Supplementary Material for Not All Points Are Equal: Learning Highly Efficient Point-based Detectors for 3D LiDAR Point Clouds

Yifan Zhang[1], Qingyong Hu[2*], Guoquan Xu[1], Yanxin Ma[1], Jianwei Wan[1], Yulan Guo[1]
[1]National University of Defense Technology, [2]University of Oxford
zhangyifan16c@nudt.edu.cn, qingyong.hu@cs.ox.ac.uk

1. Details of The Proposed IA-SSD

(1) Detailed Network Architecture. Here, we provide the detailed architecture of our IA-SSD. The proposed IA-SSD has a lightweight backbone, which consists of three SA (Set Abstraction) layers [4] with only two radii for the spherical neighbor query. The detailed architecture deployed on KITTI Dataset is as follows:

syntax: $SA(npoint, [radii], [nquery], [dimension])$

$SA(4096, [0.2, 0.8], [16, 32], [[16, 16, 32], [32, 32, 64]])$
$\rightarrow MLP(96 \rightarrow 64)$
$SA(1024, [0.8, 1.6], [16, 32], [[64, 64, 128], [64, 96, 128]])$
$\rightarrow MLP(256 \rightarrow 128)$
$SA(512, [1.6, 4.8], [16, 32], [[128, 128, 256], [128, 256, 256]])$
$\rightarrow MLP(512 \rightarrow 256)$

where $npoint$ denotes the number of sampled points, $[radii]$ denote the grouping radii, $[nquery]$ denotes the number of grouping points, $[dimension]$ denotes the feature dimensions.

The class/centroid-aware prediction layer:

$$MLP(256 \rightarrow 256 \rightarrow 3)$$

The architecture of the contextual instance centroid perception module is as follows:

$$MLP(256 \rightarrow 128 \rightarrow 3)$$

The architecture of centroid-based instance aggregation is as follows:

$SA(256, [4.8, 6.4], [16, 32], [[256, 256, 512], [256, 512, 1024]])$
$\rightarrow MLP(1536 \rightarrow 512)$

The final detection head is composed of two branches:

cls branch : $FC(512) \rightarrow FC(256) \rightarrow FC(256) \rightarrow FC(3)$
reg branch : $FC(512) \rightarrow FC(256) \rightarrow FC(256) \rightarrow FC(30)$

Considering the large-scale spatial ranges and increasing number of potential instances in the Waymo and ONCE datasets, the number of sampled points are improved to 16384, 4096, 2048, and 1024 in our framework, and the contextual centroid perception boundary is improved to 2.0m. The rest of the hyperparameters are kept consistent for a fair comparison.

2. Additional Implementation Details

(1) Data augmentation. During training, We also apply two data augmentation strategies including scene-level augmentation and object-level augmentation. The detailed settings and hyperparameters are as follows:

Scene-level augmentation:

- Random scene flip with a 50 $\%$ probability.
- Random scene rotation around z-axis with a random value from $[-\frac{\pi}{4}, \frac{\pi}{4}]$.
- Random scene scaling with a random factor from $[0.95, 1.05]$.

Object-level augmentation:

- Transform objects from other scenes. In particular, 20 cars, 15 pedestrians, and 15 cyclists are copied to the current scene. Note that, the minimum number of points for a sampled instance is 5.

(2) Training and inference. We train the proposed IA-SSD in an end-to-end fashion with a maximum of 80 epochs. Adam solver with onecycle learning strategy [8] is used for optimization. In our experiment, the batch size is set to 8, and the learning rate is set to 0.01. During inference, our IA-SSD is able to take raw point clouds and generate proposals for all objects in a single forward pass. Finally, all proposals are filtered by 3D-NMS post-processing with an IoU threshold of 0.01 on KITTI and 0.1 on Waymo/ONCE.

1^{st} layer	2^{nd} layer	3^{rd} layer	4^{th} layer	Recall Car	Recall Ped.	Recall Cyc.	Car Mod (IoU=0.7)	Ped. Mod (IoU=0.5)	Cyc. Mod (IoU=0.5)
Random	Random	Random	Random	67.4%	72.1%	57.3%	75.02	51.16	66.07
D-FPS	D-FPS	D-FPS	D-FPS	91.4%	69.1%	71.6%	78.12	50.46	65.19
D-FPS	Feat-FPS	Feat-FPS	Feat-FPS	95.3%	80.1%	91.7%	79.00	54.31	71.08
D-FPS	D-FPS	Cls-aware	Cls-aware	97.9%	97.4%	92.7%	79.19	58.81	70.15
D-FPS	D-FPS	Cls-aware	Ctr-aware	97.9%	97.7%	96.3%	79.54	58.49	71.33
D-FPS	**D-FPS**	**Ctr-aware**	**Ctr-aware**	**97.9%**	**98.4%**	**97.2%**	**79.57**	**58.91**	**71.24**

Table 1. The correlation between the instance recall ratio and the final detection performance.

Method	256p	1024p	4096p	16384p
D-FPS [3]	<0.1 ms	0.5 ms	2.8 ms	23.7 ms
Feat-FPS [11]	0.3 ms	0.7 ms	4.2 ms	40.6 ms
Cls/Ctr-aware	**0.2 ms**	**0.2 ms**	**0.3 ms**	**0.5 ms**

Method	256p	1024p	4096p	16384p
D-FPS [3]	<1 MB	<1 MB	<1 MB	<1 MB
Feat-FPS [11]	64 MB	104 MB	448 MB	6228 MB
Cls/Ctr-aware	**0.25 MB**	**1 MB**	**4 MB**	**17 MB**

Table 2. Time and memory consumption of sampling methods.

3. Additional Experimental Results

(1) Preserving more foreground points really benefits the final detection performance? As mentioned in Section 3.2, two instance-aware strategies are proposed to keep high instance recall while hierarchically downsampling the points. However, it remains unclear that whether the more foreground points really benefit the final detection performance. To this end, we further justify the motivation of our IA-SSD here. Specifically, we conduct several groups of experiments based on our framework with different sampling strategies. Note that, the network architecture and parameter settings are kept consistent. The quantitative detection results, accompanied with the instance recall ratio after the last downsampling layers by using different possible combinations of the sampling approaches are shown in Table 1.

From the results in Table 1 we can see that: (1) the instance recall ratio is positively correlated with the final detection performance, especially for small objects with a limited number of points such as *pedestrians* and *cyclists*. (2) The detection performance of *cars* is relatively robust to the variations of sampling strategies, primarily because that *car* usually has a sufficient number of foreground points remaining after downsampling, hence relatively easy to be detected. (3) Adopting the proposed instance-aware sampling strategies at the early encoding layers may negatively affect the final detection performance, primarily because of the insufficient semantic information in the early latent point features. (4) Deploying the proposed instance-aware downsampling strategies at the last two encoding layers can significantly improve the detection performance. Overall, this experiment further demonstrates that more foreground points are appealing for object detection task, especially for small but important objects.

	Method	Type	Car Mod (IoU=0.7)	Ped. Mod (IoU=0.5)	Cyc. Mod (IoU=0.5)
Voxel-based	SECOND [10]	1-stage	78.62	52.98	67.15
	PointPillars [1]	1-stage	77.28	52.29	62.28
	Part-A^2 [7]	2-stage	79.40	60.05	69.90
Point-Voxel	PV-RCNN [5]	2-stage	83.61	57.90	70.47
Point-based	PointRCNN [6]	2-stage	78.70	54.41	72.11
	3DSSD [11]	1-stage	79.06	10.49	16.93
	IA-SSD (Ours)	**1-stage**	**79.57**	**58.91**	**71.24**

Table 3. Performance comparison of different detectors based on the OpenPCDet library. Note that, all detectors are trained with multi-class objects together, and the results are achieved by using a single detection model.

Dataset	*Mem.*	*Paral.*	*Speed*$^{\perp}$	*Speed*$^{\top}$	*Input Scale*
Waymo [9]	626 MB	16	9†	14	81920
	433 MB	23	8†	20	65536
ONCE [2]	401 MB	25	11†	21	60k

Table 4. Efficiency of our IA-SSD on Waymo and ONCE Datasets. The number of input points to our framework is increased, considering the large-scale panoramic scenes compared with KITTI. Here "Mem." and "Paral." denote the GPU memory footprint per frame during inference and the maximum number of batches that can be parallelized on one RTX2080Ti (11GB). "Speed$^{\perp}$", "Speed$^{\top}$" is inference speed when processing one frame or full-loaded GPU memory. † We divide the whole scene into four parallel parts in the first sampling layer.

(2) Efficiency of Sampling. We further explore the efficiency of different sampling strategies, to have an intuitive idea of the advantages of our instance-aware sampling. Table 2 compares the time and memory consumption of different sampling strategies with a varying number of points. We can clearly see that the proposed instance-aware sampling has superior efficiency compared with the Feat-FPS [11], hence leading to a higher frame rate of our method during inference.

(3) Evaluation on KITTI validation set. We also report the detection results achieved by several representative approaches on the *validation* set of the KITTI Dataset in Table 3. Note that, all results achieved by baselines are reproduced based on the OpenPCDet[1]. In particular, all baselines are trained with multi-class objects in a single model for a fair comparison. It can be seen that our single-stage IA-SSD achieves superior detection performance compared

[1]https://github.com/open-mmlab/OpenPCDet

with other point-based baselines. We also noticed that the prior SoTA detector 3DSSD[2] achieve poor results on the class of pedestrian and cyclist, further demonstrating the advantages of our IA-SSD.

(4) Efficiency of our IA-SSD on large-scale LiDAR scenarios. To further verify the efficiency of our IA-SSD on large-scale 3D datasets, we further report the efficiency of our IA-SSD on the validation set of Waymo and ONCE datasets. As shown in Table 4, the proposed IA-SSD can still achieve satisfactory real-time performance in such complex panoramic scenes.

(5) Qualitative visualization of our instance-aware downsampling. To intuitively compare the performance of different sampling approaches, we qualitatively show the visualization of the downsampled point clouds achieved by different approaches in Figure 1. Clearly, the proposed instance-aware sampling can effectively preserve more foreground points (shown in red), especially for foreground points belonging to small and sparse instances (*e.g., pedestrian*), as well instances far away from the sensors.

(6) Visualization of the Contextual Centroid Perception. We also visualize the results produced by our contextual centroid perception module in Figure 2. It is clear that the downsampled point clouds at this stage are quite sparse and insufficient, which makes the centroid estimation and instance regression considerably difficult. Therefore, it is necessary to exploit the useful information around the instance, even outside the ground-truth bounding boxes. Thanks to the proposed contextual centroid perception module, our IA-SSD can even precept the objects with extremely indistinguishable geometry and limited points (shown in purple dotted circles). This further demonstrated the effectiveness of the proposed module.

(7) Additional qualitative detection results on the KITTI Dataset. We also show extra qualitative detection results achieved by our IA-SSD on the *validation* (Figure 3) and *test* (Figure 4) split of the KITTI Dataset. It can be seen that our IA-SSD can achieve satisfactory detection performance on this dataset, even for some challenging cases. It is also worth mentioning that the detection results of different objects are achieved by our IA-SSD in a single pass, instead of the common practice to train separate models for different objects.

(8) Additional qualitative detection results on the large-scale datasets. Here, we present extra qualitative detection results achieved by our IA-SSD on two large-scale datasets with challenging panoramic scenarios. Figure 5 and Figure 6 illustrate the detection results on the validation set of Waymo and ONCE Dataset respectively. It can be seen that our IA-SSD can also achieve promising detection performance in challenging and complex 3D scenes.

[2]https://github.com/qiqihaer/3DSSD-pytorch-openPCDet

4. Potential Negative Societal Impact

In this paper, we proposed an efficient point-based solution capable of achieving promising low-cost objects detection in autonomous driving scenarios. Our model is trained and evaluated totally based on open-sourced datasets, and there is no known potential negative impact on society.

5. Video Illustration

We provide a video demo illustrating the detection performance of our IA-SSD in 3D point clouds, which can be viewed at https://youtu.be/3jP2o9KXunA.

References

[1] 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. 2

[2] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Jie Yu, Chunjing Xu, et al. One million scenes for autonomous driving: Once dataset. 2021. 2

[3] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3D object detection in point clouds. In ICCV, pages 9277–9286, 2019. 2

[4] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In NeurIPS, pages 5099–5108, 2017. 1

[5] 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. 2

[6] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3D object proposal generation and detection from point cloud. In CVPR, pages 770–779, 2019. 2

[7] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network. IEEE TPAMI, 2020. 2

[8] 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, page 1100612, 2019. 1

[9] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In CVPR, pages 2446–2454, 2020. 2

[10] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. Sensors, page 3337, 2018. 2

[11] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3D single stage object detector. In CVPR, pages 11040–11048, 2020. 2

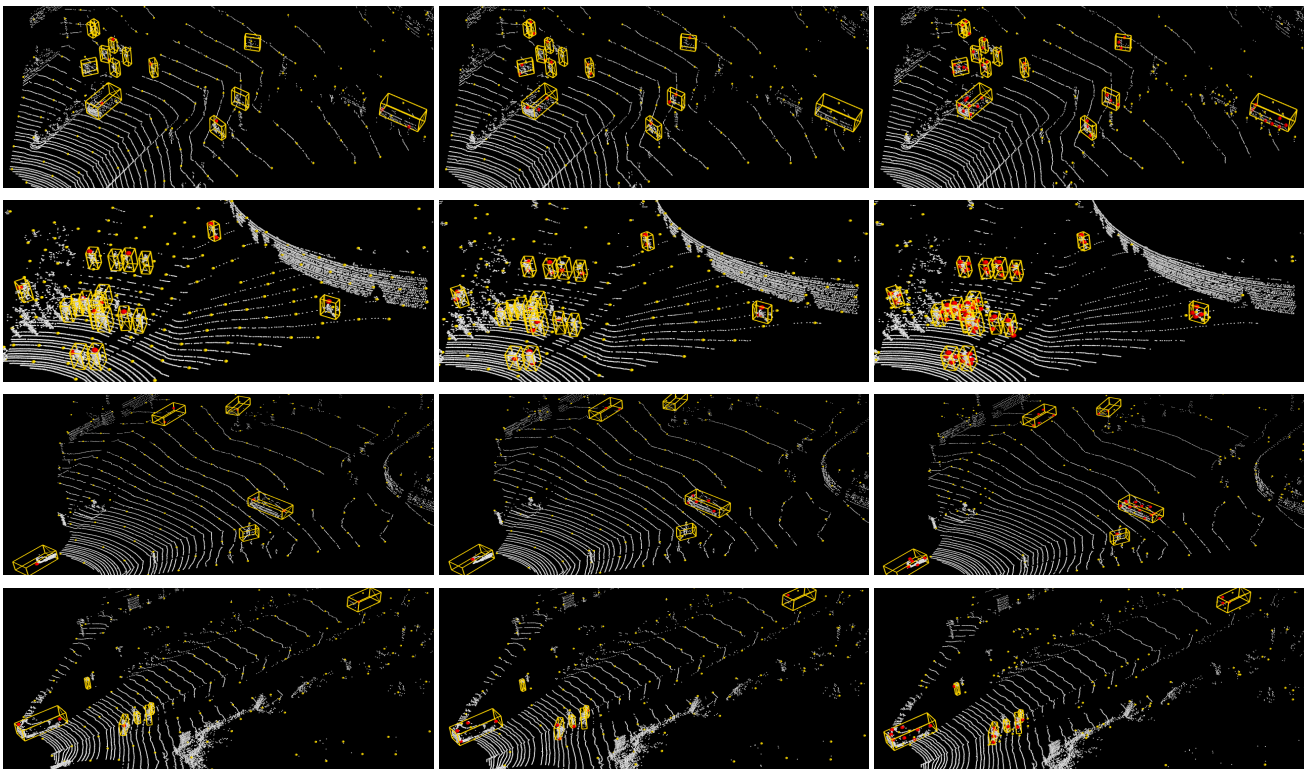

Figure 1. Qualitative visualization of the downsampled point clouds achieved by different sampling strategies (From left to right, D-FPS, F-FPS, and the proposed instance-aware sampling). Note that, the raw point clouds and representative points are colored in white and gold, respectively. Positive representative points are highlighted in red.

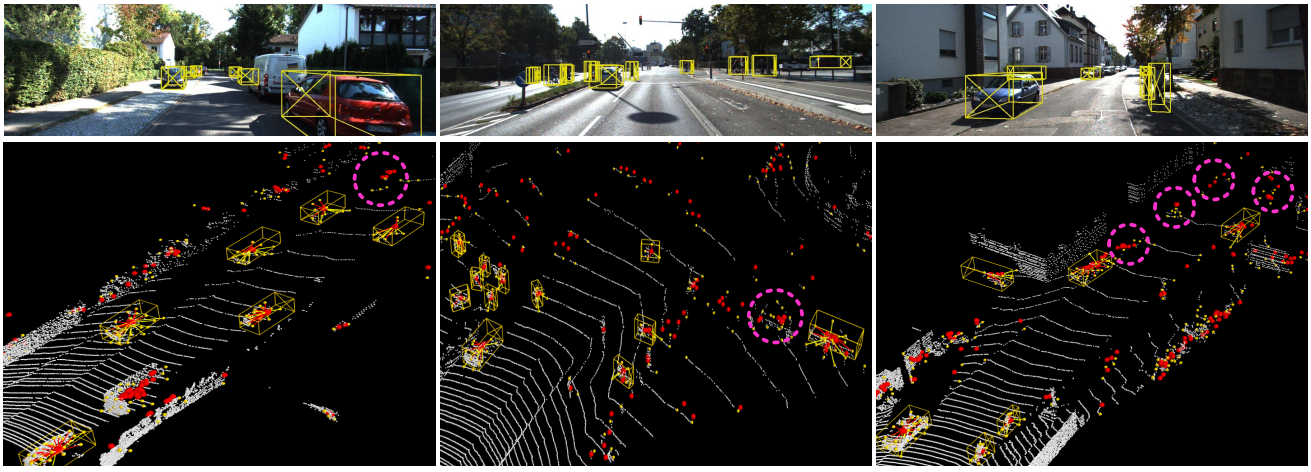

Figure 2. Visualization of the contextual centroid perception on the *validation* spit of the KITTI dataset. All representative points and predicted centroid are colored in gold and red, respectively. In particular, we also show the offsets of representative points inside/around the objects in red/gold. Best viewed in color.

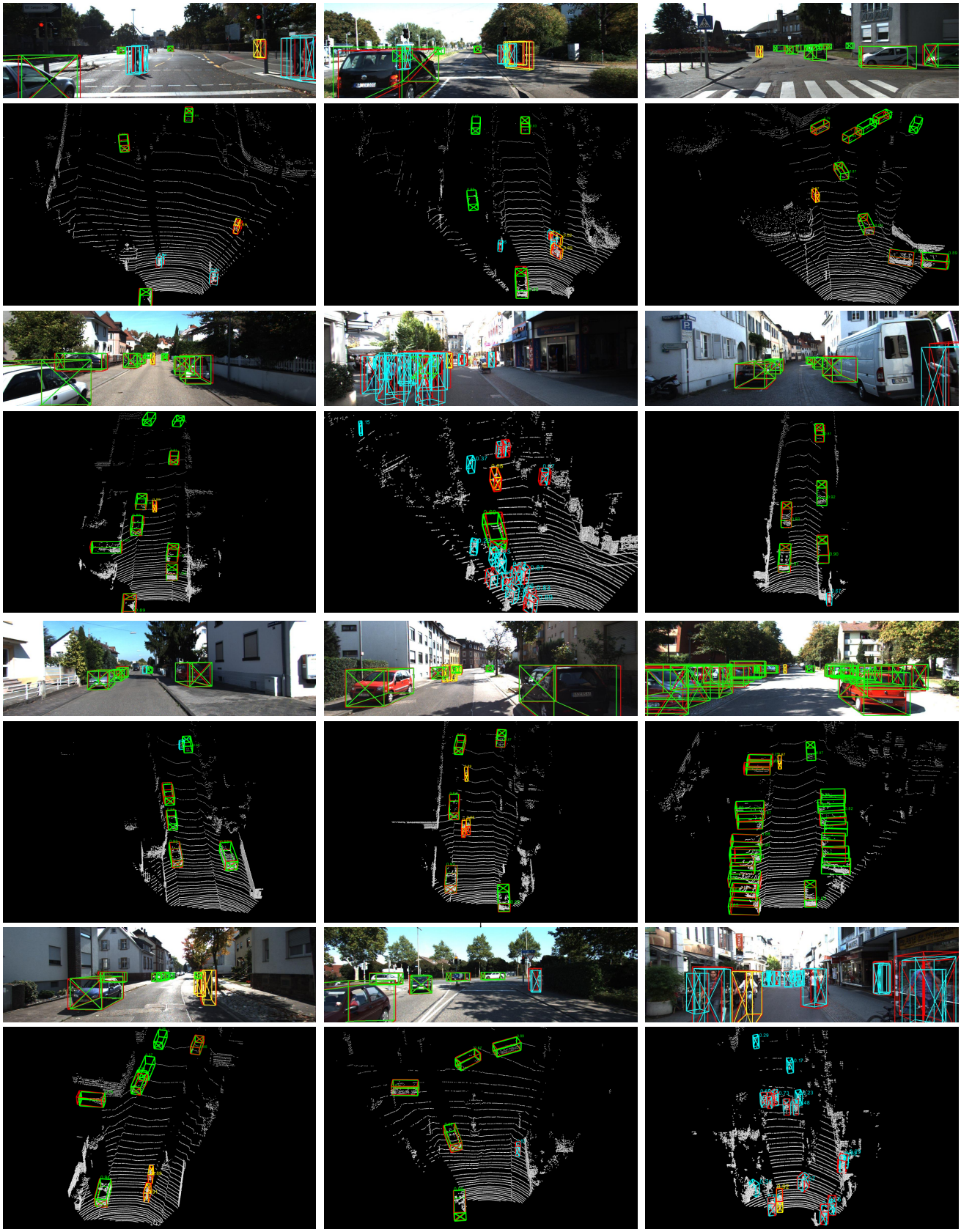

Figure 3. Extra qualitative results achieved by our IA-SSD on the *validation* set of the KITTI Dataset. We also show the corresponding projected 3D bounding boxes on images. Note that, the ground-truth bounding boxes are shown in red, and the predicted bounding boxes are shown in green for *car*, cyan for *pedestrian*, and yellow for *cyclist*. Best viewed in color.

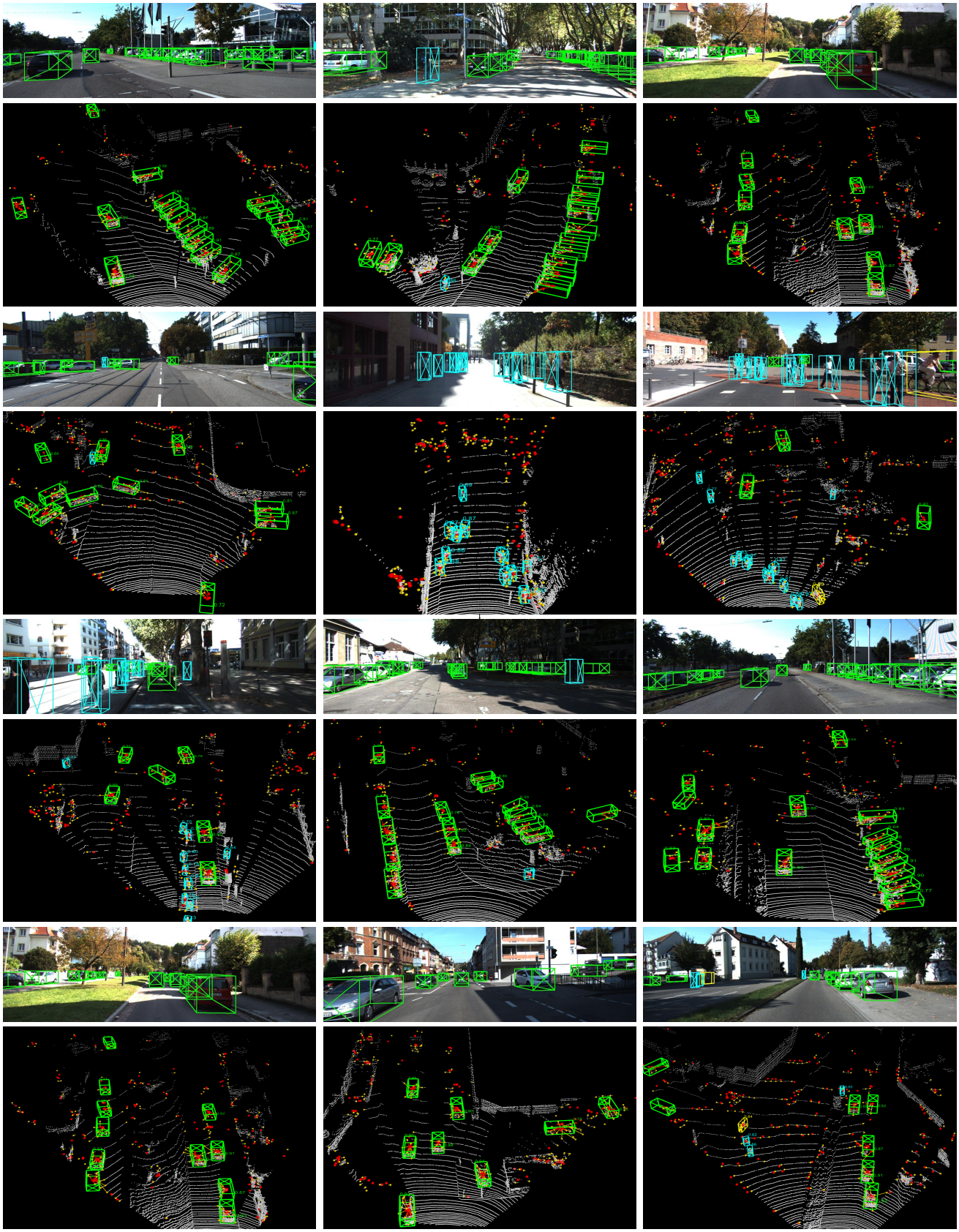

Figure 4. Extra qualitative results achieved by our IA-SSD on the *test* set of the KITTI Dataset. We also show the corresponding projected 3D bounding boxes on images. Note that, there is no ground-truth bounding boxes available, hence we only show the predicted bounding boxes in green for *car*, cyan for *pedestrian*, and yellow for *cyclist*. The centroid predictions are marked in red, while the 256 representative points are shown in gold. Best viewed in color.

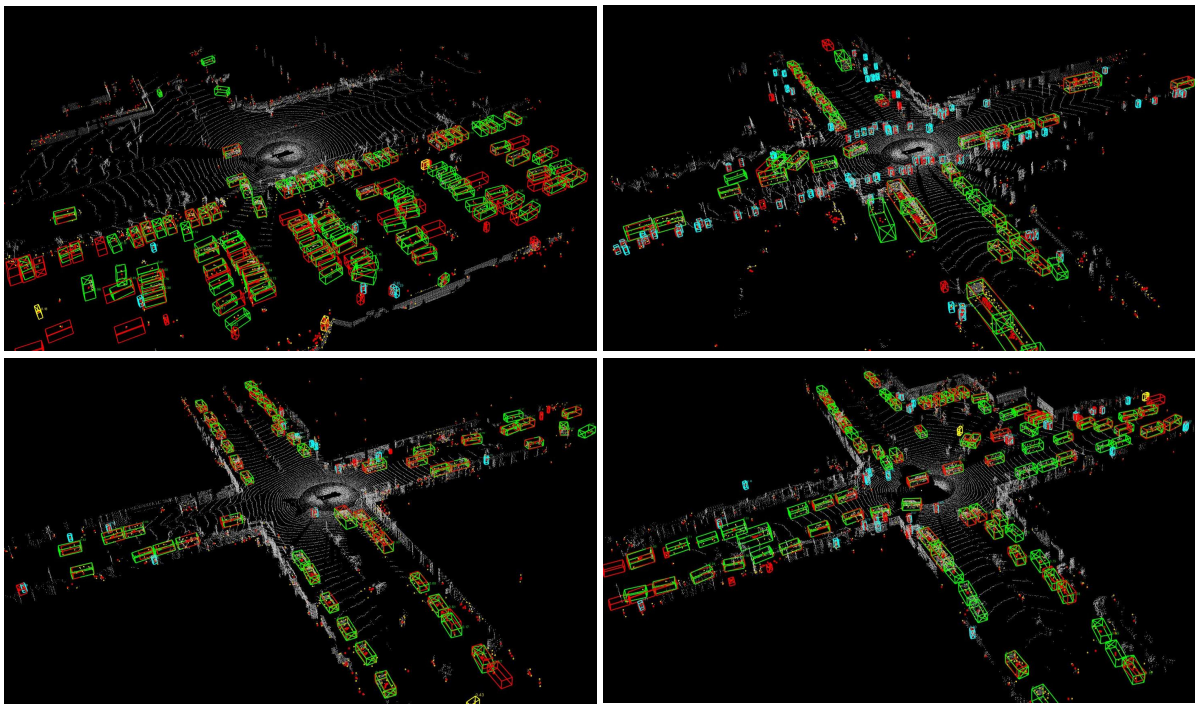

Figure 5. Extra qualitative results achieved by our IA-SSD on the *val* set of the Waymo Dataset. Here We demonstrate our detection results on some challenging scenes. Note that, the ground-truth bounding boxes are shown in red, and the predicted bounding boxes are shown in green for *vehicle*, cyan for *pedestrian*, and yellow for *cyclist*. Best viewed in color.

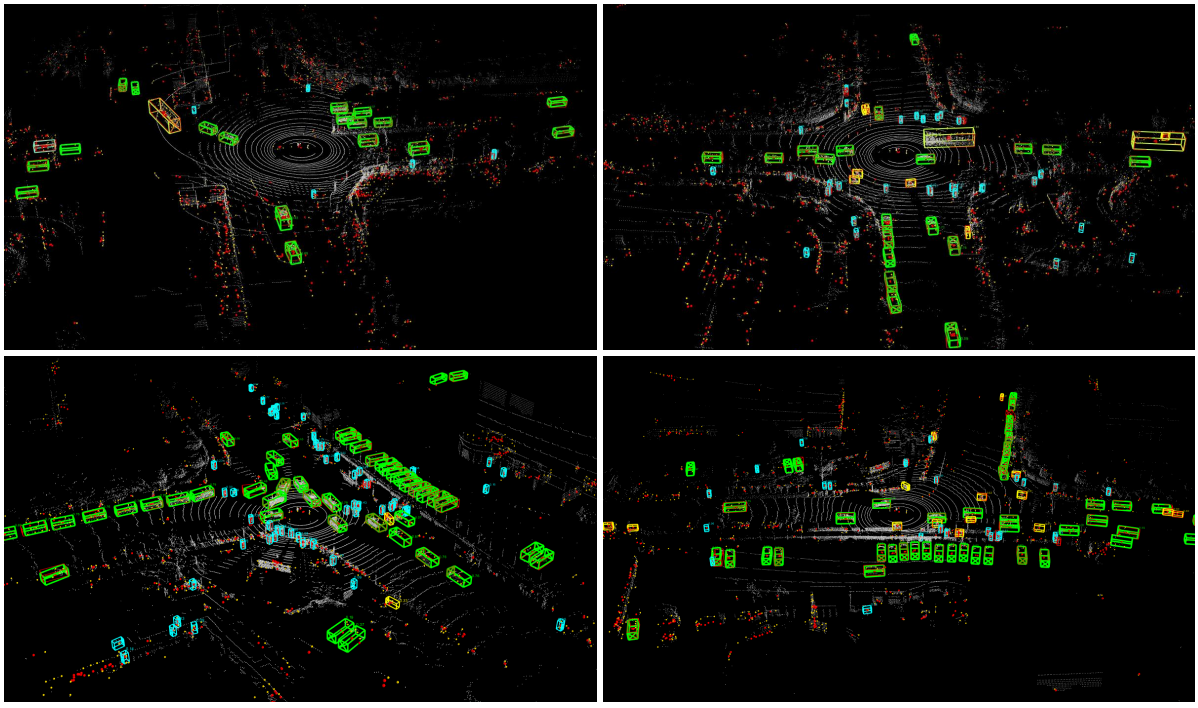

Figure 6. Extra qualitative results achieved by our IA-SSD on the *val* set of the ONCE Dataset. Here We demonstrate our detection results on some challenging scenes. Note that, the ground-truth bounding boxes are shown in red, and the predicted bounding boxes are shown in green for *vehicle*, cyan for *pedestrian*, and yellow for *cyclist*. Best viewed in color.