

On Offline Evaluation of 3D Object Detection for Autonomous Driving

Tim Schreier Katrin Renz Andreas Geiger Kashyap Chitta
University of Tübingen Tübingen AI Center

tschreier2@gmail.com {katrin.renz, a.geiger, kashyap.chitta}@uni-tuebingen.de

Abstract

Prior work in 3D object detection evaluates models using offline metrics like average precision since closed-loop online evaluation on the downstream driving task is costly. However, it is unclear how indicative offline results are of driving performance. In this work, we perform the first empirical evaluation measuring how predictive different detection metrics are of driving performance when detectors are integrated into a full self-driving stack. We conduct extensive experiments on urban driving in the CARLA simulator using 16 object detection models. We find that the nuScenes Detection Score has a higher correlation to driving performance than the widely used average precision metric. In addition, our results call for caution on the exclusive reliance on the emerging class of ‘planner-centric’ metrics.

1. Introduction

Ever since the first object detection benchmark challenges like the PASCAL VOC [12] became popular, mean average precision (mAP) has been used as the standard metric for evaluating the performance of detection models. Recent works [19, 9, 17] have criticized mAP for its task-agnostic design as the metric assigns equal importance to all objects, which does not reflect real-world priorities for self-driving. Therefore, different task-specific modifications of mAP [9, 3, 13, 32] and planner-centric approaches to detection evaluation [19, 17, 21] have been proposed. These offline metrics are useful because they are quick, cheap, and safe to evaluate compared to online tests. However, relying solely on offline evaluation is only useful if it strongly correlates with the actual driving performance. With the influx of new detection metrics, it has become unclear which metric researchers should rely on and how the metrics compare.

In this work¹, we provide the first empirical evidence of how predictive detection metrics are of downstream driving performance. We train 16 modern 3D detectors, integrate them into a self-driving pipeline, and evaluate their performance in the CARLA simulator [11]. This allows us to study how strongly these metrics correlate with driving

¹To read the full-length report, please visit: <https://t.ly/CsIrt>.

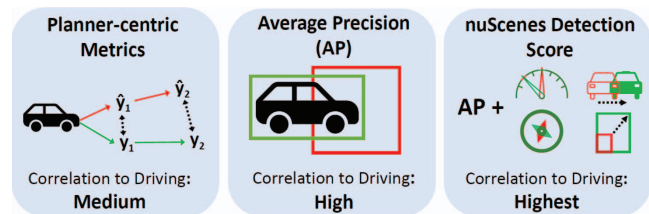


Figure 1. **Summary of findings.** nuScenes Detection Score is most predictive of driving performance in an extensive empirical study on the CARLA simulator.

outcomes. We find that even though mAP is highly correlated with driving performance, the nuScenes Detection Score [3], a task-specific variation, is even more predictive. Furthermore, the planner-centric metrics we examine, which measure the impact of inaccurate detections on planner outcomes, are significantly less indicative of driving performance. Our key findings are summarized in Fig. 1.

2. Related Work

Task-specific Detection Metrics. To track algorithmic advancements, researchers compare object detection models on dataset-based competitions [3, 5, 32, 16, 13] using the mAP metric [24], which does not take the egocentric nature and task-specific characteristics of driving into account. Therefore, prior work has proposed task-specific object detection metrics for self-driving [19, 32, 21, 2, 17]. One approach is taking mAP and adapting it to self-driving. mAPH, the principal metric of the Waymo challenge [32] and the Average Orientation Similarity (AOS) [13] are both designed to account for the importance of correct heading estimation for behaviour planning and weigh detections accordingly. Similarly, Deng et al. [9] suggest evaluating detections from an egocentric perspective and introduce the Support Distance Error (SDE).

Another approach focuses on evaluating the effects of perception errors on the planning module. The planning-KL-divergence (PKL) [23] measures the KL-divergences between distributions of waypoint locations conditioned either on noisy perception or ground truth object annotations. Li et al. [21] emphasize the importance of understand-

ing the internal reasoning of a planner, as some detection failures do not cause immediate behaviour change. However, their approach focuses on model-based planners and does not apply to neural architectures. Ivanovic and Pavone [17] describe an approach where local gradients of a hand-crafted planning function are used to assign object weights.

None of these prior works have provided quantitative results demonstrating that the metric they propose is more closely related to measures of driving performance than the standard mAP metric. We seek to address this gap in the literature by comparing offline evaluations with online tests.

Online and Offline Evaluation. Deep neural networks for self-driving applications are commonly first tested offline with the use of a pre-recorded dataset [26]. Offline results are used as proxy performance indicators for online evaluations, which are more expensive to conduct. It is often unclear how offline measurements relate to system-level functionality in embedded systems. Haq et al. [14] evaluate the correlations between online and offline performance of a camera-based end-to-end lane-keeping model. The authors conclude that offline evaluations cannot be used for safety testing in the context of steering prediction. However, a recent replication of this study with improved methodology obtains a tighter relationship between online and offline results [31]. In similar experiments, prior work has found little to no correlation when comparing online driving performance to offline prediction accuracy for agents in the CARLA [7] and nuPlan [8] simulators.

Contrary to these efforts, we do not focus on steering prediction but aim to evaluate the effects of detection errors on driving outcomes. We aim to provide the first quantitative evidence comparing detection metrics for self-driving to help the community choose relevant metrics.

3. Modular Pipeline for Autonomous Driving

In this section, we present our driving agent. We introduce the problem setting and describe the modular structure of our agent. Next, we discuss the modules for object detection, tracking, and motion planning.

Task and Setup. We address the task of urban driving with the objective of navigating safely along a predetermined route while adhering to traffic regulations. The agent predicts steering and throttle from sensor inputs. While all traffic participants need to be recognized via a LiDAR-based object detector, our agent has privileged access to simulator information concerning the ego lane and traffic light states.

Pipeline. At every timestep, the point cloud of a 360° LiDAR sensor is processed by a 3D object detector which produces a set of oriented bounding box predictions of traffic participants in the scene. Detections are tracked over time to yield consistent predictions and speed estimates for detected objects. Given its high-level navigation goal, the

planner then uses these predictions to decide on an appropriate set of target waypoints that encode the planned trajectory. A PID controller then processes the waypoints to compute appropriate lateral and longitudinal controls. To study the effects of the performance of specific object detectors on the downstream task of driving, all pipeline elements except the detector are constant across experiments.

Detection and Tracking. For our experiments, we consider eight different LiDAR-based 3D object detection architectures. To cover a breadth of architectures in our analysis, we include voxel-based detectors [10, 33], a pillar-based detector [27], and approaches that also incorporate point-based information from the point cloud [20, 28, 29, 28, 30]. This array of detection architectures further includes a mix of anchor-based and anchor-free detection heads as well as single-stage and two-stage approaches. In our setup, detections are associated across frames using Hungarian matching [18]. The speed of tracked objects is approximated via a simple heuristic: per object track, the bounding box centers of the last two timesteps are projected to the ground plane. The L2 norm between these points is divided by the timestep length to approximate speed.

Planning. For our experiments, we choose PlanT [25] for motion planning. PlanT is a transformer-based planner with state-of-the-art performance in the CARLA simulator.

4. Metrics

This section introduces the metrics we use in our experiments: the online metrics that quantify driving performance, the mAP metric, its task-specific variations, and finally two planner-centric metrics.

Online Metrics. Online evaluations provide the most reliable estimates of system-level performance. In this work, we use the CARLA Driving Score, the official metric for the CARLA leaderboard [1], and the number of collisions.

Driving Score. The CARLA Driving Score (DS) is a composite metric combining route completion with the infraction score. The route completion (RC) describes the percentage of the route the agent completed and the infraction score (IS) measures collisions or violations of traffic rules.

Collision Count. As a second metric, we count the total number of collisions per evaluation (#Col.) in which the ego vehicle is involved. This metric exclusively focuses on safety since it does not depend on the route completion.

Average Precision based Metrics. *Average precision* is the standard metric to measure performance on detection tasks. It is defined as the area under the precision-recall curve. Following the KITTI protocol [13], we use an intersection over union (IoU) threshold of 70% as the true positive criterion and compute the integral using a 40-point interpolation with equidistant recall values.

Average Orientation Similarity. As the original mAP metric

does not account for a notion of heading, we have included the average orientation similarity metric (AOS) [13] in our analysis. AOS modifies AP by weighting true positive detections according to the accuracy of their heading predictions. The heading angles of vehicles in the scene provide essential information for motion forecasting and behavior planning [13, 3, 32].

Inverse Distance weighted AP. Another approach for modifying AP is weighting detections by their inverse distance to the ego vehicle [9]. Closer objects are inherently more safety-critical than those far away. Thus, one should expect the weighting of AP by the inverse distance (ID-AP) to produce a metric that better reflects the ego-centric nature of detection for self-driving.

nuScenes Detection Score. The *nuScenes Detection Score* (NDS) [3] is a popular task-specific detection metric that uses mAP as a starting point. It relies on center distance in birds-eye view (BEV) for the true positive criterion instead of the standard IoU-based approach. Here, we use a fixed center distance of 1 meter to not confound our results by averaging across thresholds (the authors suggest averaging over four thresholds). By relying on center distances, the authors decouple mAP from object size and orientation, which they account for separately. They argue that center distance covers objects of different sizes more evenly because smaller volume objects like pedestrians can quickly achieve an IoU of zero if predictions have minor translation errors. NDS also includes five explicit detection quality measures for all true positive detections. These measures are weighted equally and are, in total, given as much weight as average precision in the NDS. The five true positive metrics are (1) Average Translation Error (ATE): Euclidean center distance in BEV in meters; (2) Average Scale Error (ASE): Calculated as $1 - \text{IOU}$ after aligning centers and orientation; (3) Average Orientation Error (AOE): The smaller yaw angle between GT and prediction in radians; (4) Average Velocity Error (AVE): Absolute velocity error in m/s; and (5) Average Attribute Error (AAE): (1-Acc) for the prediction accuracy of additional attributes. We do not include the AAE in our experiments, as the additional attributes are specific to the nuScenes dataset.

Similar to mAP, NDS does not account for an object’s distance to the ego vehicle. We thus also test an inverse distance-weighted version of the NDS (ID-NDS).

Planner-Centric Detection Metrics. Several authors have recently pursued a novel approach to detection metrics for self-driving: focusing on the planner instead of evaluating object detections directly [23, 21, 17]. To evaluate the planner-centric metrics, we first compute the planner’s output for a scene given the ground truth object information. We then also calculate the planner’s output given the noisy output of the perception stack (detection + tracking) and analyse how the two predicted trajectories (i.e., the sets of

waypoints) differ. This approach naturally down-weights distant and less relevant objects in the metrics as they bear little significance to planning.

We use the average and final displacement errors [4] to quantify differences in trajectories. The *average displacement error* (ADE) at timestep t is the average of the point-wise L2 distances between the predicted waypoints based on ground truth detections and the predicted waypoints based on noisy detections. We define the ADE for a route as the mean of all frame-based ADEs in that route. Similarly, we define the *final displacement error* (FDE) as the L2 distance between the final waypoints of two trajectories, averaged over all frames of a route.

5. Experiments

This section discusses our setup for online evaluation and explains how we train and configure the object detection models. We then present the results of our analysis.

Online Evaluation. We base our experiments on the CARLA simulator (Version: 0.9.10) [11]. Within CARLA, we gauge online performance for the detection models by integrating them into our modular pipeline to test driving performance. We use the Longest6 benchmark [6] as an online evaluation protocol. Longest6 contains 36 ~ 1.5 km long routes with high traffic density across six towns.

While evaluating the agents on Longest6, we log detailed ground truth bounding box information about the objects in the scene, the perception stack’s predictions, and the sensor information. Based on the resulting logs, we compute the offline detection metrics for every route. This enables us to compare the observed driving outcomes for a given route with the associated offline detection performance measures.

Implementation. We train eight detection architectures using the open-source LiDAR detection framework OpenPCDet [22] and the default hyperparameter configurations it provides. All architectures are trained for 72 hours on eight NVIDIA GeForce RTX 2080TIs. We include two checkpoints per architecture to increase the number of data points in our analysis. To ensure variance in checkpoints’ behavior, the first is extracted after only 36 hours of training and the second after 72 hours. In total, this results in 16 models.

We generate the training data via the CARLA simulator by observing the driving behaviour of an expert algorithm [6] with privileged access to ground truth information. At every frame, we record 360° LiDAR data with a rotation frequency of 20 Hz. We set the minimum confidence threshold for detections to 0.3, apply non-maximum suppression with an IoU threshold of 0.2 and only present object tracks to the planner that are tracked for four consecutive frames.

Metric Evaluation Protocol. After we evaluate the driving performance for all 16 models on the Longest6 benchmark, we average scores across the benchmark’s 36 routes to at-

Metric	Correlation: DS	Correlation: #Col.
nuScenes Detection Score [3]	0.852	0.907
Average Precision [15]	0.805	0.903
Avg. Displacement Error [4]	0.784	0.770
Avg. Orientation Similarity [13]	0.742	0.894
Final Displacement Error [4]	0.703	0.653

Table 1. **Pearson correlation between online and offline metrics.** Absolute values are shown for clarity.

tain one data point per detector for every metric. For the 16 detector-wise data points, we then calculate Pearson’s correlation coefficients between offline and online metrics. For ease of comparison and interpretation, we provide all Pearson’s r correlation coefficients as absolute values.

Correlation Results. Table 1 presents the correlation coefficients for the base metrics. Note that we only discuss Pearson correlation here for the sake of conciseness but found very similar relations when computing Spearman coefficients. The first clear result from the analysis is that all offline metrics correlate highly to driving outcomes. NDS achieves the highest correlation values, with the standard mAP value in second place. The planner-centric metrics achieve less impressive results than their AP-based alternatives. The strength of the correlation indicates that offline metrics can provide reasonable heuristics for a detector’s performance in online tests. Offline metrics thus seem to be a reliable proxy for rougher comparisons among models and quick hypotheses testing. This insight is significant as the computational cost for online tests is high. The offline metrics only take around four minutes to evaluate, while testing a single model on Longest6 using an NVIDIA GeForce RTX 2080Ti takes three days.

nuScenes Detection Score Ablation. Table 1 shows that NDS correlates more to the CARLA Driving Score than the standard mAP metric (0.85 vs. 0.80). Both metrics have similar correlations to the number of collisions (0.90).

In the original NDS metric, all detection quality measures are given equal importance, and their sum is given as much weight as mAP. We find that even though all detection quality metrics contribute a little bit toward the strong correlation, switching all of them off also produces a metric that correlates more with the Driving Score than mAP (0.82 vs. 0.80). This suggests that the center distance approach is indeed preferable to an IoU based TP-criterion.

Inverse Distance Weighting. We include inverse distance weighted variations for mAP and NDS. We observe that the correlation to Driving Score drops for both metrics (comparing triangles with circles in Figure 2). In contrast, the correlation to the number of collisions increases compared to the original metrics. The fact that inverse distance weighting increases predictive power for collisions is expected. Many collisions likely occur when the percep-

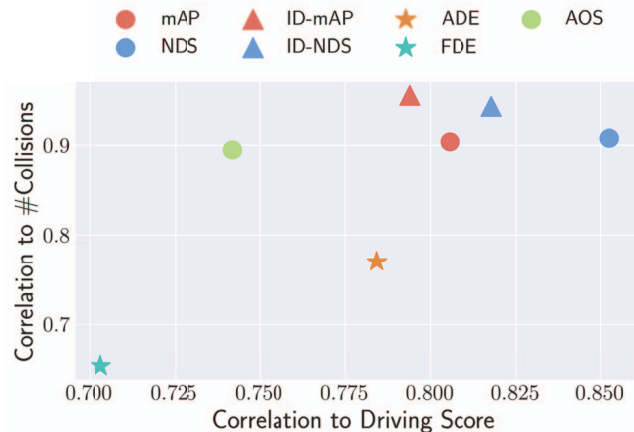


Figure 2. **Correlation analysis summary.** The plot marks correlation coefficients for planner-centric metrics with a star symbol. The inverse distance weighted metrics are marked with triangles.

tion stack misses traffic participants directly in front of the ego. However, inverse distance weighting rewards defensive agents, that achieve less route completion. ID-MAP is especially tightly connected to the collision count, with a Pearson’s r coefficient of 0.955.

Planner-Centric Metrics. Our results show that ADE marks a better indicator for driving performance than FDE. While the planner-centric ADE does correlate to Driving Score and collision count (0.78 & 0.77, respectively), these correlations pale in comparison to those of the mAP-based metrics (see Figure 2). The significant discrepancies in correlations we observe between the approaches, therefore, contra-indicate a reliance on planner-centric metrics alone when evaluating object detection for self-driving.

6. Conclusion

In this work, we demonstrate that common metrics for 3D object detection are highly correlated with online driving performance. Our extensive evaluation shows that the nuScenes Detection Score is more predictive of closed-loop outcomes than the standard mean average precision. While we find the standard mAP score to yield strong correlation nonetheless, our results invoke skepticism regarding detection benchmarks that exclusively rely on planner-centric approaches or use a strong focus on heading accuracy.

There are two important limitations that constrain the degree to which one can generalize from our results. First, we base all our experiments on the same neural planning architecture. In our experiments, PlanT acts as a mediator between the detection performance and the driving outcomes. Different planners might focus on other object cues for motion forecasting and behavior planning. Second, the online metric Driving Score is a relatively simple heuristic for evaluating overall driving performance, and more accurate online metrics might yield different correlation outcomes.

Acknowledgements. This work was supported by the BMBF (Tübingen AI Center, FKZ: 01IS18039A), the DFG (SFB 1233, TP 17, project number: 276693517), and by EXC (number 2064/1 – project number 390727645). We thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting K. Renz and K. Chitta. The authors also thank Luis Winckelmann for several helpful discussions.

References

- [1] Carla leaderboard. leaderboard.carla.org. Accessed: 2023-06-20. [2](#)
- [2] Ayoosh Bansal, Jayati Singh, Micaela Verucchi, Marco Caccamo, and Lui Sha. Risk ranked recall: Collision safety metric for object detection systems in autonomous vehicles. In *Mediterranean Conference on Embedded Computing (MECO)*, 2021. [1](#)
- [3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020. [1](#), [3](#), [4](#)
- [4] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021. [3](#), [4](#)
- [5] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019. [1](#)
- [6] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *Transactions on Pattern Analysis and Machine Intelligence*, 2022. [3](#)
- [7] Felipe Codevilla, Antonio M Lopez, Vladlen Koltun, and Alexey Dosovitskiy. On offline evaluation of vision-based driving models. In *Proceedings of the European Conference on Computer Vision*, 2018. [2](#)
- [8] Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. *arXiv preprint arXiv:2306.07962*, 2023. [2](#)
- [9] Boyang Deng, Charles R Qi, Mahyar Najibi, Thomas Funkhouser, Yin Zhou, and Dragomir Anguelov. Revisiting 3d object detection from an egocentric perspective. *Advances in Neural Information Processing Systems*, 2021. [1](#), [3](#)
- [10] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. [2](#)
- [11] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, 2017. [1](#), [3](#)
- [12] Mark Everingham, Andrew Zisserman, Christopher KI Williams, Luc Van Gool, Moray Allan, Christopher M Bishop, Olivier Chapelle, Navneet Dalal, Thomas Deselaers, Gyuri Dorkó, et al. The 2005 pascal visual object classes challenge. In *Machine Learning Challenges*, 2006. [1](#)
- [13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on computer vision and pattern recognition*, 2012. [1](#), [2](#), [3](#), [4](#)
- [14] Fitash Ul Haq, Donghwan Shin, Shiva Nejati, and Lionel Briand. Can offline testing of deep neural networks replace their online testing? a case study of automated driving systems. *Empirical Software Engineering*, 2021. [2](#)
- [15] Derek Hoiem, Santosh K Divvala, and James H Hays. Pascal voc 2008 challenge. *World Literature Today*, 2009. [4](#)
- [16] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. In *Conference on Robot Learning*, 2021. [1](#)
- [17] Boris Ivanovic and Marco Pavone. Injecting planning-awareness into prediction and detection evaluation. In *Intelligent Vehicles Symposium (IV)*, 2022. [1](#), [2](#), [3](#)
- [18] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955. [2](#)
- [19] Jacob Lambert. Project 3d_lidar.detection_evaluation. github.com/jacoblambert/3d_lidar_detection_evaluation, 2013. Accessed: 2023-05-20. [1](#)
- [20] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019. [2](#)
- [21] Wei-Xin Li and Xiaodong Yang. Transcendental idealism of planner: Evaluating perception from planning perspective for autonomous driving. *arXiv preprint arXiv:2306.07276*, 2023. [1](#), [3](#)
- [22] OD-Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. github.com/open-mmlab/OpenPCDet, 2020. Accessed: 2023-02-20. [3](#)
- [23] Jonah Philion, Amlan Kar, and Sanja Fidler. Learning to evaluate perception models using planner-centric metrics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. [1](#), [3](#)
- [24] Rui Qian, Xin Lai, and Xirong Li. 3d object detection for autonomous driving: a survey. *Pattern Recognition*, 2022. [1](#)
- [25] Katrin Renz, Kashyap Chitta, Otniel-Bogdan Mercea, A Koepke, Zeynep Akata, and Andreas Geiger. Plant: Explainable planning transformers via object-level representations. *arXiv preprint arXiv:2210.14222*, 2022. [2](#)
- [26] Vincenzo Riccio, Gunel Jahangirova, Andrea Stocco, Nar-giz Humbatova, Michael Weiss, and Paolo Tonella. Testing

- machine learning based systems: a systematic mapping. *Empirical Software Engineering*, 2020. 2
- [27] Guangsheng Shi, Ruifeng Li, and Chao Ma. Pillarnet: Real-time and high-performance pillar-based 3d object detection. In *European Conference on Computer Vision*, 2022. 2
- [28] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2
- [29] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *International Journal of Computer Vision*, 2023. 2
- [30] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 2
- [31] Andrea Stocco, Brian Pulfer, and Paolo Tonella. Model vs system level testing of autonomous driving systems: a replication and extension study. *Empirical Software Engineering*, 2023. 2
- [32] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, and Benjamin Caine. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020. 1, 3
- [33] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 2018. 2