nuScenes: A multimodal dataset for autonomous driving

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu,
Anush Krishnan, Yu Pan, Giancarlo Baldan, Oscar Beijbom
nuTonomy: an APTIV company
nuscenes@nutonomy.com

Abstract

Robust detection and tracking of objects is crucial for the deployment of autonomous vehicle technology. Image-based benchmark datasets have driven development in computer vision tasks such as object detection, tracking and segmentation of agents in the environment. Most autonomous vehicles, however, carry a combination of cameras and range sensors such as lidar and radar. As machine learning based methods for detection and tracking become more prevalent, there is a need to train and evaluate such methods on datasets containing range sensor data along with images. In this work we present nuTonomy scenes (nuScenes), the first published dataset to carry the full autonomous vehicle sensor suite: 6 cameras, 5 radars and 1 lidar, all with full 360 degree field of view. nuScenes comprises 1000 scenes, each 20s long and fully annotated with 3D bounding boxes for 23 classes and 8 attributes. It has 7x as many annotations and 100x as many images as the pioneering KITTI dataset. We define novel 3D detection and tracking metrics. We also provide careful dataset analysis as well as baselines for lidar and image based detection and tracking. Data, development kit and more information are available online at `www.nuscenes.org`.

1. Introduction

Autonomous driving has the potential to radically change the cityscape and save many human lives [73]. A crucial part of safe navigation is the detection and tracking of agents in the environment surrounding the vehicle. To achieve this, a modern self-driving vehicle deploys several sensors along with sophisticated detection and tracking algorithms. Such algorithms rely increasingly on machine learning, which drives the need for benchmark datasets. While there is a plethora of image datasets for this purpose, there is a lack of multimodal datasets that exhibit the full set of challenges associated with building an automonous driving perception system. We released the nuScenes dataset to address this gap[1].

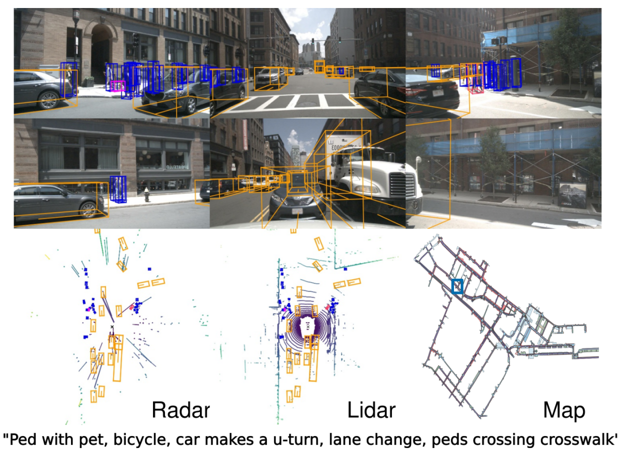

Figure 1. An example from the nuScenes dataset. We see 6 different camera views, lidar and radar data, as well as the human annotated semantic map. At the bottom we show the human written scene description.

Multimodal datasets are of particular importance as no single type of sensor is sufficient and the sensor types are complementary. Cameras allow accurate measurements of edges, color and lighting enabling classification and localization on the image plane. However, 3D localization from images is challenging [13, 12, 55, 75, 65, 62, 68]. Lidar pointclouds, on the other hand, contain less semantic information but highly accurate localization in 3D [49]. Furthermore the reflectance of lidar is an important feature [39, 49]. However, lidar data is sparse and the range is typically limited to 50-100m. Radar sensors achieve a range of 200-300m and measure the object velocity through the Doppler effect. However, the returns are even sparser than lidar and less precise in terms of localization. While radar has been used for decades [1, 3], we are not aware of any autonomous driving datasets that provide radar data.

Since the three sensor types have different failure modes during difficult conditions, the joint treatment of sensor data is essential for agent detection and tracking. Literature [44] even suggests that multimodal sensor configurations are not just complementary, but provide redundancy in the face of sabotage, failure, adverse conditions

[1]nuScenes teaser set released Sep. 2018, full release March 2019.

Figure 2. Front camera images collected from clear weather (col 1), nighttime (col 2), rain (col 3) and construction zones (col 4).

and blind spots. And while there are several works that have proposed fusion methods based on cameras and lidar [46, 14, 61, 50, 76, 70, 28], PointPillars [49] has shown that a lidar only method performs on par or better than fusion based methods. This suggests more work is required to combine multimodal measurements in a principled manner.

In order to train deep learning methods, quality data annotations are required. Most datasets provide 2D semantic annotations as boxes or masks (class or instance) [8, 18, 32, 79, 53]. At the time of the initial nuScenes release, only a few datasets annotated objects using 3D boxes [31, 40, 59], and they did not provide the full sensor suite. Following the nuScenes release, there are now several sets which contain the full sensor suite. Still, to the best of our knowledge, no other 3D dataset provides attribute annotations, such as pedestrian pose or vehicle state.

Existing AV datasets and vehicles are highly specialized for particular operational design domains. More research is required on how to generalize to "complex, cluttered and unseen environments" [35]. Therefore, there is a need to study how detection methods generalize to different countries, lighting (daytime vs. nighttime), driving directions, road markings, vegetation, precipitation and previously unseen object types.

Contextual knowledge using semantic maps is also an important prior for scene understanding [77, 2, 34]. For example, one would expect to find cars on the road, but not on the sidewalk or inside buildings. With the notable exception of [43, 10], the majority of the AV datasets do not provide semantic maps.

1.1. Contributions

From the complexities of the multimodal 3D detection challenge, and the limitations of current AV datasets, a large-scale multimodal dataset with 360° coverage across all vision and range sensors collected from diverse situations alongside map information would boost AV scene-understanding research further. nuScenes does just that, and it is the main contribution of this work.

nuScenes represents a large leap forward in terms of data volumes and complexities (Table 1), and is the first dataset to provide 360° sensor coverage from the *entire sensor suite*. It is also the first AV dataset to include *radar data* and captured using an AV *approved for public roads*. It is further the first multimodal dataset that contains data from *nighttime* and *rainy* conditions, and with *object attributes and scene descriptions* in addition to object class and location. nuScenes enables research on object detection, tracking and behavior modeling in a range of conditions.

Our second contribution is new detection and tracking metrics aimed at the AV application. We train 3D object detectors and trackers as a baseline, including a novel approach of using multiple lidar sweeps to enhance object detection. We also present and analyze the results of the nuScenes object detection and tracking challenges.

Third, we publish the devkit, evaluation code, taxonomy, annotator instructions, and database schema for industry-wide standardization. Recently, the Lyft L5 [43] dataset adopted this format to achieve compatibility between the different datasets. The nuScenes data is published under CC BY-NC-SA 4.0 license, which means that anyone can use this dataset for non-commercial research purposes. All data, code, and information is made available online at `www.nuscenes.org`.

Since the release, nuScenes has received strong interest from the AV community [84, 66, 48, 85, 9, 5, 64, 27, 47, 80, 83]. Some works extended our dataset to introduce new annotations for natural language object referral [21] and high-level scene understanding [69]. The detection challenge enabled lidar-based and camera-based detection works such as [84, 66], that improved over the state-of-the-art at the time of initial release [49, 65] by 40% and 81% (Table 4). nuScenes has been used for 3D object detection [78, 58], multi-agent forecasting [9, 64], pedestrian localization [5], weather augmentation [36], and moving pointcloud prediction [26]. Being still the only annotated AV dataset to provide radar data, nuScenes encourages researchers to explore radar and sensor fusion for object detection [26, 41].

Dataset	**Year**	**Sce-nes**	**Size (hr)**	**RGB imgs**	**PCs lidar††**	**PCs radar**	**Ann. frames**	**3D boxes**	**Night / Rain**	**Map layers**	**Clas-ses**	**Locations**
CamVid [8]	2008	4	0.4	18k	0	0	700	0	No/No	0	32	Cambridge
Cityscapes [18]	2016	n/a	-	25k	0	0	25k	0	No/No	0	30	50 cities
Vistas [32]	2017	n/a	-	25k	0	0	25k	0	Yes/Yes	0	152	Global
BDD100K [79]	2017	100k	1k	100M	0	0	100k	0	Yes/Yes	0	10	NY, SF
ApolloScape [40]	2018	-	100	144k	0**	0	144k	70k	Yes/No	0	8-35	4x China
D^2-City [11]	2019	1k†	-	700k†	0	0	700k†	0	No/Yes	0	12	5x China
KITTI [31]	2012	22	1.5	15k	15k	0	15k	200k	No/No	0	8	Karlsruhe
AS lidar [52]	2018	-	2	0	20k	0	20k	475k	-/-	0	6	China
KAIST [16]	2018	-	-	8.9k	8.9k	0	8.9k	0	**Yes**/No	0	3	Seoul
H3D [59]	2019	160	0.77	83k	27k	0	27k	1.1M	No/No	0	8	SF
nuScenes	2019	**1k**	5.5	**1.4M**	**400k**	**1.3M**	**40k**	1.4M	**Yes/Yes**	**11**	**23**	Boston, SG
Argoverse [10]	2019	113†	0.6†	490k†	44k	0	22k†	993k†	**Yes/Yes**	2	15	Miami, PT
Lyft L5 [43]	2019	366	2.5	323k	46k	0	**46k**	1.3M	No/No	7	9	Palo Alto
Waymo Open [71]	2019	**1k**	5.5	1M	200k	0	**200k**‡	**12M**‡	**Yes/Yes**	0	4	3x USA
A*3D [60]	2019	n/a	**55**	39k	39k	0	**39k**	230k	**Yes/Yes**	0	7	SG
A2D2 [33]	2019	n/a	-	-	-	0	12k	-	-/-	0	14	3x Germany

Table 1. AV dataset comparison. The top part of the table indicates datasets without range data. The middle and lower parts indicate datasets with range data released until and after the initial release of this dataset. We do not compare dataset statistics against newer datasets. Only datasets which provide annotations for at least *car*, *pedestrian* and *bicycle* are included in this comparison. (†) We report numbers only for scenes annotated with cuboids. (‡) The current Waymo Open dataset size is comparable to nuScenes, but at a 5x higher annotation frequency. (††) Lidar pointcloud count collected from *each lidar*. (**) [40] provides static depth maps. (-) indicates that no information is provided. SG: Singapore, NY: New York, SF: San Francisco, PT: Pittsburgh, AS: ApolloScape.

1.2. Related datasets

The last decade has seen the release of several driving datasets which have played a huge role in scene-understanding research for AVs. Most datasets have focused on 2D annotations (boxes, masks) for RGB camera images. CamVid [8], Cityscapes [18], Mapillary Vistas [32], D^2-City [11], BDD100k [79] and Apolloscape [40] released ever growing datasets with segmentation masks. Vistas, D^2-City and BDD100k also contain images captured during different weather and illumination settings. Other datasets focus exclusively on pedestrian annotations on images [19, 24, 74, 23, 82, 22, 56]. The ease of capturing and annotating RGB images have made the release of these large image-only datasets possible.

On the other hand, multimodal datasets, which are typically comprised of images, range sensor data (lidars, radars), and GPS/IMU data, are expensive to collect and annotate due to the difficulties of integrating, synchronizing, and calibrating multiple sensors. KITTI [31] was the pioneering multimodal dataset providing dense pointclouds from a lidar sensor as well as front-facing stereo images and GPS/IMU data. It provides 200k 3D boxes over 22 scenes which helped advance the state-of-the-art in 3D object detection. The recent H3D dataset [59] includes 160 crowded scenes with a total of 1.1M 3D boxes annotated over 27k frames. The objects are annotated in the full 360° view, as opposed to KITTI where an object is only annotated if it is present in the frontal view. The KAIST multispectral dataset [16] is a multimodal dataset that consists of RGB and thermal camera, RGB stereo, 3D lidar and GPS/IMU. It provides nighttime data, but the size of the dataset is limited and annotations are in 2D. Other notable multimodal datasets include [15] providing driving behavior labels, [42] providing place categorization labels and [6, 53] providing raw data without semantic labels.

After the initial nuScenes release, [71, 10, 60, 33, 43] followed to release their own large-scale AV datasets (Table 1). Among these datasets, only the Waymo Open dataset [71] provides significantly more annotations, mostly due to the higher annotation frequency (10Hz vs. 2Hz)[2]. A*3D takes an orthogonal approach where a similar number of frames (39k) are selected and annotated from 55 hours of data. The Lyft L5 dataset [43] is most similar to nuScenes. It was released using the nuScenes database schema and can therefore be parsed using the nuScenes devkit.

2. The nuScenes dataset

Here we describe how we plan drives, setup our vehicles, select interesting scenes, annotate the dataset and protect the privacy of third parties.

Drive planning. We drive in Boston (Seaport and South Boston) and Singapore (One North, Holland Village and Queenstown), two cities that are known for their dense traffic and highly challenging driving situations. We emphasize the diversity across locations in terms of vegetation, buildings, vehicles, road markings and right versus left-hand traffic. From a large body of training data we manually select 84 logs with 15h of driving data (242km travelled at an average of 16km/h). Driving routes are carefully chosen to

[2]In preliminary analysis we found that annotations at 2Hz are robust to interpolation to finer temporal resolution, like 10Hz or 20Hz. A similar conclusion was drawn for H3D [59] where annotations are interpolated from 2Hz to 10Hz.

Sensor	Details
6x Camera	RGB, 12Hz capture frequency, $1/1.8$" CMOS sensor, 1600×900 resolution, auto exposure, JPEG compressed
1x Lidar	Spinning, 32 beams, 20Hz capture frequency, 360° horizontal FOV, $-30°$ to $10°$ vertical FOV, $\leq 70m$ range, $\pm$2cm accuracy, up to $1.4M$ points per second.
5x Radar	$\leq 250m$ range, 77GHz, FMCW, 13Hz capture frequency, ± 0.1km/h vel. accuracy
GPS & IMU	GPS, IMU, AHRS. $0.2°$ heading, $0.1°$ roll/pitch, 20mm RTK positioning, 1000Hz update rate

Table 2. Sensor data in nuScenes.

capture a diverse set of locations (urban, residential, nature and industrial), times (day and night) and weather conditions (sun, rain and clouds).

Car setup. We use two Renault Zoe supermini electric cars with an identical sensor layout to drive in Boston and Singapore. See Figure 4 for sensor placements and Table 2 for sensor details. Front and side cameras have a 70° FOV and are offset by 55°. The rear camera has a FOV of 110°.

Sensor synchronization. To achieve good cross-modality data alignment between the lidar and the cameras, the exposure of a camera is triggered when the top lidar sweeps across the center of the camera's FOV. The timestamp of the image is the exposure trigger time; and the timestamp of the lidar scan is the time when the full rotation of the current lidar frame is achieved. Given that the camera's exposure time is nearly instantaneous, this method generally yields good data alignment[3]. We perform motion compensation using the localization algorithm described below.

Localization. Most existing datasets provide the vehicle location based on GPS and IMU [31, 40, 18, 59]. Such localization systems are vulnerable to GPS outages, as seen on the KITTI dataset [31, 7]. As we operate in dense urban areas, this problem is even more pronounced. To accurately localize our vehicle, we create a detailed HD map of lidar points in an offline step. While collecting data, we use a Monte Carlo Localization scheme from lidar and odometry information [17]. This method is very robust and we achieve localization errors of $\leq 10cm$. To encourage low-level robotics research, we also provide the raw CAN bus data (accelerations, torque, steering angles, wheel speeds).

Maps. We provide highly accurate human-annotated semantic maps of the relevant areas. The original rasterized map includes only roads and sidewalks with a resolution of 10px/m. The vectorized *map expansion* provides information on 11 semantic classes as shown in Figure 3, making it richer than the semantic maps of other datasets published since the original release [10, 43]. We encourage the use of localization and semantic maps as strong priors for all tasks. Finally, we provide the baseline routes - the idealized path an AV *should* take, assuming there are no obstacles. This

[3]The cameras run at 12Hz while the lidar runs at 20Hz. The 12 camera exposures are spread as evenly as possible across the 20 lidar scans, so not all lidar scans have a corresponding camera frame.

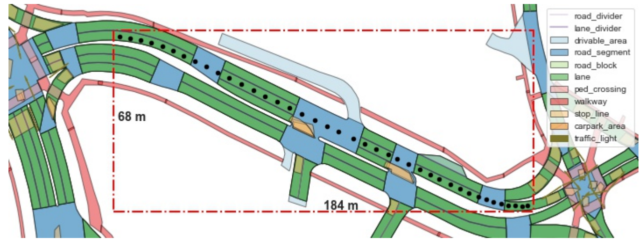

Figure 3. Semantic map of nuScenes with 11 semantic layers in different colors. To show the path of the ego vehicle we plot each keyframe ego pose from *scene-0121* with black spheres.

route may assist trajectory prediction [64], as it simplifies the problem by reducing the search space of viable routes.

Scene selection. After collecting the raw sensor data, we manually select 1000 *interesting* scenes of $20s$ duration each. Such scenes include high traffic density (e.g. intersections, construction sites), rare classes (e.g. ambulances, animals), potentially dangerous traffic situations (e.g. jaywalkers, incorrect behavior), maneuvers (e.g. lane change, turning, stopping) and situations that may be difficult for an AV. We also select some scenes to encourage diversity in terms of spatial coverage, different scene types, as well as different weather and lighting conditions. Expert annotators write textual descriptions or *captions* for each scene (e.g.: "Wait at intersection, peds on sidewalk, bicycle crossing, jaywalker, turn right, parked cars, rain").

Data annotation. Having selected the scenes, we sample keyframes (image, lidar, radar) at 2Hz. We annotate each of the 23 object classes in every keyframe with a semantic category, attributes (visibility, activity, and pose) and a cuboid modeled as x, y, z, width, length, height and yaw angle. We annotate objects continuously throughout each scene if they are covered by at least one lidar or radar point. Using expert annotators and multiple validation steps, we achieve highly accurate annotations. We also release intermediate sensor frames, which are important for tracking, prediction and object detection as shown in Section 4.2. At capture frequencies of 12Hz, 13Hz and 20Hz for camera, radar and lidar, this makes our dataset unique. Only the Waymo Open dataset provides a similarly high capture frequency of 10Hz (Table 1).

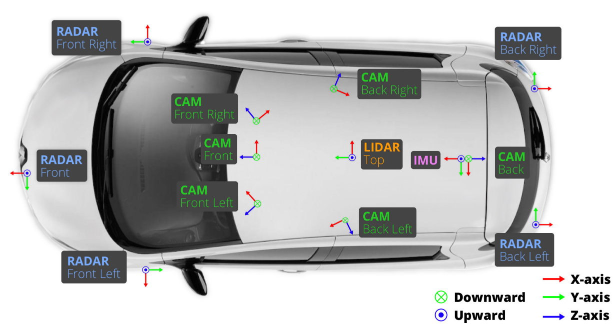

Figure 4. Sensor setup for our data collection platform.

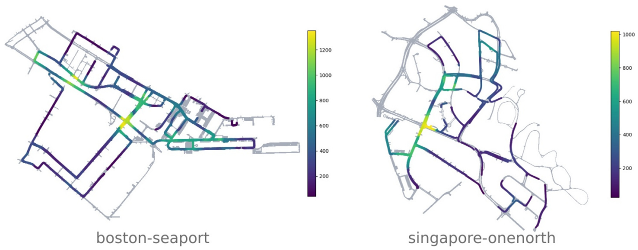

Figure 5. Spatial data coverage for two nuScenes locations. Colors indicate the number of keyframes with ego vehicle poses within a 100m radius across all scenes.

Annotation statistics. Our dataset has 23 categories including different vehicles, types of pedestrians, mobility devices and other objects (Figure 9-SM). We present statistics on geometry and frequencies of different classes (Figure 10-SM). Per keyframe there are 7 pedestrians and 20 vehicles on average. Moreover, 40k keyframes were taken from four different scene locations (Boston: 55%, SG-OneNorth: 21.5%, SG-Queenstown: 13.5%, SG-HollandVillage: 10%) with various weather and lighting conditions (rain: 19.4%, night: 11.6%). Due to the finegrained classes in nuScenes, the dataset shows severe class imbalance with a ratio of 1:10k for the least and most common class annotations (1:36 in KITTI). This encourages the community to explore this long tail problem in more depth.

Figure 5 shows spatial coverage across all scenes. We see that most data comes from intersections. Figure 11-SM shows that *car* annotations are seen at varying distances and as far as 80m from the ego-vehicle. Box orientation is also varying, with the most number in vertical and horizontal angles for cars as expected due to parked cars and cars in the same lane. Lidar and radar points statistics inside each box annotation are shown in Figure 15-SM. Annotated objects contain up to 100 lidar points even at a radial distance of 80m and at most 12k lidar points at 3m. At the same time they contain up to 40 radar returns at 10m and 10 at 50m. The radar range far exceeds the lidar range at up to 200m.

Scene reconstruction. To analyze our localization quality, we compute the merged pointcloud of an entire scene by registering approximately 800 pointclouds in global coordinates. We remove points corresponding to the ego vehicle and assign to each point the mean color value of the closest camera pixel that the point is reprojected to. Scene reconstruction can be seen in Figure 6 which demonstrate accurate synchronization and localization.

3. Tasks & Metrics

The multimodal nature of nuScenes supports a multitude of tasks including detection, tracking, prediction & localization. Here we present the detection and tracking tasks and metrics. We define the *detection* task to only operate on sensor data between $[t-0.5, t]$ seconds for an object at time t, whereas the *tracking* task operates on data between $[0, t]$.

3.1. Detection

The nuScenes detection task requires detecting 10 object classes with 3D bounding boxes, attributes (e.g. sitting vs. standing), and velocities. The 10 classes are a subset of all 23 classes annotated in nuScenes (Table 5-SM).

Average Precision metric. We use the Average Precision (AP) metric [31, 25], but define a match by thresholding the 2D center distance d on the ground plane instead of intersection over union (IOU). This is done in order to decouple detection from object size and orientation but also because objects with small footprints, like pedestrians and bikes, if detected with a small translation error, give 0 IOU (Figure 8). This makes it hard to compare the performance of vision-only methods which tend to have large localization errors [65].

We then calculate AP as the normalized area under the precision recall curve for recall and precision over 10%. Operating points where recall or precision is less than 10% are removed in order to minimize the impact of noise commonly seen in low precision and recall regions. If no operating point in this region is achieved, the AP for that class is set to zero. We then average over matching thresholds of $\mathbb{D} = \{0.5, 1, 2, 4\}$ meters and the set of classes $\mathbb{C}$:

$$\mathrm{mAP} = \frac{1}{|\mathbb{C}||\mathbb{D}|} \sum_{c \in \mathbb{C}} \sum_{d \in \mathbb{D}} \mathrm{AP}_{c,d} \tag{1}$$

True Positive metrics. In addition to AP, we measure a set of *True Positive metrics* (TP metrics) for each prediction that was matched with a ground truth box. All TP metrics are calculated using $d = 2$m center distance during matching, and they are all designed to be positive scalars. In the proposed metric, the TP metrics are all in native units (see below) which makes the results easy to interpret and compare. Matching and scoring happen independently per class and each metric is the average of the cumulative mean at each achieved recall level above 10%. If 10% recall is not achieved for a particular class, all TP errors for that class are set to 1. The following TP errors are defined:

Average Translation Error (ATE) is the Euclidean center distance in 2D (units in $meters$). Average Scale Error (ASE) is the 3D intersection over union (IOU) after aligning orientation and translation $(1 - IOU)$. Average Orientation Error (AOE) is the smallest yaw angle difference

Figure 6. Sample scene reconstruction given lidar points and camera images. We project the lidar points in an image plane with colors assigned based on the pixel color from camera data.

between prediction and ground truth ($radians$). All angles are measured on a full 360° period except for barriers where they are measured on a 180° period. Average Velocity Error (AVE) is the absolute velocity error as the L2 norm of the velocity differences in 2D (m/s). Average Attribute Error (AAE) is defined as 1 minus attribute classification accuracy ($1 - acc$). Finally, the mTP is calculated as:

$$\text{mTP} = \frac{1}{|\mathbb{C}|} \sum_{c \in \mathbb{C}} \text{TP}_c \tag{2}$$

We omit measurements for classes where they are not well defined: AVE for cones and barriers since they are stationary; AOE of cones since they do not have a well defined orientation; and AAE for cones and barriers since there are no attributes defined on these classes.

nuScenes detection score. mAP with a threshold on IOU is perhaps the most popular metric for object detection [31, 18, 20]. However, this metric can not capture all aspects of the nuScenes detection tasks, like velocity and attribute estimation. Further, it couples location, size and orientation estimates. The ApolloScape [40] 3D car instance challenge disentangles these by defining thresholds for each error type and recall threshold. This results in 10×3 thresholds, making this approach complex, arbitrary and unintuitive. We propose instead consolidating the different error types into a scalar score: the nuScenes detection score (NDS).

$$\text{NDS} = \frac{1}{10}[5\,\text{mAP} + \sum_{\text{mTP} \in \mathbb{TP}} (1 - \min(1, \text{mTP}))] \tag{3}$$

Here mAP is mean Average Precision (1), and $\mathbb{TP}$ the set of the five mean True Positive metrics (2). Half of NDS is thus based on the detection performance while the other half quantifies the quality of the detections in terms of box location, size, orientation, attributes, and velocity. Since mAVE, mAOE and mATE can be larger than 1, we bound each metric between 0 and 1 in (3).

3.2. Tracking

In this section we present the tracking task setup and metrics. The focus of the tracking task is to track all detected objects in a scene. All detection classes defined in Section 3.1 are used, except the static classes: *barrier*, *construction* and *trafficcone*.

AMOTA and AMOTP metrics. Weng and Kitani [72] presented a similar 3D MOT benchmark on KITTI [31]. They point out that traditional metrics do not take into account the confidence of a prediction. Thus they develop AMOTA and AMOTP which average MOTA and MOTP across all recall thresholds. In the updated formulation[4], MOTA is augmented by a term to adjust for the respective recall (which we call MOTAR):

$$AMOTA = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} MOTAR$$

$$MOTAR = \max\left(0,\ 1\ -\ \alpha\, \frac{IDS_r + FP_r + FN_r - (1-r)P}{rP}\right)$$

This is to guarantee that MOTAR values span the entire $[0, 1]$ range. Due to the difficulty of nuScenes, we introduce an additional factor $\alpha = 0.2$ to achieve non-zero values. For the same reason, traditional MOTA is often zero on nuScenes. We perform 40-point interpolation in the recall range $[0.1, 1]$ (the recall values are denoted as $\mathcal{R}$). AMOTA is the main metric and challenge winners are determined according to it.

Traditional metrics. We also use traditional tracking metrics such as multi object tracking accuracy (MOTA) and multi object tracking precision (MOTP) [4], false alarms per frame, mostly tracked trajectories, mostly lost trajectories, false positives, false negatives, identity switches, and track fragmentations. Similar to [72], we try all recall thresholds and then use the threshold that achieves highest MOTAR.

TID and LGD metrics. In addition, we devise two novel metrics: Track initialization duration (TID) and longest gap duration (LGD). Some trackers require a fixed window of past sensor readings or perform poorly without a good initialization. TID measures the duration from the beginning of the track until the time an object is first detected. LGD computes the longest duration of *any* detection gap in a track. If an object is not tracked, we assign the entire track duration as TID and LGD. For both metrics, we compute the average over all tracks. These metrics are relevant for AVs as many short-term track fragmentations may be more acceptable than missing an object for several seconds.

4. Experiments

In this section we present object detection and tracking experiments on the nuScenes dataset, analyze their characteristics and suggest avenues for future research.

4.1. Baselines

We present a number of baselines with different modalities for detection and tracking.

Lidar detection baseline. To demonstrate the performance of a leading algorithm on nuScenes, we train a lidar-only 3D object detector, PointPillars [49]. We take advantage of temporal data available in nuScenes by accumulating lidar sweeps for a richer pointcloud as input. A single network was trained for all classes. The network was modified to also learn velocities as an additional regression target for each 3D box. We set the box attributes to the most common attribute for each class in the training data.

Image detection baseline. To examine image-only 3D object detection, we re-implement the Orthographic Feature

[4]The authors of [72] will publish this shortly.

Transform (OFT) [65] method. A single OFT network was used for all classes. We modified the original OFT to use a SSD detection head and confirmed that this matched published results on KITTI. The network takes in a single image from which the full 360° predictions are combined together from all 6 cameras using NMS. We set the box velocity to zero and attributes to the most common attribute for each class in the train data.

Detection challenge results. We compare the results of the top submissions to the nuScenes detection challenge 2019. Among all submissions, Megvii [84] gave the best performance. It is a lidar-based class-balanced multi-head network with sparse 3D convolutions. Among image-only submissions, MonoDIS [66] was the best, significantly outperforming our image baseline and even some lidar-based methods. It uses a novel disentangling 2D and 3D detection loss. Note that the top methods all performed importance sampling, which shows the importance of addressing the class imbalance problem.

Tracking baselines. We present several baselines for tracking from camera and lidar data. From the detection challenge, we pick the best performing lidar method (Megvii [84]), the fastest reported method at inference time (PointPillars [49]), as well as the best performing camera method (MonoDIS [66]). Using the detections from each method, we setup baselines using the tracking approach described in [72]. We provide detection and tracking results for each of these methods on the train, val and test splits to facilitate more systematic research. The nuScenes tracking challenge 2019 is currently ongoing and we will add it in a future version of this paper.

4.2. Analysis

Here we analyze the properties of the methods presented in Section 4.1, as well as the dataset and matching function.

The case for a large benchmark dataset. One of the contributions of nuScenes is the dataset size, and in particular the increase compared to KITTI (Table 1). Here we examine the benefits of the larger dataset size. We train PointPillars [49], OFT [65] and an additional image baseline, SSD+3D, with varying amounts of training data. SSD+3D has the same 3D parametrization as MonoDIS [66], but use a single stage design [51]. For this ablation study we train PointPillars with 6x fewer epochs and a one cycle optimizer schedule [67] to cut down the training time. Our main finding is that the *method ordering changes* with the amount of data (Figure 7). In particular, PointPillars performs similar to SSD+3D at data volumes commensurate with KITTI, but as more data is used, it is clear that PointPillars is stronger. This suggests that the full potential of complex algorithms can only be see verified with a bigger and more diverse training set. A similar conclusion was reached by [54, 57] with [57] suggesting that the KITTI leaderboard reflects the data aug. method rather than the actual algorithms.

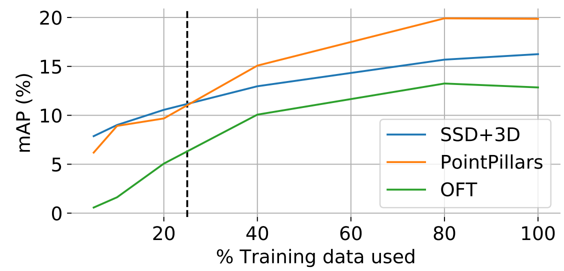

Figure 7. Amount of training data vs. mean Average Precision (mAP) on the val set of nuScenes. The dashed black line corresponds to the amount of training data in KITTI [31].

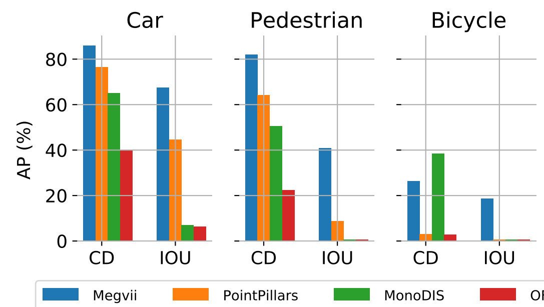

Figure 8. Average precision vs. matching function. CD: Center distance. IOU: Intersection over union. We use IOU $= 0.7$ for *car* and IOU $= 0.5$ for *pedestrian* and *bicycle* following KITTI [31]. We use CD $= 2m$ for the TP metrics in Section 3.1.

The importance of the matching function. We compare performance of published methods (Table 4) when using our proposed 2m center-distance matching versus the IOU matching used in KITTI. As expected, when using IOU matching, small object like pedestrians and bicycles fail to achieve above 0 AP, making ordering impossible (Figure 8). In contrast, center distance matching declares the more sophisticated MonoDIS a clear winner. The impact is smaller for the car class, but also in this case it is hard to resolve the difference between MonoDIS and OFT.

The matching function also changes the balance between lidar and image based methods. In fact, the ordering switches when using center distance matching to favour MonoDIS over both lidar-based methods on the bicycle class (Figure 8). This makes sense since the thin structures of bicycles make them difficult to detect in lidar. We conclude that center distance matching is more approporate in order to rank image-based detection methods alongside lidar-based methods.

Multiple lidar sweeps improve performance. According to our evaluation protocol (Section 3.1), one is only allowed to use $0.5s$ of previous data to make a detection decision. This corresponds to 10 previous lidar sweeps since the lidar is sampled at 20Hz. We device a simple way of incorporating multiple pointclouds into the PointPillars baseline

Lidar sweeps	Pretraining	NDS (%)	mAP (%)	mAVE (m/s)
1	KITTI	31.8	21.9	1.21
5	KITTI	42.9	27.7	0.34
10	KITTI	**44.8**	28.8	0.30
10	ImageNet	**44.9**	28.9	0.31
10	None	44.2	27.6	0.33

Table 3. PointPillars [49] detection performance on the val set. We can see that more lidar sweeps lead to a significant performance increase and that pretraining with ImageNet is on par with KITTI.

and investigate the performance impact. Accumulation is implemented by moving all pointclouds to the coordinate system of the keyframe and appending a scalar time-stamp to each point indicating the time delta in seconds from the keyframe. The encoder includes the time delta as an extra decoration for the lidar points. Aside from the advantage of richer pointclouds, this also provides temporal information, which helps the network in localization and enables velocity prediction. We experiment with using 1, 5, and 10 lidar sweeps. The results show that both detection and velocity estimates improve with an increasing number of lidar sweeps but with diminishing rate of return (Table 3).

Which sensor is most important? An important question for AVs is which sensors are required to achieve the best detection performance. Here we compare the performance of leading lidar and image detectors. We focus on these modalities as there are no competitive radar-only methods in the literature and our preliminary study with PointPillars on radar data did not achieve promising results. We compare PointPillars, which is a fast and light lidar detector with MonoDIS, a top image detector (Table 4). The two methods achieve similar mAP (30.5% vs. 30.4%), but PointPillars has higher NDS (45.3% vs. 38.4%). The close mAP is, of itself, notable and speaks to the recent advantage in 3D estimation from monocular vision. However, as discussed above the differences would be larger with an IOU based matching function.

Class specifc performance is in Table 6-SM. PointPillars was stronger for the two most common classes: cars (68.4% vs. 47.8% AP), and pedestrians (59.7% vs. 37.0% AP). MonoDIS, on the other hand, was stronger for the smaller classes bicycles (24.5% vs. 1.1% AP) and cones (48.7% vs. 30.8% AP). This is expected since 1) bicycles are thin objects with typically few lidar returns and 2) traffic cones are easy to detect in images, but small and easily overlooked in a lidar pointcloud. 3) MonoDIS applied importance sampling during training to boost rare classes. With similar detection performance, why was NDS lower for MonoDIS? The main reasons are the average translation errors (52cm vs. 74cm) and velocity errors ($1.55 m/s$ vs. $0.32 m/s$), both as expected. MonoDIS also had larger scale errors with mean IOU 74% vs. 71% but the difference is small, suggesting the strong ability for image-only methods to infer size from appearance.

Method	NDS (%)	mAP (%)	mATE (m)	mASE (1-iou)	mAOE (rad)	mAVE (m/s)	mAAE (1-acc)
OFT [65]†	21.2	12.6	0.82	0.36	0.85	1.73	0.48
SSD+3D†	26.8	16.4	0.90	0.33	0.62	1.31	0.29
MDIS [66]†	38.4	30.4	0.74	0.26	0.55	1.55	0.13
PP [49]	45.3	30.5	0.52	0.29	0.50	0.32	0.37
Megvii [84]	**63.3**	**52.8**	**0.30**	**0.25**	**0.38**	**0.25**	**0.14**

Table 4. Object detection results on the test set of nuScenes. PointPillars, OFT and SSD+3D are baselines provided in this paper, other methods are the top submissions to the nuScenes detection challenge leaderboard. (†) uses monocular camera images as input. All other methods use lidar. PP: PointPillars [49], MDIS: MonoDIS [66].

The importance of pre-training. Using the lidar baseline we examine the importance of pre-training when training a detector on nuScenes. No pretraining means weights are initialized randomly using a uniform distribution as in [37]. ImageNet [20] pretraining [45] uses a backbone that was first trained to accurately classify images. KITTI [31] pretraining uses a backbone that was trained on the lidar pointclouds to predict 3D boxes. Interestingly, while the KITTI pretrained network did converge faster, the final performance of the network only marginally varied between different pretrainings (Table 3). One explanation may be that while KITTI is close in domain, the size is not large enough to train a general purpose backbone.

Better detection gives better tracking. Weng and Kitani [72] presented a simple baseline that achieved state-of-the-art 3d tracking results using powerful detections on KITTI. Here we analyze whether better detections also imply better tracking performance on nuScenes, using the image and lidar baselines presented in Section 4.1. Megvii, PointPillars and MonoDIS achieve an AMOTA of 27.9%, 13.1% and 10.3%, and an AMOTP of $1.50m$, $1.69m$ and $1.79m$ on the val set. Compared to the mAP and NDS detection results in Table 4, the ranking is indeed preserved. While the performance is correlated across most metrics, we notice that MonoDIS has the shortest LGD and highest number of track fragmentations. This may indicate that despite the lower performance, image-based methods are less likely to miss an object for a protracted period of time.

5. Conclusion

In this paper we present the nuScenes dataset, detection and tracking tasks, metrics, baselines and results. This is the first dataset collected from an AV on public roads and that contains the full 360° sensor suite (lidar, images, and radar). nuScenes has the largest collection of 3D box annotations of any previously released dataset. To spur research on 3D object detection for AVs, we introduce a new detection metric that balances all aspects of detection performance. We demonstrate novel adaptations of leading lidar and image object detectors and trackers on nuScenes. Future work will add image-level and point-level semantic labels.

Acknowledgements. The nuScenes dataset was annotated by Scale.ai and we thank Alexandr Wang and Dave Morse for their support. We thank Sun Li and Karen Ngo at nuTonomy for data inspection and quality control, Bassam Helou and Thomas Roddick for OFT baseline results, and Deshraj Yadav and Rishabh Jain from EvalAI [29] for setting up the nuScenes challenges.

References

[1] Giancarlo Alessandretti, Alberto Broggi, and Pietro Cerri. Vehicle and guard rail detection using radar and vision data fusion. *IEEE Transactions on Intelligent Transportation Systems*, 2007. 1

[2] Dan Barnes, Will Maddern, and Ingmar Posner. Exploiting 3d semantic scene priors for online traffic light interpretation. In *IVS*, 2015. 2

[3] Klaus Bengler, Klaus Dietmayer, Berthold Farber, Markus Maurer, Christoph Stiller, and Hermann Winner. Three decades of driver assistance systems: Review and future perspectives. *ITSM*, 2014. 1

[4] Keni Bernardin, Alexander Elbs, and Rainer Stiefelhagen. Multiple object tracking performance metrics and evaluation in a smart room environment. In *ECCV Workshop on Visual Surveillance*, 2006. 6

[5] Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi. Monoloco: Monocular 3d pedestrian localization and uncertainty estimation. In *ICCV*, 2019. 2

[6] José-Luis Blanco-Claraco, Francisco-Ángel Moreno-Dueas, and Javier González-Jiménez. The Málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario. *IJRR*, 2014. 3

[7] Martin Brossard, Axel Barrau, and Silvère Bonnabel. AI-IMU Dead-Reckoning. *arXiv preprint arXiv:1904.06064*, 2019. 4

[8] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008. 2, 3

[9] Sergio Casas, Cole Gulino, Renjie Liao, and Raquel Urtasun. Spatially-aware graph neural networks for relational behavior forecasting from sensor data. *arXiv preprint arXiv:1910.08233*, 2019. 2

[10] 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. 2, 3, 4

[11] Z. Che, G. Li, T. Li, B. Jiang, X. Shi, X. Zhang, Y. Lu, G. Wu, Y. Liu, and J. Ye. D^2-City: A large-scale dashcam video dataset of diverse traffic scenarios. *arXiv:1904.01975*, 2019. 3

[12] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In *NIPS*, 2015. 1

[13] Xiaozhi Chen, Laustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *CVPR*, 2016. 1

[14] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, 2017. 2

[15] Yiping Chen, Jingkang Wang, Jonathan Li, Cewu Lu, Zhipeng Luo, Han Xue, and Cheng Wang. Lidar-video driving dataset: Learning driving policies effectively. In *CVPR*, 2018. 3

[16] Yukyung Choi, Namil Kim, Soonmin Hwang, Kibaek Park, Jae Shin Yoon, Kyounghwan An, and In So Kweon. KAIST multi-spectral day/night data set for autonomous and assisted driving. *IEEE Transactions on Intelligent Transportation Systems*, 2017. 3

[17] Z. J. Chong, B. Qin, T. Bandyopadhyay, M. H. Ang, E. Frazzoli, and D. Rus. Synthetic 2d lidar for precise vehicle localization in 3d urban environment. In *ICRA*, 2013. 4

[18] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 3, 4, 6, 12

[19] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 3

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 6, 8

[21] Thierry Deruyttere, Simon Vandenhende, Dusan Grujicic, Luc Van Gool, and Marie-Francine Moens. Talk2car: Taking control of your self-driving car. *arXiv preprint arXiv:1909.10838*, 2019. 2

[22] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 2012. 3

[23] Markus Enzweiler and Dariu M. Gavrila. Monocular pedestrian detection: Survey and experiments. *PAMI*, 2009. 3

[24] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. A mobile vision system for robust multi-person tracking. In *CVPR*, 2008. 3

[25] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 2010. 5

[26] Hehe Fan and Yi Yang. PointRNN: Point recurrent neural network for moving point cloud processing. *arXiv preprint arXiv:1910.08287*, 2019. 2

[27] Di Feng, Christian Haase-Schuetz, Lars Rosenbaum, Heinz Hertlein, Fabian Duffhauss, Claudius Glaeser, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *arXiv preprint arXiv:1902.07830*, 2019. 2

[28] D. Feng, C. Haase-Schuetz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *arXiv:1902.07830*, 2019. 2

[29] EvalAI: Towards Better Evaluation Systems for AI Agents. D. yadav and r. jain and h. agrawal and p. chattopadhyay and t. singh and a. jain and s. b. singh and s. lee and d. batra. *arXiv:1902.03570*, 2019. 9

[30] Andrea Frome, German Cheung, Ahmad Abdulkader, Marco Zennaro, Bo Wu, Alessandro Bissacco, Hartwig Adam, Hartmut Neven, and Luc Vincent. Large-scale privacy protection in google street view. In *ICCV*, 2009. 12

[31] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012. 2, 3, 4, 5, 6, 7, 8, 12

[32] Neuhold Gerhard, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The Mapillary Vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 2, 3

[33] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mhlegg, Sebastian Dorn, Tiffany Fernandez, Martin Jnicke, Sudesh Mirashi, Chiragkumar Savani, Martin Sturm, Oleksandr Vorobiov, and Peter Schuberth. A2D2: AEV autonomous driving dataset. http://www.a2d2.audi, 2019. 3

[34] Hugo Grimmett, Mathias Buerki, Lina Paz, Pedro Pinies, Paul Furgale, Ingmar Posner, and Paul Newman. Integrating metric and semantic maps for vision-only automated parking. In *ICRA*, 2015. 2

[35] Junyao Guo, Unmesh Kurup, and Mohak Shah. Is it safe to drive? an overview of factors, challenges, and datasets for driveability assessment in autonomous driving. *arXiv:1811.11277*, 2018. 2

[36] Shirsendu Sukanta Halder, Jean-Francois Lalonde, and Raoul de Charette. Physics-based rendering for improving robustness to rain. In *ICCV*, 2019. 2

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 8

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 12

[39] Namdar Homayounfar, Wei-Chiu Ma, Shrinidhi Kowshika Lakshmikanth, and Raquel Urtasun. Hierarchical recurrent attention networks for structured online maps. In *CVPR*, 2018. 1

[40] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *arXiv:1803.06184*, 2018. 2, 3, 4, 6, 12

[41] Vijay John and Seiichi Mita. Rvnet: Deep sensor fusion of monocular camera and radar for image-based obstacle detection in challenging environments, 2019. 2

[42] Hojung Jung, Yuki Oto, Oscar M. Mozos, Yumi Iwashita, and Ryo Kurazume. Multi-modal panoramic 3d outdoor datasets for place categorization. In *IROS*, 2016. 3

[43] 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. https://level5.lyft.com/dataset/, 2019. 2, 3, 4

[44] Jaekyum Kim, Jaehyung Choi, Yechol Kim, Junho Koh, Chung Choo Chung, and Jun Won Choi. Robust camera lidar sensor fusion via deep gated information fusion network. In *IVS*, 2018. 1

[45] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 8

[46] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven Waslander. Joint 3d proposal generation and object detection from view aggregation. In *IROS*, 2018. 2

[47] Charles-Éric Noël Laflamme, François Pomerleau, and Philippe Giguère. Driving datasets literature review. *arXiv preprint arXiv:1910.11968*, 2019. 2

[48] Nitheesh Lakshminarayana. Large scale multimodal data capture, evaluation and maintenance framework for autonomous driving datasets. In *ICCVW*, 2019. 2

[49] 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*, 2019. 1, 2, 6, 7, 8, 13, 14, 15

[50] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *ECCV*, 2018. 2

[51] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 7

[52] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents http://apolloscape.auto/tracking.html. In *AAAI*, 2019. 3

[53] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *IJRR*, 2017. 2, 3

[54] Gregory P. Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K. Wellington. LaserNet: An efficient probabilistic 3D object detector for autonomous driving. In *CVPR*, 2019. 7

[55] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *CVPR*, 2017. 1

[56] Luk Neumann, Michelle Karg, Shanshan Zhang, Christian Scharfenberger, Eric Piegert, Sarah Mistr, Olga Prokofyeva, Robert Thiel, Andrea Vedaldi, Andrew Zisserman, and Bernt Schiele. Nightowls: A pedestrians at night dataset. In *ACCV*, 2018. 3

[57] Jiquan Ngiam, Benjamin Caine, Wei Han, Brandon Yang, Yuning Chai, Pei Sun, Yin Zhou, Xi Yi, Ouais Alsharif, Patrick Nguyen, Zhifeng Chen, Jonathon Shlens, and Vijay Vasudevan. Starnet: Targeted computation for object detection in point clouds. *arXiv preprint arXiv:1908.11069*, 2019. 7

[58] Farzan Erlik Nowruzi, Prince Kapoor, Dhanvin Kolhatkar, Fahed Al Hassanat, Robert Laganiere, and Julien Rebut. How much real data do we actually need: Analyzing object detection performance using synthetic and real data. In *ICML Workshop on AI for Autonomous Driving*, 2019. 2

[59] Abhishek Patil, Srikanth Malla, Haiming Gang, and Yi-Ting Chen. The H3D dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes. In *ICRA*, 2019. 2, 3, 4, 12

[60] Quang-Hieu Pham, Pierre Sevestre, Ramanpreet Singh Pahwa, Huijing Zhan, Chun Ho Pang, Yuda Chen, Armin Mustafa, Vijay Chandrasekhar, and Jie Lin. A*3D Dataset: Towards autonomous driving in challenging environments. *arXiv:1909.07541*, 2019. 3

[61] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum pointnets for 3d object detection from RGB-D data. In *CVPR*, 2018. 2

[62] Akshay Rangesh and Mohan M. Trivedi. Ground plane polling for 6dof pose estimation of objects on the road. In *arXiv:1811.06666*, 2018. 1

[63] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 12

[64] Nicholas Rhinehart, Rowan McAllister, Kris M. Kitani, and Sergey Levine. PRECOG: Predictions conditioned on goals in visual multi-agent scenarios. In *ICCV*, 2019. 2, 4

[65] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. In *BMVC*, 2019. 1, 2, 5, 7, 8, 13

[66] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel Lopez-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. *ICCV*, 2019. 2, 7, 8, 15

[67] Leslie N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018. 7

[68] 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

[69] Ziyan Wang, Buyu Liu, Samuel Schulter, and Manmohan Chandraker. Dataset for high-level 3d scene understanding of complex road scenes in the top-view. In *CVPRW*, 2019. 2

[70] Zining Wang, Wei Zhan, and Masayoshi Tomizuka. Fusing birds eye view lidar point cloud and front view camera image for 3d object detection. In *IVS*, 2018. 2

[71] Waymo. Waymo Open Dataset: An autonomous driving dataset, 2019. 3

[72] Xinshuo Weng and Kris Kitani. A baseline for 3d multi-object tracking. *arXiv preprint arXiv:1907.03961*, 2019. 6, 7, 8

[73] L. Woensel and G. Archer. Ten technologies which could change our lives. *European Parlimentary Research Service*, 2015. 1

[74] Christian Wojek, Stefan Walk, and Bernt Schiele. Multi-cue onboard pedestrian detection. In *CVPR*, 2009. 3

[75] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *CVPR*, 2018. 1

[76] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *CVPR*, 2018. 2

[77] Bin Yang, Ming Liang, and Raquel Urtasun. HDNET: Exploiting HD maps for 3d object detection. In *CoRL*, 2018. 2

[78] Yangyang Ye, Chi Zhang, Xiaoli Hao, Houjin Chen, and Zhaoxiang Zhang. SARPNET: Shape attention regional proposal network for lidar-based 3d object detection. *Neurocomputing*, 2019. 2

[79] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving video database with scalable annotation tooling. *arXiv:1805.04687*, 2018. 2, 3

[80] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *arXiv preprint arXiv:1906.05113*, 2019. 2

[81] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *SPL*, 23(10), 2016. 12

[82] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In *CVPR*, 2017. 3

[83] Hao Zhou and Jorge Laval. Longitudinal motion planning for autonomous vehicles and its impact on congestion: A survey. *arXiv preprint arXiv:1910.06070*, 2019. 2

[84] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv:1908.09492*, 2019. 2, 7, 8

[85] Jing Zhu and Yi Fang. Learning object-specific distance from a monocular image. In *ICCV*, 2019. 2

A. The nuScenes dataset

In this section we provide more details on the nuScenes dataset, the sensor calibration, privacy protection approach, data format, class mapping and annotation statistics.

Sensor calibration. To achieve a high quality multi-sensor dataset, careful calibration of sensor intrinsic and extrinsic parameters is required. These calibration parameters are updated around twice per week over the data collection period of 6 months. Here we describe how we perform sensor calibration for our data collection platform to achieve a high-quality multimodal dataset. Specifically, we carefully calibrate the extrinsics and intrinsics of every sensor. We express extrinsic coordinates of each sensor to be relative to the *ego frame*, i.e. the midpoint of the rear vehicle axle. The most relevant steps are described below:

- Lidar extrinsics: We use a laser liner to accurately measure the relative location of the lidar to the ego frame.
- Camera extrinsics: We place a cube-shaped calibration target in front of the camera and lidar sensors. The calibration target consists of three orthogonal planes with known patterns. After detecting the patterns we compute the transformation matrix from camera to lidar by aligning the planes of the calibration target. Given the lidar to ego frame transformation computed above, we compute the camera to ego frame transformation.
- Radar extrinsics: We mount the radar in a horizontal position. Then we collect radar measurements by driving on public roads. After filtering radar returns for moving objects, we calibrate the yaw angle using a brute force approach to minimize the compensated range rates for static objects.
- Camera intrinsic calibration: We use a calibration target board with a known set of patterns to infer the intrinsic and distortion parameters of the camera.

Privacy protection. It is our priority to protect the privacy of third parties. As manual labeling of faces and license plates is prohibitively expensive for 1.4M images, we use state-of-the-art object detection techniques. Specifically for plate detection, we use Faster R-CNN [63] with ResNet-101 backbone [38] trained on Cityscapes [18][5]. For face detection, we use [81][6]. We set the classification threshold to achieve an extremely high recall (similar to [30]). To increase the precision, we remove predictions that do not overlap with the reprojections of the known *pedestrian* and *vehicle* boxes in the image. Eventually we use the predicted boxes to blur faces and license plates in the images.

[5]https://github.com/bourdakos1/Custom-Object-Detection
[6]https://github.com/TropComplique/mtcnn-pytorch

General nuScenes class	Detection class	Tracking class
animal	void	void
debris	void	void
pushable_pullable	void	void
bicycle_rack	void	void
ambulance	void	void
police	void	void
barrier	barrier	void
bicycle	bicycle	bicycle
bus.bendy	bus	bus
bus.rigid	bus	bus
car	car	car
construction	construction_vehicle	void
motorcycle	motorcycle	motorcycle
adult	pedestrian	pedestrian
child	pedestrian	pedestrian
construction_worker	pedestrian	pedestrian
police_officer	pedestrian	pedestrian
personal_mobility	void	void
stroller	void	void
wheelchair	void	void
trafficcone	traffic_ cone	void
trailer	trailer	trailer
truck	truck	truck

Table 5. Mapping from general classes in nuScenes to the classes used in the detection and tracking challenges. Note that for brevity we omit most prefixes for the general nuScenes classes.

Data format. Contrary to most existing datasets [31, 59, 40], we store the annotations and metadata (e.g. localization, timestamps, calibration data) in a relational database which avoids redundancy and allows for efficient access. The nuScenes devkit, taxonomy and annotation instructions are available online[7].

Class mapping. The nuScenes dataset comes with annotations for 23 classes. Since some of these only have a handful of annotations, we merge similar classes and remove classes that have less than 10000 annotations. This results in 10 classes for our detection task. Out of these, we omit 3 classes that are mostly static for the tracking task. Table 5-SM shows the detection classes and tracking classes and their counterpart in the general nuScenes dataset.

Annotation statistics. We present more statistics on the annotations of nuScenes. Absolute velocities are shown in Figure 12-SM. The average speed for moving *car*, *pedestrian* and *bicycle* categories are 6.6, 1.3 and 4 m/s. Note that our data was gathered from urban areas which shows reasonable velocity range for these three categories.

We analyze the distribution of box annotations around the ego-vehicle for *car*, *pedestrian* and *bicycle* categories through a polar range density map as shown in Figure 13-SM. Here, the occurrence bins are log-scaled. Generally, the annotations are well-distributed surrounding the ego-vehicle. The annotations are also denser when they are nearer to the ego-vehicle. However, the *pedestrian* and *bi-*

[7]https://github.com/nutonomy/nuscenes-devkit

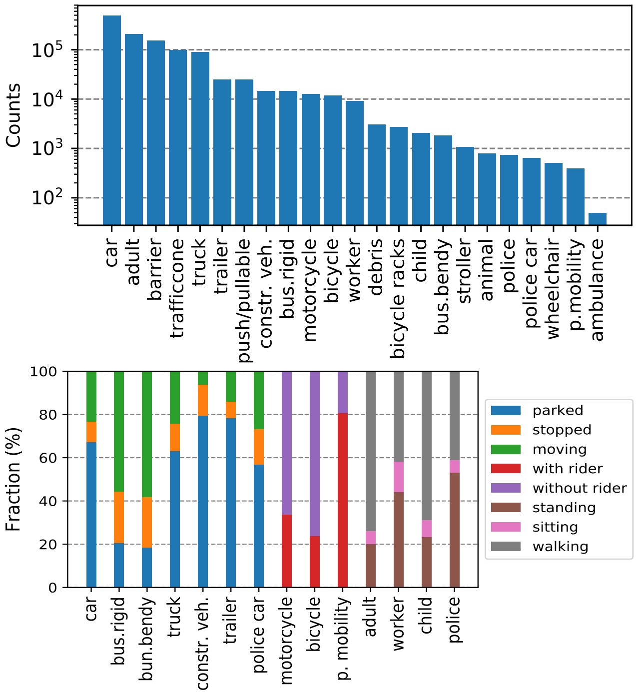

Figure 9. Top: Number of annotations per category. Bottom: Attributes distribution for selected categories. Cars and adults are the most frequent categories in our dataset, while ambulance is the least frequent. The attribute plot also shows some expected patterns: construction vehicles are rarely moving, pedestrians are rarely sitting while buses are commonly moving.

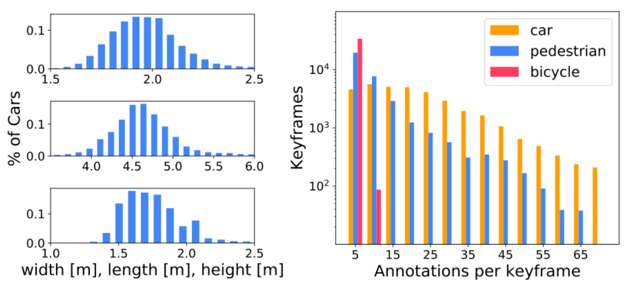

Figure 10. Left: Bounding box size distributions for *car*. Right: Category count in each keyframe for *car*, *pedestrian*, and *bicycle*.

cycle have less annotations above the 100m range. It can also be seen that the *car* category is denser in the front and back of the ego-vehicle, since most vehicles are following the same lane as the ego-vehicle.

In Section 2 we discussed the number of lidar points inside a box for all categories through a hexbin density plot, but here we present the number of lidar points of each category as shown in Figure 14-SM. Similarly, the occurrence bins are log-scaled. As can be seen, there are more lidar points found inside the box annotations for *car* at varying distances from the ego-vehicle as compared to *pedestrian* and *bicycle*. This is expected as cars have larger and more reflective surface area than the other two categories, hence more lidar points are reflected back to the sensor.

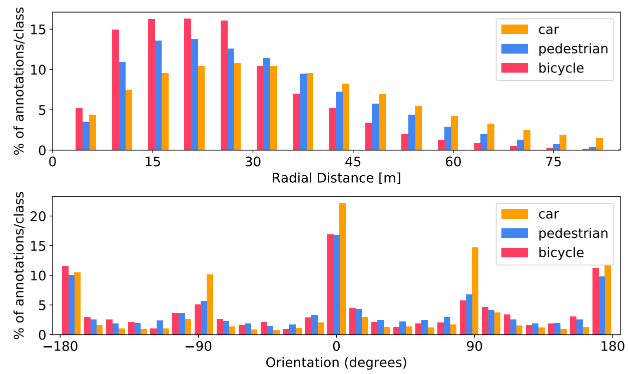

Figure 11. Top: radial distance of objects from the ego vehicle. Bottom: orientation of boxes in box coordinate frame.

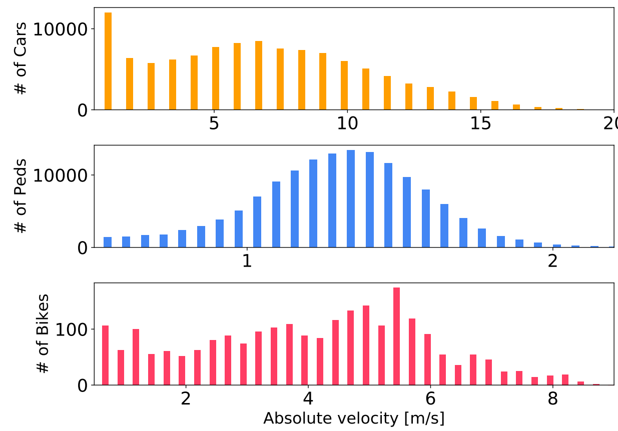

Figure 12. Absolute velocities. We only look at moving objects with speed > 0.5m/s.

B. Experiments

Here we provide additional details on training the lidar and image based 3D object detection baselines and provide more insights on the results.

PointPillars implementation details. For all experiments, our PointPillars [49] networks were trained using a pillar xy resolution of 0.25 meters and an x and y range of $[-50, 50]$ meters. The max number of pillars and batch size was varied with the number of lidar sweeps. For 1, 5, and 10 sweeps, we set the maximum number of pillars to 10000, 22000, and 30000 respectively and the batch size to 64, 64, and 48. All experiments were trained for 750 epochs. The initial learning rate was set to 10^{-3} and was reduced by a factor of 10 at epoch 600 and again at 700. Only ground truth annotations with one or more lidar points in the accumulated pointcloud were used as positive training examples. Since bikes inside of bike racks are not annotated individually and the evaluation metrics ignore bike racks, all lidar points inside bike racks were filtered out during training.

OFT implementation details. For each camera, the Orthographic Feature Transform [65] (OFT) baseline was

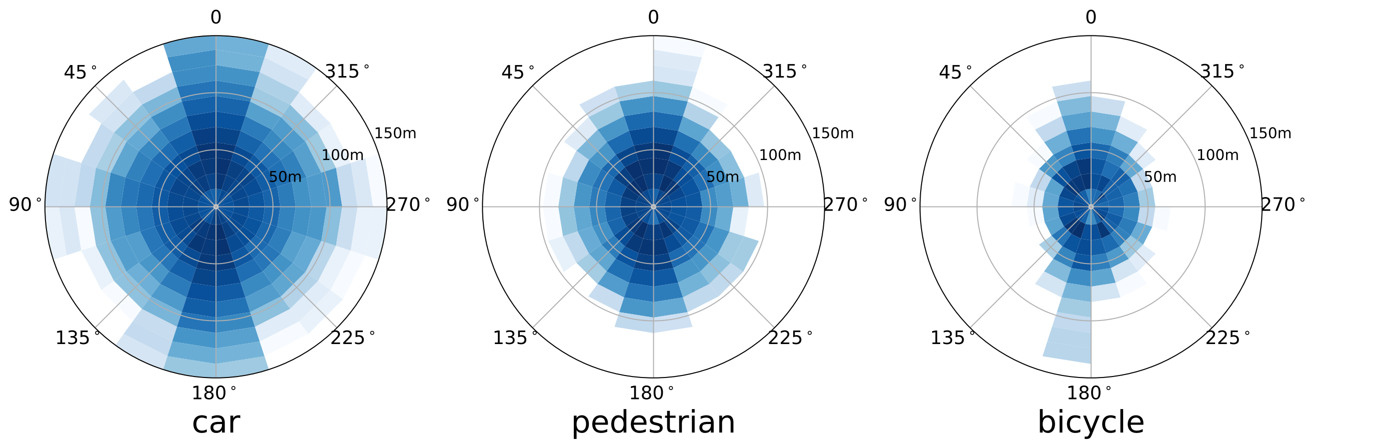

Figure 13. Polar log-scaled density map for box annotations where the radial axis is the distance from the ego-vehicle in meters and the polar axis is the yaw angle wrt to the ego-vehicle. The darker the bin is, the more box annotations in that area. Here, we only show the density up to 150m radial distance for all maps, but *car* would have annotations up to 200m.

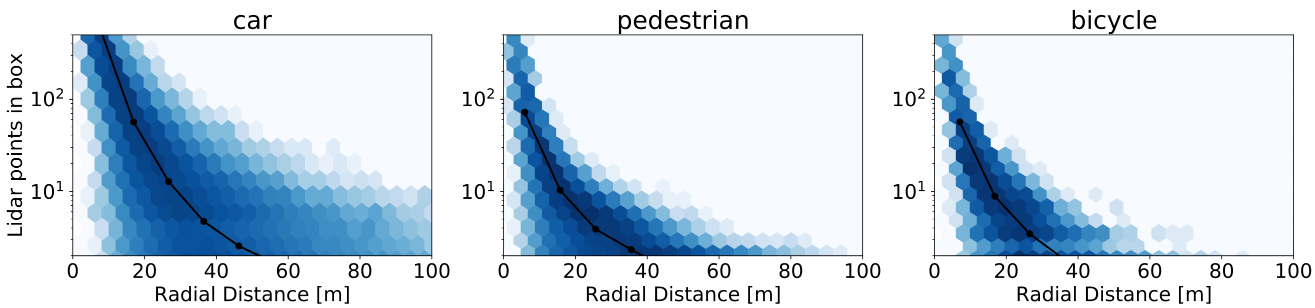

Figure 14. Hexbin log-scaled density plots of the number of lidar points inside a box annotation stratified by categories (*car*, *pedestrian* and *bicycle*.

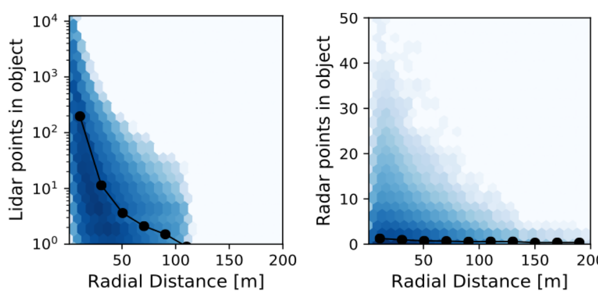

Figure 15. Hexbin log-scaled density plots of the number of lidar and radar points inside a box annotation. The black line represents the mean number of points for a given distance wrt the ego-vehicle.

trained on a voxel grid in each camera's frame with an lateral range of $[-40, 40]$ meters, a longitudinal range of $[0.1, 50.1]$ meters and a vertical range of $(-3, 1)$ meters. We trained only on annotations that were within 50 meters of the car's ego frame coordinate system's origin. Using the 'visibility' attribute in the nuScenes dataset, we also filtered out annotations that had visibility less than 40%. The network was trained for 60 epochs using a learning rate of 2×10^{-3} and used random initialization for the network weights (no ImageNet pretraining).

Per-class analysis. The per class performance of PointPillars [49] is shown in Table 6-SM (top) and Figure 17-SM. The network performed best overall on cars and pedestrians which are the two most common categories. The worst performing categories were bicycles and construction vehicles, two of the rarest categories that also present additional challenges. Construction vehicles pose a unique challenge due to their high variation in size and shape. While the translational error is similar for cars and pedestrians, the orientation error for pedestrians ($21°$) is higher than that of cars ($11°$). This smaller orientation error for cars is expected since cars have a greater distinction between their front and side profile relative to pedestrians. The vehicle velocity estimates are promising (e.g. 0.24 m/s AVE for the *car* class) considering the typical speed of a vehicle in the city would be 10 to 15 m/s.

Semantic map filtering. In Section 4.2 and Table 6-SM we show that the PointPillars baseline achieves only an AP of 1% on the *bicycle* class. However, when filtering both the predictions and ground truth to only include boxes on the semantic map prior[8], the AP increases to 30%. This observation can be seen in Figure 16-SM, where we plot the AP at different distances of the ground truth to the semantic

[8] Defined here as the union of roads and sidewalks.

PointPillars						
Class	**AP**	**ATE**	**ASE**	**AOE**	**AVE**	**AAE**
Barrier	38.9	0.71	0.30	0.08	N/A	N/A
Bicycle	1.1	0.31	0.32	0.54	0.43	0.68
Bus	28.2	0.56	0.20	0.25	0.42	0.34
Car	68.4	0.28	0.16	0.20	0.24	0.36
Constr. Veh.	4.1	0.89	0.49	1.26	0.11	0.15
Motorcycle	27.4	0.36	0.29	0.79	0.63	0.64
Pedestrian	59.7	0.28	0.31	0.37	0.25	0.16
Traffic Cone	30.8	0.40	0.39	N/A	N/A	N/A
Trailer	23.4	0.89	0.20	0.83	0.20	0.21
Truck	23.0	0.49	0.23	0.18	0.25	0.41
Mean	30.5	0.52	0.29	0.50	0.32	0.37
MonoDIS						
Class	**AP**	**ATE**	**ASE**	**AOE**	**AVE**	**AAE**
Barrier	51.1	0.53	0.29	0.15	N/A	N/A
Bicycle	24.5	0.71	0.30	1.04	0.93	0.01
Bus	18.8	0.84	0.19	0.12	2.86	0.30
Car	47.8	0.61	0.15	0.07	1.78	0.12
Constr. Veh.	7.4	1.03	0.39	0.89	0.38	0.15
Motorcycle	29.0	0.66	0.24	0.51	3.15	0.02
Pedestrian	37.0	0.70	0.31	1.27	0.89	0.18
Traffic Cone	48.7	0.50	0.36	N/A	N/A	N/A
Trailer	17.6	1.03	0.20	0.78	0.64	0.15
Truck	22.0	0.78	0.20	0.08	1.80	0.14
Mean	30.4	0.74	0.26	0.55	1.55	0.13

Table 6. Detailed detection performance for PointPillars [49] (top) and MonoDIS [66] (bottom) on the test set. AP: average precision averaged over distance thresholds (%), ATE: average translation error (m), ASE: average scale error (1-IOU), AOE: average orientation error (rad), AVE: average velocity error (m/s), AAE: average attribute error ($1 - acc.$), N/A: not applicable (Section 3.1). nuScenes Detection Score (NDS) = 45.3% (PointPillars) and 38.4% (MonoDIS).

map prior. As seen, the AP drops when the matched GT is farther from the semantic map prior. Again, this is likely because bicycles away from the semantic map tend to be parked and occluded with low visibility.

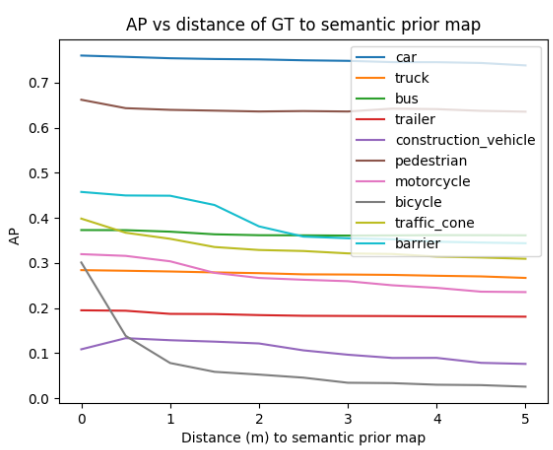

Figure 16. PointPillars [49] detection performance vs. semantic prior map location on the val set. For the best lidar network (10 lidar sweeps with ImageNet pretraining), the predictions and ground truth annotations were only included if within a given distance of the semantic prior map.

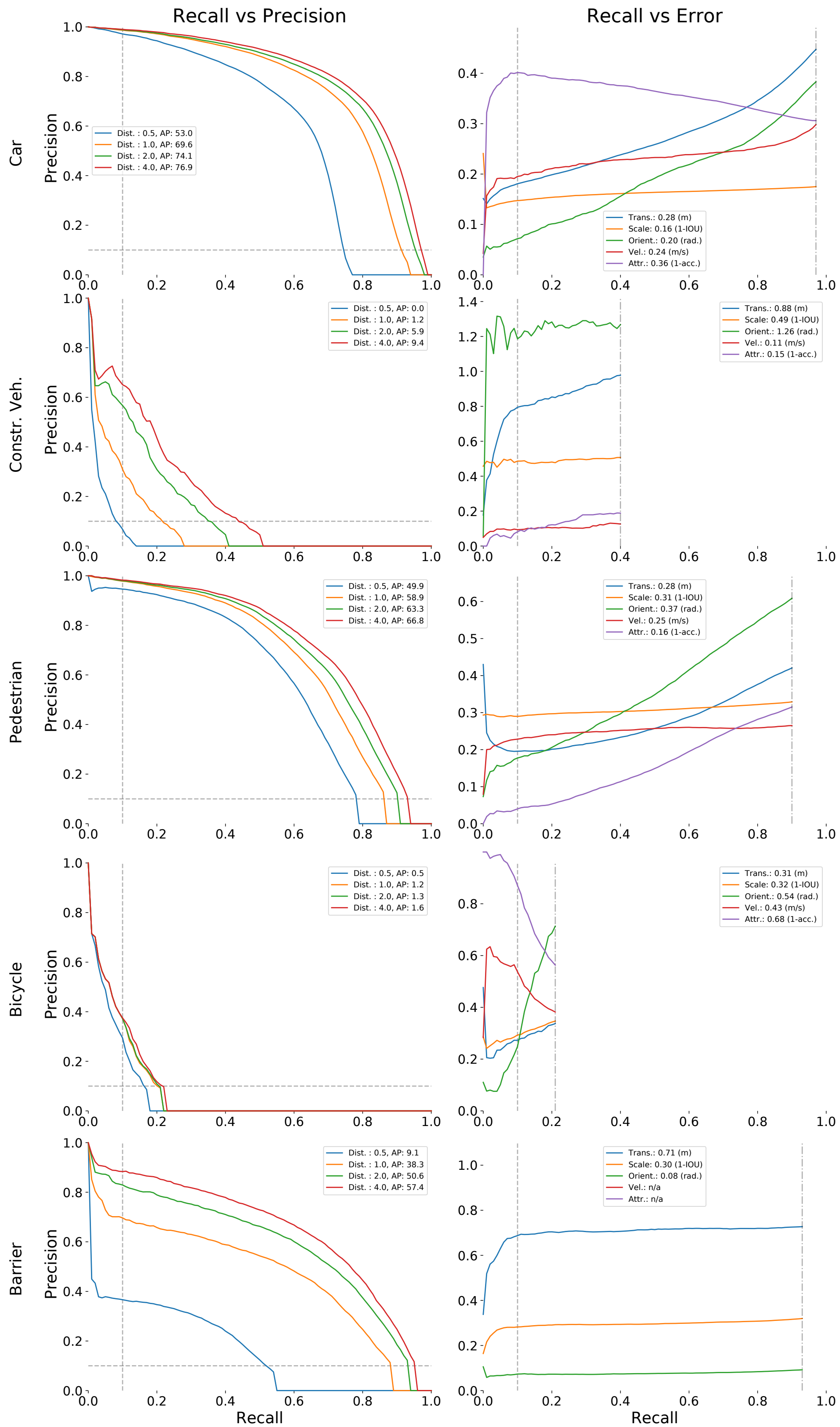

Figure 17. Per class results for PointPillars on the nuScenes test set taken from the detection leaderboard.