Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR Segmentation

Xinge Zhu[1*] Hui Zhou[1*] Tai Wang[1] Fangzhou Hong[7]
Yuexin Ma[2 5] Wei Li[3] Hongsheng Li[1 6] Dahua Lin[1 4]

[1]CUHK-SenseTime Joint Lab, CUHK [2]ShanghaiTech University
[3]Inceptio [4]Centre of Perceptual and Interactive Intelligence
[5]Shanghai Engineering Research Center of Intelligent Vision and Imaging
[6]School of CST, Xidian University [7]Nanyang Technological University

Abstract

State-of-the-art methods for large-scale driving-scene LiDAR segmentation often project the point clouds to 2D space and then process them via 2D convolution. Although this corporation shows the competitiveness in the point cloud, it inevitably alters and abandons the 3D topology and geometric relations. A natural remedy is to utilize the 3D voxelization and 3D convolution network. However, we found that in the outdoor point cloud, the improvement obtained in this way is quite limited. An important reason is the property of the outdoor point cloud, namely sparsity and varying density. Motivated by this investigation, we propose a new framework for the outdoor LiDAR segmentation, where cylindrical partition and asymmetrical 3D convolution networks are designed to explore the 3D geometric pattern while maintaining these inherent properties. Moreover, a point-wise refinement module is introduced to alleviate the interference of lossy voxel-based label encoding. We evaluate the proposed model on two large-scale datasets, i.e., *SemanticKITTI and nuScenes. Our method achieves the **1st place** in the leaderboard of SemanticKITTI and outperforms existing methods on nuScenes with a noticeable margin, about 4%. Furthermore, the proposed 3D framework also generalizes well to LiDAR panoptic segmentation and LiDAR 3D detection.*

1. Introduction

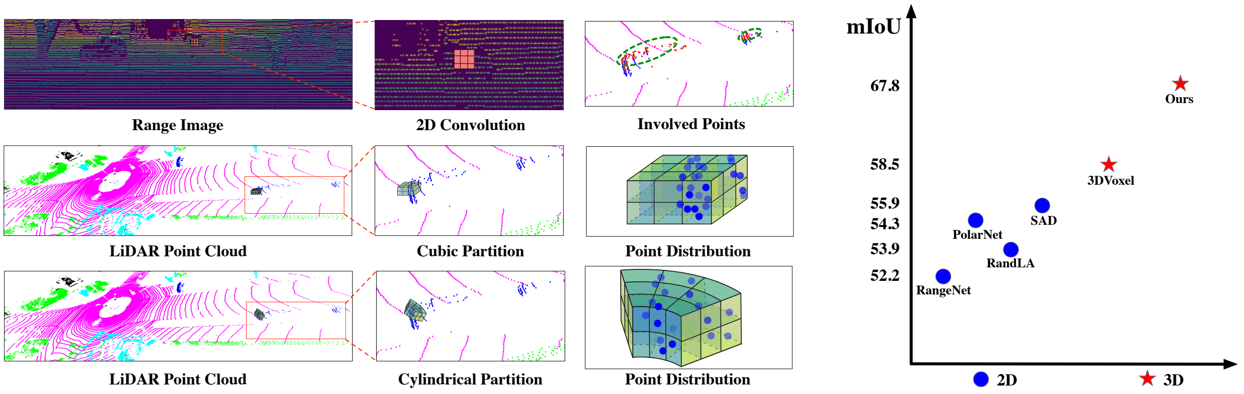
3D LiDAR sensor has become an indispensable device in modern autonomous driving vehicles. It captures more precise and farther-away distance measurements of the surrounding environments than conventional visual cameras. The measurements of the sensor naturally form 3D point clouds that can be used to realize a thorough scene understanding for autonomous driving planning and execution, in which LiDAR segmentation is crucial for driving-scene understanding. It aims to identify the pre-defined categories of each 3D point, such as car, truck *etc*., which provides point-wise perception information of the overall 3D scene.

Recently, the advances in deep learning have significantly pushed forward the state of the art in image segmentation. Some existing LiDAR segmentation approaches follow this route to project the 3D point clouds onto a 2D space and process them via 2D convolution networks, including range image based [23, 37] and bird's-eye-view image based [46]. However, this group of methods lose and alter the accurate 3D geometric information during the 3D-to-2D projection (as shown in the top row of Fig. 1a).

A natural alternative is to utilize the 3D partition and 3D convolution networks to process the point cloud and maintain their 3D geometric relations. However, in our initial attempts, we directly apply the 3D voxelization [12, 7] and 3D convolution networks to outdoor LiDAR point cloud, only to find very limited performance gain (as shown in Fig. 1b). Our investigation into this issue reveals a key difficulty of outdoor LiDAR point cloud, namely sparsity and varying density, which is also the key difference to indoor scenes with dense and uniform-density points. However, previous 3D voxelization methods consider the point cloud as a uniform one and split them via the uniform cube, while neglecting the varying-density property of outdoor point cloud. Consequently, this effect to apply the 3D partition to outdoor point cloud is met with fundamental difficulty.

Motivated by these findings, we propose a new framework to outdoor LiDAR segmentation that consists of two key components, *i.e.*, 3D cylindrical partition and asymmetrical 3D convolution networks, which maintain the 3D ge-

[*] denotes the equal contribution and code at https://github.com/xinge008/Cylinder3D. Corresponding email: zhuxinge123@gmail.com



(a) (b)

Figure 1: (a) Range Image (2D projection) v.s. Cubic Partition v.s. Cylindrical Partition. From top row, it can be found that range image abandons the 3D topology, where 2d convolution processes points in different locations (far away from each other in green circles). From bottom part, cylindrical partition generates the more balanced point distribution than cubic partition (89% v.s. 61% cells containing points). (b) Applying the regular 3D voxel partition and 3D convolution directly (*i.e.*, 3DVoxel) gets limited performance gain compared to projection-based (2D) methods [46, 14, 23, 40], while our method achieves a remarkable performance gain by further tackling the inherent difficulty of outdoor LiDAR point clouds (showing results on SemanticKITTI dataset).

ometric information and handle these issues from partition and networks, respectively. Here, cylindrical partition resorts to the cylinder coordinates to divide the point cloud dynamically according to the distance (Regions that are far away from the origin have much sparse points, thus requiring a larger cell), which produces a more balanced point distribution (as shown in Fig. 1a); while asymmetrical 3D convolution networks strengthen the horizontal and vertical kernels to match the point distribution of objects in the driving scene and enhance the robustness to the sparsity. Moreover, voxel based methods might divide the points with different categories into the same cell and cell label encoding would inevitably cause the information loss. To alleviate the interference of lossy label encoding, a point-wise module is introduced to further refine the features obtained from voxel-based network. Overall, the corporation of these components well maintains the geometric relation and tackle the difficulty of outdoor point cloud, thus improving the effectiveness of 3D frameworks.

We evaluate the proposed method on two large-scale outdoor datasets, including SemanticKITTI [2] and nuScenes [5]. Our method achieves the **1st place** in the leaderboard of SemanticKITTI and also outperforms the existing methods on nuScenes with a large margin. We also extend the proposed cylindrical partition and asymmetrical 3D convolution networks to LiDAR panoptic segmentation and LiDAR 3D detection. Experimental results show its strong generalization capability and scalability.

The contributions of this work mainly lie in three aspects: (1) We reposition the focus of outdoor LiDAR segmentation from 2D projection to 3D structure, and further investigate the inherent properties (difficulties) of outdoor point cloud. (2) We introduce a new framework to explore the 3D geometric pattern and tackle these difficulties caused by sparsity and varying density, through cylindrical partition and asymmetrical 3D convolution networks. (3) The proposed method achieves the state of art on SemanticKITTI and nuScenes, and also shows strong generalization capability on LiDAR panoptic segmentation and LiDAR 3D detection.

2. Related Work

Indoor-scene Point Cloud Segmentation. Indoor-scene point clouds carry out some properties, including generally uniform density, small number of points, and small range of the scene. Mainstream methods [25, 31, 39, 36, 32, 20, 10, 45, 41, 33, 24, 26] of indoor point cloud segmentation learn the point features based on the raw point directly, which are often based on the pioneering work, *i.e.*, PointNet, and promote the effectiveness of sampling, grouping and ordering to achieve the better performance. Another group of methods utilize the clustering algorithm [36, 32] to extract the hierarchical point features. However, these methods focusing on indoor point cloud are limited to adapt to the outdoor point cloud under the property of varying density and large range of scenes, and the large number of points also result in the computational difficulties for these methods when deploying from indoor to outdoor.

Outdoor-scene Point Cloud Segmentation. Most existing outdoor-scene point cloud segmentation methods [14, 8, 23, 1, 43, 18, 47] focus on converting the 3D point cloud to 2D grids, to enable the usage of 2D Convolutional Neural Networks. SqueezeSeg [37], Darknet [2], Squeeze-

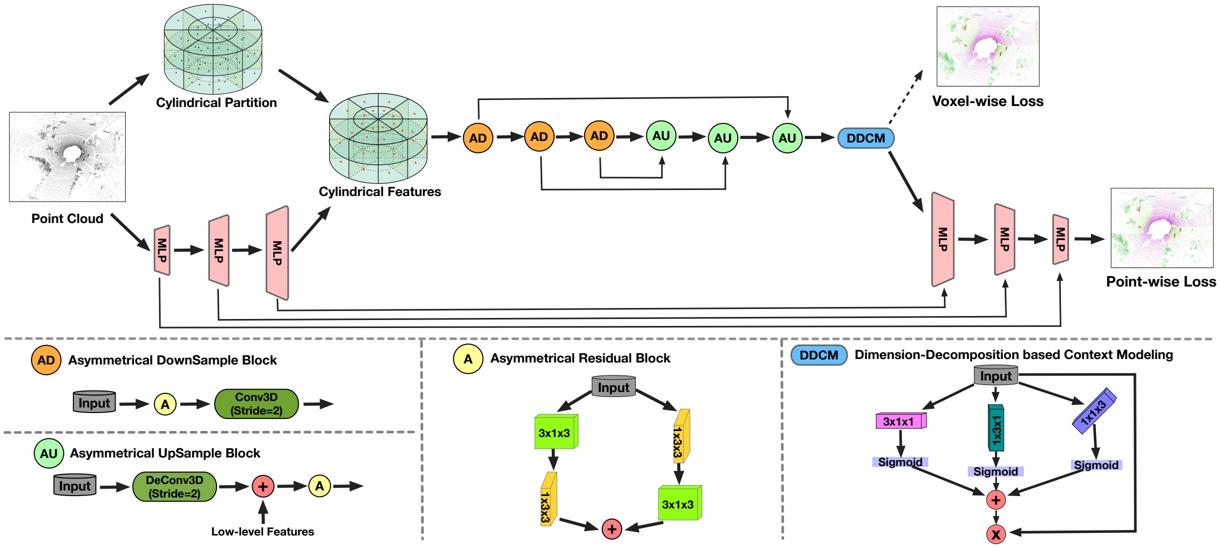


Figure 2: (1): Top half is the overall framework. Here, LiDAR point cloud is fed into MLP to get the point-wise features and then these features are reassigned based on the cylinderical partition. Asymmetrical 3D convolution networks are then used to generate the voxel-wise outputs. Finally, a point-wise module is introduced to refine these outputs. (2): Bottom half elaborates four components, including Asymmetrical Downsample block (AD), Asymmetrical Upsample blcok (AU), Asymmetrical residual block (A) and Dimension-Decomposition based Context Modeling (DDCM).

Segv2 [38], and RangeNet++ [23] utilize the spherical projection mechanism, which converts the point cloud to a frontal-view image or a range image, and adopt the 2D convolution network on the pseudo image for point cloud segmentation. PolarNet [46] follows the bird's-eye-view projection, which projects point cloud data into bird's-eye-view representation under the polar coordinates. However, these 3D-to-2D projection methods inevitably loss and alter the 3D topology and fails to model the geometric information.

3D Voxel Partition. 3D voxel partition is another routine of point cloud encoding [13, 30, 12, 7, 21]. It converts a point cloud into 3D voxels, which mainly retains the 3D geometric information. OccuSeg [13], SSCN [12] and SEG-Cloud [30] follow this line to utilize the voxel partition and apply regular 3D convolutions for LiDAR segmentation. It is worth noting that while the aforementioned efforts have shown encouraging results, the improvement in the outdoor LiDAR point cloud remains limited. As mentioned above, a common issue is that these methods neglect the inherent properties of outdoor LiDAR point cloud, namely, sparsity and varying density. Compared to these methods, our method resorts to the 3D cylindrical partition and asymmetrical 3D convolution networks to tackle these difficulties.

3. Methodology

3.1. Framework Overview

In the context of semantic segmentation, given a point cloud, our task is to assign the semantic label to each point. Based on the comparison between 2D and 3D representation and investigation of the inherent properties of outdoor LiDAR point cloud, we desire to obtain a framework which explores the 3D geometric information and handles the difficulty caused by sparsity and varying-density. To this end, we propose a new outdoor segmentation approach based on the 3D partition and 3D convolution networks. To handle these difficulties of outdoor LiDAR point cloud, namely sparsity and varying density, we first employ the cylindrical partition to generate the more balanced point distribution (more robust to varying density), then apply the asymmetrical 3D convolution networks to power the horizontal and vertical weights, thus well matching the object point distribution and enhancing the robustness to the sparsity.

As shown in the top half of Fig. 2, the framework consists of two major components, including cylindrical partition and asymmetrical 3D convolution networks. The LiDAR point cloud is first divided by the cylindrical partition and the features extracted from MLP is then reassigned based on this partition. Asymmetrical 3D convolution networks are then used to generate the voxel-wise outputs. Finally, a point-wise module is introduced to alleviate the interference of lossy cell-label encoding, thus refining the outputs. In the following sections, we will present these components in detail.

3.2. Cylindrical Partition

As mentioned above, outdoor-scene LiDAR point cloud possesses the property of varying density, where nearby

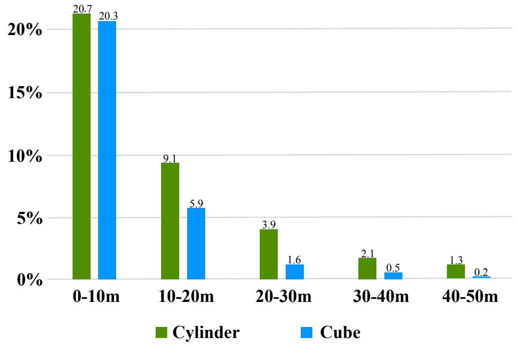


Figure 3: The proportion of non-empty cells at different distances between cylindrical and cubic partition (The results are calculated on the training set of SemanticKITTI). It can be found that cylinder partition makes a higher non-empty proportion and more balanced point distribution, especially for farther-away regions.

region has much greater density than farther-away region. Therefore, uniform cells splitting the varying-density points would fall into an imbalanced distribution (for example, larger proportion of empty cells). While in the cylinder coordinate system, it utilizes the increasing grid size to cover the farther-away region, and thus more evenly distributes the points across different regions and gives an more balanced representation against the varying density. We perform a statistic to show the proportion of non-empty cells across different distances in Fig. 3. It can be found that with the distance goes far, cylindrical partition maintains a balanced non-empty proportion due to the increasing grid size while cubic partition suffers the imbalanced distribution, especially in the farther-away regions (about 6 times less than cylindrical partition). Moreover, unlike these projection-based methods project the point to the 2D view, cylindrical partition maintains the 3D grid representation to retain the geometric structure.

The workflow is illustrated in Fig. 4. We first transform the points on Cartesian coordinate system to the Cylinder coordinate system. This step transforms the points (x, y, z) to points (ρ, θ, z), where radius ρ (distance to origin in x-y axis) and azimuth θ (angle from x-axis to y-axis) are calculated. Then cylindrical partition performs the split on these three dimensions, note that in the cylinder coordinate, more farther-away region, larger cell. Point-wise features obtained from the MLP are reassigned based on the result of this partition to get the cylindrical features. After these steps, we unroll the cylinder from 0-degree and get the 3D cylindrical representation $\mathbb{R} \in C \times H \times W \times L$, where C denotes the feature dimension and H, W, L mean the radius, azimuth and height. Subsequent asymmetrical 3D convolution networks will be performing on this representation.

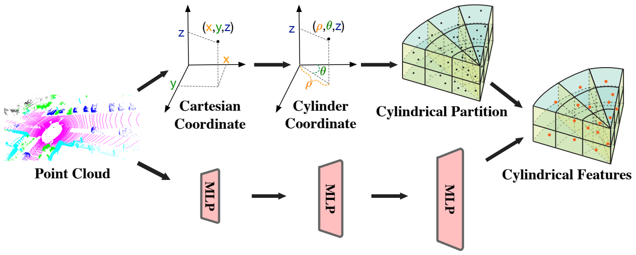


Figure 4: The pipeline of cylindrical partition. We first transform the Cartesian coordinate to Cylinder coordinate and then assign the point-wise features to the structured representation based on the cylindrical partition.

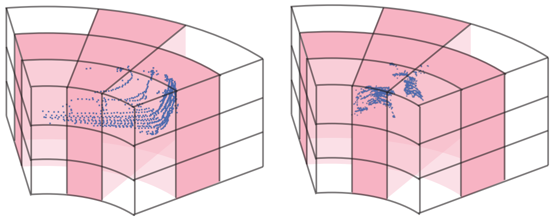

(a) Car (b) Motorcycle

Figure 5: An illustration of asymmetrical residual block, where two asymmetrical kernels are stacked to power the skeleton. It can be observed that asymmetrical residual block focuses on the horizontal and vertical kernels.

3.3. Asymmetrical 3D Convolution Network

Since the driving-scene point cloud carries out the specific object shape distribution, including car, truck, bus, motorcycle and other cubic objects, we aim to follow this observation to enhance the representational power of a standard 3D convolution. Moreover, recent literature [35, 9] also shows that the central crisscross weights count more in the square convolution kernel. In this way, we devise the asymmetrical residual block to strengthen the horizontal and vertical responses and match the object point distribution. Based on the proposed asymmetrical residual block, we further build the asymmetrical downsample block and asymmetrical upsample block to perform the downsample and upsample operation. Moreover, a dimension-decomposition based context modeling (termed as DDCM) is introduced to explore the high-rank global context in decomposite-aggregate strategy. We detail these components in the bottom of Fig. 2

Asymmetrical Residual Block Motivated by the observation and conclusion in [35, 9], the asymmetrical residual block strengthens the horizontal and vertical kernels, which matches the point distribution of object in the driving scene and explicitly makes the skeleton of the kernel powerful, thus enhancing the robustness to the sparsity of outdoor LiDAR point cloud. We use the Car and Motorcycle as the example to show the asymmetrical residual block in Fig. 5, where 3D convolutions are performing on the cylindrical grids. Moreover, the proposed asymmetrical residual block also saves the computation and memory cost compared to the regular square-kernel 3D convolution block. By incorporating the asymmetrical residual block, the asymmetrical downsample block and upsample block are designed and our asymmetrical 3D convolution networks are built via stacking these downsample and upsample blocks.

Dimension-Decomposition based Context Modeling Since the global context features should be high-rank to have enough capacity to capture the large context varieties [44], it is hard to construct these features directly. We follow the tensor decomposition theory [6] to build the high-rank context as a combination of low-rank tensors, where we use three rank-1 kernels to obtain the low-rank features and then aggregate them together to get the final global context.

3.4. Point-wise Refinement Module

Partition-based methods predict one label for each cell. Although partition-based methods effectively explore the large-range point cloud, however, this group of method, including cube-based and cylinder-based, inevitably suffers from the lossy cell-label encoding, *e.g.*, points with different categories are divided into same cell, and this case would cause the information loss. We make a statistic to show the effect of different label encoding methods in Fig. 6, where majority encoding means using the major category of points inside a cell as the cell label and minority encoding indicates using the minor category as the cell label. It can be observed that both of them cannot reach the 100 percent mIoU (ideal encoding) and inevitably have the information loss. Here, the point-wise refinement module is introduced to alleviate the interference of lossy cell-label encoding. We first project the voxel-wise features to the point-wise based on the point-voxel mapping table. Then the point-wise module takes both point features before and after 3D convolution networks as the input, and fuses them together to refine the output.

3.5. Objective Function

The total objective of our method consists of two components, including voxel-wise loss and point-wise loss. It can be formulated as $L = L_{voxel} + L_{point}$. For the voxel-wise loss (L_{voxel}), we follow the existing methods [8, 14] and use the weighted cross-entropy loss and lovasz-softmax [4] loss to maximize the point accuracy and the intersection-over-union score, respectively. For point-wise loss (L_{point}), we only use the weighted cross-entropy loss to supervise the training. During inference, the output from point-wise refinement module is used as the final output. For the optimizer, Adam with an initial learning rate of $1e^{-3}$ is employed.

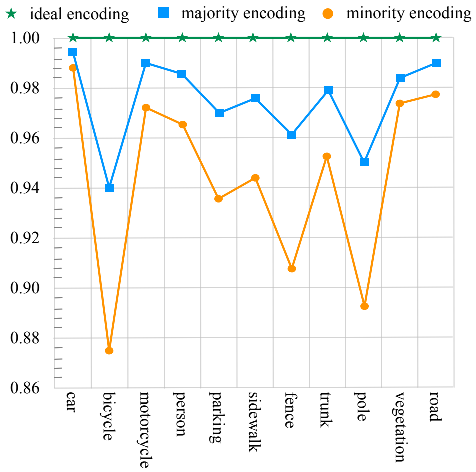


Figure 6: Upper bound of mIoU with different label encoding methods (*i.e.*, majority and minority encoding). It can be found that no matter what encoding methods are, the information loss always occurs, which is also the reason for point-wise refinement.

4. Experiments

In this section, we first provide the detailed experimental setup, then evaluate the proposed method on two large-scale datasets, *i.e.*, SemanticKITTI and nuScenes. Furthermore, extensive ablation studies are conducted to validate each component. In the end, we extend our method to LiDAR panoptic segmentation and 3D detection to verify its scalability and generalization ability.

4.1. Dataset and Metric

SemanticKITTI [2] is a large-scale driving-scene dataset for point cloud segmentation, including semantic segmentation and panoptic segmentation. It is derived from the KITTI Vision Odometry Benchmark and collected in Germany with the Velodyne-HDLE64 LiDAR. The dataset consists of 22 sequences, splitting sequences 00 to 10 as training set (where sequence 08 is used as the validation set), and sequences 11 to 21 as test set. 19 classes are remained for training and evaluation after merging classes with different moving status and ignore classes with very few points.

nuScenes [5] It collects 1000 scenes of 20s duration with 32 beams LiDAR sensor. The number of total frames is 40,000, which is sampled at 20Hz. They also officially split the data into training and validation set. After merging similar classes and removing rare classes, total 16 classes for

Table 1: Results of our proposed method and state-of-the-art LiDAR Segmentation methods on SemanticKITTI test set. Note that all results are obtained from the literature, where post-processing, flip & rotation test ensemble, *etc*. are also applied.

Methods	**mIoU**	Latency(ms)	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic
TangentConv [29]	35.9	3000	86.8	1.3	12.7	11.6	10.2	17.1	20.2	0.5	82.9	15.2	61.7	9.0	82.8	44.2	75.5	42.5	55.5	30.2	22.2
Darknet53 [2]	49.9	-	86.4	24.5	32.7	25.5	22.6	36.2	33.6	4.7	91.8	64.8	74.6	27.9	84.1	55.0	78.3	50.1	64.0	38.9	52.2
RandLA-Net [14]	50.3	800	94.0	19.8	21.4	42.7	38.7	47.5	48.8	4.6	90.4	56.9	67.9	15.5	81.1	49.7	78.3	60.3	59.0	44.2	38.1
RangeNet++ [23]	52.2	-	91.4	25.7	34.4	25.7	23.0	38.3	38.8	4.8	91.8	65.0	75.2	27.8	87.4	58.6	80.5	55.1	64.6	47.9	55.9
PolarNet [46]	54.3	-	93.8	40.3	30.1	22.9	28.5	43.2	40.2	5.6	90.8	61.7	74.4	21.7	90.0	61.3	84.0	65.5	67.8	51.8	57.5
SqueezeSegv3 [40]	55.9	-	92.5	38.7	36.5	29.6	33.0	45.6	46.2	20.1	91.7	63.4	74.8	26.4	89.0	59.4	82.0	58.7	65.4	49.6	58.9
Salsanext [8]	59.5	-	91.9	48.3	38.6	38.9	31.9	60.2	59.0	19.4	91.7	63.7	75.8	29.1	90.2	64.2	81.8	63.6	66.5	54.3	62.1
KPConv [31]	58.8	263	96.0	32.0	42.5	33.4	44.3	61.5	61.6	11.8	88.8	61.3	72.7	31.6	**95.0**	64.2	84.8	69.2	69.1	56.4	47.4
FusionNet [43]	61.3	-	95.3	47.5	37.7	41.8	34.5	59.5	56.8	11.9	91.8	68.8	77.1	30.8	92.5	**69.4**	84.5	69.8	68.5	60.4	**66.5**
KPRNet [17]	63.1	-	95.5	54.1	47.9	23.6	42.6	65.9	65.0	16.5	**93.2**	**73.9**	**80.6**	30.2	91.7	68.4	**85.7**	69.8	71.2	58.7	64.1
TORANDONet [11]	63.1	-	94.2	55.7	48.1	40.0	38.2	63.6	60.1	34.9	89.7	66.3	74.5	28.7	91.3	65.6	85.6	67.0	**71.5**	58.0	65.9
SPVNAS [28]	66.4	259	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ours	**67.8**	170	**97.1**	**67.6**	**64.0**	**59.0**	**58.6**	**73.9**	**67.9**	**36.0**	91.4	65.1	75.5	**32.3**	91.0	66.5	85.4	**71.8**	68.5	**62.6**	65.6

Table 2: Results of our proposed method and other LiDAR Segmentation methods on nuScenes validation set.

Methods	**mIoU**	barrier	bicycle	bus	car	construction	motorcycle	pedestrian	traffic-cone	trailer	truck	driveable	other	sidewalk	terrain	manmade	vegetation
RangeNet++ [23]	65.5	66.0	21.3	77.2	80.9	30.2	66.8	69.6	52.1	54.2	72.3	94.1	66.6	63.5	70.1	83.1	79.8
PolarNet [46]	71.0	74.7	28.2	85.3	90.9	35.1	77.5	71.3	58.8	57.4	76.1	96.5	71.1	74.7	74.0	87.3	85.7
Salsanext [8]	72.2	74.8	34.1	85.9	88.4	42.2	72.4	72.2	63.1	61.3	76.5	96.0	70.8	71.2	71.5	86.7	84.4
Ours	**76.1**	**76.4**	**40.3**	**91.2**	**93.8**	**51.3**	**78.0**	**78.9**	**64.9**	**62.1**	**84.4**	**96.8**	**71.6**	**76.4**	**75.4**	**90.5**	**87.4**

the LiDAR semantic segmentation are remained.

For both two datasets, cylindrical partition splits these point clouds into 3D representation with the size = 480 × 360 × 32, where three dimensions indicate the radius, angle and height, respectively.

Evaluation Metric To evaluate the proposed method, we follow the official guidance to leverage mean intersection-over-union (mIoU) as the evaluation metric defined in [2, 5], which can be formulated as: $IoU_i = \frac{TP_i}{TP_i+FP_i+FN_i}$ where TP_i, FP_i, FN_i represent true positive, false positive, and false negative predictions for class i and the mIoU is the mean value of IoU_i over all classes.

4.2. Results on SemanticKITTI

In this experiment, we compare the results of our proposed method with existing state-of-the-art LiDAR segmentation methods on SemanticKITTI test set. As shown in Table 1, our method outperforms all existing methods in term of mIoU. Compared to the projection-based methods on 2D space, including Darknet53 [2], SqueezeSegv3 [40], RangeNet++ [23] and PolarNet [46], our method achieves 8% ∼ 17% performance gain in term of mIoU due to the modeling of 3D geometric information. Compared to some voxel partition and 3D convolution based methods, including FusionNet [43], TORANDONet [11] (multi-view fusion based method) and SPVNAS [28] (utilizing the neural architecture search for LiDAR segmentation), the proposed method also performs better than these 3D convolution based methods, where the cylindrical partition and asymmetrical 3D convolution networks well handle the difficulty of driving-scene LiDAR point cloud that is neglected by these methods.

4.3. Results on nuScenes

Since nuScenes dataset does not release the evaluation server for LiDAR segmentation, we report the results on the validation set, where RangeNet++ and Salsanext perform the post-processing. As shown in Table 2, our method achieves better performance than existing methods in all categories. Specifically, the proposed method obtains about 4% ∼ 7% performance gain than projection-based methods. Moreover, for these categories with sparse points, such as bicycle and pedestrian, our method significantly outperforms existing approaches, which also demonstrates the effectiveness of the proposed method to tackle the sparsity and varying density.

4.4. Ablation Studies

In this section, we perform the thorough ablation experiments to investigate the effect of different components in

Table 3: Ablation studies for network components on SemanticKITTI validation set. PR denotes the point-wise refinement module.

Baseline	Cylinder	Asym-CNN	DDCM	PR	mIoU
✓					58.1
✓	✓				61.0
✓	✓	✓			63.8
✓	✓	✓	✓		65.2
✓	✓	✓	✓	✓	65.9

our method. We also design several variants of asymmetrical residual block to verify our claim that strengthening the horizontal and vertical kernels power the representation ability for driving-scene point cloud.

Effects of Network Components In this part, we make several variants of our model to validate the contributions of different components. The results on SemanticKITTI validation set are reported in Table 3. Baseline method denotes the framework using 3D voxel partition and 3D convolution networks. It can be observed that cylindrical partition performs much better than cubic-based partition with about 3% mIoU gain and asymmetrical 3D convolution networks also significantly boost the performance about 3% improvement, which demonstrates that both cylindrical partition and asymmetrical 3D convolution networks are crucial in the proposed method. Furthermore, dimension-decomposition based context modeling delivers the effective global context features, which yields an improvement of 1.4%. Point-wise refinement module further pushes forward the performance based on the strong model, about 0.7%.

Variants of Asymmetrical Residual Block To verify the effectiveness of the proposed asymmetrical residual block, we design several variants of asymmetrical residual block to investigate the effect of horizontal and vertical kernel enhancement (as shown in Fig. 7). The first variant is the regular residual block without any asymmetrical structure. The second one is the 1D-asymmetrical residual block, which utilizes the 1D asymmetrical kernels without height and also strengthens the horizontal and vertical kernels in one-dimension.

We conduct the ablation studies on SemanticKITTI validation set. Note that we use the cylindrical partition as the partition method and stack these proposed variants to build the 3D convolution networks. We report the results in Table 4. It can be found that although the 1D-Asymmetrical residual block only powers the horizontal and vertical kernels in one-dimension without considering the height, it still achieves 1.3% gain in term of mIoU, which demonstrates the effectiveness of horizontal and vertical kernel strengthening. After taking the height into the consideration, the proposed asymmetrical residual block further matches the object distribution in driving scene and powers the skeleton of kernels to enhance the robustness to the sparsity. From Table 4, the proposed asymmetrical residual block significantly boosts the performance with about 3% improvements, where large improvement can be observed on some instance categories (about 10% gain), including truck, person and motorcycle, because it matches the point distribution of object and enhances the representational power.

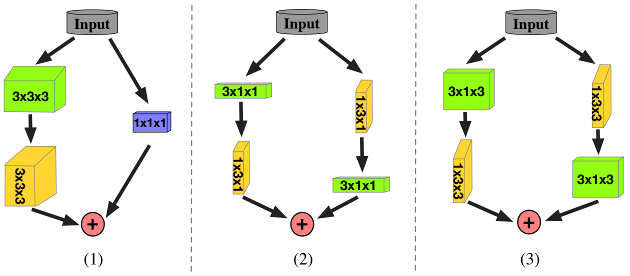


Figure 7: We design three blocks for ablation studies of asymmetrical residual block, including (1) regular residual block , (2) 1D-asymmetrical residual block without height and (3) the proposed asymmetrical residual block.

Table 4: Ablation studies for asymmetrical residual block.

Methods	mIoU
Regular residual block	61.0
1D-Asymmetrical residual block	62.0
Asymmetrical residual block	63.8

4.5. Generalization Analyses

In this section, we extend the proposed cylindrical partition and asymmetrical 3D convolution networks to other LiDAR-based tasks, including LiDAR panoptic segmentation and LiDAR 3D detection. These experimental results demonstrate the generalization ability of our proposed method and its effectiveness on LiDAR point cloud processing. In what follows, we will describe these two tasks and present how we adapt the proposed model in details.

Generalize to LiDAR Panoptic Segmentation Panoptic segmentation is first proposed in [16] as a new task, in which semantic segmentation is performed for background classes and instance segmentation for foreground classes and these two groups of category are also termed as **stuff** and **things** classes, respectively. Behley *et al.* [3] extend the task to LiDAR point clouds and propose the LiDAR panoptic segmentation. In this experiment, we conduct the panoptic segmentation on SemanticKITTI dataset and report results on the validation set. For the evaluation metrics, we follow the metrics defined in [3], where they are the same as that of image panoptic segmentation defined in [16] including Panoptic Quality (PQ), Segmentation Qual-

Table 5: LiDAR panoptic segmentation results on the validation set of SemanticKITTI.

Method	**PQ**	PQ†	RQ	SQ	PQ$^{\mathrm{Th}}$	RQ$^{\mathrm{Th}}$	SQ$^{\mathrm{Th}}$	PQ$^{\mathrm{St}}$	RQ$^{\mathrm{St}}$	SQ$^{\mathrm{St}}$	mIoU
KPConv [31] + PV-RCNN [27]	51.7	57.4	63.1	**78.9**	46.8	56.8	**81.5**	**55.2**	**67.8**	**77.1**	63.1
PointGroup [15]	46.1	54.0	56.6	74.6	47.7	55.9	73.8	45.0	57.1	75.1	55.7
LPASD [22]	36.5	46.1	-	-	-	28.2	-	-	-	-	50.7
Ours	**56.4**	**62**	**67.1**	76.5	**58.8**	**66.8**	75.8	54.8	67.4	**77.1**	**63.5**

Table 6: LiDAR 3D detection results on nuScenes dataset. CyAs denotes the **Cy**lindrical partition and **As**ymmetrical 3D convolution networks.

Methods	mAP	NDS
PointPillar [19]	30.5	45.3
PP + Reconfig [34]	32.5	50.6
SECOND [42]	31.6	46.8
SECOND + CyAs	36.4	51.7
SSN [48]	46.3	56.9
SSN + CyAs	47.7	58.2

ity (SQ) and Recognition Quality (RQ) which are calculated across all classes. PQ† is defined by swapping PQ of each **stuff** class to its IoU and averaging over all classes like PQ does. Since the categories in panoptic segmentation contain two groups, *i.e.*, **stuff** and **things**, these metrics are also performed separately on these two groups, including PQ$^{\mathrm{Th}}$, PQ$^{\mathrm{St}}$, RQ$^{\mathrm{Th}}$, RQ$^{\mathrm{St}}$, SQ$^{\mathrm{Th}}$ and SQ$^{\mathrm{St}}$, where Panoptic Quality (PQ) is usually used as the first criteria.

In this experiment, we use the proposed cylindrical partition as the partition method and asymmetrical 3D convolution networks as the backbone. Moreover, a semantic branch is used to output the semantic labels for stuff categories, and an instance branch is introduced to generate the instance-level features and further extract their instance IDs for things categories through heuristic clustering algorithms (we use mean-shift in the implementation).

We report the results in Table 5. It can be found that our method achieves much better performance than existing methods [22, 15]. In terms of PQ, we have about 4.7% point improvement, and particularly for the thing categories, our method significantly outperforms state-of-the-art in terms of PQ$^{\mathrm{Th}}$ and PQ$^{\mathrm{St}}$ with a large margin of 10% points. Experimental results demonstrate the effectiveness of the proposed method and its good generalization ability.

Generalize to LiDAR 3D Detection LiDAR 3D detection aims to localize and classify the multi-class objects in the point cloud. SECOND [42] first utilizes the 3D voxelization and 3D convolution networks to perform the single-stage 3D detection. In this experiment, we follow SECOND method and replace the regular voxelization and 3D convolution with the proposed cylindrical partition and asymmetrical 3D convolution networks, respectively. Similarly, to verify its scalability, we also extend the proposed modules to SSN [48]. The experiments are conducted on nuScenes dataset and the cylindrical partition also generates the $480 \times 360 \times 32$ representation. For the evaluation metrics, we follow the official metrics defined in nuScenes, *i.e.*, mean average precision (mAP) and nuScenes detection score (NDS).

The results are shown in Table 6. PP + Reconfig [34] is a partition enhancement approach based on PointPillar [19], while our SECOND + CyAs performs better with similar backbone, which indicates the superiority of the cylindrical partition. We then extend the proposed method (*i.e.*, CyAs) to two baseline methods, termed as SECOND + CyAs and SSN + CyAs, respectively. By comparing these two models with their extensions, it can be observed that the proposed Cylindrical partition and Asymmetrical 3D convolution networks boost the performance consistently, even for the strong baseline *i.e.*, SSN, which demonstrates the effectiveness and scalability of our model.

5. Conclusion

In this paper, we have proposed a cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation, where it maintains the 3D geometric relation. Specifically, two key components, the cylinder partition and asymmetrical 3D convolution networks, are designed to handle the inherent difficulties in outdoor LiDAR point cloud, namely sparsity and varying density, effectively and robustly. We conduct the extensive experiments and ablation studies, where the model achieves the 1st place in SemanticKITTI and state-of-the-art in nuScenes, and keeps good generalization ability to other LiDAR based tasks, including LiDAR panoptic segmentation and LiDAR 3D detection.

Acknowledgments This work is supported in part by Centre for Perceptual and Interactive Intelligence Limited, in part by the GRF through the Research Grants Council of Hong Kong under Grants (Nos. 14208417, 14207319 and 14203518) and ITS/431/18FX, in part by CUHK Strategic Fund and CUHK Agreement TS1712093, in part by the Shanghai Committee of Science and Technology, China (Grant No. 20DZ1100800).

References

[1] Iñigo Alonso, Luis Riazuelo, Luis Montesano, and Ana C Murillo. 3d-mininet: Learning a 2d representation from point clouds for fast and efficient 3d lidar semantic segmentation. *arXiv preprint arXiv:2002.10893*, 2020. 2

[2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, pages 9297–9307, 2019. 2, 5, 6

[3] Jens Behley, Andres Milioto, and Cyrill Stachniss. A benchmark for lidar-based panoptic segmentation based on kitti. *arXiv preprint arXiv:2003.02371*, 2020. 7

[4] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*, pages 4413–4421, 2018. 5

[5] 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. *arXiv preprint arXiv:1903.11027*, 2019. 2, 5, 6

[6] Wanli Chen, Xinge Zhu, Ruoqi Sun, Junjun He, Ruiyu Li, Xiaoyong Shen, and Bei Yu. Tensor low-rank reconstruction for semantic segmentation. *arXiv preprint arXiv:2008.00490*, 2020. 5

[7] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *MICCAI*, pages 424–432. Springer, 2016. 1, 3

[8] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds for autonomous driving, 2020. 2, 5, 6

[9] Xiaohan Ding, Yuchen Guo, Guiguang Ding, and Jungong Han. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In *ICCV*, pages 1911–1920, 2019. 4

[10] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *CVPR*, pages 9031–9040, 2020. 2

[11] Martin Gerdzhev, Ryan Razani, Ehsan Taghavi, and Bingbing Liu. Tornado-net: multiview total variation semantic segmentation with diamond inception module. *arXiv preprint arXiv:2008.10544*, 2020. 6

[12] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, pages 9224–9232, 2018. 1, 3

[13] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *CVPR*, pages 2940–2949, 2020. 3

[14] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *CVPR*, pages 11108–11117, 2020. 2, 5, 6

[15] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *CVPR*, pages 4867–4876, 2020. 8

[16] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, pages 9404–9413, 2019. 7

[17] Deyvid Kochanov, Fatemeh Karimi Nejadasl, and Olaf Booij. Kprnet: Improving projection-based lidar semantic segmentation. *arXiv preprint arXiv:2007.12668*, 2020. 6

[18] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, pages 4558–4567, 2018. 2

[19] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019. 8

[20] Yecheng Lyu, Xinming Huang, and Ziming Zhang. Learning to segment 3d point clouds in 2d image space. In *CVPR*, pages 12255–12264, 2020. 2

[21] Hsien-Yu Meng, Lin Gao, Yu-Kun Lai, and Dinesh Manocha. Vv-net: Voxel vae net with group convolutions for point cloud segmentation. In *ICCV*, pages 8500–8508, 2019. 3

[22] A. Milioto, J. Behley, C. McCool, and C. Stachniss. LiDAR Panoptic Segmentation for Autonomous Driving. 2020. 8

[23] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *IROS*, pages 4213–4220. IEEE, 2019. 1, 2, 3, 6

[24] Quang-Hieu Pham, Thanh Nguyen, Binh-Son Hua, Gemma Roig, and Sai-Kit Yeung. Jsis3d: joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In *CVPR*, pages 8827–8836, 2019. 2

[25] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017. 2

[26] Xiaojuan Qi, Renjie Liao, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. 3d graph neural networks for rgbd semantic segmentation. In *ICCV*, pages 5199–5208, 2017. 2

[27] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*, pages 10529–10538, 2020. 8

[28] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. *arXiv preprint arXiv:2007.16100*, 2020. 6

[29] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *CVPR*, pages 3887–3896, 2018. 6

[30] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *3DV*, pages 537–547. IEEE, 2017. 3

[31] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, pages 6411–6420, 2019. 2, 6, 8

[32] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 2

[33] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *CVPR*, pages 10296–10305, 2019. 2

[34] Tai Wang, Xinge Zhu, and Dahua Lin. Reconfigurable voxels: A new representation for lidar-based point clouds. *Conference on Robot Learning*, 2020. 8

[35] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. In *CVPR*, pages 9336–9345, 2019. 4

[36] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 2

[37] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *ICRA*, pages 1887–1893. IEEE, 2018. 1, 2

[38] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *ICRA*, pages 4376–4382. IEEE, 2019. 3

[39] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, pages 9621–9630, 2019. 2

[40] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. *arXiv preprint arXiv:2004.01803*, 2020. 2, 6

[41] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *CVPR*, pages 5589–5598, 2020. 2

[42] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 8

[43] Feihu Zhang, Jin Fang, Benjamin Wah, and Philip Torr. Deep fusionnet for point cloud semantic segmentation. *ECCV*, 2020. 2, 6

[44] Hang Zhang, Han Zhang, Chenguang Wang, and Junyuan Xie. Co-occurrent features in semantic segmentation. In *CVPR*, pages 548–557, 2019. 5

[45] Jiazhao Zhang, Chenyang Zhu, Lintao Zheng, and Kai Xu. Fusion-aware point convolution for online semantic 3d scene segmentation. In *CVPR*, pages 4534–4543, 2020. 2

[46] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *CVPR*, pages 9601–9610, 2020. 1, 2, 3, 6

[47] Hui Zhou, Xinge Zhu, X. Song, Yuexin Ma, Zhe Wang, Hongsheng Li, and D. Lin. Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation. *ArXiv*, abs/2008.01550, 2020. 2

[48] Xinge Zhu, Yuexin Ma, Tai Wang, Yan Xu, Jianping Shi, and Dahua Lin. Ssn: Shape signature networks for multi-class object detection from point clouds. *ECCV*, 2020. 8