Supplementary Material:
End-to-End Pseudo-LiDAR for Image-Based 3D Object Detection

Rui Qian*[1,2] Divyansh Garg*[1] Yan Wang*[1] Yurong You*[1]
Serge Belongie[1,2] Bharath Hariharan[1] Mark Campbell[1] Kilian Q. Weinberger[1] Wei-Lun Chao[3]
[1] Cornell Univeristy [2] Cornell Tech [3] The Ohio State University
{rq49, dg595, yw763, yy785, sjb344, bh497, mc288, kqw4}@cornell.edu chao.209@osu.edu

We provide details and results omitted in the main text.

- section S1: results in pedestrians and cyclists (subsection 4.3 of the main paper).
- section S2: results at different depth ranges (subsection 4.3 of the main paper).
- section S3: KITTI testset results (subsection 4.3 of the main paper).
- section S4: additional qualitative results (subsection 4.5 of the main paper).
- section S5: gradient visualization (subsection 4.5 of the main paper).
- section S6: other results (subsection 4.6 of the main paper).

S1. Results on Pedestrians and Cyclists

In addition to 3D object detection on Car category, in Table S1 we show the results on Pedestrian and Cyclist categories in KITTI object detection validation set [2, 3]. To be consistent with the main paper, we apply P-RCNN [4] as the object detector. Our approach (E2E-PL) outperforms the baseline one without end-to-end training (PL++) [7] by a notable margin for image-based 3D detection.

Category	Model	Easy	Moderate	Hard
Pedestrian	PL++	31.9 / 26.5	25.2 / 21.3	21.0 / 18.1
	E2E-PL	**35.7 / 32.3**	**27.8 / 24.9**	**23.4 / 21.5**
Cyclist	PL++	36.7 / 33.5	23.9 / 22.5	22.7 / 20.8
	E2E-PL	**42.8 / 38.4**	**26.2 / 24.1**	**24.5 / 22.7**

Table S1: **Results on pedestrians and cyclists (KITTI validation set).** We report AP_{BEV} / AP_{3D} (in %) of the two categories at IoU=0.5, following existing works [4, 5]. PL++ denotes the PSEUDO-LIDAR ++ pipeline with images only (i.e., SDN alone) [5]. Both approaches use P-RCNN [4] as the object detector.

S2. Evaluation at Different Depth Ranges

We analyze 3D object detection of Car category for ground truths at different depth ranges (i.e., 0-30 or 30-70 meters). We report results with the point-cloud-based pipelines in Table S2 and the quantization-based pipeline in Table S3. E2E-PL achieves better performance at both depth ranges (except for 30-70 meters, moderate, AP_{3D}). Specifically, on AP_{BEV}, the relative gain between E2E-PL and the baseline becomes larger for the far-away range and the hard setting.

Range	Model	Easy	Moderate	Hard	#Objs
0-30	PL++	82.9 / 68.7	76.8 / 64.1	67.9 / 55.7	7379
	E2E-PL	**86.2 / 72.7**	**78.6 / 66.5**	**69.4 / 57.7**	
30-70	PL++	19.7 / 11.0	29.5 / 18.1	27.5 / 16.4	3583
	E2E-PL	**23.8 / 15.1**	**31.8 / 18.0**	**31.0 / 16.9**	

Table S2: **3D object detection via the point-cloud-based pipeline with P-RCNN at different depth ranges.** We report AP_{BEV} / AP_{3D} (in %) of the **car** category at IoU=0.7, using P-RCNN for detection. In the last column we show the number of car objects in KITTI object validation set within different ranges.

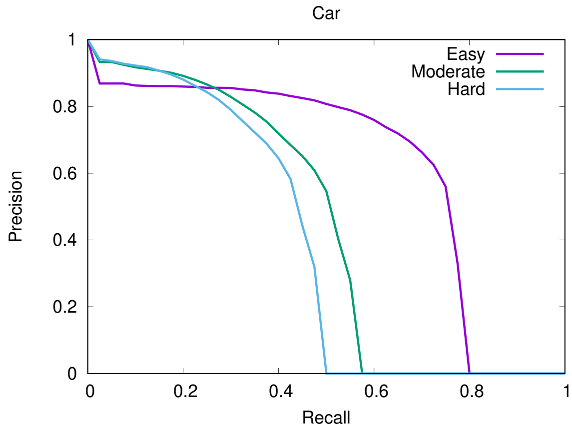
Range	Model	Easy	Moderate	Hard	#Objs
0-30	PL++	81.4 / -	75.5 / -	65.8 / -	7379
	E2E-PL	**82.1 / -**	**76.4 / -**	**67.5 / -**	
30-70	PL++	26.1 / -	23.9 / -	20.5 / -	3583
	E2E-PL	**26.8 / -**	**36.1 / -**	**31.7 / -**	

Table S3: **3D object detection via the quantization-based pipeline with PIXOR* at different depth ranges.** The setup is the same as in S2, except that PIXOR* does not have height prediction and therefore no AP_{3D} is reported.

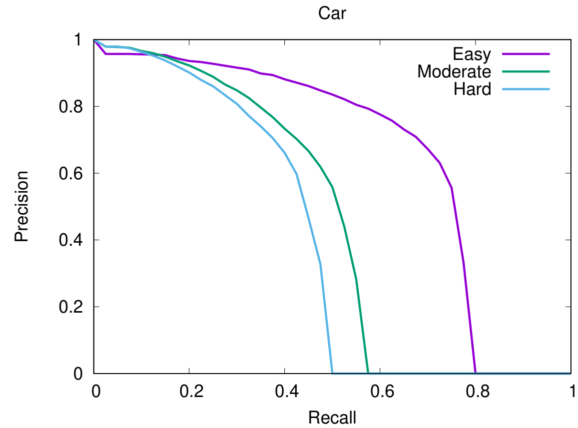
S3. On KITTI Test Set

In Figure S1, we compare the precision-recall curves of our E2E-PL and PSEUDO-LIDAR ++ (named Pseudo-LiDAR V2 on the leaderboard). On the 3D object detection

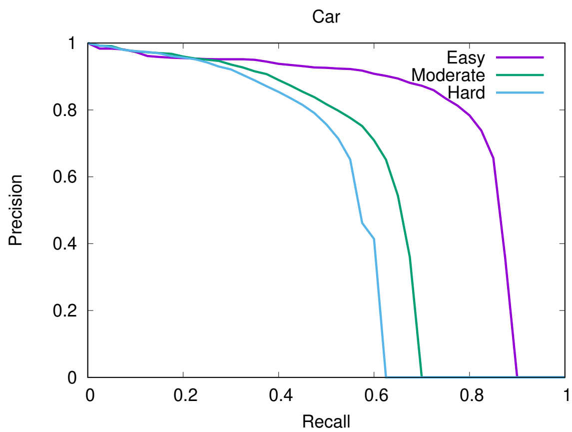
* Equal contributions



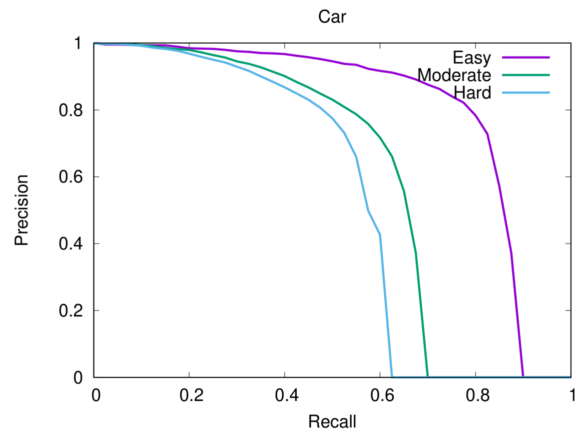
(a) PL++ on 3D Track (AP_{3D})



(b) E2E-PL on 3D Track (AP_{3D})



(c) PL++ on BEV Track (AP_{BEV})



(d) E2E-PL on BEV Track (AP_{BEV})

Figure S1: **Precision-recall curves on KITTI test dataset.** We here compare E2E-PL with PL++ on the 3D object detection track and bird's eye view detection track.

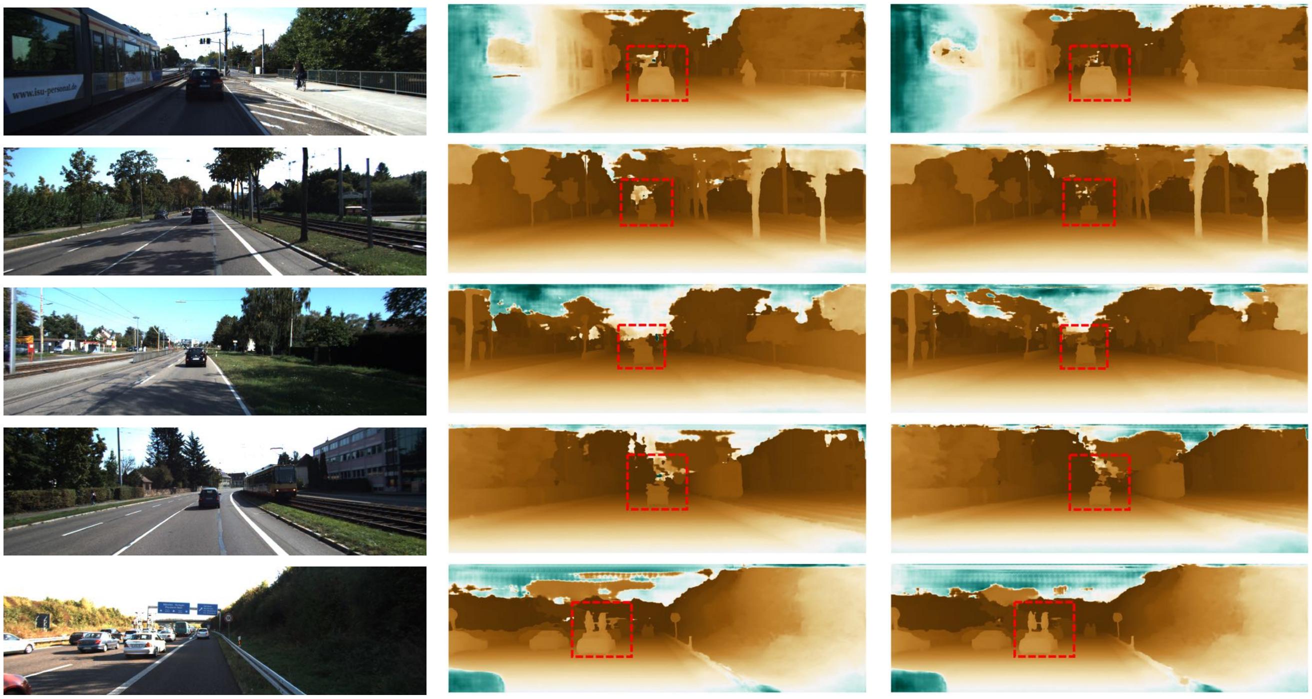
track (first row of Figure S1), PSEUDO-LIDAR ++ has a notable drop of precision on easy cars even at low recalls, meaning that PSEUDO-LIDAR ++ has many high-confident false positive predictions. The same situation happens to moderate and hard cars. Our E2E-PL suppresses the false positive predictions, resulting in more smoother precision-recall curves. On the bird's-eye view detection track (second row of Figure S1), the precision of E2E-PL is over 97% within recall interval 0.0 to 0.2, which is higher than the precision of PSEUDO-LIDAR ++, indicating that E2E-PL has fewer false positives.

S4. Additional Qualitative Results

We show more qualitative depth comparisons in Figure S2. We use red bounding boxes to highlight the depth improvement in car related areas. We also show detection comparisons in Figure S3, where our E2E-PL has fewer false positive and negative predictions.

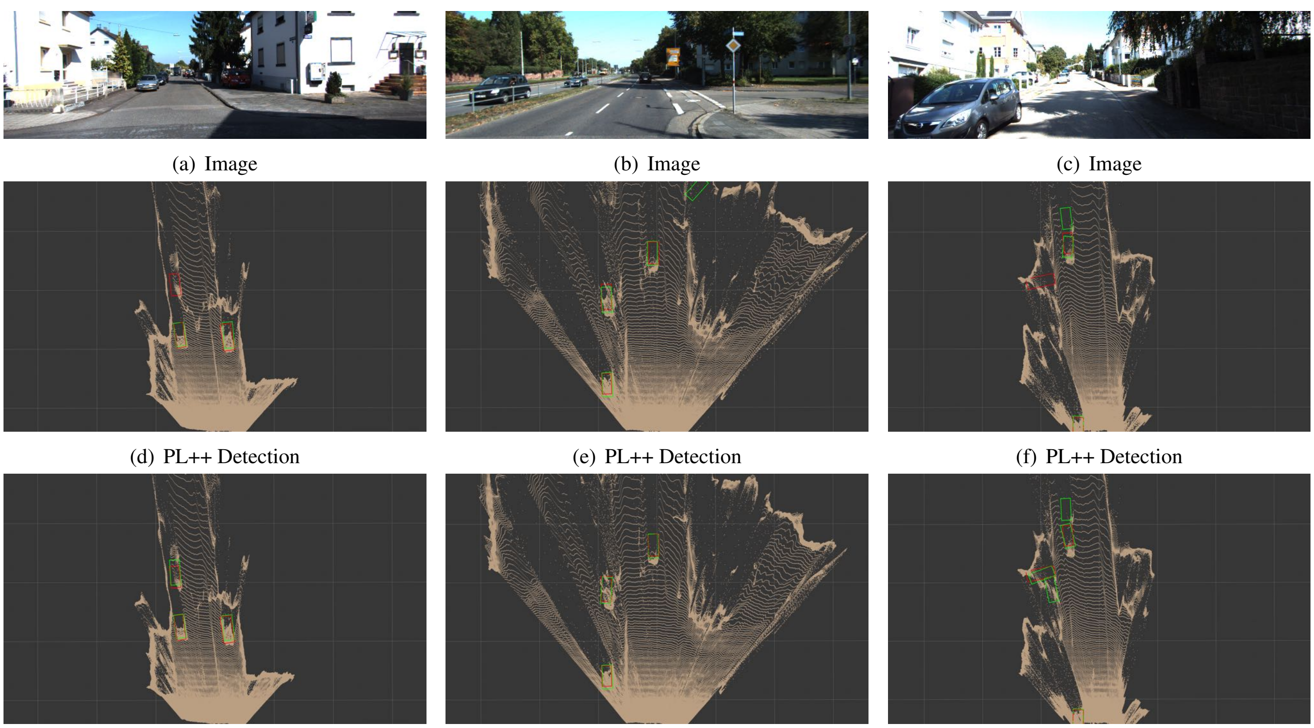
S5. Gradient Visualization on Depth Maps

We also visualize the gradients of the detection loss with respect to the depth map to indicate the effectiveness of our E2E-PL pipeline, as illustrated in Figure S4. We use JET Colormap to indicate the relative absolute value of gradients, where red color indicates higher values while blue color indicates lower values. The gradients from the detector focus heavily around cars.



(a) Input (b) Depth estimation of PL++ (c) Depth estimation of E2E-PL

Figure S2: **Qualitative comparison of depth estimation.** We here compare PL++ with E2E-PL.

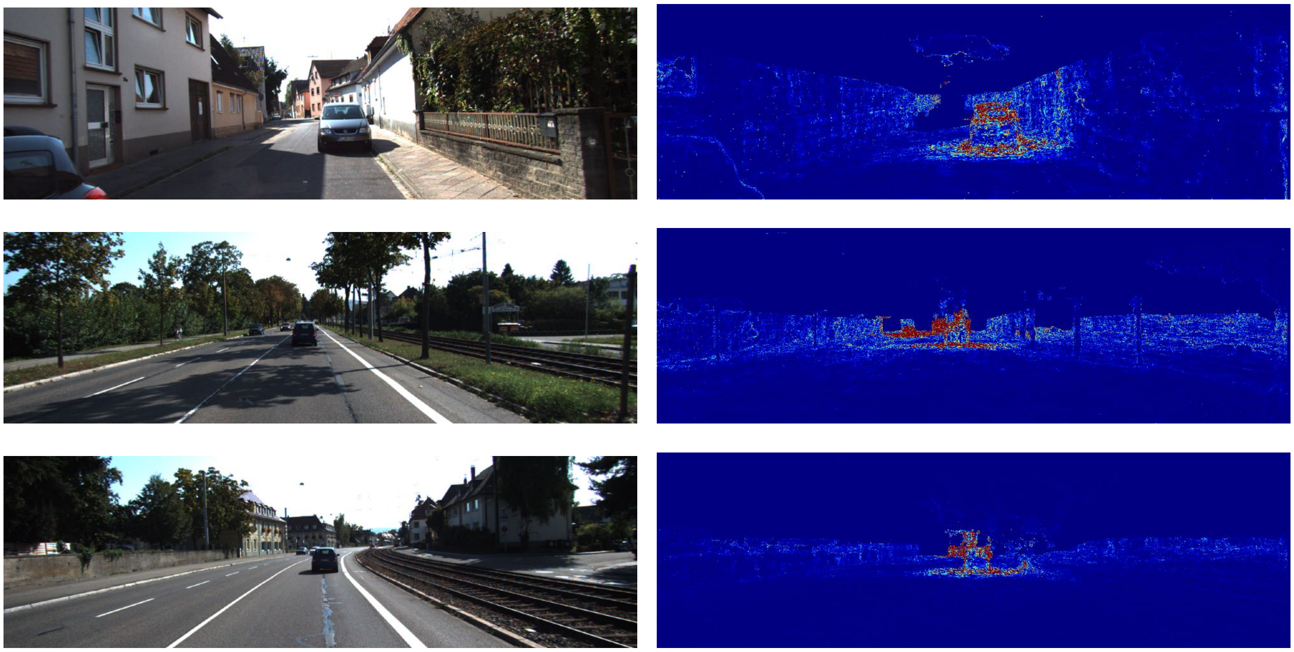


(a) Image (b) Image (c) Image

(d) PL++ Detection (e) PL++ Detection (f) PL++ Detection

(g) E2E-PL Detection (h) E2E-PL Detection (i) E2E-PL Detection

Figure S3: **Qualitative comparison of detection results.** We here compare PL++ with E2E-PL. The red bounding boxes are ground truth and the green bounding boxes are predictions.



(a) Image (b) Gradient from detector

Figure S4: **Visualization of absolute gradient values by the detection loss.** We use JET colormap to indicate the relative absolute values of resulting gradients, where red color indicates larger values; blue, otherwise.

S6. Other results

S6.1. Depth estimation

We summarize the quantitative results of depth estimation (w/o or w/ end-to-end training) in Table S4. As the detection loss only provides semantic information to the foreground objects, which occupy merely 10% of pixels (Figure 2), its improvement to the overall depth estimation is limited. But for pixels around the objects, we do see improvement at certain depth ranges. We hypothesize that the detection loss may not directly improve the metric depth, but will sharpen the object boundaries in 3D to facilitate object detection and localization.

Range(meters)	Model	Mean error	Std
0-10	PL++	0.728	2.485
	E2E-PL	**0.728**	**2.435**
10-20	PL++	1.000	3.113
	E2E-PL	**0.984**	**2.926**
20-30	PL++	2.318	4.885
	E2E-PL	**2.259**	**4.679**

Table S4: Quantitative results on depth estimation.

S6.2. Argoverse dataset [1]

We also experiment with Argoverse [1]. We convert the Argoverse dataset into KITTI format, following the original split, which results in $6,572$ and $2,511$ scenes (*i.e.*, stereo images with the corresponding synchronized LiDAR point clouds) for training and validation. We use the same training scheme and hyperparameters as those in KITTI experiments, and report the validation results in Table S5. We define the easy, moderate, and hard settings following [6]. Note that since the synchronization rate of stereo images in Argoverse is 5Hz instead of 10Hz, the dataset used here is smaller than that used in [6]. We note that the sensor calibration in Argoverse may not register the stereo images into the perfect epipolar correspondence (as indicated in argoverse-api). Our experimental results in Table S5 also confirmed the issue: image-based results are much worse than the LiDAR-based ones. Nevertheless, our E2E-PL pipeline still outperforms PL++. We note that, most existing image-based detectors only report results on KITTI.

References

[1] Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argo-

Method	Input	IoU=0.5			IoU=0.7		
		Easy	Moderate	Hard	Easy	Moderate	Hard
PL++: P-RCNN	S	68.7 / 55.6	46.3 / 36.6	43.5 / 35.1	17.2 / 6.9	17.0 / 11.1	17.0 / 11.6
E2E-PL: P-RCNN	S	**73.6 / 61.3**	**47.9 / 39.1**	**44.6 / 35.7**	**30.2 / 16.1**	**18.8 / 11.3**	**17.9** / 11.5
P-RCNN	L	93.2 / 89.7	85.1 / 79.4	84.5 / 76.8	73.8 / 42.3	66.5 / 34.6	63.7 / 37.4

Table S5: **3D object detection via the point-cloud-based pipeline with P-RCNN on Argoverse dataset.** We report AP_{BEV} / AP_{3D} (in %) of the **car** category, using P-RCNN for detection. We arrange methods according to the input signals: S for stereo images, L for 64-beam LiDAR. PL stands for PSEUDO-LIDAR. *Results of our end-to-end* PSEUDO-LIDAR *are in blue*. Methods with 64-beam LiDAR are in gray. Best viewed in color.

verse: 3d tracking and forecasting with rich maps. In *CVPR*, 2019. 4

[2] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1

[3] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 1

[4] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019. 1

[5] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *CVPR*, 2019. 1

[6] Yan Wang, Xiangyu Chen, Yurong You, Erran Li Li, Bharath Hariharan, Mark Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. Train in germany, test in the usa: Making 3d object detectors generalize. In *CVPR*, 2020. 4

[7] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. In *ICLR*, 2020. 1