3D Object Detection in Point Clouds

Jiantao Gao

2020.4.10

3 works by Shaoshuai Shi (CUHK)

- PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud (CVPR 2019)
- Part-A2 Net: 3D Part-Aware and Aggregation Neural Network for Object Detection from Point Cloud (T-PAMI 2020)
- PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection (CVPR 2020)

PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud

Shaoshuai Shi Xiaogang Wang Hongsheng Li
The Chinese University of Hong Kong
{ssshi, xgwang, hsli}@ee.cuhk.edu.hk

Motivation:

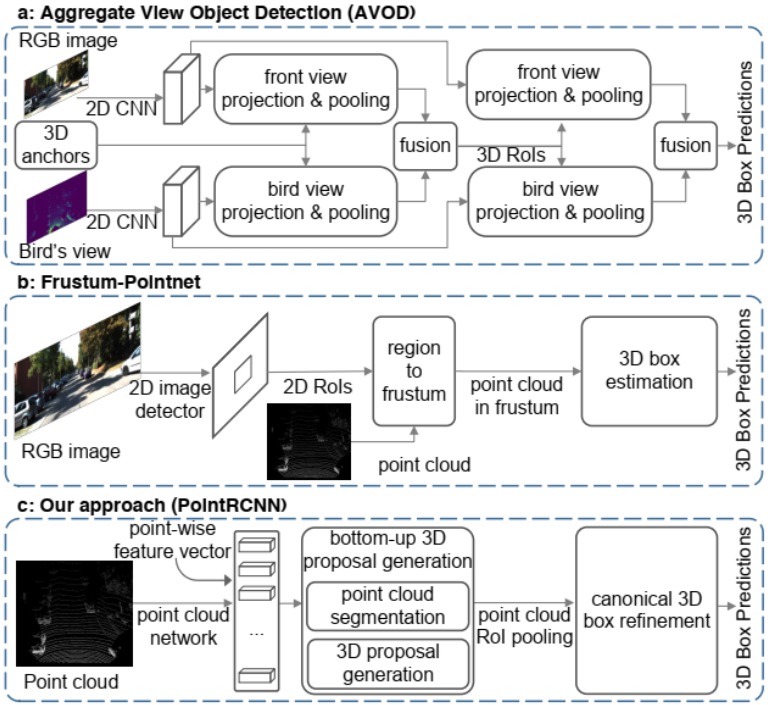


Figure 1. Comparison with state-of-the-art methods. Instead of generating proposals from fused feature maps of bird's view and front view [14], or RGB images [25], our method directly generates 3D proposals from raw point cloud in a bottom-up manner.

1. Generating proposals from multi-views:

- Suffer from information loss during the quantization

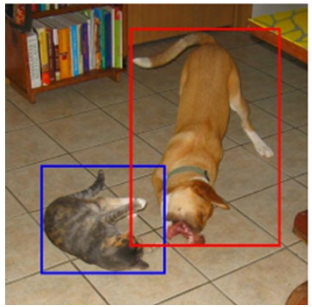
2. Generating proposals from 2D RGB detection results:

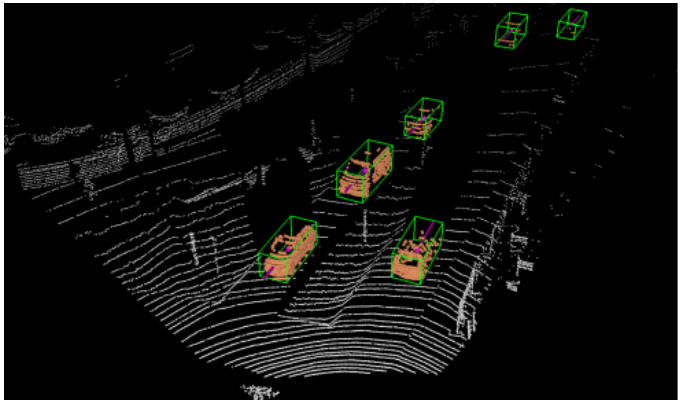
- Heavily relies on the 2D detection performance
- Cannot take the advantages of 3D information

Generating proposals from 3D point clouds directly ?

Motivation:

How to generate proposals from 3D point clouds directly ?





Unlike object detection from 2D images, 3D objects in autonomous driving scenes are naturally and well separated by annotated 3D bounding boxes.

↓

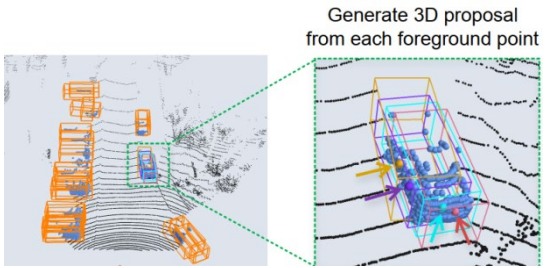
The training data for 3D object detection directly provides ***the semantic masks for 3D object segmentation***

↓

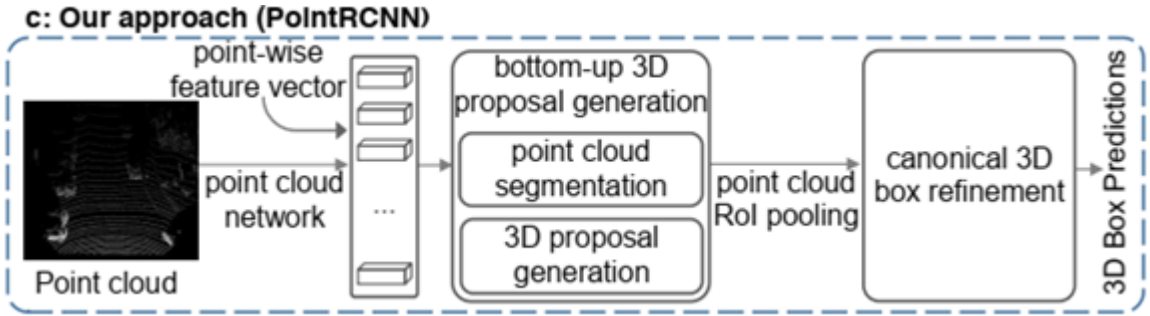
The foreground points are closed to their corresponding object centers

Firstly, segmentation

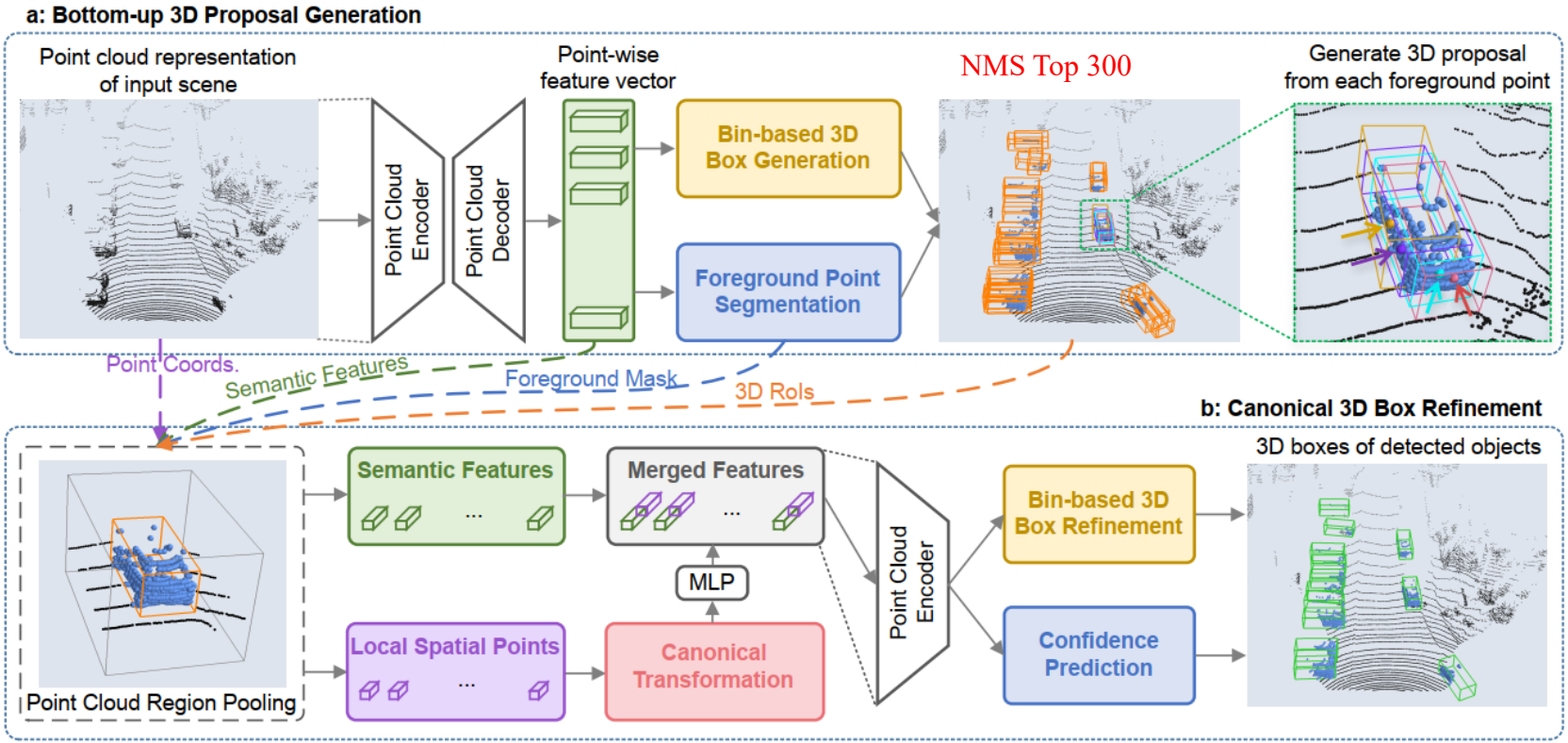
Second, regression



Pipeline of PointRCNN:



Pipeline of PointRCNN:



$(x_i, y_i, z_i, h_i + \eta, w_i + \eta, l_i + \eta, \theta_i)$

Bin-based 3D Box Generation:

- A 3D bounding box : (x; y; z; h; w; l; θ)
- the coordinates of a interest foreground point:

$$(x^{(p)}, y^{(p)}, z^{(p)})$$

↓

the center coordinates of its corresponding object:

$$(x^{p}, y^{p}, z^{p})$$

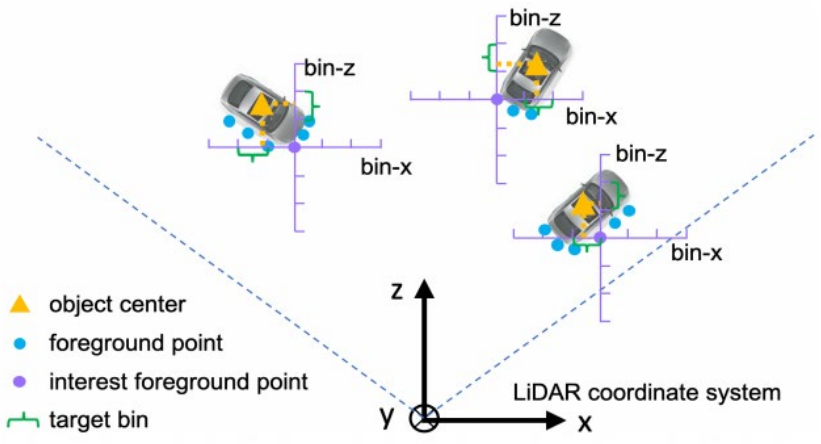


Figure 3. Illustration of bin-based localization. The surrounding area along X and Z axes of each foreground point is split into a series of bins to locate the object center.

Bin-based Classification and Regression on X-Z plane

a. set a search range S for each X and Z axis of the current foreground point

b. each 1D search range is divided into bins of uniform length δ on the X-Z plane

c.

$$\text{bin}_x^{(p)} = \left\lfloor \frac{x^p - x^{(p)} + \mathcal{S}}{\delta} \right\rfloor, \ \text{bin}_z^{(p)} = \left\lfloor \frac{z^p - z^{(p)} + \mathcal{S}}{\delta} \right\rfloor,$$

$$\underset{u\in\{x,z\}}{\text{res}_u^{(p)}} = \frac{1}{\mathcal{C}} \left(u^p - u^{(p)} + \mathcal{S} - \left(\text{bin}_u^{(p)} \cdot \delta + \frac{\delta}{2} \right) \right), \quad (2)$$

$$\text{res}_y^{(p)} = y^p - y^{(p)}$$

Loss Function:

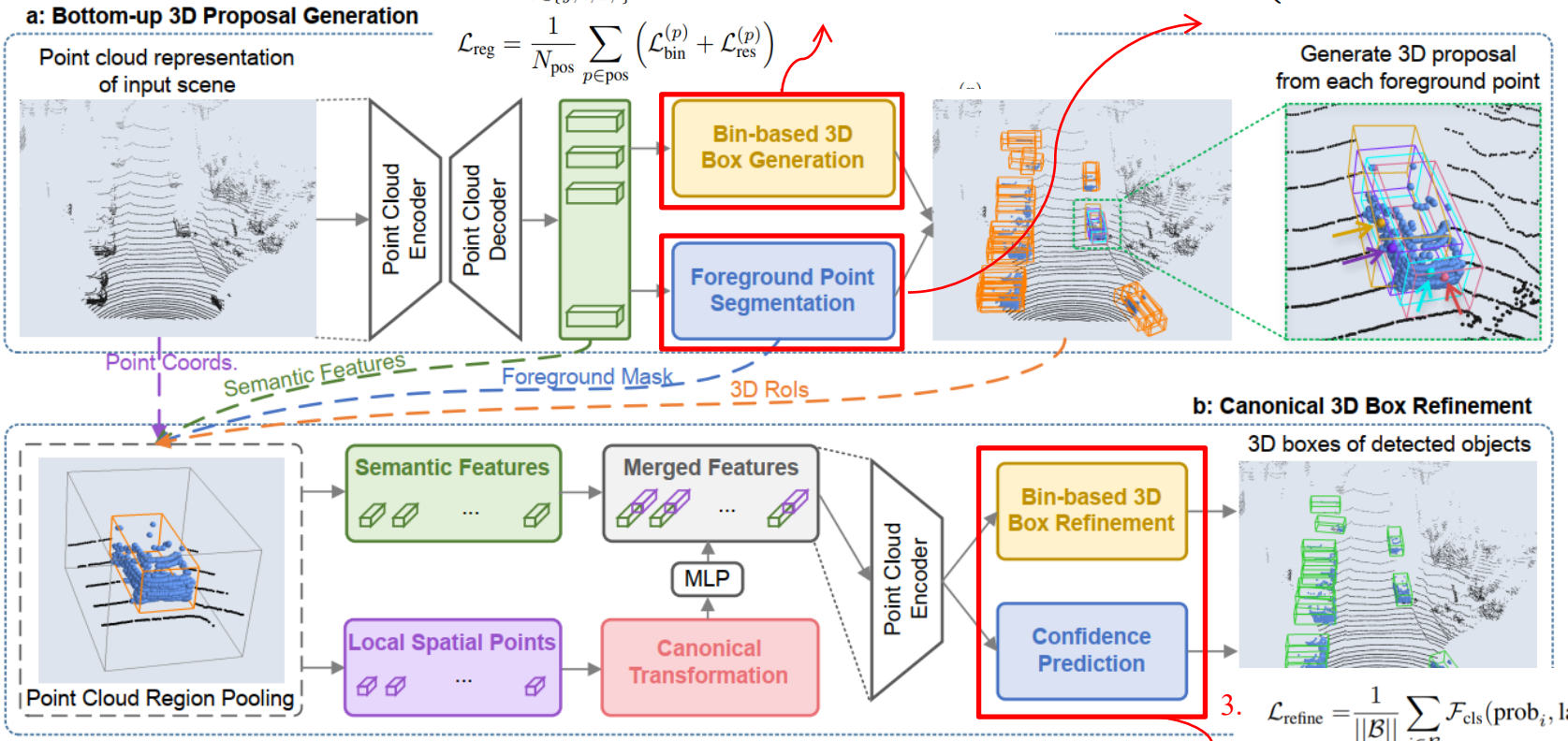
1. $$\mathcal{L}_{\text{bin}}^{(p)} = \sum_{u\in\{x,z,\theta\}} (\mathcal{F}_{\text{cls}}(\widehat{\text{bin}}_u^{(p)}, \text{bin}_u^{(p)}) + \mathcal{F}_{\text{reg}}(\widehat{\text{res}}_u^{(p)}, \text{res}_u^{(p)})),$$

$$\mathcal{L}_{\text{res}}^{(p)} = \sum_{v\in\{y,h,w,l\}} \mathcal{F}_{\text{reg}}(\widehat{\text{res}}_v^{(p)}, \text{res}_v^{(p)}),$$

$$\mathcal{L}_{\text{reg}} = \frac{1}{N_{\text{pos}}} \sum_{p\in\text{pos}} \left(\mathcal{L}_{\text{bin}}^{(p)} + \mathcal{L}_{\text{res}}^{(p)}\right)$$

2. $$\mathcal{L}_{\text{focal}}(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t),$$

$$\text{where } p_t = \begin{cases} p & \text{for forground point} \\ 1-p & \text{otherwise} \end{cases}$$



3. $$\mathcal{L}_{\text{refine}} = \frac{1}{||\mathcal{B}||}\sum_{i\in\mathcal{B}} \mathcal{F}_{\text{cls}}(\text{prob}_i, \text{label}_i)$$

$$+ \frac{1}{||\mathcal{B}_{pos}||}\sum_{i\in\mathcal{B}_{pos}} (\tilde{\mathcal{L}}_{\text{bin}}^{(i)} + \tilde{\mathcal{L}}_{\text{res}}^{(i)})$$

Experiments:

Method	Modality	Car (IoU=0.7)			Pedestrian (IoU=0.5)			Cyclist (IoU=0.5)		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
MV3D [4]	RGB + LiDAR	71.09	62.35	55.12	-	-	-	-	-	-
UberATG-ContFuse [17]	RGB + LiDAR	82.54	66.22	64.04	-	-	-	-	-	-
AVOD-FPN [14]	RGB + LiDAR	81.94	71.88	66.38	50.80	42.81	**40.88**	64.00	52.18	46.61
F-PointNet [25]	RGB + LiDAR	81.20	70.39	62.19	**51.21**	**44.89**	40.23	71.96	56.77	50.39
VoxelNet [43]	LiDAR	77.47	65.11	57.73	39.48	33.69	31.51	61.22	48.36	44.37
SECOND [40]	LiDAR	83.13	73.66	66.20	51.07	42.56	37.29	70.51	53.85	46.90
Ours	LiDAR	**85.94**	**75.76**	**68.32**	49.43	41.78	38.63	**73.93**	**59.60**	**53.59**

Table 1. Performance comparison of 3D object detection with previous methods on KITTI *test* split by submitting to official test server. The evaluation metric is Average Precision(AP) with IoU threshold 0.7 for car and 0.5 for pedestrian/cyclist.

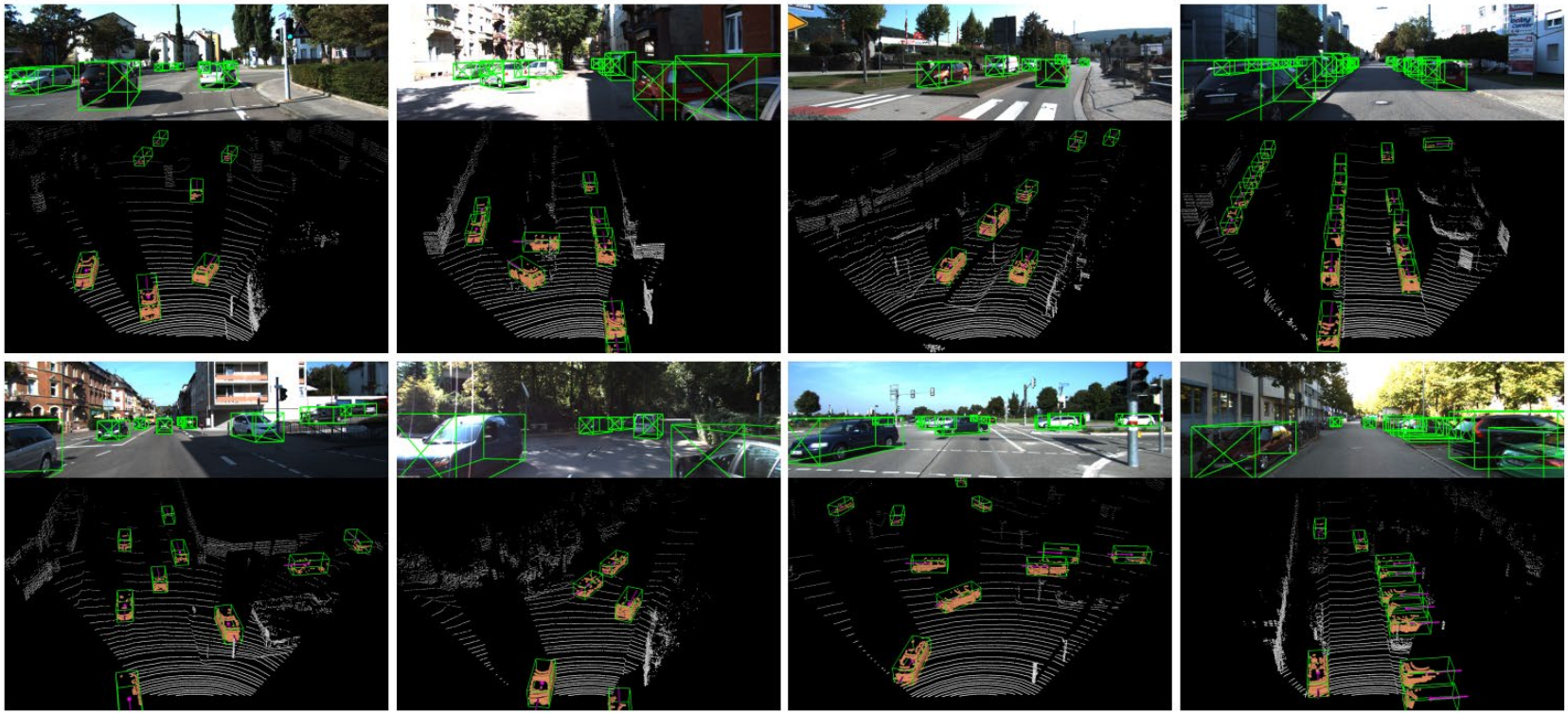
Method	AP(IoU=0.7)		
	Easy	Moderate	Hard
MV3D [4]	71.29	62.68	56.56
VoxelNet [43]	81.98	65.46	62.85
SECOND [40]	87.43	76.48	69.10
AVOD-FPN [14]	84.41	74.44	68.65
F-PointNet [25]	83.76	70.92	63.65
Ours (no GT-AUG)	88.45	77.67	76.30
Ours	**88.88**	**78.63**	**77.38**

Table 2. Performance comparison of 3D object detection with previous methods on the car class of KITTI *val* split set.

RoIs #	Recall(IoU=0.5)			Recall(IoU=0.7)
	MV3D	AVOD	Ours	Ours
10	-	86.00	**86.66**	29.87
20	-	-	**91.83**	32.55
30	-	-	**93.31**	32.76
40	-	-	**95.55**	40.04
50	-	91.00	**96.01**	40.28
100	-	-	**96.79**	74.81
200	-	-	**98.03**	76.29
300	91.00	-	**98.21**	82.29

Table 3. Recall of proposal generation network with different number of RoIs and 3D IoU threshold for the car class on the *val* split at moderate difficulty. Note that only MV3D [4] and AVOD [14] of previous methods reported the number of recall.

Experiments:



Ablation Study:

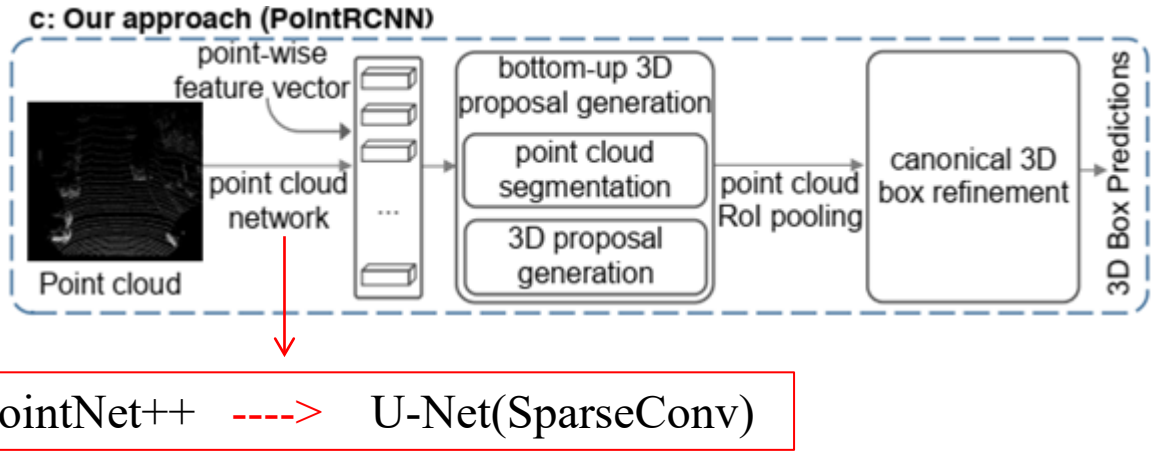
CT	RPN features	camera depth	seg. mask	AP_E	AP_M	AP_H
×	✓	✓	✓	7.64	13.68	13.94
✓	×	✓	✓	84.75	74.96	74.29
✓	✓	×	✓	87.34	76.79	75.46
✓	✓	✓	×	86.25	76.64	75.86
✓	✓	✓	✓	**88.45**	**77.67**	**76.30**

Table 4. Performance for different input combinations of refinement network. AP_E, AP_M, AP_H denote the average precision for easy, moderate, hard difficulty on KITTI *val* split, respectively. CT denotes canonical transformation.

η (context width)	AP_E	AP_M	AP_H
no context	86.65	75.68	68.92
0.5m	87.87	77.12	75.61
0.8m	88.27	77.40	76.07
1.0m	**88.45**	**77.67**	**76.30**
1.5m	86.82	76.87	75.88
2.0m	86.47	76.61	75.53

Table 5. Performance of adopting different context width η of context-aware point cloud pooling.

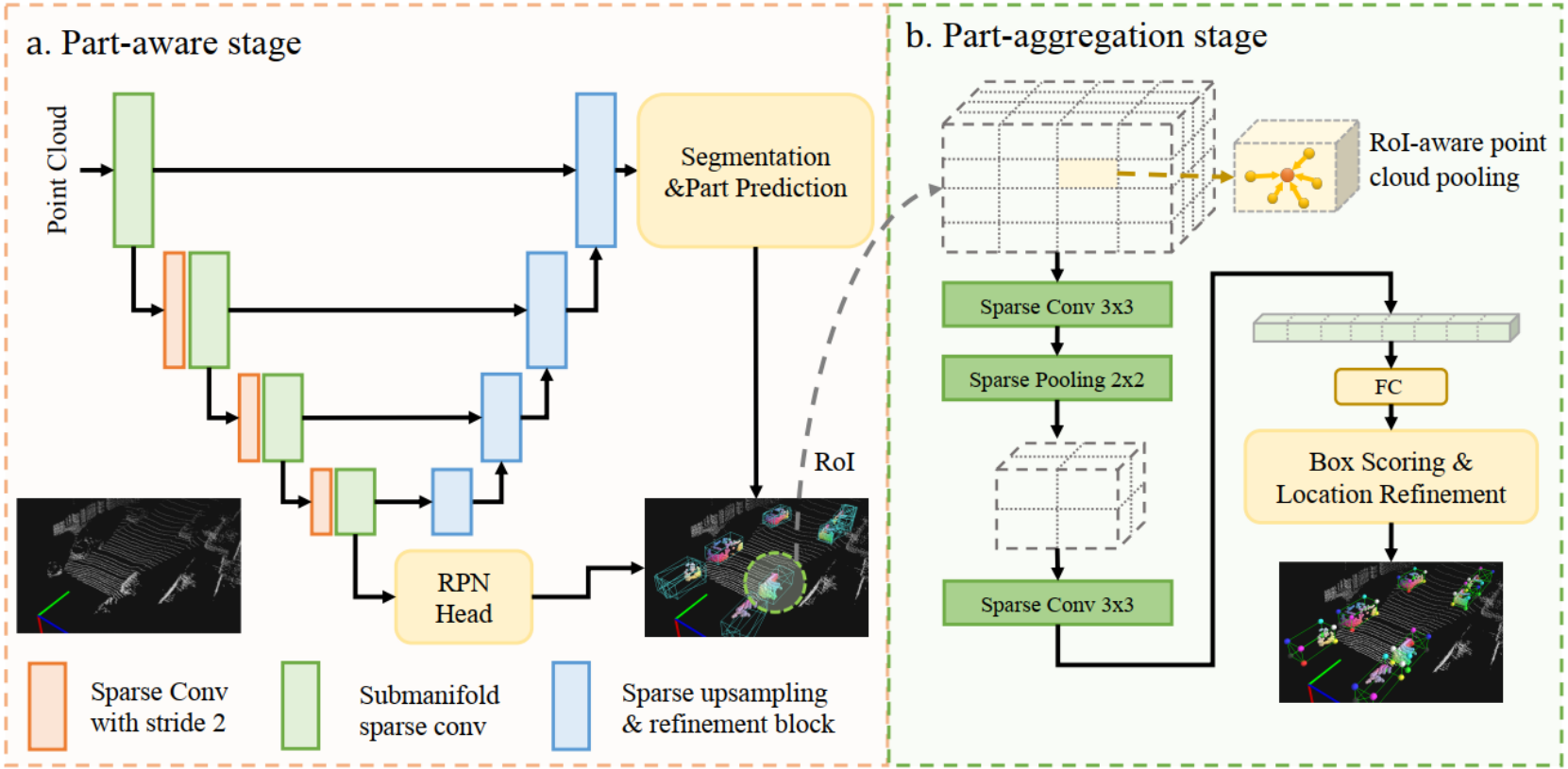
Limitation:


c: Our approach (PointRCNN)
point-wise
feature vector
point cloud
network
Point cloud
bottom-up 3D
proposal generation
point cloud
segmentation
3D proposal
generation
point cloud
RoI pooling
canonical 3D
box refinement
3D Box Predictions
PointNet++ ----> U-Net(SparseConv)

Part-A^2 Net: 3D Part-Aware and Aggregation Neural Network for Object Detection from Point Cloud

Shaoshuai Shi[1] Zhe Wang[2] Xiaogang Wang[1] Hongsheng Li[1]
[1]The Chinese University of Hong Kong [2]SenseTime Research
{ssshi, xgwang, hsli}@ee.cuhk.edu.hk wangzhe@sensetime.com

Pipeline of Part-A^2:



Intra-object part locations :

However, we observed that the 3D box annotations not only **provide the segmentation masks**, but also **imply accurate intra-object part locations** for all the points within the 3D boxes.

A 3D bounding box: $(C_x, C_y, C_z, h, w, l, \theta)$

A foreground point: (p_x, p_y, p_z)

The intra-object part locations of the point:

$$[t_x \quad t_y] = [p_x - C_x \quad p_y - C_y] \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix},$$

$$O_x = \frac{t_x}{w} + 0.5,\ O_y = \frac{t_y}{l} + 0.5,\ O_z = \frac{p_z - O_z}{h} + 0.5 \tag{1}$$

where we have $O_x, O_y, O_z \in [0, 1]$, and then the part location of the object center is $(0.5, 0.5, 0.5)$.

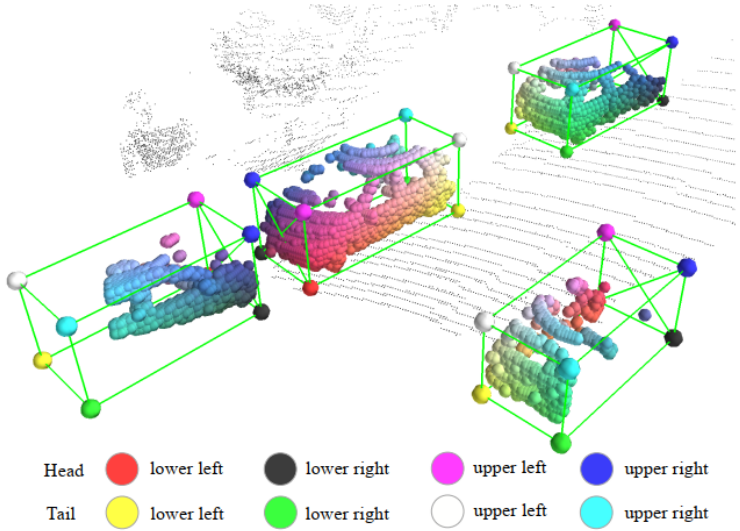


Figure 1. Intra-object part locations and segmentation masks can be robustly predicted by the proposed part-aware and aggregation network even when objects are partially occluded. Such part locations can assist accurate 3D object detection. Best view in colors.

RoI-aware Point Cloud Pooling:

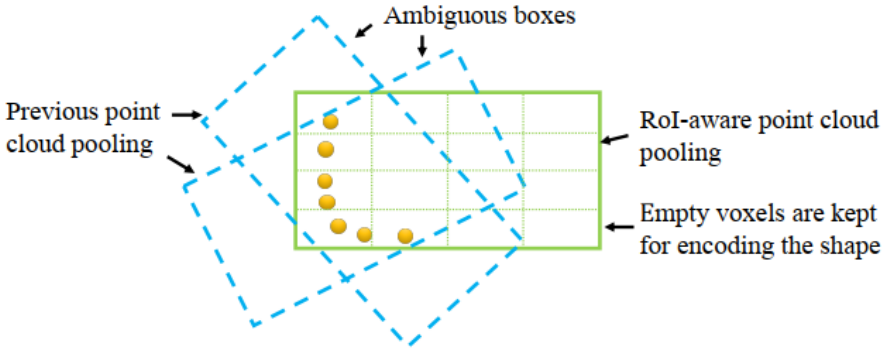


Figure 4. Illustration of RoI-aware point cloud feature pooling. Due to the ambiguity showed in the above BEV figure, We could not recover the original box shape by using previous point cloud pooling method. Our proposed RoI-aware point cloud pooling method could encode the box shape by keeping the empty voxels, which could be efficiently processed by following sparse convolution.

Experiments:

Method	Modality	3D Detection (Car)			BEV Detection (Car)			3D Detection (Cyclist)			BEV Detection (Cyclist)		
		Mod.	Easy	Hard	Mod.	Easy	Hard	Mod.	Easy	Hard	Mod.	Easy	Hard
MV3D [2]	RGB + LiDAR	62.35	71.09	55.12	76.90	86.02	68.49	-	-	-	-	-	-
ContFuse [15]	RGB + LiDAR	66.22	82.54	64.04	85.83	88.81	77.33	-	-	-	-	-	-
AVOD-FPN [11]	RGB + LiDAR	71.88	81.94	66.38	83.79	88.53	77.90	52.18	64.00	46.61	57.48	68.09	50.77
F-PointNet [20]	RGB + LiDAR	70.39	81.20	62.19	84.00	88.70	75.33	56.77	71.96	50.39	61.96	75.38	54.68
UberATG-MMF [14]	RGB + LiDAR	76.75	86.81	68.41	87.47	89.49	79.10	-	-	-	-	-	-
VoxelNet [35]	LiDAR only	65.11	77.47	57.73	79.26	89.35	77.39	48.36	61.22	44.37	54.76	66.70	50.55
SECOND [32]	LiDAR only	73.66	83.13	66.20	79.37	88.07	77.95	53.85	70.51	40.90	56.04	73.67	48.78
PointPillars [12]	LiDAR only	74.99	79.05	68.30	**86.10**	88.35	79.83	59.07	75.78	52.92	62.25	79.14	56.00
PointRCNN [27]	LiDAR only	75.76	**85.94**	68.32	85.68	89.47	79.10	59.60	73.93	53.59	66.77	81.52	60.78
Part-A^2(Ours)	LiDAR only	**77.86**	**85.94**	**72.00**	84.76	**89.52**	**81.47**	**62.73**	**78.58**	**57.74**	**68.12**	**81.91**	**61.92**

Table 1. Performance evaluation on KITTI 3D object detection test server (*test* split). The 3D object detection and bird's eye view detection are evaluated by average precision with rotated IoU threshold 0.7 for car and 0.5 for cyclist.

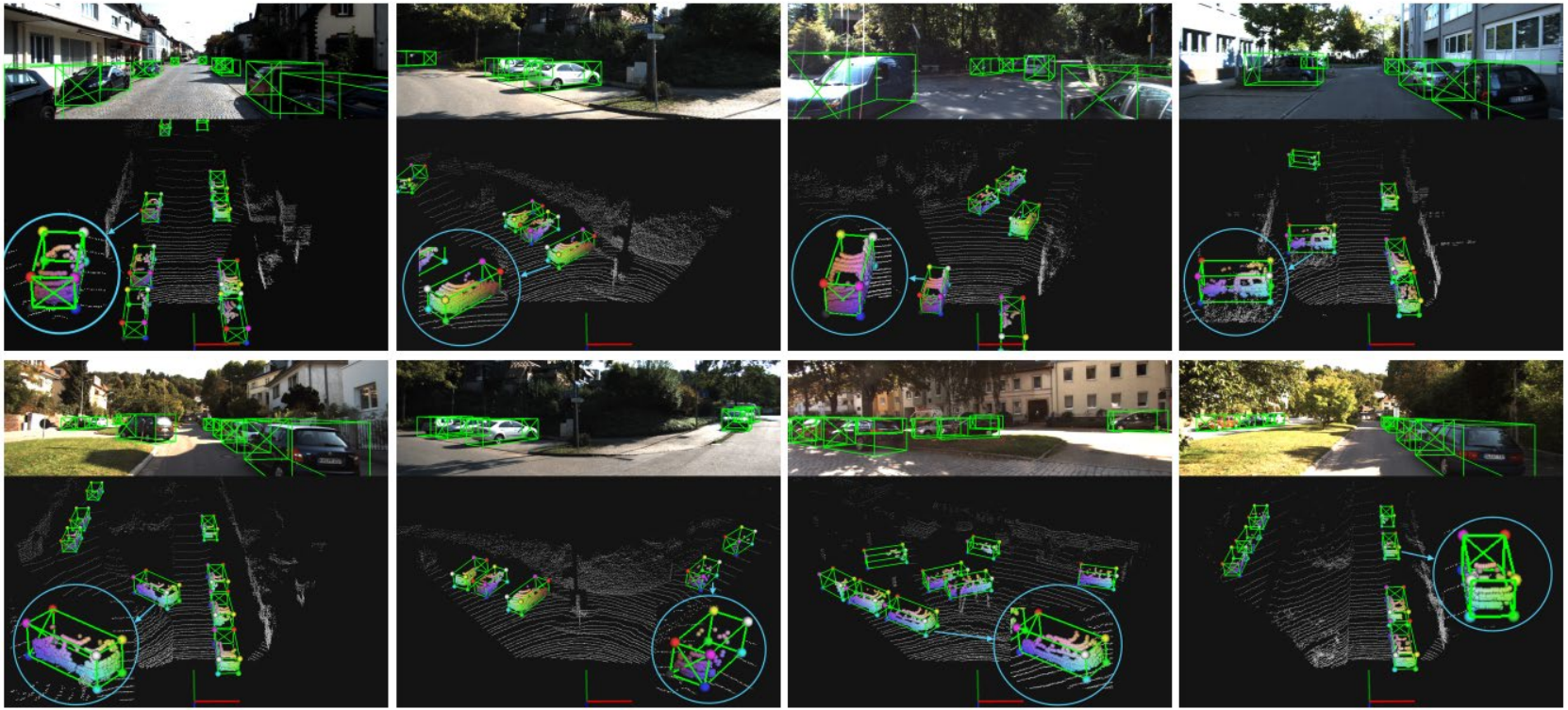
Method	Modality	AP (IoU=0.7)		
		Moderate	Easy	Hard
MV3D [2]	RGB & LiDAR	62.68	71.29	56.56
ContFuse[15]	RGB & LiDAR	73.25	86.32	67.81
AVOD-FPN [11]	RGB & LiDAR	74.44	84.41	68.65
F-PointNet [20]	RGB & LiDAR	70.92	83.76	63.65
VoxelNet [35]	LiDAR	65.46	81.98	62.85
SECOND [32]	LiDAR	76.48	87.43	69.10
PointRCNN [27]	LiDAR	78.63	88.88	77.38
Part-A^2(Ours)	LiDAR	**79.47**	**89.47**	**78.54**

Table 2. Performance comparison of 3D object detection on the car class of the KITTI *val* split set.

Method	Modality	AP (IoU=0.7)		
		Moderate	Easy	Hard
MV3D [2]	RGB & LiDAR	78.10	86.55	76.67
F-PointNet [20]	RGB & LiDAR	84.02	88.16	76.44
ContFusion [15]	RGB& LiDAR	87.34	95.44	82.43
VoxelNet [35]	LiDAR	84.81	89.60	78.57
SECOND [32]	LiDAR	87.07	89.96	79.66
Part-A^2(Ours)	LiDAR	**88.61**	**90.42**	**87.31**

Table 3. Performance comparison of bird-view object detection on the car class of the KITTI *val* split set.

Experiments:



Ablation Study:

Method	Recall (IoU=0.7) box#100	AP (IoU=0.7) Easy	Moderate	Hard
Part-aware w/o parts	80.90	88.48	77.97	75.84
Part-aware	80.99	88.90	78.54	76.44
Part-A^2 w/o parts	82.92	89.23	79.00	77.66
Part-A^2	**84.33**	**89.47**	**79.47**	**78.54**

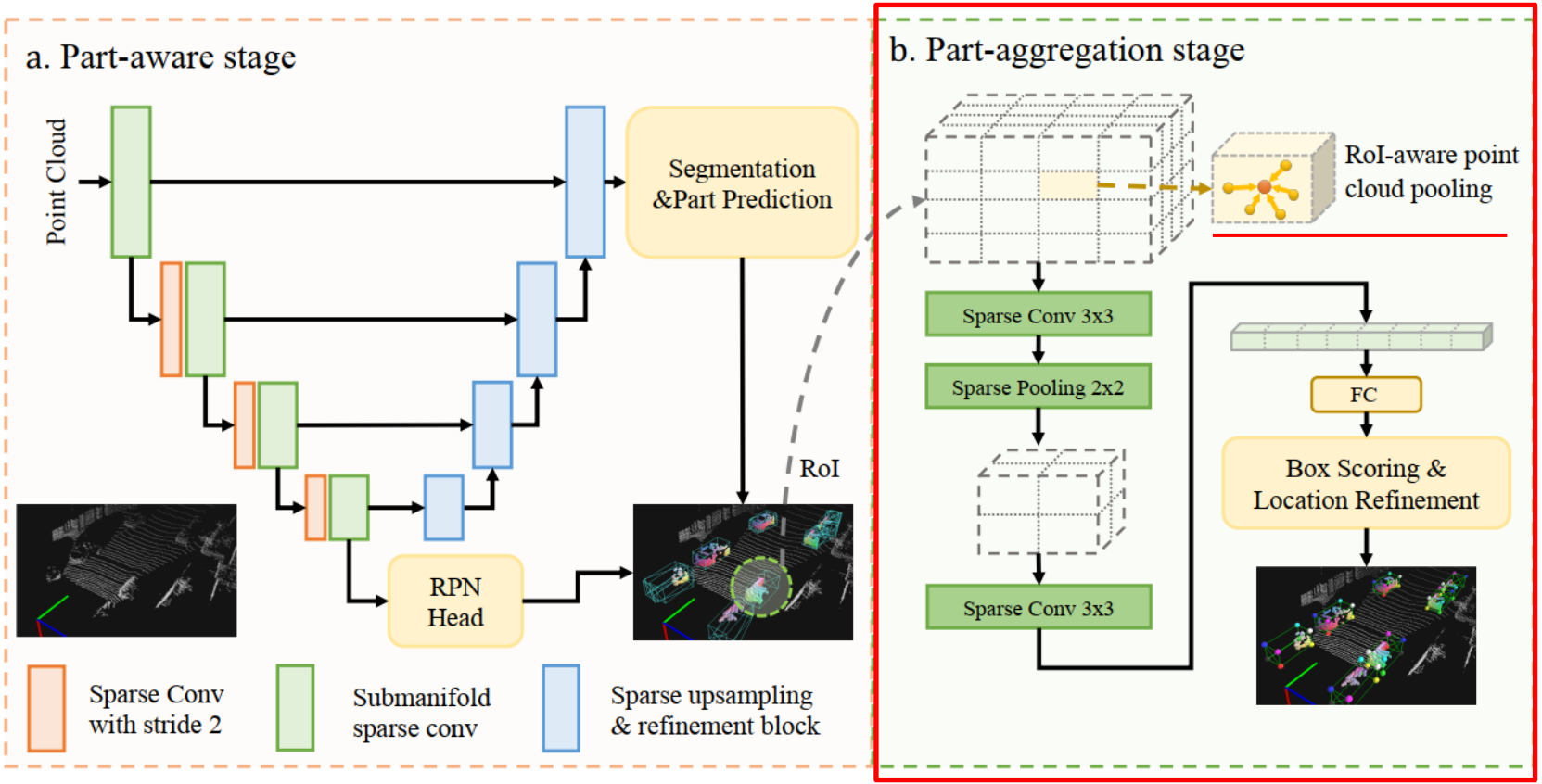
Table 5. Effects of removing the intra-object part location supervisions from our Part-A^2 net.

RoI pooling size	AP_{Easy}	$AP_{Mod.}$	AP_{Hard}
$6 \times 6 \times 6$	89.02	78.85	78.04
$8 \times 8 \times 8$	89.09	78.97	78.15
$10 \times 10 \times 10$	89.44	79.15	78.42
$12 \times 12 \times 12$	**89.61**	79.35	78.50
$14 \times 14 \times 14$	89.47	**79.47**	78.54
$16 \times 16 \times 16$	89.52	79.45	**78.56**

Table 8. Effects of using different RoI-aware pooling sizes in our part-aggregation stage.

Limitation:

- Avg/Max pooling the point in each voxel: location information loss



PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection

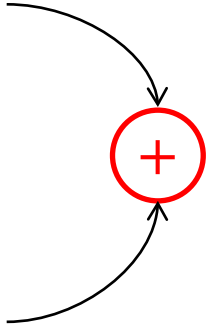
Shaoshuai Shi[1] Chaoxu Guo[2,3] Li Jiang[4]
Zhe Wang[2] Jianping Shi[2] Xiaogang Wang[1] Hongsheng Li[1]

[1]CUHK-SenseTime Joint Laboratory, The Chinese University of Hong Kong
[2]SenseTime Research [3]NLPR, CASIA [4]CSE, CUHK

Motivation:

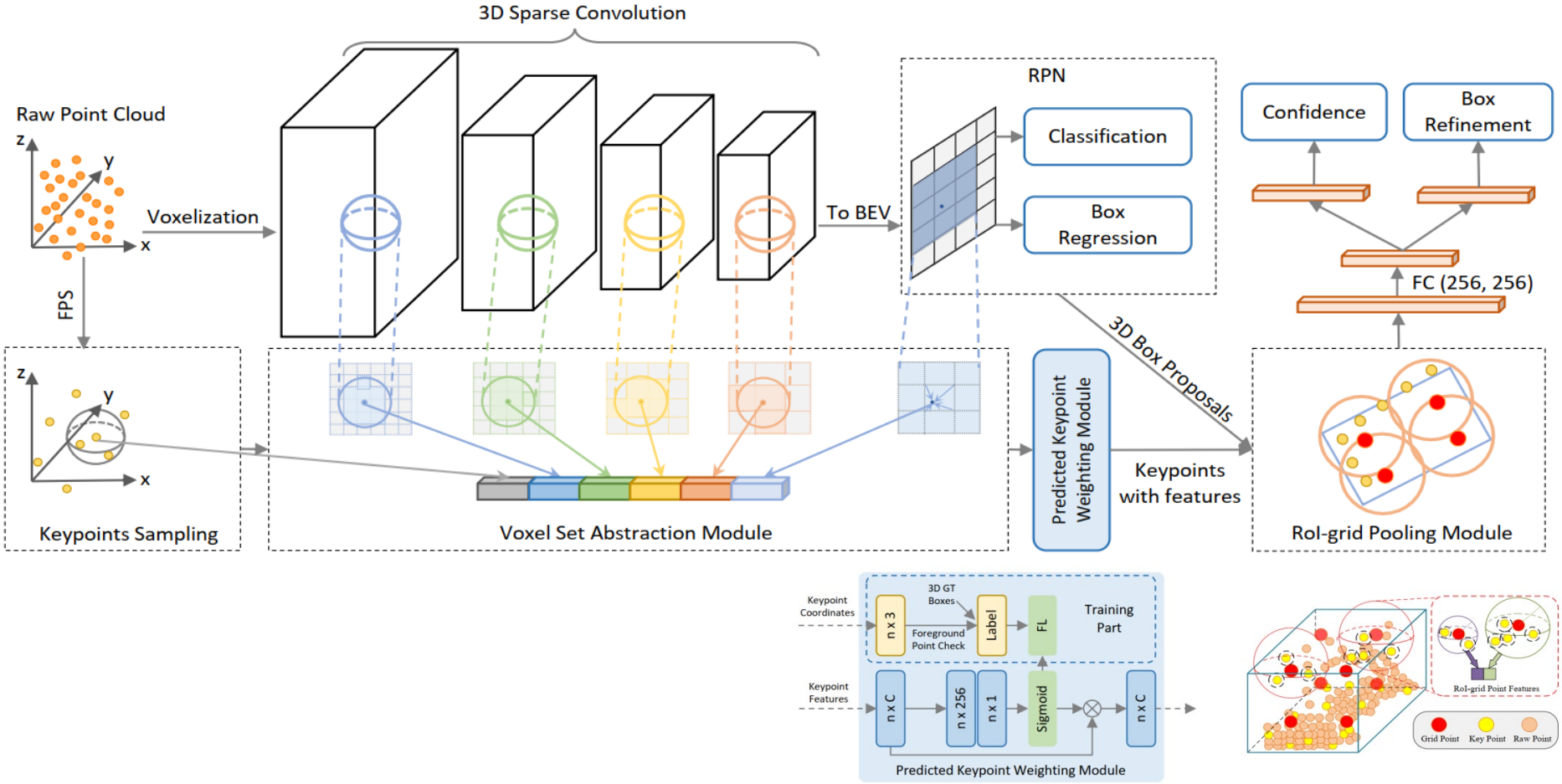
- The voxel-based operation efficiently encodes multi-scale feature representations and can generate high-quality 3D proposals

- The PointNet-based set abstraction operation preserves accurate location information with flexible receptive fields



Combine ?

Pipeline of PV-RCNN:



Experiments:

Method	Reference	Modality	Car - 3D Detection			Car - BEV Detection			Cyclist - 3D Detection			Cyclist - BEV Detection		
			Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
MV3D [1]	CVPR 2017	RGB + LiDAR	74.97	63.63	54.00	86.62	78.93	69.80	-	-	-	-	-	-
ContFuse [17]	ECCV 2018	RGB + LiDAR	83.68	68.78	61.67	94.07	85.35	75.88	-	-	-	-	-	-
AVOD-FPN [11]	IROS 2018	RGB + LiDAR	83.07	71.76	65.73	90.99	84.82	79.62	63.76	50.55	44.93	69.39	57.12	51.09
F-PointNet [22]	CVPR 2018	RGB + LiDAR	82.19	69.79	60.59	91.17	84.67	74.77	72.27	56.12	49.01	77.26	61.37	53.78
UberATG-MMF [16]	CVPR 2019	RGB + LiDAR	88.40	77.43	70.22	93.67	88.21	81.99	-	-	-	-	-	-
SECOND [34]	Sensors 2018	LiDAR only	83.34	72.55	65.82	89.39	83.77	78.59	71.33	52.08	45.83	76.50	56.05	49.45
PointPillars [12]	CVPR 2019	LiDAR only	82.58	74.31	68.99	90.07	86.56	82.81	77.10	58.65	51.92	79.90	62.73	55.58
PointRCNN [25]	CVPR 2019	LiDAR only	86.96	75.64	70.70	92.13	87.39	82.72	74.96	58.82	52.53	**82.56**	67.24	60.28
3D IoU Loss [39]	3DV 2019	LiDAR only	86.16	76.50	71.39	91.36	86.22	81.20	-	-	-	-	-	-
Fast Point R-CNN [2]	ICCV 2019	LiDAR only	85.29	77.40	70.24	90.87	87.84	80.52	-	-	-	-	-	-
STD [37]	ICCV 2019	LiDAR only	87.95	79.71	75.09	94.74	89.19	**86.42**	**78.69**	61.59	55.30	81.36	67.23	59.35
Patches [13]	Arxiv 2019	LiDAR only	88.67	77.20	71.82	92.72	88.39	83.19	-	-	-	-	-	-
Part-A^2 [26]	Arxiv 2019	LiDAR only	87.81	78.49	73.51	91.70	87.79	84.61	-	-	-	-	-	-
PV-RCNN (Ours)	-	LiDAR only	**90.25**	**81.43**	**76.82**	**94.98**	**90.65**	86.14	78.60	**63.71**	**57.65**	82.49	**68.89**	**62.41**
Improvement	-	-	*+1.58*	*+1.72*	*+1.73*	*+0.24*	*+1.46*	*-0.28*	*-0.06*	*+2.12*	*+2.35*	*-0.07*	*+1.65*	*+2.13*

Table 1. Performance comparison on the KITTI *test* set. The results are evaluated by the mean Average Precision with 40 recall positions.

Method	Reference	Modality	3D mAP
MV3D [1]	CVPR 2017	RGB + LiDAR	62.68
ContFuse [17]	ECCV 2018	RGB + LiDAR	73.25
AVOD-FPN [11]	IROS 2018	RGB + LiDAR	74.44
F-PointNet [22]	CVPR 2018	RGB + LiDAR	70.92
VoxelNet [41]	CVPR 2018	LiDAR only	65.46
SECOND [34]	Sensors 2018	LiDAR only	76.48
PointRCNN [25]	CVPR 2019	LiDAR only	78.63
Fast Point R-CNN [2]	ICCV 2019	LiDAR only	79.00
STD [37]	ICCV 2019	LiDAR only	79.80
PV-RCNN (Ours)	-	LiDAR only	**83.90**

Table 2. Performance comparison on the moderate level car class of KITTI *val* split with mAP calculated by 11 recall positions.

IoU Thresh.	3D mAP			BEV mAP		
	Easy	Moderate	Hard	Easy	Moderate	Hard
0.7	92.57	84.83	82.69	95.76	91.11	88.93

Table 3. Performance on the KITTI *val* split set with mAP calculated by 40 recall positions for car class.

Method	PointRCNN [25]	STD [37]	PV-RCNN (Ours)
Recall (IoU=0.7)	74.8	76.8	85.5

Table 4. Recall of different proposal generation networks on the car class at moderate difficulty level of the KITTI *val* split set.

Experiments:

Difficulty	Method	3D mAP (IoU=0.7)				3D mAPH (IoU=0.7)				BEV mAP (IoU=0.7)				BEV mAPH (IoU=0.7)			
		Overall	0-30m	30-50m	50m-Inf	Overall	0-30m	30-50m	50m-Inf	Overall	0-30m	30-50m	50m-Inf	Overall	0-30m	30-50m	50m-Inf
LEVEL_1	PointPillar [12]	56.62	81.01	51.75	27.94	-	-	-	-	75.57	92.1	74.06	55.47	-	-	-	-
	MVF [40]	62.93	86.30	60.02	36.02	-	-	-	-	80.40	93.59	79.21	63.09	-	-	-	-
	PV-RCNN (Ours)	**70.30**	**91.92**	**69.21**	**42.17**	**69.69**	**91.34**	**68.53**	**41.31**	**82.96**	**97.35**	**82.99**	**64.97**	**82.06**	**96.71**	**82.01**	**63.15**
	Improvement	*+7.37*	*+5.62*	*+9.19*	*+6.15*	-	-	-	-	*+2.56*	*+3.76*	*+3.78*	*+1.88*	-	-	-	-
LEVEL_2	PV-RCNN (Ours)	65.36	91.58	65.13	36.46	64.79	91.00	64.49	35.70	77.45	94.64	80.39	55.39	76.60	94.03	79.40	53.82

Table 5. Performance comparison on the Waymo Open Dataset with 202 validation sequences for the vehicle detection. Note that the results of PointPillar [12] on the Waymo Open Dataset are reproduced by [40].

Ablation Study:

Method	RPN with 3D Voxel CNN	Keypoints Encoding	RoI-grid Pooling	Easy	Mod.	Hard
RPN Baseline	✓			90.46	80.87	77.30
Pool from Encoder	✓		✓	91.88	82.86	80.52
PV-RCNN	✓	✓	✓	**92.57**	**84.83**	**82.69**

Table 6. Effects of voxel-to-keypoint scene encoding strategy and RoI-grid pooling refinement.

$f_i^{(pv_1)}$	$f_i^{(pv_2)}$	$f_i^{(pv_3)}$	$f_i^{(pv_4)}$	$f_i^{(bev)}$	$f_i^{(raw)}$	Moderate mAP
					✓	81.98
				✓		83.32
			✓			83.17
		✓	✓			84.54
		✓	✓	✓		84.69
		✓	✓	✓	✓	84.72
	✓	✓	✓	✓	✓	84.75
✓	✓	✓	✓	✓	✓	**84.83**

Table 7. Effects of different feature components for VSA module.

Thanks