestrians than the methods with fine-grained geometrical modeling and complex temporal modeling. This clearly verifies the importance of network designs in terms of basic architecture and appropriate training.

In addition to 3D results, we also report BEV metrics in Table 5. BEV representation is widely used in autonomous

Method	Frames	All L2		Vehicle L1		Vehicle L2		Pedestrian L1		Pedestrian L2		Cyclist L1		Cyclist L2	
		mAP	mAPH	AP	APH	AP	APH	AP	APH	AP	APH	AP	APH	AP	APH
SWFormer [37]	3	-	-	82.89	82.49	75.02	74.65	82.13	78.13	75.87	72.07	-	-	-	-
PillarNet-34† [32]	3	73.98	72.48	83.23	82.80	76.09	75.69	82.38	79.02	76.66	73.46	71.44	70.51	69.20	68.29
CenterPoint++ [45]	3	74.20	72.80	82.80	82.30	75.50	75.10	81.00	78.20	75.10	72.40	74.40	73.30	72.00	71.00
AFDetV2 [13]	2	74.60	73.12	81.65	81.22	74.30	73.89	81.26	78.05	75.47	72.41	76.41	75.37	74.05	73.04
PV-RCNN++* [33]	2	75.00	73.52	83.74	83.32	76.31	75.92	82.60	79.38	76.63	73.55	74.44	73.43	72.06	71.09
PillarNeXt-B	3	**75.53**	**74.10**	83.28	82.38	76.18	75.76	84.40	81.44	78.84	75.98	73.77	72.73	71.56	70.55

Table 6. Comparison of PillarNeXt-B and the state-of-the-art methods under the 3D metrics on the test set of WOD. * denotes the two-stage method and † indicates using test-time augmentations.

Method	Encoder	Grid Size	NDS	mAP	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
CenterPoint [45]	V	0.075	66.8	59.6	0.292	0.255	0.302	0.259	0.193
OHS [5]	V	0.1	66.0	59.5	-	-	-	-	-
PillarNet-18 [32]	P	0.075	67.4	59.9	-	-	-	-	-
Transfusion-L [1]	V	0.075	66.8	60.0	-	-	-	-	-
UVTR-L [15]	V	0.075	67.7	60.9	0.334	0.257	0.300	0.204	0.182
VISTA [8]	V+R	0.1	68.1	60.8	-	-	-	-	-
PillarNeXt-B	P	0.075	**68.4**	62.2	0.286	0.256	0.285	0.251	0.192
Our Voxel-B	V	0.075	67.8	**62.3**	0.299	0.254	0.316	0.271	0.195

Table 7. Comparison of PillarNeXt-B and the state-of-the-art methods on the validation set of nuScenes. P/V/R denotes the pillar, voxel and range view based grid encoder, respectively. Most leading methods adopt the voxel based representations.

Method	Car	Truck	Bus	Trailer	CV	Ped	Motor	Bicycle	TC	Barrier	mAP
PillarNeXt-B	85.0	57.4	67.6	35.6	20.6	86.8	68.6	53.1	77.3	69.7	62.2
Our Voxel-B	84.8	58.0	68.3	37.1	21.8	86.1	68.4	56.5	74.2	68.2	62.3

Table 8. Comparison of our proposed pillar and voxel based models under per-class AP and mAP on the validation set of nuScenes. Abbreviations are construction vehicle (CV), pedestrian (Ped), motorcycle (Motor), and traffic cone (TC).

driving as the downstream tasks are naturally carried out in the space of BEV. Interestingly, our single-frame model already outperforms many multi-frame methods. As compared with the offboard method 3DAL [30], which takes the whole sequence (around 200 frames) for refinement, our 3-frame model achieves better performance. This again validates the efficacy of our succinct single-stage network.

In Table 6, we further demonstrate our results on the test set of WOD to evaluate the generalization of our approach. We do not use any test-time augmentation or model ensembling. PillarNeXt-B is also found to outperform the state-of-the-art methods without bells and whistles.

4.4. Comparison with State-of-the-Art on nuScenes

We in the end compare PillarNeXt-B with the state-of-the-art methods on nuScenes. Our model is trained for 20 epochs with the commonly used re-sampling CBGS [49] and the faded copy-and-paste data augmentation. We report the results on the validation set in Table 7. Here we use a simpler model by removing the upsampling layer and IoU score branch in the detection head. Our approach achieves the superior performance of 68.4% NDS, which shows the exceptional generalizability of the model across different datasets. It is noteworthy that apart from PillarNet-18, all high-performing methods are voxel based. Our pillar based model outperforms the leading voxel and multi-view based methods by a large margin in mAP. For deeper analysis, we also compare with our voxel based model (Voxel-B) under exactly the same setting. PillarNeXt-B achieves a higher NDS and almost the same mAP. In particular, for the per-class performance shown in Table 8, PillarNeXt-B outperforms Voxel-B in pedestrians and traffic cones. This further verifies that pillar model can be highly effective in accurately detecting small objects.

5. Conclusions

In this paper, we challenge the common belief that a high-performing 3D object detection model requires the fine-grained local geometrical modeling. We systematically study three local point aggregators, and find that the simplest pillar encoder with enhanced strategy performs the best in both accuracy and latency. We also show that enlarging receptive field and operating resolutions play the key roles. We hope our findings can serve as a solid baseline and encourage the research community to rethink what should be focused on for LiDAR based 3D object detection.

References

[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1090–1099, 2022. 8

[2] Eli Bronstein, Mark Palatucci, Dominik Notz, Brandyn White, Alex Kuefler, Yiren Lu, Supratik Paul, Payam Nikdel, Paul Mougin, Hongge Chen, Justin Fu, Austin Abrams, Punit Shah, Evan Racah, Benjamin Frenkel, Shimon Whiteson, and Dragomir Anguelov. Hierarchical model-based imitation learning for planning in autonomous driving. In *IROS*, 2022. 1

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 4

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. 2, 4, 5, 6

[5] Qi Chen, Lin Sun, Zhixin Wang, Kui Jia, and Alan Yuille. Object as hotspots: An anchor-free 3d object detection approach via firing of hotspots. In *European conference on computer vision*, pages 68–84. Springer, 2020. 8

[6] Qiang Chen, Yingming Wang, Tong Yang, Xiangyu Zhang, Jian Cheng, and Jian Sun. You only look one-level feature. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13039–13048, 2021. 1, 2, 5, 6

[7] Xuesong Chen, Shaoshuai Shi, Benjin Zhu, Ka Chun Cheung, Hang Xu, and Hongsheng Li. Mppnet: Multi-frame feature intertwining with proxy points for 3d temporal object detection. *arXiv preprint arXiv:2205.05979*, 2022. 7

[8] Shengheng Deng, Zhihao Liang, Lin Sun, and Kui Jia. Vista: Boosting 3d object detection via dual cross-view spatial attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8448–8457, 2022. 8

[9] Piotr Dollár, Mannat Singh, and Ross Girshick. Fast and accurate model scaling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 924–932, 2021. 3

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[11] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing single stride 3d object detector with sparse transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8458–8468, 2022. 7

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[13] Yihan Hu, Zhuangzhuang Ding, Runzhou Ge, Wenxin Shao, Li Huang, Kun Li, and Qiang Liu. Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 969–979, 2022. 5, 7, 8

[14] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 1, 2, 3, 4

[15] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3d object detection. *arXiv preprint arXiv:2206.00630*, 2022. 8

[16] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2

[17] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1, 2, 5, 6

[18] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018. 2

[19] Ze Liu, Han Hu, Yue Cao, Zheng Zhang, and Xin Tong. A closer look at local aggregation operators in point cloud analysis. In *European Conference on Computer Vision*, pages 326–342. Springer, 2020. 2

[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1

[21] Zhijian Liu, Haotian Tang, Shengyu Zhao, Kevin Shao, and Song Han. Pvnas: 3d neural architecture search with point-voxel convolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8552–8568, 2021. 2

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 4

[23] Yiren Lu, Justin Fu, George Tucker, Xinlei Pan, Eli Bronstein, Becca Roelofs, Benjamin Sapp, Brandyn White, Aleksandra Faust, Shimon Whiteson, Dragomir Anguelov, and Sergey Levine. Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios. In *NeurIPS*, 2022. 1

[24] Chenxu Luo, Xiaodong Yang, and Alan Yuille. Exploring simple 3d multi-object tracking for autonomous driving. In *ICCV*, 2021. 1

[25] Chenxu Luo, Xiaodong Yang, and Alan Yuille. Self-supervised pillar motion learning for autonomous driving. In *CVPR*, 2021. 1

[26] Gregory P Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K Wellington. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12677–12686, 2019. 1, 2

[27] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple and efficient attention networks. *arXiv:2207.05844*, 2022. 1

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 4

[29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[30] Charles R Qi, Yin Zhou, Mahyar Najibi, Pei Sun, Khoa Vo, Boyang Deng, and Dragomir Anguelov. Offboard 3d object detection from point cloud sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6134–6144, 2021. 2, 7, 8

[31] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *arXiv preprint arXiv:2206.04670*, 2022. 2

[32] Guangsheng Shi, Ruifeng Li, and Chao Ma. Pillarnet: High-performance pillar-based 3d object detection. *arXiv preprint arXiv:2205.07403*, 2022. 1, 2, 3, 5, 6, 7, 8

[33] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *arXiv preprint arXiv:2102.00463*, 2021. 1, 2, 7, 8

[34] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019. 2

[35] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019. 4

[36] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 4

[37] Pei Sun, Mingxing Tan, Weiyue Wang, Chenxi Liu, Fei Xia, Zhaoqi Leng, and Dragomir Anguelov. Swformer: Sparse window transformer for 3d object detection in point clouds. *arXiv preprint arXiv:2210.07372*, 2022. 7, 8

[38] Pei Sun, Weiyue Wang, Yuning Chai, Gamaleldin Elsayed, Alex Bewley, Xiao Zhang, Cristian Sminchisescu, and Dragomir Anguelov. Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5725–5734, 2021. 7

[39] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 3

[40] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 1, 2, 3, 4, 5, 6

[41] Xiaofang Wang and Kris M Kitani. Cost-aware comparison of lidar-based 3d object detectors. *arXiv preprint arXiv:2205.01142*, 2022. 3

[42] Xishun Wang, Tong Su, Fang Da, and Xiaodong Yang. Prophnet: Efficient agent-centric motion forecasting with anchor-informed proposals. In *CVPR*, 2023. 1

[43] Yue Wang, Alireza Fathi, Abhijit Kundu, David A Ross, Caroline Pantofaru, Tom Funkhouser, and Justin Solomon. Pillar-based object detection for autonomous driving. In *European Conference on Computer Vision*, pages 18–34. Springer, 2020. 1, 2, 3

[44] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1, 2, 3, 4

[45] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021. 3, 4, 8

[46] Yin Zhou, Pei Sun, Yu Zhang, Dragomir Anguelov, Jiyang Gao, Tom Ouyang, James Guo, Jiquan Ngiam, and Vijay Vasudevan. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In *Conference on Robot Learning*, pages 923–932. PMLR, 2020. 2

[47] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018. 1, 2, 3, 6

[48] Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. Centerformer: Center-based transformer for 3d object detection. In *European Conference on Computer Vision*, pages 496–513. Springer, 2022. 7

[49] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019. 3, 8