On Camera and LiDAR Positions in End-to-End Autonomous Driving

Malte Stelzer[1], Jan Pirklbauer[1], Jan Bickerdt[2], Volker P. Schomerus[2], Jan Piewek[2], Thorsten Bagdonat[2], and Tim Fingscheidt[1]

[1] Technische Universität Braunschweig, Germany
{malte.stelzer, jan.pirklbauer, t.fingscheidt}@tu-bs.de
[2] Group Innovation Volkswagen AG, Germany
{jan.bickerdt, volker.patricio.schomerus, jan.piewek, thorsten.bagdonat}@volkswagen.de

Abstract. Autonomous vehicles rely on various sensors for environment perception. The placement of the sensors turns out to be critical, as their position determines which environmental information can be perceived. In this paper, we investigate the influence of the position of three front-oriented RGB cameras and a LiDAR sensor on the end-to-end autonomous driving performance. Furthermore, we explore the effects of a mismatch in positioning during training and test (i.e., vehicle operation). In total, four sensor configurations are investigated. We employ the `CARLA` simulator and the recently published `TransFuser` architecture for end-to-end autonomous driving. To ensure comparability between runs despite `CARLA`'s non-deterministic traffic manager, we collect the sensor data for all configurations in a single simulation run. We discover that sensor positions close to and above the rear mirror excel both the roof center and the very high (impractical) baseline position w.r.t. the overall driving score. A sensor position mismatch between training and testing leads to a drop in all performance metrics. However, multi-condition models trained on a mix of sensor positions significantly regain performance regarding the infraction score, thereby improving the model's robustness against domain shifts caused by sensor mismatches during training and test.

Keywords: End-to-End Autonomous Driving · Sensors · CARLA

1 Introduction

End-to-end approaches seek to solve the task of autonomous driving in a holistic manner [29], in contrast to the modular approach, where a pipeline of interconnected modules collaboratively solves the task, with each module being dedicated to a specific sub-task [34]. While end-to-end approaches may lack interpretability [29], they require less engineering effort [34].
The task of autonomous driving in an urban environment is complex, involving challenging scenarios with numerous other traffic participants. In this environment, maintaining safety is critical [12]. In recent years, transformer-based imitation learning approaches [8,25,27,28] have frequently achieved state-of-the-art

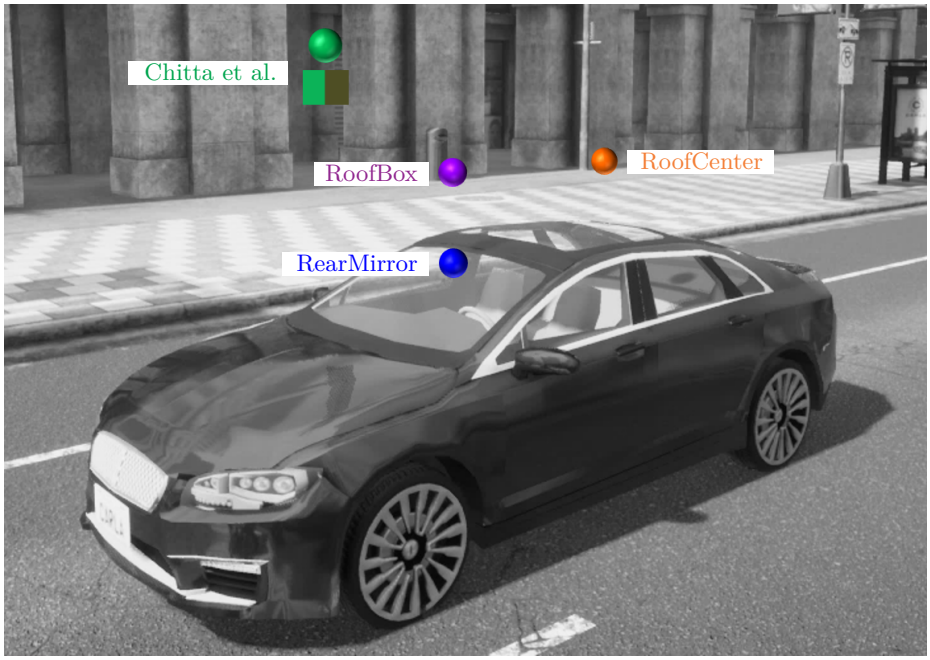

Fig. 1: Investigated sensor positions

results on the CARLA Autonomous Driving Leaderboard [2] and other benchmarks. Various works aim to optimize the training process of imitation learning by enhancing the expert [37] or refining the selection of training data [19]. Other studies explore the integration of RGB cameras and LiDAR, for instance, for tasks such as object tracking [36] or risk analysis [35]. While many approaches utilize a similar combination of one or several RGB cameras with RaDAR or LiDAR, there is no universal concept for how to position these sensors. Recent studies have shown that perception models are sensitive to changes in camera viewpoints [18] and have also addressed issues by mitigating the impact of sensor failures [13].

In this paper, we investigate how the position of RGB cameras and LiDAR influences the driving performance of an end-to-end driving model, as sensor positioning is a crucial decision for researchers and practitioners in automotive industry. Our chosen sensor positions, along with the very high sensor positions from the baseline [8], are illustrated in Figure 1, where in our setups the sensors are placed in a practical manner close to the rearview mirror (RearMirror) or in typical research positions higher in the front or center of the vehicle's roof (RoofBox and RoofCenter, respectively). Additionally, we explore the driving performance in the case of a positional mismatch of the sensors between the training and testing phases. The vehicle capturing data for model training may not nec-

essarily have the exact same sensor setup as the testing vehicle, which can be considered a domain shift for the models. Finally, we evaluate multi-condition models on our three investigated setups, where data of all three different sensor setups is used in training.

2 Related Work

This section provides a brief overview on end-to-end autonomous driving methods and sensor positions.

2.1 End-to-End Autonomous Driving

In the task of end-to-end autonomous driving in an urban environment, two major learning paradigms stand out: reinforcement learning (RL) and imitation learning (IL).
Chen et al. [6] train an agent using maximum-entropy reinforcement learning. The utilized latent space can be decoded into a semantic bird's eye mask, providing interpretability. Toromanoff et al. [31] initially train a `ResNet-18` encoder [16] to predict affordances, such as the traffic light state. In a subsequent training step, these affordances are used as the RL state through a conditional network [10] employing Rainbow IQN [30]. Liang et al. [21] introduced controllable imitative reinforcement learning (CIRL), which is a combination of IL and RL. In a first stage, the model learns to imitate human driving behaviour based on videos and commands. Subsequently, the model undergoes further optimization in an RL stage. This helps to improve the model and to accelerate the RL process. Zhang et al. [37] introduced a privileged RL expert to improve the IL process that does not predict actions but action distributions.
Codevilla et al. [10] introduced conditional imitation learning (CIL), where a command can be utilized either as an additional input or as a switch in a branched architecture. Chen et al. [5] proposed a two-stage approach for IL: In the first stage, a privileged agent learns to imitate expert actions. In the second stage, a sensorimotor agent without privileged information mimics the privileged agent. This offers the advantage that the privileged agent is interpretable and can be used for online training. Utilizing transformers, Prakash et al. [25] introduced the `TransFuser`, where RGB images and LiDAR pseudo-images are processed in two different encoder branches and fused at several intermediate feature resolutions. Chitta et al. [8] further enhanced the architecture by employing a different backbone for the encoder branches and they incorporated multitask learning. An improved version, `TransFuser++`, was introduced by Jaeger et al. [14] with modifications to the architecture and output representation. Wu et al. [33] proposed a branched architecture named `TCP` (trajectory-guided control prediction). Image encodings and encodings for navigational information and speed are shared across two branches, which predict a future trajectory or direct vehicle controls, respectively. The output of the trajectory branch is transformed into vehicle controls using two PID controllers and fused with the output from the other branch

in a weighted manner. Shao et al. [27] employ a single transformer and a safety controller in their `InterFuser` architecture, which includes motion prediction of other traffic participants and handcrafted safety constraints. It also utilizes RGB images and LiDAR point clouds as inputs but is scalable to any number of input modalities. However, its training is computationally expensive due to the need of a large dataset. The `ReasonNet` by Shao et al. [28] holds the top position on the CARLA Autonomous Driving Leaderboard [2]. It incorporates modules to store and fuse temporal information about other objects, capturing relationships and interactions among other objects and the environment. `LAV` (learning from all vehicles) by Chen and Krähenbühl [4] comprises a perception model that learns viewpoint-invariant information and a motion planner that learns to predict future waypoints. `TransFuser`, `TCP`, `InterFuser`, `ReasonNet`, and `LAV` all fall into the category of IL as they train with expert data.
We chose the `TransFuser` architecture from Chitta et al. [8] for our experiments, as it is quickly trainable, and therefore allows to investigate the impact of various different sensor positions on the driving performance.

2.2 Sensor Positions

With increasing autonomy, also the number of sensors in vehicles rises. Prominent examples include cameras, RaDAR, and LiDAR. Cameras are a well established technology and provide semantic understanding of traffic scenery. However, in the case of monocular cameras, a 3D understanding is difficult. Therefore, RaDAR and LiDAR sensors are employed, both offering spatial information. While LiDAR provides better resolution, it is more susceptible to external factors such as weather [38].
Full autonomy in terms of SAE level 5 [1] has not yet reached the consumer market. In consumer cars of lower autonomy level, sensors are often discretely placed, for example, in the bumper or behind the rearview mirror. Although LiDAR is planned for integration into many new cars, its current cost is a hindrance to widespread adoption in the market. Thus, fully autonomous vehicles are still only a research topic. The aim with sensors is to collect as much information as necessary at the lowest cost possible. Many approaches position their sensors on the front part of the vehicle's roof [11,17,22,23]. Particularly, LiDAR requires a high positioning for a full 360° view. In some cases, additional sensors are mounted on the side mirrors [24] or the frontal bender [22]. Generally, there is no fixed or dominant concept for the number and positioning of sensors.
Autonomous driving functions are developed and subject to first tests in simulations such as `CARLA` [3]. In a simulation environment, sensors can be positioned relatively freely without restrictions on realism. Additionally, often only 180° of the LiDAR sensor is used [8,27], allowing for a lower positioning.
Following the works of Chitta et al. [8] and Shao et al. [27], we chose three front-oriented RGB cameras and one LiDAR sensor. These sensor types are complementary to each other, which is a common approach both in simulation and in real vehicles.

3 Dataset Generation

For the generation of training datasets, we built upon the scripts provided by Chitta et al. [8] for both sensor configuration and the employed route-based scenarios.

Data Structure: An expert policy with privileged information [8] is deployed into `CARLA v0.9.10`, capturing observations along with corresponding actions in the form of the next $T = 4$ waypoints $\overline{\mathbf{w}}_{t+1}^{t+T}$ at a frame rate of 2 fps. The observations contain multiple modalities: First, three RGB images with a resolution of 320×160 are captured by three front-facing cameras: one facing directly forward at 0° and the other two angled 60° to each side, each covering a field of view (FOV) of 60°. Additionally, depth and semantic segmentation images with the same camera settings as the RGB images are provided by `CARLA`. Note that during inference, an FOV of 120° and a resolution of 960×480 is used. From this, a central section with a resolution of 320×160 is cropped to avoid lens distortions at the image edges, as required by the official Leaderboard. This guarantees same inputs in training and inference. Second, a LiDAR sensor with a rotation frequency of 600 rpm (which resembles a frame rate of 10 fps), an upper FOV of 10°, and a lower FOV of -30° captures a point cloud. Third, metadata, including top-down views, velocity, and bounding boxes, are also captured. Only RGB, depth, semantic segmentation images, and the LiDAR point cloud are dependent on the sensor position. We then mount *all possible* sensors, as described in Section 4.1, on the expert vehicle to allow simultaneous capturing. This results in four datasets with identical size and statistics, despite the non-determinism of the `CARLA` version used, which allows for a fair comparison of the investigated sensor setups. The expert vehicle is solely equipped with sensors for capturing data; its underlying driving policy only operates based on privileged information.

Route-Based Scenarios: The data is recorded within predefined route-based scenarios. The expert follows routes defined by sparse goal locations, and once the expert reaches a specific trigger point, a scenario begins. These scenarios contain events such as control loss of the vehicle, left turns, encounters with dynamic objects (e.g., a pedestrian), and even traffic rule violations by other participants. Data collection occurs across routes in eight `CARLA` towns, featuring seven distinct scenarios (a scenario for lane changing is not actively included as it occurs naturally in some situations). In addition to standard scenarios, supplementary scenarios involving lane changes on a highway are also captured. For robustness against domain shifts due to weather variations, the weather changes with each frame.

4 Methods and Metrics

In the following, we will describe the employed sensor positions and the concept of sensor position mismatch, along with their roles in the training process. Furthermore, we will provide details about the chosen model architecture and the metrics used for evaluation.

4.1 Sensor Positions

The positioning of sensors impacts the information available to the vehicle. It requires finding a compromise that addresses blind spots close to the vehicle, provides a comprehensive view of the traffic environment, and ensures practicality.
In this paper, we investigate three setups where in each of them the RGB cameras and the LiDAR sensor are positioned at the same location. This enables that the sensors capture similar information and allows for future investigations with less modalities. Figure 1 shows sensor positions investigated in this paper, including ours and those from Chitta et al. [8]. Our RearMirror setup allocates both the LiDAR sensor and the RGB cameras at the same location on the windshield, directly above the rearview mirror, at a height of 1.35 m (blue sphere). It is the most practical of the investigated setups as it positions the sensors close to the vehicle. Other types of sensors are already positioned at this location in vehicles, and advances in solid-state LiDAR, which is smaller and lighter than conventional mechanical LiDAR [20], also allow for this positioning. Our RoofBox setup locates both sensor types precisely 50 cm above the aforementioned setup (violet sphere). In our RoofCenter setup, the sensors are also positioned at a height of 1.85 m, but at the center of the ego vehicle's roof (orange sphere). The latter two setups represent typical positions in current research on autonomous driving [15, 23]. Finally, the sensor setup by Chitta et al. [8] places the RGB cameras at a practically unrealistic height of 2.3 m (green cube) and the LiDAR sensor at 2.5 m (green sphere).
To examine the influence of sensor positions on driving performance, models based on the `TransFuser` [8] architecture are trained on datasets corresponding to the respective sensor positions. For each sensor setup, three models are trained with different random seeds. Subsequently, they are evaluated in $B = 3$ runs in ensembles of three on the `Longest6 Benchmark` [8].

4.2 Sensor Position Mismatch

In addition to the positioning itself, the behavior of an autonomous vehicle in case of a mismatch between sensor positioning in training and test is crucial. The sensor positioning used during training data acquisition may not necessarily perfectly align with the sensor positioning in the test vehicle. A small disparity in sensor positions should ideally not result in poor driving performance. In our experimental setup, however, we explore *how much* the performance declines, if training and test positions are clearly different.
In another plausible scenario, an autonomous vehicle can also be trained on a diverse training dataset from various sources, encompassing different sensor setups. This multi-condition approach may offer the advantage of increased robustness versus (slight) deviations in sensor positions between training and test. However, it presents a challenge for the autonomous vehicle as it needs to learn a sensor position-invariant representation of the traffic scenery.
To explore the impact of a sensor position mismatch between the training and

test phases, the models trained on one sensor setup (cf. Section 4.1) are again evaluated in $B = 3$ runs in ensembles of three on the `Longest6 Benchmark` [8], but with a different sensor setup in test as compared to training.
To examine the performance of a multi-condition so-called Mixed model, training is performed on the setups RearMirror, RoofBox, and RoofCenter jointly. The evaluation is conducted in ensembles of three models in $B = 3$ runs on the `Longest6 Benchmark` [8].

4.3 Model & Training

Task Description: The investigated general task is point-to-point navigation in an urban environment [7,8,19,27]. For navigation, the model is given a number of G 2D sparse goal locations $\mathbf{u}_1^G = (\mathbf{u}_1, ..., \mathbf{u}_g, ..., \mathbf{u}_G)$, with $\mathbf{u}_g \in \mathbb{R}^2$, and $\mathbf{u}_1$ representing the starting point and $\mathbf{u}_G$ the endpoint. Along these routes, the model must cope with various scenarios, some of which may be hazardous.
Input Processing: We selected the `TransFuser` architecture [8] for our experiments. At each discrete time step t, the model receives observations previously captured by the expert. The RGB images from the three front-oriented cameras are merged, resulting in a single RGB image $\mathbf{x}_t \in \mathbb{I}^{H \times W \times C}$ with height $H = 160$, width $W = 960$, $C = 3$ RGB channels, and a FOV of 180°, see examples in Figure 2 (a)-(c). The input image is cropped to a resolution of $160 \times 704 \times 3$ at a horizontal position corresponding to a randomly selected angle for augmentation. To maintain consistency between the various input and output modalities, the LiDAR point cloud and ground truth data are adjusted accordingly to reflect the change in viewpoint. Each RGB image (a, b, c) corresponds to one LiDAR visualization (A, B, C, respectively), while all images portray the exact same traffic scene. In the (a) RearMirror setup, occlusions from the ego vehicle are visible, while the (c) RoofCenter setup provides the best overall view. The (b) RoofBox setup is a compromise between the (a) RearMirror and (c) RoofCenter setups with no occlusions from the ego vehicle but a slightly worse overview than the (c) RoofCenter setup. For a LiDAR pseudo-image, only the points 32 m in front of the sensor and 16 m to the sides are considered. This point cloud is then transformed into a histogram with $C' = 2$ bins, one bin above and one below the ground plane. The grid is further divided into 0.125 m $\times$ 0.125 m patches, resulting in a LiDAR pseudo-image $\mathbf{v}_t \in \mathbb{I}^{H' \times W' \times C'}$ with height and width $H' = W' = 256$. Visualizations of LiDAR point cloud examples can be seen in Figure 2 (A)-(C), where not the pseudo-images $\mathbf{v}_t$ are shown but the projection of the point clouds onto the ground plane for (A) the RearMirror, (B) RoofBox, and (C) RoofCenter setups. The visualizations cover the same range as the pseudo-images. Occlusions stemming from the ego vehicle are visible in both the (A) RearMirror and (C) RoofCenter setups. These setups also exhibit large blind spots due to shading from the ego vehicle. In the (B) RoofBox setup, there are no occlusions from the ego vehicle due to its sensor positioning higher than the (A) RearMirror setup and more in the front part of the vehicle than the (C) RoofCenter setup. With the current velocity ν_t and the next goal location $\mathbf{u}_{g=g(t)}$, this yields in measurements $\mathcal{X}_t = \{\mathbf{x}_t, \mathbf{v}_t, \nu_t, \mathbf{u}_{g=g(t)}\}$. The depth and

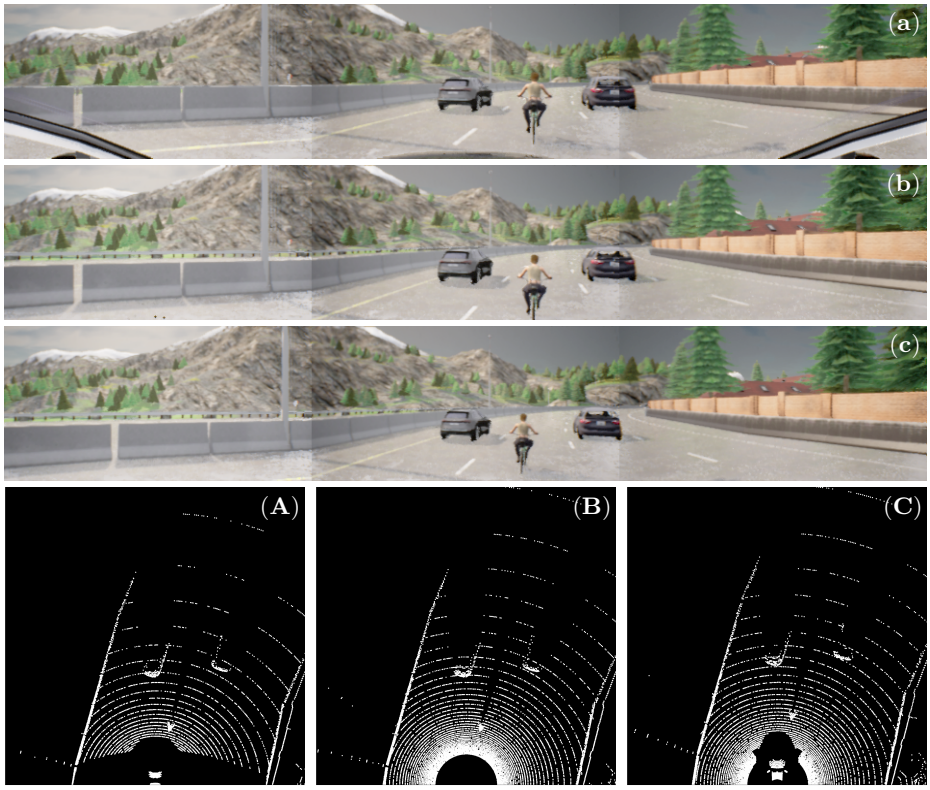

Fig. 2: RGB images (a-c) and LiDAR visualizations (A-C) for investigated sensor setups: (a, A) RearMirror, (b, B) RoofBox, (c, C) RoofCenter

semantic segmentation images undergo processing in the same manner as the RGB images. Along with an HD map extracted from the top-down view and the bounding boxes, this yields auxiliary measurements $\overline{\mathcal{Y}}_t$. Each of the four datasets used in training includes a total of 3040 routes in 225k frames.

End-to-End Driving Model: The model policy DNN $\boldsymbol{\pi}$ in Figures 3 and 4 reads in the measurements $\mathcal{X}_t$. The RGB image $\mathbf{x}_t$ and the LiDAR pseudo-image $\mathbf{v}_t$ undergo internal processing in two separate branches, each containing multiple `RegNet` encoders [26]. At four intermediate feature resolutions, information from the two branches is exchanged using attention modules [32], and the output is fed back into the corresponding branches through element-wise summation. The outputs $\mathbf{x}'_t$ and $\mathbf{v}'_t$ of the two branches are then combined via element-wise summation, resulting in a 512-dimensional feature vector. This feature is further compressed using an MLP, yielding a 64-dimensional feature vector. The resulting vector undergoes additional processing in a GRU-based decoder [9] that calculates the $T = 4$ future waypoints $\mathbf{w}_{t+1}^{t+T}$ in an autoregressive manner. For

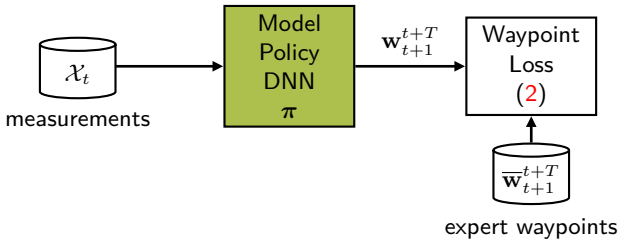

Fig. 3: Open-loop training with waypoint loss (2): For each measurement $\mathcal{X}_t$ in each time step t, the model policy DNN $\boldsymbol{\pi}$ learns via error backpropagation to minimize the distance between its predicted waypoints $\mathbf{w}_{t+1}^{t+T}$ and the waypoints $\overline{\mathbf{w}}_{t+1}^{t+T}$ from the expert. Further auxiliary losses are omitted here for clarity.

further details about the architecture, the interested reader is referred to [8].
Training by Imitation Learning: Imitation learning aims to find a model policy DNN $\boldsymbol{\pi}$ that best mimics an expert policy $\overline{\boldsymbol{\pi}}$ (utilized for training data collection). As shown in Figure 3, the model policy $\boldsymbol{\pi}$ produces waypoints $\mathbf{w}_{t+1}^{t+T}$ based on the measurements $\mathcal{X}_t$, which can be described by

$$\boldsymbol{\pi}(\mathcal{X}_t) = \mathbf{w}_{t+1}^{t+T}. \tag{1}$$

During training, the distance between the waypoints $\mathbf{w}_{t+1}^{t+T}$ (1) and the ground truth waypoints $\overline{\mathbf{w}}_{t+1}^{t+T}$ produced by the expert policy $\overline{\boldsymbol{\pi}}$ is minimized and stored for training purposes using the mean absolute error (i.e., the waypoint L_1 loss)

$$J_t^{\mathrm{WP}} = J^{\mathrm{WP}}\left(\boldsymbol{\pi}(\mathcal{X}_t), \overline{\boldsymbol{\pi}}(\mathcal{X}_t)\right) = \sum_{\tau=1}^{T} ||\mathbf{w}_{t+\tau} - \overline{\mathbf{w}}_{t+\tau}||_1. \tag{2}$$

In addition to waypoint prediction, auxiliary tasks are employed to enhance model optimization. The output $\mathbf{x}_t'$ of the RGB image branch serves as input for two decoders based on auto-regressive image-based waypoint prediction with multi task (`AIM-MT`) [7], predicting depth and semantic segmentation images. The output $\mathbf{v}_t'$ of the LiDAR branch is utilized as input for a convolutional decoder and a `CenterNet` decoder [39], predicting the HD map and bounding boxes, respectively. These decoders do not affect the driving itself. Their purpose is purely for model optimization during training and interpretability during testing. The training is in analogy to Figure 3 with different losses and the auxiliary measurements $\overline{\mathcal{Y}}_t$ as the ground truth from the expert. For detailed information about the auxiliary tasks and the corresponding losses, we refer the reader to [8]. The training is conducted for 31 epochs for the setups from Chitta et al. [8], RearMirror, RoofBox, and RoofCenter. For fairness of comparison, the models for the Mixed setup are trained for 10 epochs only, as they see three times the training material in one epoch compared to a single sensor setup model. Here, we change the sensor setup in training with every batch. The used batch size is 96 for all methods.

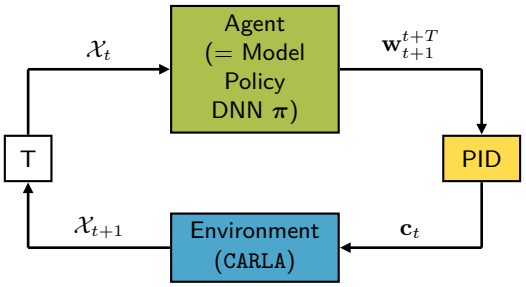

Fig. 4: Closed-loop test: At each time step t, the agent DNN $\boldsymbol{\pi}$ generates $T = 4$ waypoints $\mathbf{w}_{t+1}^{t+T}$, which are subsequently transformed into vehicle control commands $\mathbf{c}_t$ by two PID controllers (depicted as one for simplicity). Subsequently, based on these commands, the environment outputs new measurements $\mathcal{X}_{t+1}$ to the agent.

Testing with a PID Controller: As shown in Figure 4, the test is conducted in a closed-loop manner, where the model is deployed into `CARLA` as an agent. Unlike during training, the model in the test actively interacts with the environment and performs driving tasks. The $T = 4$ waypoints $\mathbf{w}_{t+1}^{t+T}$ predicted by the GRU-based decoder need to be transformed into vehicle control commands $\mathbf{c}_t = (c_t^{\text{ste}}, c_t^{\text{thr}}, c_t^{\text{bra}})$. For this purpose, two PID controllers are utilized. The lateral PID controller calculates the steering command $c_t^{\text{ste}} \in (-1, 1)$ based on the midpoint of waypoints $\mathbf{w}_{t+1}$ and $\mathbf{w}_{t+2}$. The longitudinal PID controller generates throttle $c_t^{\text{thr}} \in (0, 1)$ and brake commands $c_t^{\text{bra}} \in (0, 1)$ based on a velocity parameter $\gamma_t = \frac{1}{\Delta t}\|\mathbf{w}_{t+2} - \mathbf{w}_{t+1}\|_2$ with $\Delta t = 1$. Further details can be found in [8] and its supplementary.

4.4 Metrics

To address non-determinism in the trainings and in test, we report the mean and sample standard deviation for all of the three following metrics.
Route completion RC $\in [0, 100]$ serves as an *efficiency* measure for an autonomous vehicle, representing the percentage of the entire route that has actually been travelled. It naturally decreases if the vehicle encounters blockages, timeouts, or drives off-road or outside the designated route, as there is a limit on the maximum time to reach the goal. The route completion over all $B = 3$ runs

$$\text{RC} = \frac{1}{B \cdot N} \sum_{b \in \mathcal{B}} \sum_{n \in \mathcal{N}} \text{RC}_{b,n} \tag{3}$$

is defined as an average of single route completions $RC_{b,n}$ with run index $b \in \mathcal{B} = \{1, ..., B\}$ and route index $n \in \mathcal{N} = \{1, 2, ..., N\}$.
The infraction score IS $\in [0, 1]$ is defined as a *safety* measure for the autonomous vehicle. The IS decreases once collisions happen or when there are violations of traffic lights and stop signs. The IS is calculated as

$$\mathrm{IS} = \frac{1}{B \cdot N} \sum_{b \in \mathcal{B}} \sum_{n \in \mathcal{N}} \mathrm{IS}_{b,n}, \tag{4}$$

where $\mathrm{IS}_{b,n}$ is the infraction multiplier for a single run and route described as

$$\mathrm{IS}_{b,n} = \prod_{j \in \mathcal{J}} p_j^{c_{j,b,n}}. \tag{5}$$

Here, p_j is the specific penalty for an infraction type (e.g., collision with a pedestrian) $j \in \mathcal{J} = \{1, 2, ..., J\}$ and $c_{j,b,n}$ represents the number of type j infractions in run b on route n. Specific penalties for the infraction types can be found in [2]. The driving score $\mathrm{DS} \in [0, 100]$ is the primary metric as the product of the RC and IS, thereby considering both efficiency and safety. It is calculated as

$$\mathrm{DS} = \frac{1}{B \cdot N} \sum_{b \in \mathcal{B}} \sum_{n \in \mathcal{N}} \mathrm{RC}_{b,n} \cdot \mathrm{IS}_{b,n}. \tag{6}$$

Unlike the training setup shown in Figure 3, where the loss is calculated on a frame (or batch) basis, the evaluation metrics are calculated on a route basis. Further details about the computation of the metrics, particularly the reported sample standard deviations, can be found in the supplementary Section A.

5 Results and Discussion

Table 1 displays the results of the test runs on the `Longest6 Benchmark`. The mean and standard deviation of $B = 3$ test runs are reported for the metrics DS (6), RC (3), and IS (4). The upper segment of Table 1 presents the outcomes when sensor setups for training and test are matched, while the lower segment of Table 1 shows the results for mismatched setups. We mark best values in bold font and second-best by underlining. However, as we also report the sample standard deviation, some of the setups do not turn out to deviate significantly in performance. Accordingly, for each metric (column), we denote statistically equivalent performing setups by the colors green—blue—gray—red in performance-descending order. Values that fall into several statistically equivalent ranges are marked with hatched patterns in the corresponding colors. Values that fall significantly below this scale, which are not necessarily statistically equivalent, are marked in violet.

Matched Setups: We start with discussing the matched sensor setups in Table 1 (upper segment). In terms of route completion (RC), the setup from Chitta et al. [8] performs worst (75.10, gray). The RearMirror setup, with an RC of 83.22 (blue), performs better, while the RoofCenter setup, with an RC of 87.28 (green), turns out to be best. Note that the RoofBox setup, with an RC of 84.69 (green/blue hatched), is statistically equivalent to both. *Independent of sensor height, route completion seems to be better the more the sensors are mounted towards the rear of the vehicle (oversight!).*

Regarding the infraction score (IS), the setup from Chitta et al. [8], as well as

	Models (Training Setup)	Test Setup	Performance Metrics DS ↑	RC ↑	IS ↑
Matched Setups	Chitta et al. [8]	Chitta et al. [8]	44.66 ± 2.39	75.10 ± 3.38	**0.62** ± 0.06
	RearMirror	RearMirror	51.64 ± 3.37	83.22 ± 0.82	0.60 ± 0.04
	RoofBox	RoofBox	**52.87** ± 3.29	84.69 ± 3.33	0.61 ± 0.05
	RoofCenter	RoofCenter	45.52 ± 2.66	**87.28** ± 2.12	0.51 ± 0.02
Mismatched Setups	RearMirror	RoofBox	21.41 ± 2.96	60.19 ± 2.92	0.36 ± 0.08
	RearMirror	RoofCenter	28.19 ± 2.99	69.52 ± 4.01	0.36 ± 0.02
	RoofBox	RearMirror	16.60 ± 1.82	43.30 ± 2.73	0.37 ± 0.04
	RoofBox	RoofCenter	**38.37** ± 2.31	**82.39** ± 3.75	**0.47** ± 0.01
	RoofCenter	RearMirror	20.86 ± 3.30	55.64 ± 1.24	0.34 ± 0.03
	RoofCenter	RoofBox	24.28 ± 2.50	72.05 ± 5.50	0.37 ± 0.05
	Mixed	RearMirror	**46.66** ± 3.11	**81.40** ± 1.68	0.56 ± 0.03
	Mixed	RoofBox	41.06 ± 8.37	74.93 ± 4.38	**0.57** ± 0.06
	Mixed	RoofCenter	42.84 ± 3.29	73.75 ± 3.27	**0.57** ± 0.01

Table 1: `Longest6 Benchmark` results: The driving score (DS), route completion (RC), and infraction score (IS) are reported with the mean and standard deviation of three runs on the `Longest6 Benchmark` for each sensor setup. The upper table segment shows setups that are matched in training and test, whereas the bottom segment shows results for mismatches. The best values are **bold**, the second-best underlined. For each metric (higher is better), we denote setups with statistically equivalent performance by the colors green — blue — gray — red in performance-descending order. Values that fall into several statistically equivalent ranges of values are marked with hatched patterns in the corresponding colors. Values that fall below this scale, which are not necessarily statistically equivalent, are marked in violet.

the RearMirror and RoofBox setups, perform statistically equivalent (green). The RoofCenter setup performs worst with an IS of 0.51 (blue), falling significantly behind the other three setups. *Independent of sensor height, we observe higher infraction scores for front-mounted sensors in the matched case.*

When it comes to the overall driving score (DS), the setup from Chitta et al. exhibits a DS of 44.66 (blue), which is slightly lower than in the corresponding publication [8] (47.30). This difference is caused by the use of different datasets. Given the standard deviations of both (5.72 [8] and 2.39 (our setup)), this difference shows no significance. Together with the RoofCenter setup (blue), which performs statistically equivalent but slightly better on average, these two setups show the lowest DS. The better performing RearMirror and RoofBox setups (green), which are only 50 cm apart vertically, perform statistically equivalent regarding the DS, while the RoofBox setup has a higher mean DS.

Overall, both the setup from Chitta et al. [8] and the RoofCenter setup exhibit imbalanced performance, with either RC or IS being poor. *Only the* RoofBox *setup performs best (*green*) or statistically equivalent to the best (*green*/*blue hatched*) setup in all metrics in the matched case.*

In summary for the upper segment of Table 1, the closer the sensors are positioned to the front part of the vehicle (from front to rear: Chitta et al. - RearMir-

ror, RoofBox - RoofCenter*), the better the mean IS performs, while the mean RC deteriorates.* Taking into account the standard deviation, the IS is equivalent for setups where the sensors are mounted in the front part of the vehicle but declines if mounted in the center part. *Interestingly, on the other hand, there is no general statement regarding the influence of the sensor's height (from high to low: Chitta et al.* - RoofBox, RoofCenter - RearMirror*) on the driving performance.*
Investigating a bit deeper the second row of Table 1, the RC shows a small standard deviation (0.82), while the DS and the IS show standard deviations that can be considered normal across all setups (3.37 and 0.04, respectively). This can happen since the computation of the standard deviation of the DS is disentangled from the respective computations for the RC and IS. Each of the three metrics is individually cumulated and averaged across all routes and test runs. Additionally, if IS and RC deviate in the same direction from the mean and not opposite, a small standard deviation in the RC in connection with a normal standard deviation in the IS can cause a normal standard deviation in the DS. Details about the computation of metrics, mean and standard deviation can be found in the supplementary Section A.
Mismatched Setups (Default): Now let's turn to the case of a mismatched sensor setup between training and test (lower segments of Table 1). We observe a steep decline in performance of nearly all models. Almost all model setups show a statistically equivalent poor DS and IS performance (red). Again, the models trained in the RoofBox setup and now tested in the RoofCenter setup perform best with a DS of 38.37 (gray), and an RC of 82.39 (blue). Also, the IS of this setup is best with 0.47 (gray), only slightly below the IS of the matched RoofCenter setup (blue). The IS of the models trained on RoofBox and RoofBox respectively, but tested on RoofBox, severely declines (violet).
In general, most metrics decline in the case of a mismatch of sensor setups between training and test. The models trained in the strongest setup (RoofBox) and tested in the weakest setup (RoofCenter) perform the best among the investigated mismatched setups.
Interestingly, in some cases, the metrics of mismatched sensor setups are statistically equivalent to matched sensor setups: The RC of the models trained in the RoofBox setup but tested in the RoofCenter setup is statistically equivalent to the RCs of the matched RearMirror and RoofBox setups (blue). The RC of the mismatched setup of RearMirror and RoofCenter, as well as the mismatched setup of RoofCenter and RoofBox (gray), is also statistically equivalent to the matched setup from Chitta et al.
Mismatched Setups (Mixed): Now we turn to the Mixed training setup (lowest segment of Table 1): The same models were tested in three different sensor setups. The IS of these models is almost identical for all three setups and even statistically equivalent to the best-performing matched setups (green). The most interesting observation is that the poor IS of 0.51 (blue) for the RoofCenter test setup in the matched case could significantly be raised to 0.57 (green) by using the obviously robust Mixed models. Regarding the RC, the Mixed models in the RoofBox and RearMirror test setups are already statistically equivalent to the

(blue) performance region. Although the mean DS of the RoofCenter test setup (42.84) is better than that of the RoofBox test setup, the latter and the RearMirror test setup (46.66) are statistically equivalent to the best-performing matched setups (green). Compared to the matched training and test setting (upper segment of Table 1), the performance drops but remains closer across different test setups. *For the* Mixed *models, we can conclude that their multi-condition training helps to regain performance w.r.t. infraction score, particularly where sensor positions (here:* RoofCenter*) are suboptimal.*

6 Limitations

For the investigations in this paper, only sensor configurations and positions from industry and existing literature are used. Exploring different, potentially better sensor combinations and placements is left for future work. Additionally, no quantitative analysis of the differences between sensor positions has been conducted.
This paper relies solely on simulation for the investigations. We believe that the results and conclusions are applicable to the real world under ideal conditions. However, how environmental factors influence sensor performance depending on their positions remains to be investigated.

7 Conclusions

We investigated the influence of different positions of RGB cameras and LiDAR on the driving performance of an end-to-end autonomous driving approach. Creating sensor-specific datasets from the same simulation run, we trained and evaluated models for autonomous driving in `CARLA`. We find that the positions of the aforementioned sensors have a significant impact on driving performance. In a setup, where sensor positions in training and evaluation match, a position above the rearview mirror outperforms other sensor positions including even unrealistically high sensor positions in common baselines. In the case of a mismatched sensor setup between the training and evaluation phases, the overall driving performance of models across all sensor setups decreases. In the case of a multi-condition training using data from mixed sensor setups, robustness against variations in the evaluation sensor setup improves. Also performance remains consistent across various sensor setups in evaluation and improves compared to the mismatched setups. Multi-condition training may significantly improve infraction score performance for suboptimal sensor positions (which is the typical case in practice!). Finally, we show that in the matched case, the infraction score is better for front-mounted sensor setups, while their route completion suffers. Although there is a clear connection between those two metrics and the horizontal sensor position, no trend is observed for the height of the sensors.

References

1. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles: J3016. Tech. rep., SAE International (Apr 2021) 4
2. CARLA Autonomous Driving Leaderboard (2023–), https://leaderboard.carla.org/, Online; accessed 20.02.2024 2, 4, 11
3. Alexey Dosovitskiy and German Ros and Felipe Codevilla and Antonio Lopez and Vladlen Koltun: CARLA: An Open Urban Driving Simulator. In: Proc. of CoRL. pp. 1–16. Mountain View, CA, USA (2017) 4
4. Chen, D., Krähenbühl, P.: Learning from All Vehicles. In: Proc. of CVPR. pp. 17201–17210. Los Alamitos, CA, USA (Jun 2022) 4
5. Chen, D., Zhou, B., Koltun, V., Krähenbühl, P.: Learning by Cheating. In: Proc. of CoRL. pp. 66–75. virtual (Nov 2020) 3
6. Chen, J., Li, S.E., Tomizuka, M.: Interpretable End-to-End Urban Autonomous Driving With Latent Deep Reinforcement Learning. IEEE Trans. on Intelligent Transportation Systems **23**(6), 5068–5078 (Feb 2022) 3
7. Chitta, K., Prakash, A., Geiger, A.: NEAT: Neural Attention Fields for End-to-End Autonomous Driving. In: Proc. of ICCV. pp. 15793–15803. Virtual (Oct 2021) 7, 9
8. Chitta, K., Prakash, A., Jaeger, B., Yu, Z., Renz, K., Geiger, A.: TransFuser: Imitation With Transformer-Based Sensor Fusion for Autonomous Driving. IEEE Trans. on PAMI pp. 12878–12895 (Aug 2023) 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12
9. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. In: Proc. of EMNLP. pp. 1724–1734. Doha, Qatar (Oct 2014) 8
10. Codevilla, F., Müller, M., López, A., Koltun, V., Dosovitskiy, A.: End-to-End Driving via Conditional Imitation Learning. In: Proc. of ICRA. pp. 4693–4700. Brisbane, Australia (May 2018) 3
11. Cruise: 2022 Self-Driving Safety Report (2022–), https://assets.ctfassets.net/95kuvdv8zn1v/zKJHD7X22fNzpAJztpd5K/ac6cd2419f2665000e4eac3b7d16ad1c/Cruise_Safety_Report_2022_sm-optimized.pdf, Online; accessed 20.02.2024 4
12. Fingscheidt, T., Gottschalk, H., Houben, S. (eds.): Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety. Springer International Publishing, Cham (2022) 1
13. Ge, C., Chen, J., Xie, E., Wang, Z., Hong, L., Lu, H., Li, Z., Luo, P.: MetaBEV: Solving Sensor Failures for 3D Detection and Map Segmentation. In: Proc. of ICCV. pp. 8721–8731. Paris, France (Oct 2023) 2
14. Jaeger, B., Chitta, K., Geiger, A.: Hidden biases of end-to-end driving models. In: Proc. of ICCV. pp. 8206–8215. Paris, France (Oct 2023) 3
15. Jo, K., Kim, J., Kim, D., Jang, C., Sunwoo, M.: Development of Autonomous Car—Part II: A Case Study on the Implementation of an Autonomous Driving System Based on Distributed Architecture. IEEE Transactions on Industrial Electronics **62**(8), 5119–5132 (Mar 2015) 6
16. Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun: Deep Residual Learning for Image Recognition. arXiv **1512.03385** (Dec 2015) 3
17. Karle, P., Betz, T., Bosk, M., Fent, F., Gehrke, N., Geisslinger, M., Gressenbuch, L., Hafemann, P., Huber, S., Hübner, M., Huch, S., Kaljavesi, G., Kerbl, T., Kulmer, D., Mascetta, T., Maierhofer, S., Pfab, F., Rezabek, F., Rivera, E., Sagmeister,

S., Seidlitz, L., Sauerbeck, F., Tahiraj, I., Trauth, R., Uhlemann, N., Würsching, G., Zarrouki, B., Althoff, M., Betz, J., Bengler, K., Carle, G., Diermeyer, F., Ott, J., Lienkamp, M.: EDGAR: An Autonomous Driving Research Platform – From Feature Development to Real-World Application. arXiv **2309.15492** (Jan 2024) 4

18. Klinghoffer, T., Philion, J., Chen, W., Litany, O., Gojcic, Z., Joo, J., Raskar, R., Fidler, S., Alvarez, J.M.: Towards Viewpoint Robustness in Bird's Eye View Segmentation. In: Proc. of ICCV. pp. 8515–8524. Paris, France (Oct 2023) 2
19. Klingner, M., Müller, K., Mirzaie, M., Breitenstein, J., Termohlen, J.A., Fingscheidt, T.: On the Choice of Data for Efficient Training and Validation of End-to-End Driving Models. In: Proc. of CVPRW. pp. 4802–4811. Los Alamitos, CA, USA (Jun 2022) 2, 7
20. Li, N., Ho, C.P., Xue, J., Lim, L.W., Chen, G., Fu, Y.H., Lee, L.Y.T.: A Progress Review on Solid-State LiDAR and Nanophotonics-Based LiDAR Sensors. Laser & Photonics Reviews **16**(11), 2100511 (Aug 2022) 6
21. Liang, X., Wang, T., Yang, L., Xing, E.: CIRL: Controllable Imitative Reinforcement Learning for Vision-Based Self-driving. In: Proc. of ECCV. pp. 584–599. Munich, Germany (Sep 2018) 3
22. LLC, W.: Waymo Safety Report - February 2021 (2021), https://downloads.ctfassets.net/sv23gofxcuiz/4gZ7ZUxd4SRj1D1W6z3rpR/2ea16814cdb42f9e8eb34cae4f30b35d/2021-03-waymo-safety-report.pdf, Online; accessed 20.02.2024 4
23. Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., Haehnel, D., Hilden, T., Hoffmann, G., Huhnke, B., et al.: Junior: The Stanford Entry in the Urban Challenge. Journal of field Robotics **25**(9), 569–597 (2008) 4, 6
24. Motors, G.: 2022 Self-Driving Safety Report (2018–), https://www.gm.com/content/dam/company/docs/us/en/gmcom/gmsafetyreport.pdf, Online; accessed 20.02.2024 4
25. Prakash, A., Chitta, K., Geiger, A.: Multi-Modal Fusion Transformer for End-to-End Autonomous Driving. In: Proc. of CVPR. pp. 7073–7083. Nashville, TN, USA (Jun 2021) 1, 3
26. Radosavovic, I., Kosaraju, R.P., Girshick, R., He, K., Dollár, P.: Designing Network Design Spaces. In: Proc. of CVPR. pp. 10425–10433. Seattle, WA, USA (Jun 2020) 8
27. Shao, H., Wang, L., Chen, R., Li, H., Liu, Y.: Safety-Enhanced Autonomous Driving Using Interpretable Sensor Fusion Transformer. arXiv **2207.14024** (Dec 2022) 1, 4, 7
28. Shao, H., Wang, L., Chen, R., Waslander, S.L., Li, H., Liu, Y.: ReasonNet: End-to-End Driving with Temporal and Global Reasoning. In: Proc. of CVPR. pp. 13723–13733. Los Alamitos, CA, USA (Jun 2023) 1, 4
29. Tampuu, A., Semikin, M., Muhammad, N., Fishman, D., Matiisen, T.: A Survey of End-to-End Driving: Architectures and Training Methods. IEEE Trans. Neural Netw. Learn. Syst. **33.4**, 1364–1384 (Apr 2022) 1
30. Toromanoff, M., Wirbel, E., Moutarde, F.: Is Deep Reinforcement Learning Really Superhuman on Atari? Leveling the Playing Field. arXiv **1908.04683** (Nov 2019) 3
31. Toromanoff, M., Wirbel, E., Moutarde, F.: End-to-End Model-Free Reinforcement Learning for Urban Driving Using Implicit Affordances. In: Proc. of CVPR. pp. 7151–7160. Seattle, WA, USA (Jun 2020) 3

32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is All You Need. In: Proc. of NIPS. pp. 1–11. Long Beach, CA, USA (Dec 2017) 8
33. Wu, P., Jia, X., Chen, L., Yan, J., Li, H., Qiao, Y.: Trajectory-Guided Control Prediction for End-to-End Autonomous Driving: A Simple yet Strong Baseline. In: Proc. of NeurIPS. pp. 6119–6132. New Orleans, LA, USA (Dec 2022) 3
34. Xiao, Y., Codevilla, F., Gurram, A., Urfalioglu, O., López, A.M.: Multimodal End-to-End Autonomous Driving. IEEE Trans. Intell. Transp. Syst. **23.1**, 537–547 (Jan 2022) 1
35. Zendel, O., Huemer, J., Murschitz, M., Dominguez, G.F., Lobe, A.: Joint Camera and LiDAR Risk Analysis. In: Proc. of CVPRW. pp. 88–97. Vancouver, Canada (Jun 2023) 2
36. Zhang, C., Zhang, C., Guo, Y., Chen, L., Happold, M.: MotionTrack: End-to-End Transformer-based Multi-Object Tracking with LiDAR-Camera Fusion. In: Proc. of CVPRW. pp. 151–160. Vancouver, Canada (Jun 2023) 2
37. Zhang, Z., Liniger, A., Dai, D., Yu, F., Van Gool, L.: End-to-End Urban Driving by Imitating a Reinforcement Learning Coach. In: Proc. of ICCV. pp. 15202–15212. Montreal, QC, Canada (Oct 2021) 2, 3
38. Zhaohua, L., Bochao, G.: Radar Sensors in Automatic Driving Cars. In: Proc. of ICECTT. pp. 239–242. Nanchang, China (Nov 2020) 4
39. Zhou, X., Wang, D., Krähenbühl, P.: Objects as Points. arXiv **1904.07850** (Apr 2019) 9