Supplementary Material for
Learning to Detect Mobile Objects from LiDAR Scans Without Labels

S1. Implementation details

We set $[-H_s, H_e]$ to $[0, 70]$ m since we experiment with frontal-view detection only. We combine only one scan into the dense point cloud $\boldsymbol{S}_c^t$ every 2 m within this range. In calculating PP score, we use as many traversals as possible (≥ 2) and set $r = 0.3$m. For clustering, we use $K = 70$ and $r' = 2.0$m in the graph, and $\epsilon = 0.1$, min_samples $= 10$ for DBSCAN. For filtering, we use a loose threshold of $\alpha = 20$ percentile and $\gamma = 0.7$. Other common sense properties are simply implemented as follows:

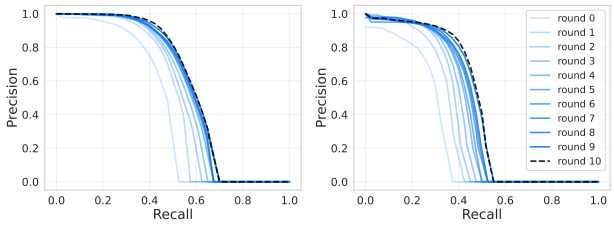
- # points in the cluster >= 10;
- Volume of fitted bounding boxes $\in [0.5, 120] m^3$;
- The height (upright distance against the ground plane) of points $\text{Height}_{\text{max}} > 0.5m$ and $\text{Height}_{\text{min}} < 1.0m$ to ensure clusters not floating in the air or beneath the ground due to errors in LiDAR.

We did not tune these parameters except qualitatively checked the fitted bounding boxes in few scenes in the Lyft "train" set. For detection models, we use the default hyper-parameters tuned on KITTI[1] with few exceptions listed in the paper. We will open-source the code upon acceptance.

S2. Experiments with other detectors

Besides the PointRCNN detector [49], We experiment with two other detectors PointPillars [31] and VoxelNet (SECOND) [60, 70], and show their results in Table S2 and Table S1. We apply the default hyper-parameters of these two models tuned on KITTI, and apply the same procedure as that on PointRCNN models. Note that PointPillars and VoxelNet model need a pre-defined anchor size for different types of objects, which we picked (length, width, height) as $(2.0, 1.0, 1.7)$ m without tuning. We observe that generally the PointPillars and VoxelNet yield worse results than PointRCNN models (possibly due to the fixed anchor size for all mobile objects), but we still observe significant gains from self-training.

[1] https://github.com/open-mmlab/OpenPCDet/tree/master/tools/cfgs/kitti_models



(a) IoU=0.25 (b) IoU=0.5

Figure S1. **Number of self-training rounds *vs*. precision-recall curves.** We show the precision-recall curves under AP_{BEV} with IoU=0.5 and IoU=0.25 for mobile objects in 0-80 m on Lyft test set from models trained with different rounds of self-training.

S3. Detailed evaluation by object types

In Table S3, we include detailed evaluations (BEV IoU= 0.5, BEV IoU= 0.25 and by different depth ranges) of the recall of different object types in the Lyft test set. This corresponds to Table 5 in the main paper.

S4. Corresponding IoU=0.5 results

We list the IoU=0.5 counterparts of Tables 1 to 6 in Tables S4 to S8.

S5. Precision-recall evaluation

In Figure S1, we show how PR curve changes with different rounds of self-training: the max recall improves gradually while keeping high precision. This aligns with the expanded recall of the training set described above, and with what we observe qualitatively in Figure 1.

S6. More qualitative results

We show visualizations for additional qualitative results in Figure S2 for 5 additional LiDAR scenes. Visualizations show the progression of MODEST from seed generation, to detector trained on seed label set, to detector after 10 rounds of self training, and finally the ground truth bounding boxes. Observe that the detections obtain higher recall and learns a correct prior over object shapes as the method progresses.

Table S1. **Detection performance with PointPillars [31] on the Lyft dataset.** Please refer to Table 1 for naming.

Method	AP_{BEV}/ AP_{3D} @ IoU = 0.25				AP_{BEV}/ AP_{3D} @ IoU = 0.5			
	0-30	30-50	50-80	0-80	0-30	30-50	50-80	0-80
MODEST (R0)	56.3 / 51.3	26.6 / 19.5	5.4 / 3.0	30.4 / 24.6	32.1 / 25.2	10.0 / 4.2	1.2 / 0.2	13.8 / 8.5
MODEST (R10)	55.7 / 49.1	43.1 / 38.4	8.8 / 7.5	33.9 / 29.9	37.4 / 27.7	28.8 / 10.0	5.0 / 1.1	22.1 / 10.7
Sup. (Lyft)	78.7 / 77.9	64.6 / 63.7	45.4 / 44.1	64.7 / 63.6	72.9 / 68.9	55.5 / 50.3	41.5 / 35.4	58.0 / 52.8

Table S2. **Detection performance with VoxelNet (SECOND) [60, 70] on the Lyft dataset.** Please refer to Table 1 for naming.

Method	AP_{BEV}/ AP_{3D} @ IoU = 0.25				AP_{BEV}/ AP_{3D} @ IoU = 0.5			
	0-30	30-50	50-80	0-80	0-30	30-50	50-80	0-80
MODEST (R0)	54.3 / 49.7	27.8 / 21.4	4.9 / 2.8	30.2 / 24.8	30.3 / 24.9	11.7 / 5.1	1.1 / 0.2	14.0 / 8.8
MODEST (R10)	54.9 / 44.8	38.7 / 31.5	8.3 / 6.2	32.5 / 26.0	32.1 / 24.3	20.0 / 7.7	3.4 / 0.8	17.0 / 8.9
Sup. (Lyft)	81.6 / 81.1	67.8 / 66.3	45.5 / 44.6	65.9 / 64.9	76.7 / 73.9	59.7 / 55.3	41.8 / 36.3	60.1 / 55.4

Table S3. **Max recall with different methods on the Lyft dataset.** Please refer to Table 1 for naming. This corresponds to the counterpart Table 5 in the main paper.

(a) Recall @ IoU=0.5

Method	Car				Truck				Pedestrian				Cyclist			
	0-30	30-50	50-80	0-80	0-30	30-50	50-80	0-80	0-30	30-50	50-80	0-80	0-30	30-50	50-80	0-80
MODEST-PP (R0)	57.6	27.3	3.0	30.1	36.1	5.0	0.2	9.1	0.9	1.1	0.8	1.0	14.4	10.7	2.1	10.7
MODEST-PP (R10)	63.2	49.1	8.0	40.9	39.0	21.4	5.6	20.1	0.0	0.0	0.2	0.1	8.1	11.5	2.9	8.2
MODEST (R0)	63.7	45.5	11.2	41.0	29.8	18.3	2.4	17.2	5.2	0.6	0.0	1.7	34.1	11.7	1.2	20.2
MODEST (R10)	67.5	70.7	40.9	60.6	35.1	28.2	13.3	25.3	35.1	35.0	7.6	27.5	62.9	41.1	7.1	44.7
MODEST (R40)	69.9	78.8	68.9	72.9	25.4	27.4	20.1	28.0	39.4	38.8	6.2	28.6	70.6	32.5	1.5	44.4
Sup. (KITTI)	82.1	76.3	53.3	71.2	45.9	23.8	23.3	30.0	56.9	32.1	2.5	29.7	59.5	16.1	1.2	33.8
Sup. (Lyft)	85.7	82.5	75.2	81.5	64.9	52.0	50.3	54.7	60.8	55.4	18.9	45.2	71.0	43.2	6.8	49.2

(b) Recall @ IoU=0.25

Method	Car				Truck				Pedestrian				Cyclist			
	0-30	30-50	50-80	0-80	0-30	30-50	50-80	0-80	0-30	30-50	50-80	0-80	0-30	30-50	50-80	0-80
MODEST-PP (R0)	65.5	53.0	8.8	43.7	52.2	19.1	1.9	19.5	2.8	6.1	4.9	4.9	33.9	31.7	8.3	28.1
MODEST-PP (R10)	73.0	66.2	14.0	52.4	62.9	36.0	9.2	35.1	0.9	0.5	0.2	0.5	15.5	15.3	4.1	13.2
MODEST (R0)	73.9	63.6	22.9	54.6	46.8	30.5	12.2	31.8	22.5	8.6	1.5	10.2	75.3	42.3	4.4	50.5
MODEST (R10)	79.0	80.8	55.1	72.6	59.5	43.1	33.6	46.8	61.7	51.4	14.6	42.5	71.8	66.0	26.0	60.7
MODEST (R40)	81.2	83.1	78.1	81.4	53.2	43.3	31.7	46.4	63.9	51.8	11.0	42.1	81.6	62.0	19.8	62.9
Sup. (KITTI)	82.8	78.3	57.5	73.6	71.7	42.3	33.4	49.4	66.0	35.0	2.8	33.5	84.1	45.3	9.7	56.7
Sup. (Lyft)	86.9	84.1	78.5	83.6	73.7	66.3	58.0	65.4	72.3	63.7	25.2	53.8	83.1	68.1	29.2	67.5

Table S4. **Detection performance with different methods on the Lyft dataset.** We report AP_{BEV}/ AP_{3D} with IoU=0.5 for mobile objects under various ranges. Please refer to Table 1 for naming. This corresponds to the counterpart Table 1 in the main paper.

Method	AP_{BEV}/ AP_{3D} @ IoU = 0.25			
	0-30	30-50	50-80	0-80
MODEST-PP (R0)	34.1 / 31.3	5.1 / 3.0	0.0 / 0.0	12.0 / 9.7
MODEST-PP (R10)	42.1 / 38.3	21.9 / 19.2	1.0 / 0.9	22.8 / 20.6
MODEST (R0)	44.9 / 40.4	24.5 / 14.8	2.7 / 0.7	26.3 / 19.8
MODEST (R10)	56.8 / 51.3	51.4 / 40.5	19.2 / 9.0	44.1 / 35.5
MODEST (R40)	61.1 / 56.2	57.5 / 53.4	41.2 / 29.8	54.1 / 47.6
Sup. (KITTI)	72.3 / 69.5	53.2 / 48.1	27.9 / 20.5	53.1 / 48.1
Sup. (Lyft)	79.6 / 77.5	66.4 / 64.4	47.8 / 43.8	65.5 / 63.2

Table S5. **Detection results on the nuScenes Dataset.** We report AP_{BEV} / AP_{3D} at IoU=0.5 for mobile objects under various ranges. Please refer to Table 1 for naming. This corresponds to the counterpart Table 2 in the main paper.

Method	AP_{BEV} / AP_{3D} @IoU = 0.5			
	0-30	30-50	50-80	0-80
MODEST-PP(R0)	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
MODEST-PP(R10)	-	-	-	-
MODEST (R0)	8.4 / 2.9	0.3 / 0.1	0.1 / 0.0	3.0 / 0.9
MODEST (R10)	11.0 / 7.6	0.4 / 0.0	0.0 / 0.0	3.9 / 2.2
Sup. (nuScenes)	29.5 / 26.3	8.4 / 6.1	2.4 / 1.1	15.5 / 13.3

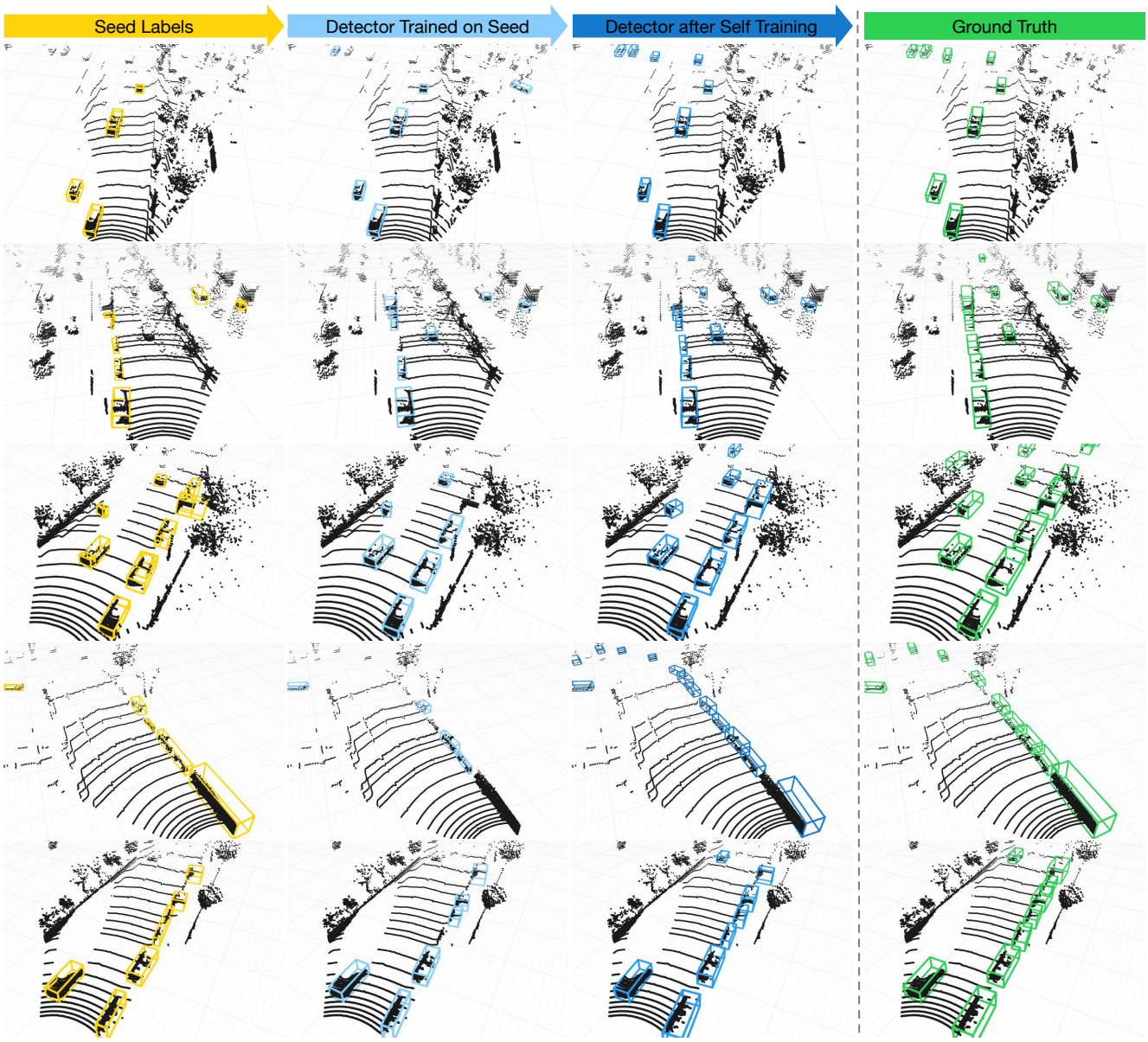


Figure S2. **Visualizations of MODEST outputs.** We show additional visualizations of LiDAR scans from three scenes in the Lyft dataset. From left to right: seed labels, detections trained on seed, detections after self training, and ground truth bounding boxes.

Table S6. **Detection performance on the KITTI validation set with models trained on the Lyft dataset.** We report AP_{BEV} / AP_{3D} with IoU=0.5 for mobile objects under various ranges. Please refer to Table 1 for naming. This corresponds to the counterpart Table 3 in the main paper.

Method	AP_{BEV} / AP_{3D} @ IoU = 0.5			
	0-30	30-50	50-80	0-80
MODEST-PP (R10)	50.5 / 48.5	10.3 / 8.7	0.2 / 0.1	35.6 / 33.8
MODEST (R10)	62.1 / 57.6	41.7 / 32.3	5.3 / 2.0	51.1 / 46.0
MODEST (R40)	57.7 / 52.7	44.1 / 40.1	11.6 / 7.2	49.3 / 44.6
Sup. (Lyft)	79.0 / 76.7	47.7 / 42.8	19.0 / 12.4	65.3 / 62.5
Sup. (KITTI)	85.4 / 83.3	69.5 / 66.9	41.2 / 35.0	78.3 / 76.0

Table S7. **The precision and recall of the "labels" on the Lyft dataset "train" split.** We report the ***precision / recall*** rate with BEV IoU=0.5 for mobile objects under various ranges. Please refer to Table 1 for naming. This corresponds to the counterpart Table 4 in the main paper.

Method	Precision / Recall @ IoU = 0.5			
	0-30	30-50	50-80	0-80
MODEST-PP (seed)	44.4 / 50.3	8.5 / 16.1	2.3 / 2.6	16.3 / 22.7
MODEST (seed)	55.8 / 43.5	28.8 / 19.3	15.6 / 4.4	38.9 / 22.2
MODEST (R0)	79.4 / 55.9	51.3 / 34.0	30.0 / 9.6	59.3 / 33.0
MODEST (R10)	82.2 / 63.4	65.9 / 55.0	38.7 / 28.9	62.9 / 49.3
MODEST (R40)	83.1 / 66.2	77.6 / 67.0	69.0 / 55.3	77.2 / 63.3

Table S8. **Common sense vs self-training.** We report AP_{BEV} / AP_{3D} with IoU=0.5 for mobile objects under various ranges. Seed and ST column mean how much data are used as seed data and self-training data respectively; FT stands for filtering by PP score during self-training. This corresponds to the counterpart Table 6 in the main paper.

Combinations			AP_{BEV} / AP_{3D} @ IoU = 0.5			
Seed	ST	FT	0-30	30-50	50-80	0-80
5%	5%		43.9 / 40.3	35.6 / 30.8	7.8 / 5.1	30.2 / 26.2
5%	5%	✓	47.7 / 40.0	40.8 / 37.7	10.1 / 8.6	33.9 / 29.8
5%	100%		55.2 / 51.1	46.4 / 38.0	13.8 / 8.4	40.2 / 34.2
100%	5%		44.6 / 40.4	38.7 / 33.0	16.3 / 7.2	34.4 / 28.0
100%	5%	✓	48.5 / 43.0	43.4 / 35.6	18.2 / 8.3	38.2 / 30.2
100%	100%		57.8 / 52.2	46.4 / 38.4	14.4 / 8.6	41.4 / 34.3
100%	100%	✓	56.8 / 51.3	51.4 / 40.5	19.2 / 9.0	44.1 / 35.5