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 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.

1. Introduction

Autonomous driving has the potential to radically change the cityscape and save many human lives [78]. 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 (Table 1), there is a lack of multimodal datasets that exhibit the full set of challenges associated with building an autonomous driving perception system. We released the nuScenes dataset to address this gap.

1 nuscenes.org
and blind spots. And while there are several works that have proposed fusion methods based on cameras and lidar [48, 14, 64, 52, 81, 75, 29], PointPillars [51] showed a lidar-only method that performed on par with existing 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) [85, 8, 33, 55]. At the time of the initial nuScenes release, only a few datasets annotated objects using 3D boxes [32, 41, 61], and they did not provide the full sensor suite. Following the nuScenes release, there are now several sets which contain the full sensor suite (Table 1). 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 focused on particular operational design domains. More research is required on generalizing to “complex, cluttered and unseen environments” [36]. Hence 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 [82, 2, 35]. For example, one would expect to find cars on the road, but not on the sidewalk or inside buildings. With the notable exception of [45, 10], most 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. Similar to [85], nuScenes is a holistic scene understanding benchmark for AVs. It enables research on multiple tasks such as 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 [45] 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 online³.

Since the release, nuScenes has received strong interest from the AV community [90, 70, 50, 91, 9, 5, 68, 28, 49, 86, 89]. Some works extended our dataset to introduce new annotations for natural language object referral [22] and high-level scene understanding [74]. The detection challenge enabled lidar based and camera based detection works such as [90, 70], that improved over the state-of-the-art at the time of initial release [51, 69] by 40% and 81% (Table 4). nuScenes has been used for 3D object detection [83, 60], multi-agent forecasting [9, 68], pedestrian localization [5], weather augmentation [37], and moving pointcloud prediction [27]. Being still the only annotated AV dataset to provide radar data, nuScenes encourages researchers to explore radar and sensor fusion for object detection [27, 42, 72].

³github.com/nutonomy/nuscenes-devkit
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Year</th>
<th>Scenes (hr)</th>
<th>Size (GB)</th>
<th>RGB images</th>
<th>PCs lidar</th>
<th>PCs radar</th>
<th>Ann. frames</th>
<th>3D boxes</th>
<th>Night / Rain</th>
<th>Map layers</th>
<th>Classes</th>
<th>Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>CamVid [8]</td>
<td>2008</td>
<td>4</td>
<td>0.4</td>
<td>18k</td>
<td>0</td>
<td>0</td>
<td>700</td>
<td>0</td>
<td>No/No</td>
<td>0</td>
<td>32</td>
<td>Cambridge</td>
</tr>
<tr>
<td>Cityscapes [19]</td>
<td>2016</td>
<td>n/a</td>
<td>-</td>
<td>25k</td>
<td>0</td>
<td>0</td>
<td>25k</td>
<td>0</td>
<td>No/No</td>
<td>0</td>
<td>30</td>
<td>50 cities</td>
</tr>
<tr>
<td>Vistas [33]</td>
<td>2017</td>
<td>n/a</td>
<td>-</td>
<td>25k</td>
<td>0</td>
<td>0</td>
<td>25k</td>
<td>0</td>
<td>Yes/Yes</td>
<td>0</td>
<td>152</td>
<td>Global</td>
</tr>
<tr>
<td>BDD100K [85]</td>
<td>2017</td>
<td>100k</td>
<td>1k</td>
<td>100M</td>
<td>0</td>
<td>0</td>
<td>100k</td>
<td>0</td>
<td>Yes/Yes</td>
<td>0</td>
<td>10</td>
<td>NY, SF</td>
</tr>
<tr>
<td>ApolloScape [41]</td>
<td>2018</td>
<td>100k</td>
<td>144k</td>
<td>0</td>
<td>0</td>
<td>144k</td>
<td>70k</td>
<td>0</td>
<td>Yes/No</td>
<td>0</td>
<td>8-35</td>
<td>4x China</td>
</tr>
<tr>
<td>D²-City [11]</td>
<td>2019</td>
<td>1k</td>
<td>700k</td>
<td>0</td>
<td>0</td>
<td>700k</td>
<td>0</td>
<td>No/No</td>
<td>0</td>
<td>12</td>
<td>5x China</td>
<td></td>
</tr>
<tr>
<td>KITTI [32]</td>
<td>2012</td>
<td>22</td>
<td>1.5</td>
<td>15k</td>
<td>15k</td>
<td>0</td>
<td>15k</td>
<td>200k</td>
<td>No/No</td>
<td>0</td>
<td>8</td>
<td>Karlsruhe</td>
</tr>
<tr>
<td>AS lidar [53]</td>
<td>2018</td>
<td>-</td>
<td>2</td>
<td>0</td>
<td>20k</td>
<td>0</td>
<td>20k</td>
<td>4750</td>
<td>4</td>
<td>0</td>
<td>6</td>
<td>China</td>
</tr>
<tr>
<td>KAIST [17]</td>
<td>2018</td>
<td>-</td>
<td>-</td>
<td>8.9k</td>
<td>8.9k</td>
<td>0</td>
<td>8.9k</td>
<td>0</td>
<td>Yes/No</td>
<td>0</td>
<td>3</td>
<td>Seoul</td>
</tr>
<tr>
<td>H3D [61]</td>
<td>2019</td>
<td>160</td>
<td>0.77</td>
<td>83k</td>
<td>27k</td>
<td>0</td>
<td>27k</td>
<td>1.1M</td>
<td>No/No</td>
<td>0</td>
<td>8</td>
<td>SF</td>
</tr>
<tr>
<td>nuScenes</td>
<td>2019</td>
<td>1k</td>
<td>5.5</td>
<td>1.4M</td>
<td>400k</td>
<td>1.3M</td>
<td>40k</td>
<td>1.4M</td>
<td>Yes/Yes</td>
<td>11</td>
<td>23</td>
<td>Boston, SG</td>
</tr>
<tr>
<td>Argoverse [10]</td>
<td>2019</td>
<td>113</td>
<td>0.6</td>
<td>490k</td>
<td>44k</td>
<td>0</td>
<td>22k</td>
<td>993k</td>
<td>Yes/Yes</td>
<td>2</td>
<td>15</td>
<td>Miami, PT</td>
</tr>
<tr>
<td>Lyft L5 [45]</td>
<td>2019</td>
<td>366</td>
<td>2.5</td>
<td>323k</td>
<td>46k</td>
<td>0</td>
<td>46k</td>
<td>1.3M</td>
<td>No/No</td>
<td>7</td>
<td>9</td>
<td>Palo Alto</td>
</tr>
<tr>
<td>Waymo Open [76]</td>
<td>2019</td>
<td>1k</td>
<td>5.5</td>
<td>1M</td>
<td>200k</td>
<td>0</td>
<td>200k</td>
<td>12M</td>
<td>Yes/Yes</td>
<td>0</td>
<td>4</td>
<td>3x USA</td>
</tr>
<tr>
<td>A²3D [62]</td>
<td>2019</td>
<td>n/a</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>39k</td>
<td>2.3M</td>
<td>Yes/Yes</td>
<td>0</td>
<td>7</td>
<td>SG</td>
</tr>
<tr>
<td>A²D² [54]</td>
<td>2019</td>
<td>n/a</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>12k</td>
<td>-</td>
<td>No/No</td>
<td>0</td>
<td>14</td>
<td>3x Germany</td>
</tr>
</tbody>
</table>

Table 1. AV dataset comparison. The top part of the table indicates datasets without range data. The middle and lower part indicate datasets (not publications) with range data released until and after the initial release of this dataset. We use bold highlights to indicate the best entries in every column among the datasets with range data. 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. (*** [41]) 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 [19], Mapillary Vistas [33], D²-City [11], BDD100k [85] and ApolloScape [41] released ever growing datasets with segmentation masks. Vistas, D²-City and BDD100k also contain images captured during different weather and illumination settings. Other datasets focus exclusively on pedestrian annotations on images [20, 25, 79, 24, 88, 23, 58]. 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 [32] 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 [61] 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 [17] 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, [43] providing place categorization labels and [6, 55] providing raw data without semantic labels.

After the initial nuScenes release, [76, 10, 62, 34, 45] followed to release their own large-scale AV datasets (Table 1). Among these datasets, only the Waymo Open dataset [76] provides significantly more annotations, mostly due to the higher annotation frequency (10Hz vs. 2Hz). A²3D takes an orthogonal approach where a similar number of frames (39k) are selected and annotated from 55 hours of data. The Lyft L5 dataset [45] 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 100k km).

*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 [61] where annotations are interpolated from 2Hz to 10Hz.*
We perform motion compensation for sensor placements and Table ≤4119 RGB, 7 GPS, IMU, AHRS. Spinning, on the KITTI dataset localization systems are vulnerable to GPS outages, as seen good data alignment time is nearly instantaneous, this method generally yields frame is achieved. Given that the camera’s exposure of a camera is triggered when the top lidar sweeps across the center of the camera’s FOV. 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 alignment5. We perform motion compensation using the localization algorithm described below.

**Localization.** Most existing datasets provide the vehicle location based on GPS and IMU [32, 41, 19, 61]. Such localization systems are vulnerable to GPS outages, as seen on the KITTI dataset [32, 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 [18]. This method is very robust and we achieve localization errors of ≤10cm. To encourage robotics research, we also provide the raw CAN bus data (e.g. velocities, accelerations, torque, steering angles, wheel speeds) similar to [65].

**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, 45]. We encourage the use of localization and semantic maps as strong priors for all tasks.

---

5The 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.](image)

![Figure 4. Sensor setup for our data collection platform.](image)
The radar range far exceeds the lidar range at up to 200m. At the same time they contain up to 40 radar returns at 10m and 10 at 50m. Lidar and radar points statistics inside each box vary, with the most number in vertical and horizontal as far as 80m from the ego-vehicle. Box orientation is also varying, with the most number in vertical and horizontal

**Annotation statistics.** Our dataset has 23 categories including different vehicles, types of pedestrians, mobility devices and other objects (Figure 8-SM). We present statistics on geometry and frequencies of different classes (Figure 9-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 10-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 14-SM. Annotated objects contain up to 100 lidar points even at a radial distance of 80m and at most 12k lidar points at 5m. 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.

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 [32, 26], 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 7). This makes it hard to compare the performance of vision-only methods which tend to have large localization errors [69].

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 \( \mathbb{C} \):

\[
\text{mAP} = \frac{1}{|\mathbb{C}|} \sum_{c \in \mathbb{C}} \sum_{d \in \mathbb{D}} \text{AP}_{c,d}
\]

**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 \( d = 2m \) 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 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). For each TP metric we compute the mean TP metric (mTP) over all classes:

\[
\text{mTP} = \frac{1}{|\mathbb{C}|} \sum_{c \in \mathbb{C}} \text{TP}_{c}
\]

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 [32, 19, 21]. 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 [41] 3D car instance challenge disentangles these by defining thresholds for each error type and recall threshold. This results in $10 \times 3$ 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).

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

Here mAP is mean Average Precision (1), and 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 [77] presented a similar 3D MOT benchmark on KITTI [32]. They point out that traditional metrics do not take into account the confidence of a prediction. Thus they develop Average Multi Object Tracking Accuracy (AMOTA) and Average Multi Object Tracking Precision (AMOTP), which average MOTA and MOTP across all recall thresholds. By comparing the KITTI and nuScenes leaderboards for detection and tracking, we find that nuScenes is significantly more difficult. Due to the difficulty of nuScenes, the traditional MOTA metric is often zero. In the updated formulation sMOTA,[77] MOTA is therefore augmented by a term to adjust for the respective recall:

$$s\text{MOTA} = \max \left( 0, 1 - \frac{\text{IDS} + \text{FP} + \text{FN} - (1 - r)P}{rP} \right)$$

This is to guarantee that sMOTA values span the entire [0, 1] range. We perform 40-point interpolation in the recall range $[0.1, 1]$ (the recall values are denoted as $R$). The resulting sAMOTA metric is the main metric for the tracking task:

$$s\text{AMOTA} = \frac{1}{|R|} \sum_{r \in R} s\text{MOTA}_r$$

Traditional metrics. We also use traditional tracking metrics such as MOTA and MOTP [4], false alarms per frame, mostly tracked trajectories, mostly lost trajectories, false positives, false negatives, identity switches, and track fragmentations. Similar to [77], we try all recall thresholds and then use the threshold that achieves highest sMOTA.$\gamma$.

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 [51]. 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 Transform (OFT) [69] 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 $360^\circ$ predictions are combined together from all 6 cameras using non-maximum suppression (NMS). We set the box velocity to zero and attributes to the most common attribute for each class in the training data.

Detection challenge results. We compare the results of the top submissions to the nuScenes detection challenge 2019. Among all submissions, Megvii [90] gave the best performance. It is a lidar based class-balanced multi-head network with sparse 3D convolutions. Among image-only
submissions, MonoDIS [70] was the best, significantly out-performing 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 [90]), the fastest reported method at inference time (PointPillars [51]), as well as the best performing camera method (MonoDIS [70]). Using the detections from each method, we setup baselines using the tracking approach described in [77]. We provide detection and tracking results for each of these methods on the train, val and test splits to facilitate more systematic research. See the Supplementary Material for the results of the 2019 nuScenes tracking challenge.

### 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 [51], OFT [69] and an additional image baseline, SSD+3D, with varying amounts of training data. SSD+3D has the same 3D parametrization as MonoDIS [70], but use a single stage design [53]. For this ablation study we train PointPillars with 6x fewer epochs and a one cycle optimizer schedule [71] to cut down the training time. Our main finding is that the *method ordering changes* with the amount of data (Figure 6). 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 verified with a bigger and more diverse training set. A similar conclusion was reached by [56, 59] with [59] suggesting that the KITTI leaderboard reflects the data aug. method rather than the actual algorithms.

**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 objects like pedestrians and bicycles fail to achieve above 0 AP, making ordering impossible (Figure 7). In contrast, center distance matching declares 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.

![Figure 6. 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 [32].](image1)

![Figure 7. 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 [32]. We use CD = 2m for the TP metrics in Section 3.1.](image2)

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 7). This makes sense since the thin structures of bicycles make them difficult to detect in lidar. We conclude that center distance matching is more appropriate to rank image based 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 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).

![Diagram](image3)
<table>
<thead>
<tr>
<th>Lidar sweeps</th>
<th>Pretraining</th>
<th>NDS (%)</th>
<th>mAP (%)</th>
<th>mA TE</th>
<th>mAAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>KITTI</td>
<td>31.8</td>
<td>21.9</td>
<td>1.21</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>KITTI</td>
<td>42.9</td>
<td>27.7</td>
<td>0.34</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>KITTI</td>
<td>44.8</td>
<td>28.8</td>
<td>0.30</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ImageNet</td>
<td>44.9</td>
<td>28.9</td>
<td>0.31</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>None</td>
<td>44.2</td>
<td>27.6</td>
<td>0.33</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. PointPillars [51] 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.

**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 specific performance is in Table 7-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. 7cm) and velocity errors (1.55m/s vs. 0.32m/s), both as expected. MonoDIS also had larger scale errors with mean IOU 11% vs. 71% but the difference is small, suggesting the strong ability for image-only methods to infer size from appearance.

**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 [38]. ImageNet [21] pretraining [47] uses a backbone that was first trained to accurately classify images. KITTI [32] 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.

---

<table>
<thead>
<tr>
<th>Method</th>
<th>NDS (%)</th>
<th>mAP (%)</th>
<th>mA TE</th>
<th>mAAE (m)</th>
<th>mA VE (m/s)</th>
<th>mAAE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KITTI</td>
<td>21.2</td>
<td>12.6</td>
<td>0.82</td>
<td>0.36</td>
<td>0.85</td>
<td>1.73</td>
</tr>
<tr>
<td>SSD+3D†</td>
<td>26.8</td>
<td>16.4</td>
<td>0.90</td>
<td>0.33</td>
<td>0.62</td>
<td>1.31</td>
</tr>
<tr>
<td>MDIS [70]†</td>
<td>38.4</td>
<td>30.4</td>
<td>0.74</td>
<td>0.26</td>
<td>0.55</td>
<td>1.55</td>
</tr>
<tr>
<td>PP [51]</td>
<td>45.3</td>
<td>30.5</td>
<td>0.52</td>
<td>0.29</td>
<td>0.30</td>
<td>0.32</td>
</tr>
<tr>
<td>Megvi [59]</td>
<td>62.5</td>
<td>52.8</td>
<td>0.30</td>
<td>0.23</td>
<td>0.38</td>
<td>0.25</td>
</tr>
</tbody>
</table>

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. (†) use only monocular camera images as input. All other methods use lidar. PP: PointPillars [51], MDIS: MonoDIS [70].

**Better detection gives better tracking.** Weng and Kiti [77] 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. Megvi, PointPillars and MonoDIS achieve an sAMOTA of 17.9%, 3.5% and 4.5%, 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 similar. 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 prolonged 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 approved for testing 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 and a benchmark for trajectory prediction [63].

**Acknowledgements.** The nuScenes dataset was annotated by Scale.ai and we thank Alexandr Wang and Dave Morse for their support. We thank Sun Li, Serene Chen and Karen Ngo at nuTonomy for data inspection and quality control, Bassam Helou and Thomas Roddick for OFT baseline results, Sergi Widjaja and Kiwoo Shin for the tutorials, and Deshraj Yadav and Rishabh Jain from EvalAI [30] for setting up the nuScenes challenges.
References


[19] 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


[31] 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


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

[51] Alex H. Lang, Sourabh Vora, Holger Caesar, Luhan Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In CVPR, 2019. 1, 2, 6, 7, 8, 14, 15, 16


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


