Supplementary Material:
Train in Germany, Test in The USA: Making 3D Object Detectors Generalize

Yan Wang*[1] Xiangyu Chen*[1] Yurong You[1] Li Erran Li[2,3]
Bharath Hariharan[1] Mark Campbell[1] Kilian Q. Weinberger[1] Wei-Lun Chao[4]
[1]Cornell University [2]Scale AI [3]Columbia University [4]The Ohio State University
{yw763, xc429, yy785, bh497, mc288, kqw4}@cornell.edu erranlli@gmail.com chao.209@osu.edu

In this Supplementary Material, we provide details omitted in the main paper.

- section S1: data format conversion (section 3 of the main paper).
- section S2: evaluation metric (section 4.1 of the main paper).
- section S3: additional results on dataset discrepancy (section 4.4 and 4.5 of the main paper).
- section S4: object detection using PIXOR [9] (section 4.2 and 4.3 of the main paper).
- section S5: object detection using POINTRCNN with different adaptation methods (section 4.5 and 5 of the main paper).
- section S6: additional qualitative results (section 5 of the main paper).

S1. Converting Datasets into KITTI Format

In this section we describe in detail how we convert Argoverse [3], nuScenes [2], Lyft [6], and Waymo [1] into KITTI [4, 5] format. As the formatting of images, point clouds and camera calibration information is trivial, and label fields such as $alpha$ and $rotation_y$ have been well-defined, we only discuss the labeling process with non-deterministic definitions.

S1.1. Object filtering

Due to the fact that KITTI focuses on objects that appear in the camera view, we follow its setting and discard all object annotations outside the frontal camera view. To allow truncated objects, we project the 8 corners of each object's 3D bounding box onto the image plane. An object will be discarded if all its 8 corners fall out of the image boundary. To make other datasets consistent with KITTI, we do not consider labeled objects farther than 70 meters.

* Equal contributions

Table S1: The original categories in each dataset that we include into the *car* and *truck* categories following the KITTI label formulation.

Dataset	Car	Truck
Argoverse	{VEHICLE}	{LARGE_VEHICLE, BUS, TRAILER, SCHOOL_BUS}
nuScenes	{car}	{bus, trailer, construction_vehicle, truck}
Lyft	{Car}	{other_vehicle, truck, bus, emergency_vehicle}
Waymo	{Car}	∅

S1.2. Matching categories with KITTI

Since the taxonomy of object categories among datasets are misaligned, it is necessary to re-label each dataset in the same way as KITTI does. As we focus on car detection, here we describe how we construct the new *car* and *truck* categories for each dataset except KITTI in Table S1. The *truck* category is also important since detected trucks are treated as false positives when we look at the *car* category. We would like to point out that Waymo labels all kinds of vehicles as *car*s. A model trained on Waymo thus will tend to predict trucks or other vehicles as cars. Therefore, directly applying a model trained on Waymo to other datasets will lead to higher false positive rates. For other datasets, the definition between categories can vary (*e.g.*, Argoverse label Ford F-Series as cars; nuScenes labels some as trucks) and result in cross-domain accuracy drop even if the data are collected at similar locations with similar sensors.

S1.3. Handling missing 2D bounding boxes

To annotate each object in the image with a 2D bounding box (the information is used by the original KITTI metric),

we first compute 8 corners of its 3D bounding box, and then calculate their pixel coordinates $\{(cx_n, cy_n), 1 \leq n \leq 8\}$. We then draw the smallest bounding box (x_1, y_1, x_2, y_2) that contains all corners whose projections fall in the image plane:

$$\begin{aligned} x_1 &= \max(\min_{0 \leq n < 8} cx_n, 0), \\ y_1 &= \max(\min_{0 \leq n < 8} cy_n, 0), \\ x_2 &= \min(\max_{0 \leq n < 8} cx_n, width), \\ y_2 &= \min(\max_{0 \leq n < 8} cy_n, height), \end{aligned} \tag{1}$$

where $width$ and $height$ denote the width and height of the 2D image, respectively.

S1.4. Calculating truncation values

Following the KITTI formulation, the *truncation* value refers to how much of an object locates beyond image boundary. With Equation 1 we estimate it by calculating how much of the object's 2D uncropped bounding box is outside the image boundary:

$$\begin{aligned} x'_1 &= \min_{0 \leq n < 8} cx_n, \\ y'_1 &= \min_{0 \leq n < 8} cy_n, \\ x'_2 &= \max_{0 \leq n < 8} cx_n, \\ y'_2 &= \max_{0 \leq n < 8} cy_n, \end{aligned} \tag{2}$$

$$\text{truncation} = 1 - \frac{(x_2 - x_1) \times (y_2 - y_1)}{(x'_2 - x'_1) \times (y'_2 - y'_1)}.$$

S1.5. Calculating occlusion values

We estimate the occlusion value of objects by approximating car shapes with corresponding 2D bounding boxes. The *occlusion* value is thus derived by computing the percentage of pixels occluded by bounding boxes from closer objects. We discretize the 0–1 occlusion value into KITTI's $\{0, 1, 2, 3\}$ labels by equally dividing the interval into 4 parts. We describe in algorithm 1 the detail of how we compute occlusion value for each object.

Algorithm 1: Computing occlusion of objects from a single scene

Input : Image height H, image width W, object list $objs$.

```
canvas ← Array([H, W]);
for x ∈ {0, 1, ..., W − 1} do
    for y ∈ {0, 1, ..., H − 1} do
        canvas[x, y] ← −1;
objs ← Sort(objs, key = depth, order = descending);
for obj ∈ objs do
    [x1, x2, y1, y2] = obj.bounding_bbox;
    for x ∈ {x1, x1 + 1, ..., x2 − 1} do
        for y ∈ {y1, y1 + 1, ..., y2 − 1} do
            canvas[x, y] ← obj.id;
for obj ∈ objs do
    [x1, x2, y1, y2] = obj.bounding_bbox;
    cnt ← 0;
    for x ∈ {x1, x1 + 1, ..., x2 − 1} do
        for y ∈ {y1, y1 + 1, ..., y2 − 1} do
            if canvas[x, y] = obj.id then
                cnt ← cnt + 1;
    obj.occlusion ← 1 − cnt / ((x2 − x1) × (y2 − y1));
```

S2. The New Difficulty Metric

In section 4.1 of the main paper, we develop a new difficulty metric to evaluate object detection (*i.e.*, how to define easy, moderate, and hard cases) so as to better align different datasets. Concretely, KITTI defines its *easy*, *moderate*, and *hard* cases according to truncation, occlusion, and 2D bounding box height (in pixels) of ground-truth annotations. The 2D box height (the threshold at 40 pixels) is meant to differentiate far-away and nearby objects: the *easy* cases only contain nearby objects. However, since the datasets we compare are collected using cameras of different focal lengths and contain images of different resolutions, directly applying the KITTI definition may not well align datasets. For example, a car at 50 meters is treated as a moderate case in KITTI but may be treated as a easy case in other datasets.

To resolve this issue, we re-define detection difficulty based on object truncation, occlusion, and **depth range (in meters)**, which completely removes the influences of cameras. In developing this new metric we hope to achieve similar case partitions to the original metric of KITTI. To this end, we estimate the distance thresholds with

$$D = \frac{f_v \times H}{h}, \tag{3}$$

where D denotes depth, f_v denotes vertical camera focal length, and H and h are object height in the 3D camera space and the 2D image space, respectively. For a car of average height (1.53 meters) in KITTI, the corresponding depth for 40 pixels is 27.03 meters. We therefore select 30 meters as the new threshold to differentiate *easy* from *moderate* and *hard* cases. For *moderate* and *hard* cases, we disregard cars with depths larger than 70 meters since most of the annotated cars in KITTI are within this range. Table S3

Table S2: 3D object detection results across multiple datasets using the original KITTI evaluation metric (pixel thresholds). We apply POINTRCNN [8]. We report average precision (AP) of the *Car* category in bird's-eye view and 3D (AP_{BEV} / AP_{3D}, IoU $= 0.7$) and compare object detection accuracy of different difficulties. The results are less comparable due to misaligned difficulty partitions among datasets. Red color: best generalization (per column and per setting); blue color: worst generalization; bold font: within-domain results.

Setting	Source\Target	KITTI	Argoverse	nuScenes	Lyft	Waymo
Easy	KITTI	**88.0 / 82.3**	44.2 / 21.4	27.5 / 7.1	72.3 / 45.5	42.1 / 10.6
	Argoverse	68.6 / 31.5	**69.9 / 43.6**	28.3 / 11.4	76.8 / 56.4	73.5 / 34.2
	nuScenes	49.4 / 13.2	57.0 / 16.5	**43.4 / 21.3**	83.0 / 31.8	71.7 / 28.2
	Lyft	72.6 / 38.9	66.9 / 33.2	35.5 / 13.1	**86.4 / 77.1**	78.0 / 54.6
	Waymo	52.0 / 13.1	64.9 / 29.4	31.5 / 14.3	82.5 / 68.8	**85.3 / 71.7**
Moderate	KITTI	**86.0 / 74.7**	44.9 / 22.3	26.2 / 8.3	63.2 / 36.3	43.9 / 12.3
	Argoverse	65.2 / 36.6	**69.8 / 44.2**	27.6 / 11.8	68.5 / 43.6	72.1 / 35.1
	nuScenes	45.4 / 12.1	56.5 / 17.1	**40.7 / 21.2**	73.4 / 26.3	68.1 / 30.7
	Lyft	67.3 / 38.3	62.4 / 35.3	33.6 / 12.3	**79.6 / 66.8**	77.3 / 53.1
	Waymo	51.5 / 14.9	64.4 / 29.8	28.9 / 13.7	75.5 / 58.2	**85.6 / 67.9**
Hard	KITTI	**85.7 / 74.8**	42.5 / 22.2	24.9 / 8.8	62.0 / 34.9	41.4 / 12.6
	Argoverse	63.5 / 37.8	**69.8 / 42.8**	26.8 / 14.5	65.9 / 44.4	68.5 / 36.7
	nuScenes	42.2 / 11.1	53.2 / 16.7	**40.2 / 20.5**	73.0 / 27.8	66.8 / 29.0
	Lyft	65.0 / 37.0	62.8 / 35.8	30.6 / 11.7	**79.7 / 67.3**	76.6 / 53.8
	Waymo	48.9 / 14.4	61.6 / 29.0	28.4 / 14.1	75.5 / 55.8	**80.2 / 67.6**

Table S3: Percentage (%) of data (total annotated cars with depths $\in [0, 70]$ meters) in each difficult partition with old / new difficulty metric. The new *easy* threshold selects much fewer data than the old metric on all datasets except KITTI.

	Dataset	Easy	Moderate	Hard
Training Set	KITTI	21.7 / 21.6	55.5 / 67.8	76.1 / 91.0
	Argoverse	27.7 / 14.9	40.5 / 40.5	59.6 / 59.6
	nuScenes	31.9 / 13.9	47.2 / 47.2	64.8 / 64.8
	Lyft	25.0 / 15.5	50.4 / 54.9	64.9 / 70.5
	Waymo	29.0 / 10.7	40.1 / 40.1	58.7 / 58.7
Validation Set	KITTI	20.4 / 20.4	55.4 / 65.5	77.1 / 88.4
	Argoverse	29.2 / 14.3	41.7 / 41.7	60.6 / 60.6
	nuScenes	38.3 / 18.4	53.6 / 53.6	68.5 / 68.5
	Lyft	25.3 / 15.5	52.5 / 57.7	66.9 / 73.3
	Waymo	30.3 / 10.3	42.3 / 42.3	60.9 / 60.9

shows the comparison between old and new difficulty partitions. The new metric contains fewer easy cases than the old metric for all but the KITTI dataset. This is because that the other datasets use either larger focal lengths or resolutions: the objects in images are therefore larger than in KITTI. We note that, the moderate cases contain all the easy cases, and the hard cases contain all the easy and moderate cases.

We also report in Table S2 the detection results within and across datasets using the old metric, in comparison to Table 2 of the main paper which uses the new metric. One notable difference is that for the easy cases in the old metric, both the within and across domain performances drop for all but KITTI datasets, since many far-away cars (which are hard to detect) in the other datasets are treated as easy cases in the old metric.

Table S4: The average size (meters) of 3D ground truth bounding boxes of the five datasets.

Dataset	Width	Height	Length
KITTI	1.62	1.53	3.89
Argoverse	1.96	1.69	4.51
nuScenes	1.96	1.73	4.64
Lyft	1.91	1.71	4.73
Waymo	2.11	1.79	4.80

S3. Dataset discrepancy

We have shown the box size distributions of each dataset in Figure 3 of the main paper. We also calculate the mean of the bounding box sizes in Table S4. There is a huge gap of size between KITTI and the other four datasets. In addition, we train an SVM classifier with the RBF kernel to predict which dataset a bounding box belongs to and present the confusion matrix result in Figure S1 (row: ground truth; column: prediction). The model has a very high confidence to distinguish KITTI from the other datasets.

We further train a point cloud classifier to tell which dataset a point cloud of car belongs to, using PointNet++ [7] as the backbone. For each dataset, we sample 8,000 object point cloud instances as training examples and 1,000 as test-

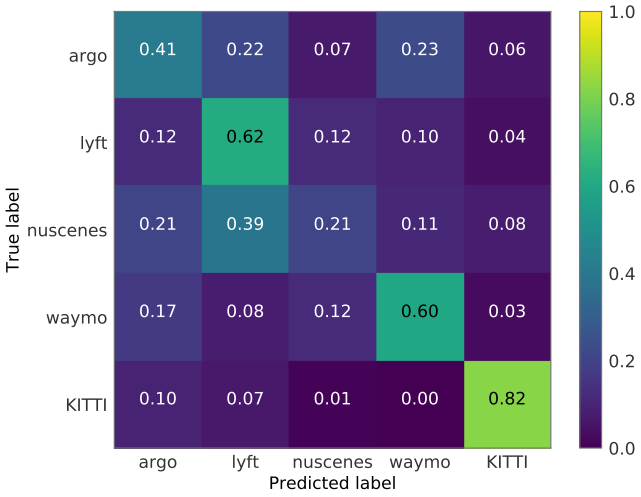

Figure S1: The confusion matrix of predicting which dataset an object belongs to using SVM with the RBF kernel. We take the $(\text{height}, \text{width}, \text{length})$ of car objects as inputs and the corresponding labels are the datasets the objects belong to.

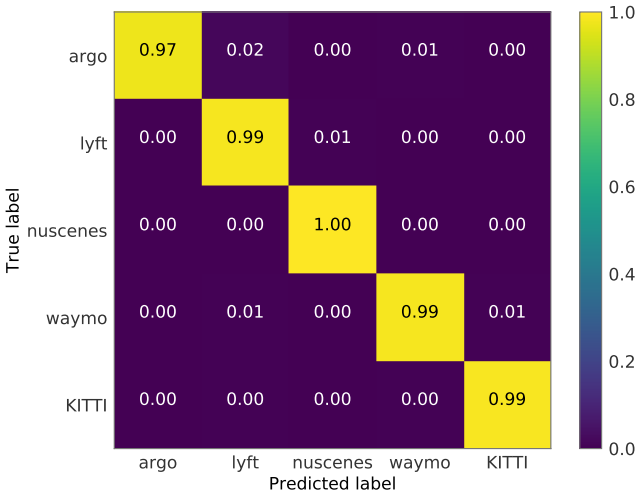

Figure S2: The confusion matrix of predicting which dataset a **Car** point cloud belongs to using PointNet++. We extract the points inside the ground truth **Car** bounding box as inputs, and the corresponding labels are the datasets the bounding boxes belong to.

ing examples. We show the confusion matrix in Figure S2. The classifier can almost perfectly classify the point clouds. Compared to Figure S1, we argue that not only the bounding box sizes, but also the point cloud styles (e.g., density, number of laser beams, etc) of cars contribute to dataset discrepancy. Interestingly, while the second factor seems to be more informative in differentiating datasets, the first factor is indeed the main cause of poor transfer performance among datasets[1].

S4. PIXOR Results

We report object detection results using PXIOR [9], which takes voxelized tensors instead of point clouds as input. We implement the algorithm ourselves, achieving comparable results as the ones in [9]. We report the results in Table S5. PXIOR performs pretty well if the model is trained and tested within the same dataset, suggesting that its model design does not over-fit to KITTI.

We also see a clear performance drop when we train a model on one dataset and test it on the other datasets. The drop is more severe than applying the PointRCNN detector in many cases. We surmise that the re-sampling operation used in PointRCNN might make the difference. We therefore apply the same re-sampling operation on the input point cloud before inputting it to PIXOR. Table S6 shows the results: re-sampling does improve the performance in applying the Waymo detector to other datasets. This is likely because Waymo has the most LiDAR points on average and re-sampling reduces the number, making it more similar to that of the other datasets. We expect that tuning the number of points in re-sampling can further boost the performance.

S5. Additional Results Using PointRCNN

S5.1. Complete tables

We show the complete tables across five datasets by replacing the predicted box sizes with the ground truth sizes (cf. section 4.5 in the main paper) in Table S7, by few-shot fine-tuning in Table S8, by statistical normalization in Table S9, and by output transformation in Table S10. For statistical normalization and output transformation, we see smaller improvements (or even some degradation) among datasets collected in the USA than between datasets collected in Germany and the USA.

S5.2. Online sales data

In the main paper, for statistical normalization we leverage the average car size of each dataset. Here we collect car sales data from Germany and the USA in the past four years. The average car size (h, w, l) is $(1.75, 1.93, 5.15)$ in the USA and $(1.49, 1.79, 4.40)$ in Germany. The difference is $(0.26, 0.14, 0.75)$, not far from $(0.20, 0.37, 0.78)$ between KITTI and the other datasets. The gap can be reduced by further considering locations (e.g., Argoverse from Miami and Pittsburgh, USA) and earlier data (KITTI was collected in 2011).

[1] As mentioned in the main paper, PointRCNN applies point re-sampling so that every scene (in RPN) and object proposal (in RCNN) will have the same numbers of input points. Such an operation could reduce the point cloud differences across domains.

Table S5: 3D object detection across multiple datasets (evaluated on the validation sets). We report average precision (AP) of the *Car* category in the bird's-eye view (AP_{BEV}) at IoU $= 0.7$, using PIXOR [9]. We report results at different difficulties (following the KITTI benchmark, but we replace the $40, 25, 25$ pixel thresholds on 2D bounding boxes with $30, 70, 70$ meters on object depths, for *Easy*, *Moderate*, and *Hard* cases, respectively) and different depth ranges (using the same truncation and occlusion thresholds as KITTI *Hard* case). The results show a significant performance drop in cross-dataset inference. We indicate the best generalization results per column and per setting by red fonts and the worst by blue fonts. We indicate in-domain results by bold fonts.

Setting	Source\Target	KITTI	Argoverse	nuScenes	Lyft	Waymo
Easy	KITTI	**87.2**	31.5	39.4	65.1	28.3
	Argoverse	57.4	**79.2**	49.0	89.4	69.3
	nuScenes	40.4	66.3	**56.8**	79.7	40.7
	Lyft	53.1	71.4	45.5	**90.7**	75.2
	Waymo	9.8	63.6	38.1	82.5	**87.2**
Moderate	KITTI	**72.8**	25.9	20.0	37.4	23.4
	Argoverse	42.5	**67.3**	24.4	57.9	55.0
	nuScenes	30.3	46.3	**30.1**	50.5	35.3
	Lyft	38.7	58.5	24.9	**78.1**	56.8
	Waymo	10.0	47.4	21.0	62.4	**76.4**
Hard	KITTI	**68.2**	28.4	18.8	34.5	24.1
	Argoverse	43.0	**64.7**	22.7	57.7	55.2
	nuScenes	27.1	46.0	**29.8**	50.6	35.8
	Lyft	35.8	54.3	24.8	**78.0**	53.7
	Waymo	11.1	46.9	21.3	63.4	**74.8**
0-30m	KITTI	**87.2**	39.5	38.9	62.2	32.1
	Argoverse	60.0	**82.4**	49.9	88.0	72.8
	nuScenes	38.7	63.0	**55.1**	79.4	43.9
	Lyft	50.7	73.5	48.4	**90.5**	76.4
	Waymo	12.9	65.7	42.7	83.1	**88.3**
30m-50m	KITTI	**50.3**	29.4	9.1	31.0	26.1
	Argoverse	23.7	**66.1**	0.8	54.5	56.5
	nuScenes	18.6	44.9	**12.3**	48.1	37.3
	Lyft	17.5	50.4	7.0	**77.0**	53.0
	Waymo	8.1	41.3	4.5	62.1	**78.0**
50m-70m	KITTI	**12.0**	3.0	3.0	9.0	10.1
	Argoverse	4.8	**31.7**	0.3	20.6	31.3
	nuScenes	9.1	13.4	**9.1**	21.6	20.9
	Lyft	6.5	19.1	9.1	**61.2**	29.9
	Waymo	1.7	20.6	9.1	39.7	**53.3**

In Table S11, we show the results of adapting a detector trained on KITTI to other datasets using statistical normalization with the car sales data: $(\Delta h, \Delta w, \Delta l)$ is $(0.26, 0.15, 0.75)$. The performance is slightly worse than using the statistics of the datasets. Nevertheless, compared to directly applying the source domain detector, statistical normalization with the car sales data still shows notable improvements.

S5.3. Pedestrian

We calculate the statistics of pedestrians, as in Table S12. There are smaller differences among datasets. We therefore expect a smaller improvement by statistical normalization.

S6. Qualitative Results

We further show qualitative results of statistical normalization refinement. We train a POINTRCNN detector on Waymo and test it on KITTI. We compare its car detection before and after statistical normalization refinement in Figure S3. Statistical normalization can not only improve the predicted bounding box sizes, but also reduce false positive rates.

Table S6: 3D object detection across multiple datasets (evaluated on the validation sets). The setting is exactly the same as Table S5, except that we perform POINTRCNN re-sampling on the input point cloud before applying the PIXOR detector.

Setting	Source\Target	KITTI	Argoverse	nuScenes	Lyft	Waymo
Easy	KITTI	**85.9**	22.3	35.7	56.2	13.4
	Argoverse	59.4	**80.5**	47.1	89.3	66.5
	nuScenes	14.5	57.1	**66.2**	73.4	44.2
	Lyft	66.6	73.8	52.2	**90.7**	77.3
	Waymo	28.6	66.0	52.0	84.2	**86.7**
Moderate	KITTI	**70.3**	19.1	18.9	33.5	14.8
	Argoverse	43.0	**66.5**	24.1	57.9	52.6
	nuScenes	12.6	46.9	**36.5**	52.6	35.7
	Lyft	49.3	54.4	28.6	**79.4**	59.2
	Waymo	23.8	51.4	26.7	69.0	**77.1**
Hard	KITTI	**67.2**	20.0	17.4	33.1	15.0
	Argoverse	42.8	**63.8**	22.3	57.7	52.5
	nuScenes	14.4	44.6	**35.7**	53.1	36.0
	Lyft	45.5	54.5	27.6	**79.3**	58.5
	Waymo	24.0	54.2	26.4	70.3	**77.3**
0-30m	KITTI	**85.8**	28.6	33.2	56.6	14.7
	Argoverse	61.5	**82.7**	48.6	88.3	65.0
	nuScenes	20.2	61.5	**64.4**	75.4	48.4
	Lyft	62.9	71.9	54.3	**90.7**	78.3
	Waymo	31.0	65.9	55.4	85.9	**88.2**
30m-50m	KITTI	**48.8**	22.0	4.5	29.3	16.8
	Argoverse	21.1	**69.7**	2.3	55.1	54.6
	nuScenes	8.6	42.8	**15.9**	52.7	40.5
	Lyft	25.1	53.1	10.3	**78.6**	59.9
	Waymo	16.7	52.4	9.8	68.8	**78.8**
50m-70m	KITTI	**15.7**	3.4	0.4	7.5	12.0
	Argoverse	9.4	**29.5**	0.1	22.2	30.4
	nuScenes	0.7	12.8	**9.1**	23.1	16.4
	Lyft	7.3	18.8	3.0	**63.6**	30.8
	Waymo	2.3	23.5	9.1	45.1	**56.8**

Table S7: Cross-dataset performance by assigning ground-truth box sizes to detected cars while keeping their centers and rotations unchanged. We report AP_{BEV}/ AP_{3D} of the *Car* category at IoU $= 0.7$, using POINTRCNN [8]. We indicate the best generalization results per column and per setting by red fonts and the worst by blue fonts. We indicate in-domain results by bold fonts.

Setting	Source\Target	KITTI	Argoverse	nuScenes	Lyft	Waymo
Easy	KITTI	**95.6 / 84.6**	80.5 / 65.7	66.5 / 33.5	89.8 / 74.8	90.3 / 77.1
	Argoverse	80.0 / 59.2	**83.1 / 77.3**	54.2 / 26.7	87.5 / 75.6	89.1 / 74.7
	nuScenes	80.5 / 63.9	77.4 / 52.5	**74.8 / 46.4**	89.4 / 65.2	85.6 / 62.9
	Lyft	83.9 / 58.4	80.2 / 67.2	65.2 / 29.2	**90.3 / 87.3**	89.9 / 73.9
	Waymo	86.1 / 78.2	79.2 / 72.7	63.1 / 30.0	88.3 / 86.1	**90.2 / 86.2**
Moderate	KITTI	**81.4 / 72.6**	64.5 / 50.9	35.0 / 18.2	74.6 / 54.3	79.4 / 63.0
	Argoverse	66.9 / 51.0	**73.6 / 60.1**	28.2 / 17.6	67.6 / 52.3	77.3 / 61.5
	nuScenes	61.4 / 47.3	59.0 / 36.2	**41.7 / 25.4**	72.4 / 45.1	69.2 / 50.6
	Lyft	71.4 / 49.4	68.5 / 49.3	34.6 / 17.4	**84.2 / 66.9**	79.7 / 64.7
	Waymo	73.7 / 60.6	68.0 / 54.9	30.8 / 18.4	75.0 / 63.2	**86.4 / 74.4**
Hard	KITTI	**82.5 / 71.9**	64.0 / 49.3	31.4 / 17.7	73.1 / 53.0	77.2 / 59.1
	Argoverse	65.6 / 52.5	**73.6 / 59.2**	27.5 / 16.6	65.3 / 52.2	75.8 / 58.4
	nuScenes	61.3 / 45.7	55.3 / 33.5	**40.9 / 25.4**	72.6 / 43.6	68.3 / 46.2
	Lyft	72.0 / 52.0	65.4 / 49.8	31.2 / 16.5	**84.8 / 67.2**	78.2 / 63.6
	Waymo	75.3 / 60.7	67.8 / 51.9	30.2 / 17.0	75.2 / 61.9	**80.8 / 68.9**
0-30m	KITTI	**89.2 / 86.7**	82.3 / 70.2	62.8 / 35.1	89.8 / 76.2	90.4 / 78.7
	Argoverse	83.3 / 68.7	**86.1 / 80.2**	56.8 / 31.9	88.3 / 77.3	89.8 / 78.1
	nuScenes	76.5 / 62.5	80.9 / 55.2	**74.2 / 49.1**	89.4 / 67.7	87.5 / 62.6
	Lyft	86.7 / 62.7	84.2 / 69.3	63.1 / 31.7	**90.5 / 88.5**	90.2 / 77.2
	Waymo	88.0 / 75.8	82.8 / 76.3	62.0 / 32.9	88.7 / 86.8	**90.5 / 88.1**
30m-50m	KITTI	**71.6 / 56.4**	63.1 / 40.4	11.1 / 9.1	74.3 / 52.9	80.4 / 64.3
	Argoverse	43.2 / 27.0	**74.5 / 53.9**	9.5 / 9.1	67.4 / 49.3	78.8 / 62.9
	nuScenes	37.1 / 25.0	49.0 / 18.3	**17.4 / 10.4**	71.0 / 42.2	75.4 / 50.5
	Lyft	52.8 / 31.4	61.9 / 35.6	11.3 / 9.1	**85.1 / 65.9**	80.4 / 65.4
	Waymo	57.6 / 38.5	63.5 / 45.8	10.0 / 9.1	75.4 / 62.5	**87.7 / 74.9**
50m-70m	KITTI	**30.8 / 15.1**	23.6 / 9.7	1.6 / 1.0	50.0 / 24.0	53.3 / 31.2
	Argoverse	13.7 / 10.1	**34.2 / 11.7**	0.5 / 0.0	37.2 / 19.8	51.9 / 31.7
	nuScenes	9.2 / 5.7	15.8 / 4.3	**9.8 / 9.1**	46.9 / 20.1	43.4 / 23.2
	Lyft	17.2 / 8.1	28.2 / 11.4	1.1 / 0.1	**64.2 / 39.9**	57.8 / 36.3
	Waymo	13.1 / 4.9	29.2 / 11.4	0.9 / 0.0	53.0 / 29.4	**65.9 / 45.2**

Table S8: Cross-dataset performance by few-shot fine-tuning using 10 labeled target domain instances (average over five rounds of experiments). We report AP_{BEV}/ AP_{3D} of the *Car* category at IoU $= 0.7$, using POINTRCNN [8]. We indicate the best generalization results per column and per setting by red fonts and the worst by blue fonts. We indicate in-domain results by bold fonts.

Setting	Source\Target	KITTI	Argoverse	nuScenes	Lyft	Waymo
Easy	KITTI	**88.0 / 82.5**	75.8 / 49.2	54.7 / 21.7	89.0 / 78.1	87.4 / 70.9
	Argoverse	80.0 / 49.7	**74.2 / 42.0**	54.0 / 19.2	86.6 / 63.5	86.6 / 56.3
	nuScenes	83.8 / 58.7	68.7 / 33.7	**73.4 / 38.1**	88.4 / 67.7	84.3 / 59.8
	Lyft	85.3 / 72.5	73.5 / 48.9	56.5 / 17.7	**90.2 / 87.3**	89.1 / 70.4
	Waymo	81.0 / 67.0	76.9 / 55.2	51.0 / 16.7	88.3 / 81.0	**90.1 / 85.3**
Moderate	KITTI	**80.6 / 68.9**	60.7 / 37.3	28.7 / 12.5	74.2 / 53.4	75.9 / 55.3
	Argoverse	68.8 / 42.8	**66.5 / 34.4**	27.5 / 11.2	65.4 / 40.2	75.3 / 46.7
	nuScenes	67.2 / 45.5	54.5 / 24.2	**40.7 / 21.2**	71.9 / 44.0	72.8 / 47.0
	Lyft	73.9 / 56.2	61.0 / 35.3	30.3 / 10.6	**83.7 / 65.5**	78.3 / 57.9
	Waymo	66.8 / 51.8	65.7 / 41.8	26.7 / 11.0	75.1 / 54.8	**85.9 / 67.9**
Hard	KITTI	**81.9 / 66.7**	59.8 / 36.5	27.5 / 12.4	71.8 / 52.9	70.1 / 54.4
	Argoverse	66.3 / 43.0	**67.9 / 37.3**	26.9 / 11.8	66.0 / 42.0	70.3 / 43.9
	nuScenes	64.7 / 44.5	52.0 / 23.4	**40.2 / 20.5**	71.0 / 44.3	68.7 / 44.3
	Lyft	74.1 / 56.2	61.9 / 37.0	28.6 / 11.1	**79.3 / 65.5**	76.9 / 55.6
	Waymo	68.1 / 52.9	62.3 / 39.3	26.7 / 11.7	74.7 / 55.2	**80.4 / 67.7**
0-30m	KITTI	**88.8 / 84.9**	73.6 / 55.2	54.0 / 23.6	89.3 / 77.6	88.7 / 74.1
	Argoverse	84.0 / 56.9	**81.2 / 52.2**	54.0 / 22.6	87.7 / 68.7	88.3 / 60.7
	nuScenes	81.2 / 59.8	70.5 / 40.1	**73.2 / 42.8**	88.8 / 69.6	86.2 / 62.4
	Lyft	87.5 / 73.9	78.1 / 54.3	56.9 / 21.2	**90.4 / 88.5**	89.4 / 74.8
	Waymo	84.8 / 71.0	79.4 / 56.6	52.8 / 20.8	88.8 / 79.1	**90.4 / 87.2**
30m-50m	KITTI	**70.2 / 51.4**	59.0 / 29.9	9.5 / 6.1	73.7 / 50.4	78.1 / 57.2
	Argoverse	47.9 / 23.8	**70.8 / 34.0**	7.3 / 2.0	65.4 / 36.9	78.1 / 48.5
	nuScenes	45.0 / 25.1	51.4 / 17.1	**17.1 / 4.1**	71.5 / 41.5	74.2 / 48.0
	Lyft	57.7 / 33.3	62.4 / 29.5	6.5 / 3.3	**83.8 / 62.7**	79.7 / 59.9
	Waymo	49.2 / 29.2	60.6 / 34.7	9.4 / 6.3	75.1 / 52.6	**87.5 / 68.8**
50m-70m	KITTI	**28.8 / 12.0**	20.1 / 6.3	3.3 / 1.2	46.8 / 19.4	45.2 / 24.3
	Argoverse	8.1 / 3.8	**33.0 / 12.7**	0.4 / 0.0	38.0 / 10.3	51.1 / 23.4
	nuScenes	12.9 / 5.7	15.5 / 2.6	**9.1 / 9.1**	47.0 / 14.9	44.3 / 19.3
	Lyft	17.5 / 8.0	26.8 / 9.1	2.5 / 0.0	**62.7 / 33.1**	54.0 / 27.2
	Waymo	10.5 / 4.8	27.6 / 7.3	1.3 / 0.0	51.2 / 19.9	**63.5 / 41.1**

Table S9: Cross-dataset performance by fine-tuning with source data after statistical normalization. We report AP_{BEV}/ AP_{3D} of the *Car* category at IoU $= 0.7$, using POINTRCNN [8]. We indicate the best generalization results per column and per setting by red fonts and the worst by blue fonts. We indicate in-domain results by bold fonts.

Setting	Source\Target	KITTI	Argoverse	nuScenes	Lyft	Waymo
Easy	KITTI	**88.0 / 82.5**	74.7 / 48.2	60.8 / 23.9	88.3 / 73.3	84.6 / 53.3
	Argoverse	76.2 / 46.1	**79.2 / 57.8**	48.3 / 18.6	84.8 / 65.0	84.8 / 49.2
	nuScenes	83.2 / 35.6	72.0 / 25.3	**73.4 / 38.1**	88.7 / 38.1	76.6 / 43.3
	Lyft	83.5 / 72.1	74.4 / 44.0	57.8 / 21.1	**90.2 / 87.3**	86.3 / 66.4
	Waymo	82.1 / 48.7	75.0 / 44.4	54.9 / 20.7	85.7 / 80.0	**90.1 / 85.3**
Moderate	KITTI	**80.6 / 68.9**	61.5 / 38.2	32.9 / 16.4	73.7 / 53.1	74.9 / 49.4
	Argoverse	67.2 / 40.5	**69.9 / 44.2**	24.7 / 11.1	63.3 / 38.9	72.0 / 43.6
	nuScenes	67.4 / 31.0	55.6 / 17.9	**40.7 / 21.2**	71.1 / 24.5	66.6 / 32.2
	Lyft	73.6 / 57.9	59.7 / 33.3	30.4 / 10.9	**83.7 / 65.5**	75.5 / 51.3
	Waymo	71.3 / 47.1	62.3 / 31.7	28.8 / 11.5	71.5 / 52.6	**85.9 / 67.9**
Hard	KITTI	**81.9 / 66.7**	60.6 / 37.1	31.9 / 15.8	73.1 / 53.5	69.4 / 49.4
	Argoverse	68.5 / 41.9	**69.9 / 42.8**	24.3 / 10.9	61.6 / 40.2	68.2 / 42.7
	nuScenes	65.2 / 30.8	52.5 / 17.2	**40.2 / 20.5**	67.3 / 28.6	65.7 / 30.4
	Lyft	75.2 / 58.9	60.8 / 31.8	29.5 / 14.4	**79.3 / 65.5**	75.5 / 53.2
	Waymo	73.0 / 49.7	60.2 / 32.5	28.4 / 10.9	71.6 / 53.3	**80.4 / 67.7**
0-30m	KITTI	**88.8 / 84.9**	73.1 / 54.2	60.0 / 29.2	88.8 / 75.4	87.1 / 60.1
	Argoverse	83.3 / 53.9	**83.3 / 63.3**	51.5 / 23.0	86.3 / 68.4	87.3 / 59.7
	nuScenes	83.6 / 42.8	72.8 / 27.2	**73.2 / 42.8**	88.9 / 47.1	78.5 / 45.9
	Lyft	87.4 / 73.6	78.7 / 51.8	58.7 / 26.8	**90.4 / 88.5**	87.9 / 72.4
	Waymo	85.7 / 59.0	79.9 / 50.5	57.6 / 24.3	87.2 / 75.8	**90.4 / 87.2**
30m-50m	KITTI	**70.2 / 51.4**	61.5 / 31.5	11.0 / 2.3	73.8 / 52.2	78.1 / 54.9
	Argoverse	48.9 / 25.7	**72.2 / 39.5**	5.0 / 4.5	61.0 / 32.4	74.4 / 46.2
	nuScenes	44.9 / 18.6	45.6 / 7.3	**17.1 / 4.1**	70.1 / 18.1	67.9 / 31.6
	Lyft	58.3 / 38.0	57.2 / 18.5	6.5 / 4.5	**83.8 / 62.7**	77.2 / 52.4
	Waymo	57.3 / 36.3	54.9 / 20.1	9.1 / 1.5	71.3 / 48.4	**87.5 / 68.8**
50m-70m	KITTI	**28.8 / 12.0**	23.8 / 5.6	3.0 / 2.3	49.9 / 22.2	46.8 / 25.1
	Argoverse	9.1 / 2.6	**29.9 / 6.9**	0.2 / 0.1	28.9 / 8.8	46.2 / 21.2
	nuScenes	9.4 / 5.1	14.8 / 2.3	**9.1 / 9.1**	40.7 / 5.2	36.4 / 14.9
	Lyft	21.1 / 6.7	21.2 / 4.9	4.5 / 0.0	**62.7 / 33.1**	52.1 / 25.3
	Waymo	14.4 / 5.7	27.7 / 11.0	1.0 / 0.0	46.9 / 22.0	**63.5 / 41.1**

Table S10: Cross-dataset performance by output transformation: directly adjusting the predicted box size by adding the difference of mean sizes between domains. We report AP_{BEV}/ AP_{3D} of the *Car* category at IoU $= 0.7$, using POINTRCNN [8]. We indicate the best generalization results per column and per setting by red fonts and the worst by blue fonts. We indicate in-domain results by bold fonts.

Setting	Source\Target	KITTI	Argoverse	nuScenes	Lyft	Waymo
Easy	KITTI	**88.0 / 82.5**	72.7 / 9.0	55.0 / 10.4	88.2 / 23.5	86.1 / 16.2
	Argoverse	53.3 / 5.7	**79.2 / 57.8**	52.6 / 21.3	87.1 / 66.1	87.6 / 56.1
	nuScenes	75.4 / 31.5	73.3 / 27.9	**73.4 / 38.1**	89.2 / 44.3	78.4 / 35.5
	Lyft	71.9 / 4.7	77.1 / 48.0	63.1 / 24.5	**90.2 / 87.3**	89.2 / 73.9
	Waymo	64.0 / 3.9	74.3 / 54.8	58.8 / 25.2	88.3 / 85.3	**90.1 / 85.3**
Moderate	KITTI	**80.6 / 68.9**	59.9 / 7.9	30.8 / 6.8	70.1 / 17.8	69.1 / 13.1
	Argoverse	52.2 / 7.3	**69.9 / 44.2**	27.5 / 11.7	66.9 / 42.1	74.3 / 45.5
	nuScenes	58.5 / 27.3	56.8 / 20.4	**40.7 / 21.2**	71.3 / 27.3	67.8 / 26.2
	Lyft	60.8 / 5.6	62.7 / 37.6	33.5 / 12.5	**83.7 / 65.5**	78.4 / 60.8
	Waymo	54.9 / 3.7	62.9 / 40.4	30.1 / 14.5	74.3 / 59.8	**85.9 / 67.9**
Hard	KITTI	**81.9 / 66.7**	59.3 / 9.3	27.8 / 7.6	66.5 / 19.1	68.7 / 13.9
	Argoverse	53.5 / 8.6	**69.9 / 42.8**	26.7 / 14.5	64.6 / 43.0	70.0 / 44.2
	nuScenes	59.5 / 27.8	53.6 / 19.9	**40.2 / 20.5**	67.6 / 28.5	66.3 / 26.0
	Lyft	63.1 / 6.9	63.4 / 38.6	30.4 / 13.3	**79.3 / 65.5**	77.3 / 57.3
	Waymo	58.0 / 4.1	60.5 / 39.2	29.4 / 14.6	74.0 / 57.2	**80.4 / 67.7**
0-30m	KITTI	**88.8 / 84.9**	73.0 / 13.7	56.2 / 13.9	88.4 / 27.5	87.7 / 22.2
	Argoverse	64.9 / 10.1	**83.3 / 63.3**	55.2 / 27.0	87.8 / 69.9	87.9 / 62.6
	nuScenes	74.6 / 36.6	73.7 / 32.0	**73.2 / 42.8**	89.2 / 46.2	79.6 / 41.6
	Lyft	74.8 / 9.1	81.2 / 55.8	61.2 / 27.2	**90.4 / 88.5**	89.6 / 77.2
	Waymo	71.3 / 4.4	78.4 / 55.7	60.5 / 25.8	88.7 / 85.0	**90.4 / 87.2**
30m-50m	KITTI	**70.2 / 51.4**	56.1 / 5.4	10.8 / 9.1	67.4 / 10.7	73.6 / 10.4
	Argoverse	35.1 / 9.1	**72.2 / 39.5**	9.5 / 0.3	66.3 / 39.1	77.5 / 44.9
	nuScenes	35.5 / 15.5	47.4 / 7.8	**17.1 / 4.1**	69.9 / 22.5	68.7 / 21.1
	Lyft	43.3 / 3.9	60.8 / 25.4	11.2 / 9.1	**83.8 / 62.7**	79.5 / 61.4
	Waymo	39.8 / 4.5	58.1 / 34.9	9.9 / 9.1	74.5 / 57.5	**87.5 / 68.8**
50m-70m	KITTI	**28.8 / 12.0**	20.5 / 1.0	1.5 / 1.0	41.3 / 6.8	42.6 / 4.2
	Argoverse	8.0 / 0.8	**29.9 / 6.9**	0.5 / 0.0	35.6 / 14.2	49.2 / 20.3
	nuScenes	7.8 / 5.1	15.3 / 3.0	**9.1 / 9.1**	41.4 / 5.6	37.0 / 12.0
	Lyft	12.7 / 0.9	25.6 / 6.0	1.1 / 0.0	**62.7 / 33.1**	54.9 / 30.4
	Waymo	7.7 / 1.1	25.5 / 6.5	0.9 / 0.0	50.8 / 22.3	**63.5 / 41.1**

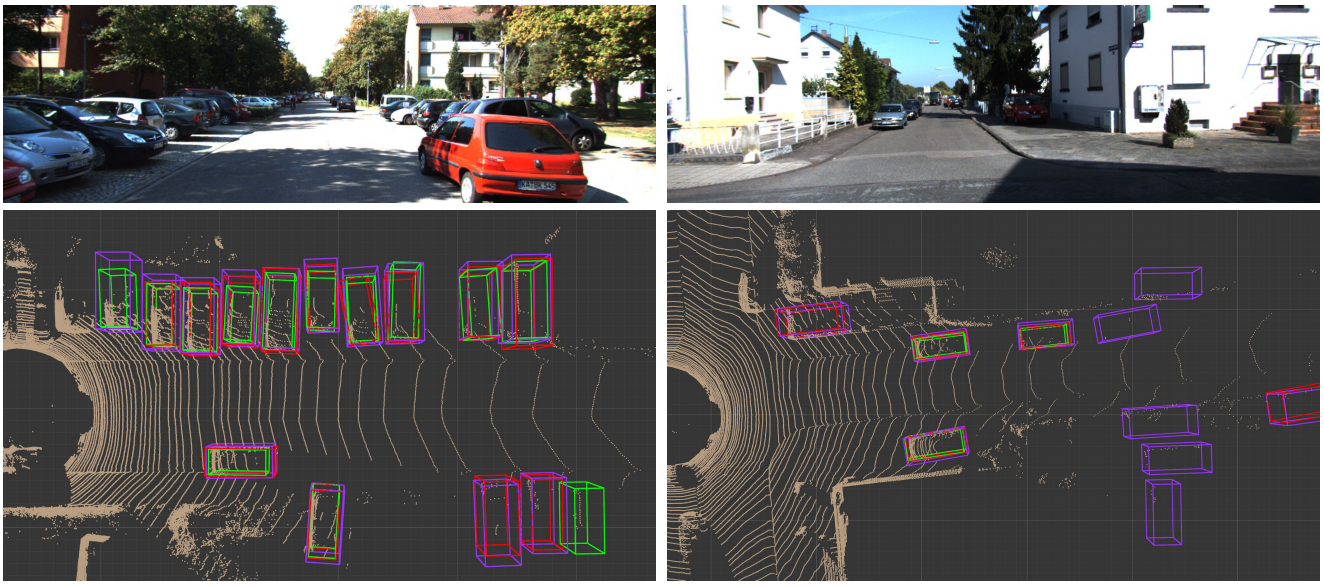

Figure S3: 3D prediction on KITTI using the model trained on Waymo before and after statistical normalization refinement. The green boxes are the ground truth *car* bounding boxes. Purple boxes and magenta boxes are predictions by the model before and after statistical normalization refinement, respectively. The left column demonstrates that statistical normalization is able to resize bounding box predictions to the correct sizes, while the right case shows that it also can reduce false positive rates.

Table S11: Statistical normalization using the mean sizes of datasets versus car sales data. Direct: directly applying the source domain detector.

		From KITTI (KITTI as the source)		
Setting	Dataset	Direct	Datasets	Car sales data
Easy	Argoverse	55.8 / 27.7	74.7 / 48.2	68.6 / 32.8
	nuScenes	47.4 / 13.3	60.8 / 23.9	62.0 / 24.4
	Lyft	81.7 / 51.8	88.3 / 73.3	88.9 / 69.9
	Waymo	45.2 / 11.9	84.6 / 53.3	66.7 / 22.8
Moderate	Argoverse	44.9 / 22.3	61.5 / 38.2	57.7 / 29.1
	nuScenes	26.2 / 8.3	32.9 / 16.4	32.6 / 13.0
	Lyft	61.8 / 33.7	73.7 / 53.1	72.6 / 47.6
	Waymo	43.9 / 12.3	74.9 / 49.4	61.8 / 22.9
Hard	Argoverse	42.5 / 22.2	60.6 / 37.1	54.0 / 30.0
	nuScenes	24.9 / 8.8	31.9 / 15.8	29.8 / 13.2
	Lyft	57.4 / 34.2	73.1 / 53.5	71.7 / 45.7
	Waymo	41.5 / 12.6	69.4 / 49.4	62.7 / 25.1
0-30m	Argoverse	58.4 / 34.7	73.1 / 54.2	71.0 / 44.0
	nuScenes	47.9 / 14.9	60.0 / 29.2	60.1 / 26.1
	Lyft	77.8 / 54.2	88.8 / 75.4	89.2 / 72.5
	Waymo	48.0 / 14.0	87.1 / 60.1	72.4 / 30.2
30m-50m	Argoverse	46.5 / 19.0	61.5 / 31.5	57.4 / 20.0
	nuScenes	9.8 / 4.5	11.0 / 2.3	5.7 / 3.0
	Lyft	60.1 / 34.5	73.8 / 52.2	72.2 / 42.7
	Waymo	50.5 / 21.4	78.1 / 54.9	66.8 / 35.5
50m-70m	Argoverse	9.2 / 3.0	23.8 / 5.6	16.8 / 4.5
	nuScenes	1.1 / 0.0	3.0 / 2.3	1.0 / 0.1
	Lyft	33.2 / 9.6	49.9 / 22.2	46.0 / 18.8
	Waymo	27.1 / 12.0	46.8 / 25.1	44.2 / 18.0

Table S12: Dataset statistics on pedestrians (meters)

	KITTI	Argoverse	nuScenes	Lyft	Waymo
H	1.76±0.11	1.84±0.15	1.78±0.18	1.76±0.18	1.75±0.20
W	0.66±0.14	0.78±0.14	0.67±0.14	0.76±0.14	0.85±0.15
L	0.84±0.23	0.78±0.14	0.73±0.19	0.78±0.17	0.90±0.19

References

[1] Waymo open dataset: An autonomous driving dataset, 2019. 1

[2] 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. 1

[3] 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. Argoverse: 3d tracking and forecasting with rich maps. In *CVPR*, 2019. 1

[4] 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

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

[6] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Lyft level 5 av dataset 2019. urlhttps://level5.lyft.com/dataset/, 2019. 1

[7] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 3

[8] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019. 3, 7, 8, 9, 10

[9] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *CVPR*, 2018. 1, 4, 5