CODA: A Real-World Road Corner Case Dataset for Object Detection in Autonomous Driving

Kaican Li[1⋆], Kai Chen[3⋆], Haoyu Wang[1⋆], Lanqing Hong[1†], Chaoqiang Ye[1], Jianhua Han[1], Yukuai Chen[2], Wei Zhang[1], Chunjing Xu[1], Dit-Yan Yeung[3], Xiaodan Liang[5], Zhenguo Li[1], and Hang Xu[1]

[1] Huawei Noah's Ark Lab
[2] Huawei Intelligent Automotive Solution BU
[3] Hong Kong University of Science and Technology
[4] Sun Yat-sen University

Abstract. Contemporary deep-learning object detection methods for autonomous driving usually presume fixed categories of common traffic participants, such as pedestrians and cars. Most existing detectors are unable to detect uncommon objects and corner cases (*e.g.*, a dog crossing a street), which may lead to severe accidents in some situations, making the timeline for the real-world application of reliable autonomous driving uncertain. One main reason that impedes the development of truly reliably self-driving systems is the lack of public datasets for evaluating the performance of object detectors on corner cases. Hence, we introduce a challenging dataset named CODA that exposes this critical problem of vision-based detectors. The dataset consists of 1500 carefully selected real-world driving scenes, each containing four object-level corner cases (on average), spanning more than 30 object categories. On CODA, the performance of standard object detectors trained on large-scale autonomous driving datasets significantly drops to no more than 12.8% in mAR. Moreover, we experiment with the state-of-the-art open-world object detector and find that it also fails to reliably identify the novel objects in CODA, suggesting that a robust perception system for autonomous driving is probably still far from reach. We expect our CODA dataset to facilitate further research in reliable detection for real-world autonomous driving. Our dataset is available at https://coda-dataset.github.io.

Keywords: autonomous driving, object detection, corner case.

1 Introduction

Deep learning has achieved prominent success in object detection for autonomous driving in the wild [5,17,38,47]. The success is mainly attributed to deep neural networks trained on an extensive amount of data extracted from real-life

⋆ Equal contribution.
† Corresponding author at honglanqing@huawei.com.

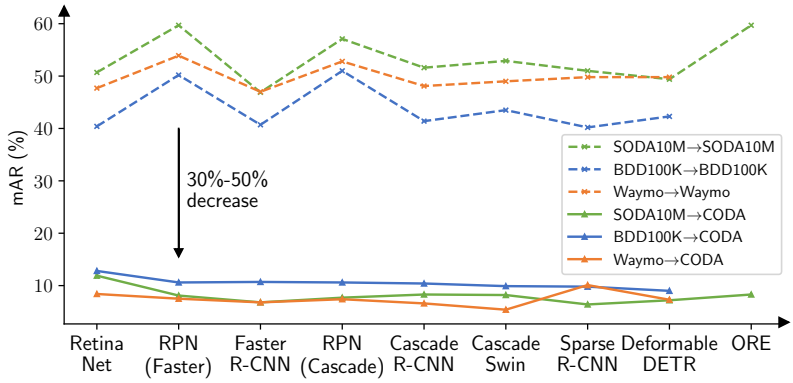

Fig. 1. Detection results on CODA compared with common autonomous driving datasets. All detectors suffer from a significant 30%-50% performance drop, with the best achieved at 12.8% mAR, which is definitely far from solved. Here $A \rightarrow B$ represents that the detector is trained on dataset A and evaluated on dataset B.

driving scenarios, which have become an indispensable component of existing autonomous driving systems [6,13,28]. Though such models are proficient in detecting common traffic participants (*e.g.*, cars, pedestrians, and cyclists), they are generally incapable of detecting novel objects that are not seen or rarely seen in the training process, i.e., the out-of-distribution samples [43,45,46]. For instance, a vehicle equipped with state-of-the-art detectors galloping on the highway may fail to detect a runaway tire or an overturned truck straight ahead of the road. These failure cases of object detection in autonomous driving may result in severe consequences, putting lives at risk.

To address the problem, we introduce CODA, a novel dataset of *object-level corner cases*[5] in real-world driving scenes. CODA is constructed from three major object detection benchmarks for autonomous driving—KITTI [11], nuScenes [4], and ONCE [28]. In Fig. 2, the examples from CODA exhibit a diverse set of scenes and a great variety of novel objects. In total, 1500 scenes (images) are selected from the combined dataset of over one million scenes, leading to nearly 6000 high-quality annotated road corner cases. The selection process of CODA consists of two stages: a fully-automated generation of proposals on potential corner cases followed by manual inspections and corrections on the proposals. Our approach for corner-case proposal generation, COPG, which significantly reduces the amount of human labor in the second stage, is a generic pipeline that only requires raw sensory data from camera and lidar sensor, *i.e.*, no annotation is needed. We believe that the approach can be utilized to efficiently produce more corner case datasets in the future.

On CODA, we have evaluated various kinds of object detection methods including standard (closed-world) detectors such as Faster R-CNN [33]; a recently-proposed open-world detector, ORE [18], which is capable of detecting certain

[5] We adopt the definition of object-level corner case proposed in [3].

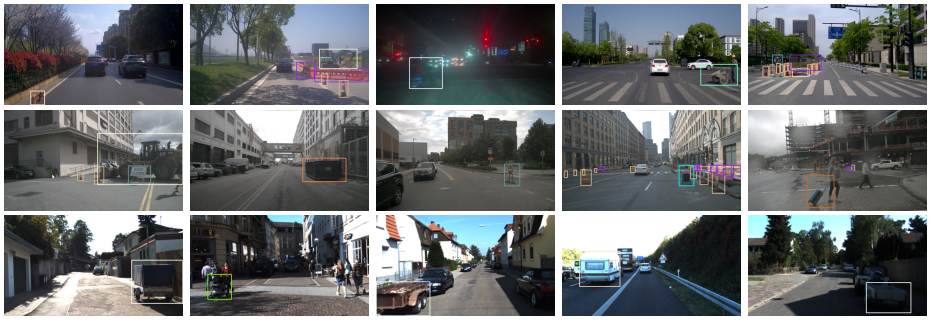

Fig. 2. Examples from CODA. Corner cases are indicated by the bounding boxes, while each color stands for a different object class. CODA contains both *instances of novel classes* (e.g., the dog in the top-left image) and *novel instances of common classes* (e.g., the cyclist in the top-middle image).

objects of unseen classes; and two anomaly detection methods [12,42] which are also in some sense suited to the task. Our experiment results show that none of the methods can consistently detect the novel objects in CODA, demonstrating how challenging CODA is. In general, there is no clear winner among the methods, even though ORE shows some improvements over the closed-world detectors. Finally, we hope that CODA can serve as an effective means for evaluating the robustness of machine perception in autonomous driving, and in turn, facilitate the development of truly reliably self-driving systems. The main contribution of this work can be summarized as follows:

- We propose CODA, the first real-world road corner case dataset, serving as a benchmark for the development of fully reliable self-driving vehicles.
- We evaluate various state-of-the-art object detectors (*e.g.*, Cascade R-CNN [5], Deformable DETR [47], and Sparse R-CNN [38]), suggesting that truly reliably self-driving systems are probably still far from reach.
- We introduce COPG, a generic pipeline for corner-case discovery, reducing human labeling effort by nearly 90% on a large-scale dataset.

2 Related Work

Road anomaly and corner case dataset. One of the pioneering datasets in road anomaly and corner case detection is the Lost and Found dataset [29] which features small objects in artificial scenes. Later introduced datasets mainly focus on semantic segmentation. Notable ones include the road anomaly dataset of Lis *et al.* [23] containing 60 real-world scenes, and Fishyscapes [1], a synthesized dataset created by overlaying objects crawled from the web onto the scenes of Cityscapes [9] and the Lost and Found dataset. StreetHazards [16] is another synthesized dataset where the scenes are simulated by computer graphics. In the same paper, the authors also introduced BDD-Anomaly, a subset of BDD100K [44], treating trains and motorcycles as anomalous objects.

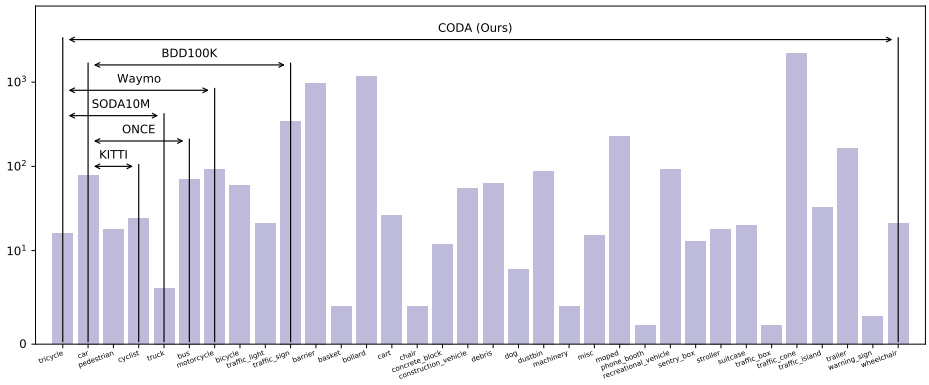

Fig. 3. Class distribution of CODA and annotation coverage of common large-scale autonomous driving benchmarks in comparison with ours. The distribution is inherently long-tailed as suggested by Zipf's law. Class *tram* in SODA10M and class *train* in BDD100K are omitted because CODA does not contain such instance.

Object detection. Existing methods can be generally categorized into one-stage and two-stage based on how the proposals are generated. One-stage detectors [21,24,32] densely predict class distributions and box coordinates on each position of a given image, while two-stage detectors [5,20,33] utilize the Region Proposal Network (RPN) to generate regions of interest (RoI), which are then fed into multi-head networks for class and coordinate offset prediction. Cascade R-CNN [5] further improves by adding a sequence of heads trained with increasing IoU thresholds. ImageNet-supervised pre-training is adopd to accelerate training, while self-supervised pre-training [6,14,27] has recently demonstrated better transfer performance. Previous detectors are mostly trained in the closed-world setting, which can only detect objects belonging to a pre-defined semantic class set. To build a real-world perception system, open-world detection [18] has raised more attention, which can explicitly detect objects of unseen classes as *unknown*.

3 Properties of CODA

Composition. The scenes in CODA are carefully selected from three large-scale autonomous driving datasets: KITTI [11], nuScenes [4], and ONCE [28]. Together, they contribute 1500 diverse scenes to CODA, each containing at least one object-level corner case that is hazardous to self-driving vehicles or their surrounding lives and assets. The corner cases can be generally grouped into 7 super-classes: *vehicle*, *pedestrian*, *cyclist*, *animal*, *traffic facility*, *obstruction*, and *misc*, governing the 34 fine-grained classes listed in Fig. 3. Moreover, these classes can be divided into *novel classes* and *common classes*. Common classes stand for common object categories (e.g., cars and pedestrians) of existing autonomous driving benchmarks; whereas novel classes stand for the opposites, such as dogs

Table 1. Comparison with other datasets. CODA is the largest dataset of its kind in multiple aspects. Here we do not compare with the Fishyscapes Web dataset [1], which is neither publicly available nor with detailed statistics. "†" means rough estimates.

Dataset	#Scenes	Real	Weather	Period	#Classes	#Instances
Lis *et al.* [23]	60	✓	✗	✗	2	300†
Fishyscapes L&F [1]	375	✓	✗	✗	3	500†
Fishyscapes Static [1]	1030	✗	✗	✗	3	1200†
StreetHazards [16]	**1500**	✗	✗	✗	1	1500†
BDD-Anomaly (v1) [16]	361	✓	✗	✗	2	4476
CODA-KITTI (Ours)	309	✓	✓	✓	6	399
CODA-nuScenes (Ours)	134	✓	✓	✓	17	1125
CODA-ONCE (Ours)	1057	✓	✓	✓	32	4413
CODA (Ours)	**1500**	✓	✓	✓	**34**	**5937**

and strollers. More than 90% of the instances in CODA are of novel classes. On one hand, instances of novel classes are inherently undetectable by (closed-world) object detectors that are trained on the common classes. On the other hand, the detectors ought to correctly identify novel instances of common classes, but often fail in doing so. Detailed definitions of common/novel classes will be introduced in Sec. 5, which is important to the evaluation of prevalent object detectors.

Diversity. The data diversity of CODA can be seen from both object level and scene level. On the object level, CODA comprises a wide range of object classes, most of which are neglected by the existing benchmarks (see Fig. 3). Though some class only has several instances (due to the natural scarcity of corner cases), they constitute a nontrivial portion of real-world driving environments. Notably, traffic facilities such as *traffic cone* and *barrier* take up a majority of the corner cases because they are indeed more common and often appear in large quantities.

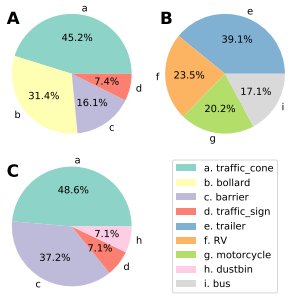

Fig. 4. Distribution of the top-4 classes in the three domains of CODA: **A** ONCE, **B** KITTI, and **C** nuScenes. The distribution largely differs across the domains.

On the scene level, CODA contains scenes from three different countries[6], which are distinct from one another as shown by the examples in Fig. 2. As a result, they introduce more novelty to the corner cases as the difference in object appearance is also a part of the domain shift of the scenes. The disparity between the domains can be seen from Fig. 4, where the distribution of top-4 common classes largely differs. In addition, the scenes in CODA exhibit different weather conditions, of which 75% are clear, 22% are cloudy, and 4% are rainy. Lastly, 9% of the scenes are night scenes apart from the daytime scenes.

[6] KITTI are captured in a mid-size city of Germany, nuScenes are captured in Singapore, and ONCE are captured in various cities of China.

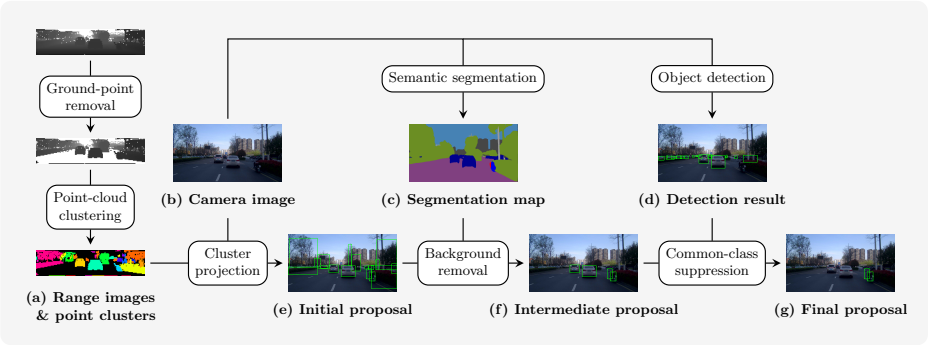

Fig. 5. Pipeline for generating proposals of corner case (COPG). The input to the pipeline is the point cloud and the camera image of a given scene. The point cloud is used to compute **(a)**, whereas the camera image **(b)** is used to produce **(c)** and **(d)**, which then help remove invalid proposals. The output **(g)** is a set of bounding boxes indicating the proposed corner cases in the camera image.

Comparison with road anomaly datasets. In Tab. 1, we compare CODA against several prominent road anomaly datasets that also have object-level annotations. In contrast to CODA, the datasets are either synthetic or small in scale. The largest one of real-world road anomalies, BDD-Anomaly (v1) [16] only contains two object classes, albeit it is comparable to CODA on the number of instances.

4 Construction of CODA

As mentioned earlier, CODA is constructed from three autonomous driving benchmarks, of which most scenes are captured in well-regulated urban areas and therefore contain very few corner cases. To identify them in the large pools of data, we must first define what "corner cases" are in a clearer sense. The main criteria we use for determining whether an object is a corner case are as follows:

- **Risk:** The object blocks or is about to block a potential path of the self-driving vehicle mounted with the camera. Static objects not on the road such as trees and buildings are not considered to block the vehicle.
- **Novelty:** The object does not belong to any of the common classes of autonomous driving benchmarks, or it is a novel instance of the common classes. For simplicity, we take the classes of SODA10M [13] as the common classes.

If an object satisfies both criteria then it is a corner case. The first criterion suggests that the object could be hit by the vehicle and the second criterion suggests that the object is difficult to detect.

4.1 Overview

Adhering to the high-level criteria above, the construction of CODA is carried out in two main stages. The first stage is an automatic generation of proposals that identifies potential corner cases from initial data, followed by the second stage, a manual selection and labeling process that eliminates the false positives of the proposals, and then classifies the remaining true positives while adjusting their bounding boxes to be more precise.

For ONCE [28] consisting of a million scenes, the first stage helps filter out nearly 90% of scenes that are unlikely to contain any corner case, significantly reducing human efforts in the subsequent stage. For KITTI [11] and nuScenes [4], which are considerably smaller than ONCE, we skip the first stage by adopting the ground-truth annotations of uncommon objects that are already provided by the datasets as proposals.

Next, we introduce COPG, our pipeline for corner-case proposal generation (illustrated in Fig. 5). It only requires raw sensory data from a camera and a lidar sensor, *i.e.*, 2D images and 3D point clouds, to identify potential corner cases in any given dataset.

4.2 Identifying Potential Corner Cases

Unsupervised point-cloud clustering. To reliably identify objects satisfying the first criterion, the first step is to learn the location of nearby objects that could obstruct the road. Hence, we turn to lidar point clouds. Since we do not assume any annotation on the points, we start by clustering them so as to separate the objects in the cloud. But before that, we remove all ground-level points by RANSAC [10] to avoid ground points being then clustered as parts of other objects and to suppress the noise from insignificant objects (*e.g.*, tin cans and small branches) on the ground.

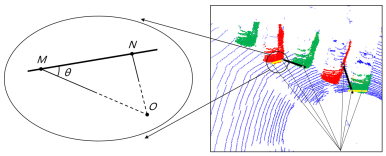

Fig. 6. Abstraction of the point-cloud clustering algorithm [2]. The right figure is a top-view example separating five cars. In the left figure, O denotes the location of the lidar sensor, M and N denote two points in the cloud, while OM and ON denote two lidar beams (OM is the longer beam). If the angle θ is greater than a fixed threshold, then the algorithm labels M and N as points belonging to the same object. The rule is based on the observation that in most cases, if M and N are from the same object, θ is relatively large; however, for those from different objects, θ turns out to be substantially smaller.

Given a point cloud with ground-level points removed, we adopt the algorithm proposed by Bogoslavskyi and Stachniss [2] to cluster the remaining points. The algorithm operates on the range image of the point cloud. A range image is a 2D image showing the distance to points in a scene from a specific point (which is the location of the lidar sensor in our case) and the image has pixel values that correspond to the distance. Given a range image, the algorithm conducts a breadth-first search over the pixels of

the image, and eventually assigns every pixel to a cluster. Specifically, the algorithm compares each pixel p with its four neighboring pixels during the search. If a neighbor p' is sufficiently *close* to p, then they are given the same cluster label. The closeness between pixels is determined by the geometric relationship between the underlying points (in the 3D cloud) of these pixels. See Fig. 6 for a detailed explanation.

After separating points of different objects apart by the clustering algorithm, the points are projected onto the camera images. 2D bounding boxes are then generated for each of the clusters, except those that are too small or too far away from the lidar sensor. These bounding boxes are our initial proposals of corner cases, which is a superset of the final proposals. Next, we apply two other techniques to remove the proposals that violate the predefined criteria.

Background removal. Not all objects found by the point-cloud clustering algorithm satisfy the first criterion since most of the objects are usually off the road. Static objects in the background (*e.g.*, vegetation and buildings) are the most common ones in this category. Discerning these objects from the others requires a semantic understanding of the scene, which could not be derived from merely point-cloud data. Instead, we find semantic segmentation on camera images particularly useful. We utilize a DeepLabv3+ [8] model pre-trained on Cityscapes [9] to produce fine segmentation maps, and then filter out the proposals that has a large overlap (over some threshold) with background regions in the corresponding segmentation map. The following classes are considered as backgrounds: *road*, *sidewalk*, *building*, *wall*, *fence*, *pole*, *vegetation*, *terrain*, and *sky*. After removing the backgrounds, we obtain a set of objects that mostly agrees with our first criterion.

Common-class suppression. To meet the second criterion, the one on the novelty of corner case, we make use of object detectors used in autonomous driving systems to filter out objects that are not considered novel by our standard. Specifically, we utilize Cascade R-CNN [5] with SP-Net backbone [17] trained on a private dataset that is similar to ONCE and consists of millions of scenes to detect common-class objects, producing a set of bounding boxes for each scene. The bounding boxes are subsequently compared with the proposals from the previous step, and those proposals that have IoUs over a threshold with any of the detected common objects are removed.

Note that we do not use ground-truth annotations to suppress the common-class proposals. Our approach has two important advantages: 1) it applies to unlabeled data as long as there is a working detector trained on a similar dataset of the same task; and 2) it keeps some novel instances of the common classes, *i.e.* the "hard cases" in object detection, that would otherwise be suppressed by the ground truth. The effectiveness of COPG is demonstrated in Sec. 6.

4.3 Further Examination

In the previous subsection, we have discussed how to extract potential corner cases from an abundance of unlabeled data. On ONCE [28], the process leaves only around 10% of the scenes for further examination. It is perhaps worth noting that by increasing the thresholds of background removal and common-class suppression, one can further reduce the number of candidate scenes, but it would also cause more corner cases to be neglected. In some sense, the thresholds control the trade-off between the final amount of true positives and the amount of human labor required to pick them out (see Appendix B for relevant ablations).

Selection. Given the generated proposals of ONCE, we start by examining the scenes containing these proposals. Those that do not contain any valid corner case (according to our criteria) are discarded. After the process, we finally arrive at the 1057 scenes of CODA-ONCE, roughly 0.1% of the one million scenes in the original dataset. This shows that corner cases are indeed rare in real-world data. As for KITTI [11] and nuScenes [4], without undergoing the automatic generation of proposals, all data are manually selected, resulting in 309 and 134 scenes respectively.

Labeling. To ease the labeling process, we use CLIP [31] to pre-label the objects in CODA. After that, we use the toolkit [39] inspired by LabelMe [35] to label the class of each corner case and to revise the bounding boxes since the proposals in each scene may not all be valid and the projection from point clouds to camera images is often inaccurate. Meanwhile, some bounding boxes are also added to corner-case objects missed by the proposals in the selected scenes. For quality assurance, the output of each annotator is verified by two other annotators. In the end, most of the corner cases are given a label of a specific class, except the ones that are either unrecognizable or difficult to categorize, which are placed under the *misc* class.

5 Experiment

5.1 Implementation details

Baselines. Four categories of baselines are evaluated on CODA: 1) for *closed-world object detectors*, state-of-the-art detectors of both one-stage (*e.g.*, RetinaNet [21]) and two-stage (*e.g.*, Faster [33] and Cascade R-CNN [5]) pre-trained on SODA10M [13], BDD100K [44] and Waymo [37] are selected; 2) *region proposal network (RPN)* [33] can recognize foreground objects in a class-agnostic manner, which might learn a more generalizable representation, so we further report the performance of the RPN of Faster R-CNN and Cascade R-CNN; 3) for *open-world object detectors*, we adopt the state-of-the-art ORE model [18] but without incremental learning; and 4) for *anomaly detection*, we modify the synthesize then compare [42] and memory-based OOD detection [12] to generate anomaly bounding boxes based on the proposals of a pre-trained RPN.

Table 2. Detection results (%) on CODA. The best performance is achieved at 12.8% AR, suggesting that truly reliable object detection is probably still far from reach. Definitions of ORIGIN, CORNER, COMMON, and NOVEL are provided in "class separation" of Sec. 5.1. "D-DETR" is short for Deformable DETR and "Cascade Swin" stands for Swin-Tiny-based Cascade R-CNN. **Bold** values highlight the best performance among detectors pre-trained on the same dataset, and "†" means official checkpoints are adopted. "*" indicates that AR is the primary evaluation metric on CODA, while "-" suggests that the detector cannot report the corresponding values, with reasons explained in "evaluation" of Sec. 5.1. See more results in Appendix D.

CODA		ORIGIN		CORNER				COMMON				NOVEL			
Method	Dataset	AP	AR	AR^*	AR_{50}	AR_{75}	AR^{10}	AR^*	AR_{50}	AR_{75}	AR^{10}	AR^*	AR_{50}	AR_{75}	AR^{10}
RetinaNet† [21]	SODA10M [13]	34.0	50.7	**11.9**	25.2	9.5	5.4	28.7	58.9	23.5	23.9	-	-	-	-
Faster R-CNN† [33]		36.7	46.9	6.8	13.0	6.4	4.9	23.9	46.8	20.1	23.1	-	-	-	-
Cascade R-CNN† [5]		39.4	51.6	8.3	15.5	7.6	5.5	27.2	47.0	29.4	25.3	-	-	-	-
D-DETR [47]		31.8	49.4	7.2	16.7	4.9	3.6	**34.6**	60.2	36.5	29.6	-	-	-	-
Sparse R-CNN [38]		31.2	51.0	6.4	13.2	5.4	3.9	26.4	47.1	25.6	23.0	-	-	-	-
Cascade Swin [26]		41.1	52.9	8.2	15.5	7.6	5.7	30.4	51.3	32.2	29.3	-	-	-	-
RPN (Faster)† [33]		-	59.7	8.1	16.2	7.4	3.1	-	-	-	-	-	-	-	-
RPN (Cascade)† [5]		-	57.1	7.7	16.0	6.8	2.8	-	-	-	-	-	-	-	-
ORE [18]		**49.2**	**59.7**	8.3	16.4	7.4	5.6	18.5	35.5	18.2	18.1	**3.4**	7.6	2.8	2.9
RetinaNet† [21]	BDD100K [44]	28.6	40.4	**12.8**	23.2	11.9	4.8	27.5	58.1	21.5	23.6	**9.7**	17.7	9.1	5.9
Faster R-CNN† [33]		31.0	40.7	10.7	19.2	10.2	4.3	24.4	48.1	20.9	22.0	7.2	13.3	6.8	5.9
Cascade R-CNN† [5]		32.4	41.4	10.4	18.5	9.7	4.5	25.7	48.4	23.3	23.6	6.9	12.5	6.5	5.7
D-DETR [47]		28.5	42.3	9.0	22.2	5.6	2.8	28.5	63.0	22.3	26.2	7.0	17.3	4.3	3.9
Sparse R-CNN† [38]		26.7	40.2	9.8	19.0	8.9	4.5	27.4	51.7	25.8	24.3	8.0	15.4	7.4	5.1
Cascade Swin [26]		**34.5**	43.5	9.9	17.2	9.7	4.9	**31.0**	55.0	29.9	29.4	6.5	11.4	6.4	5.9
RPN (Faster)† [33]		-	50.2	10.6	20.0	10.2	3.7	-	-	-	-	-	-	-	-
RPN (Cascade)† [5]		-	**51.0**	10.6	20.0	10.2	3.9	-	-	-	-	-	-	-	-
RetinaNet [21]	Waymo [37]	39.7	47.7	8.4	15.6	7.7	5.1	24.5	43.2	24.4	22.2	6.7	11.9	6.4	4.6
Faster R-CNN [33]		40.9	47.0	6.8	12.4	6.4	4.8	20.9	36.0	19.6	19.1	5.5	9.6	5.2	4.3
Cascade R-CNN [5]		42.6	48.1	6.6	11.4	6.6	5.0	18.9	32.6	20.1	17.6	5.3	8.7	5.5	4.4
D-DETR [47]		40.4	49.8	7.3	15.8	5.4	3.6	28.5	49.4	24.6	22.5	5.2	11.5	4.0	3.0
Sparse R-CNN [38]		38.8	49.8	**10.1**	19.6	9.0	4.7	**29.5**	51.8	27.0	22.1	**7.6**	14.3	7.1	4.2
Cascade Swin [26]		**44.2**	49.0	5.4	8.7	5.5	4.4	21.8	38.1	18.8	21.3	4.3	6.7	4.6	3.7
RPN (Faster) [33]		-	**53.9**	7.5	13.7	7.5	3.6	-	-	-	-	-	-	-	-
RPN (Cascade) [5]		-	52.8	7.4	13.8	7.3	3.9	-	-	-	-	-	-	-	-

Optimization. We adopt ResNet-50 [15] initialized with ImageNet-supervised pre-trained weights as the backbone for all baselines except Swin Transformer [26] based Cascade R-CNN, denoted as *Cascade Swin* in Tab. 2. We utilize the officially released checkpoints of closed-world detectors pre-trained on SODA10M and BDD100K, while re-implementing all selected baselines on Waymo, whose official checkpoints are not available, using the MMDetection [7] toolbox. All the BDD100K and Waymo baselines are trained with a batch size of 16 for 12 epochs with an 1000-iteration warmup using the SGD optimizer. The learning rate is set as 0.02, decreased by a factor of 10 at the 8th and 11th epoch. Lastly, we construct ORE based on Faster R-CNN using Detectron2 [41] following the original paper, which is then trained on SODA10M with a batch size of 8 for 24 epochs, the same with the closed-world counterparts. More optimization details are provided in Appendix A.

Class separation. Considering the fact that the semantic class sets of SODA10M, BDD100K, and Waymo differ from each other, all of which are just subsets of the CODA class set, a unified separation of common and novel classes is necessary for a fair comparison of different detectors. Without loss of generality, we define: **1) COMMON** classes as the class set of SODA10M (*i.e.*, *pedestrian*, *cyclist*, *car*, *truck*, *tram*, and *tricycle*), since ORE is trained on SODA10M; **2) NOVEL** classes as the remaining classes of CODA beyond COMMON; **3) CORNER** combines all COMMON and NOVEL classes to match detector predictions in a class-agnostic manner since it is more important to detect an obstacle before distinguishing its semantic class; and **4) ORIGIN** reports detector performance on their pre-trained datasets for reference (*i.e.*, SODA10M test set for SODA10M detectors and the corresponding validation sets for BDD100K and Waymo) since robustness to corner cases should not come at a high cost of detection precision.

Evaluation. By the class separation described above, we divide detector predictions according to the corresponding semantic classes. Specifically, we treat all predictions but the ORE *unknown*, which should be considered as predictions for NOVEL objects, of SODA10M detectors as predictions for COMMON objects. Predictions of *pedestrian*, *rider*, *car*, *truck*, and *train* of BDD100K detectors are considered as COMMON, while the remaining ones are considered as NOVEL. Such a disjoint division, however, is not applicable for Waymo. According to the official document, all recognizable vehicles are annotated as *vehicle* uniformly, suggesting that Waymo baselines can only detect vehicles in a class-agnostic manner. So here, the *vehicle* predictions of Waymo detectors are not only considered as COMMON (along with *pedestrian* and *cyclist*), but also considered as NOVEL, which might put Waymo detectors at advantage, especially for the recall-based evaluation described below, but it does not affect our conclusion. We further project all detected COMMON vehicles to a unified *vehicle* class so that detectors of different datasets have the same COMMON class set, *i.e.* *pedestrian*, *cyclist*, and *vehicle*; while we combine all NOVEL objects to evaluate in a class-agnostic manner since detectors cannot discriminate unseen classes.

Note that under two circumstances, detectors cannot be evaluated (marked as "-" in Tab. 2), including: 1) RPNs can only perform class-agnostic detection, which are only evaluated under ORIGIN and CORNER; and 2) closed-world detectors pre-trained on SODA10M cannot recognize any NOVEL objects, whose semantic class set is considered as CODA COMMON class set.

We utilize the COCO-style Average Recall (AR) as the evaluation metrics instead of Average Precision (AP) since the annotated objects are the most challenging **subset** of all CODA foreground objects. A model that can detect all foreground objects, including those not obstructing the road, would in fact have low AP on CODA. Hence, AR is much more informative than AP. We also consider 1) AR_{50} and AR_{75} for IoU thresholds of 0.5 and 0.75; 2) AR^1 and AR^{10} for at most 1 and 10 boxes per image; and 3) AR^s, AR^m and AR^l for different box scales following COCO definition [22].

5.2 Results

Significance of CODA. Experiment results are reported in Tab. 2. As summarized in Fig. 1, detectors suffer from a significant performance drop of 30%-50% AR when deployed on CODA (*e.g.*, 43.3% decrease for SODA10M Cascade R-CNN). Even for COMMON classes, the average decrease has also exceeded 21%. The best performance is achieved at 12.8% AR, which is still far from solved even considering the domain gap between CODA and pre-trained datasets. See more complete performance statistics in Appendix D.

Detectors. As shown in Tab. 2, Cascade R-CNN outperforms Faster R-CNN on CODA in general, not only for COMMON classes but also in the setting of CORNER class with a consistent improvement on the ORIGIN datasets, demonstrating the possibility to achieve higher AR on CODA without a decrease of AP on common datasets for more powerful detectors. On the contrary, RetinaNet exceeds Cascade R-CNN at the expense of AP drop, probably due to the dense prediction design. Note that Cascade R-CNN performs comparably or even better than RetinaNet referring to AR^{10} (*e.g.*, 5.5% vs. 5.4% pre-trained on SODA10M), suggesting that the AR improvement might come from more box predictions (*e.g.*, averaged 86 and 21 boxes/image for SODA10M RetinaNet and Cascade R-CNN). RPN brings minor improvement but is significantly surpassed by RetinaNet even though RPN generates more box predictions (*e.g.*, 1000 vs. 86 boxes/image on SODA10M), showing that class-aware training might be beneficial to learn a more discriminative and robust detector. Surprisingly, we observe that ORE, the open-world detector, brings improvement on both CODA and SODA10M test set, about which more analyses are provided in Sec. 6.

Pre-train datasets. BDD100K detectors perform the best among three datasets, especially for the NOVEL class since BDD100K has the largest annotated semantic class set, which is definitely beneficial to detect more complicated objects and learn a more discriminative representation as previously discussed. However, it is impossible to annotate all possible semantic classes due to the complexity of real-world road scenes. So we hope CODA can motivate researchers to consider more scalable and effective solutions to build a robust perception system.

6 Discussion

Effectiveness of COPG. The examples in Appendix E qualitatively demonstrates the effectiveness of COPG, reliably identifying nearby objects and retaining corner cases in a progressive manner. We further quantitatively study the effectiveness of COPG by considering it as a *corner-case detector*, instead of a *corner-case proposal generator*. The evaluation result is shown in Tab. 3, where COPG is compared with other object and anomaly detectors on detecting the corner cases in CODA-KITTI. Note that CODA-KITTI is curated by manually examining all the "misc"-category annotation of KITTI [11]. In other words,

Table 3. Evaluation of COPG and other object/anomaly detectors on detecting corner cases. The experiments are conducted on CODA-KITTI whose construction does not involve COPG (whereas the construction of CODA-ONCE does). Here AR^m_{50} and AR^l_{50} represent AR_{50} for medium and large objects, since no small corner cases are included in CODA-KITTI, with the same definition for AR^m_{30} and AR^l_{30} under 0.3 IoU threshold.

Method	AR^m_{50}	AR^l_{50}	AR^m_{30}	AR^l_{30}
Faster R-CNN [33]	6.7	8.3	26.4	28.8
Memory-based OOD [12]	2.2	21.8	6.6	39.5
Synthesize then Compare [42]	9.0	17.7	12.3	33.3
COPG (Ours)	**23.8**	**44.9**	**39.6**	**63.9**

the construction of CODA-KITTI does not involve COPG. As reported, COPG shows significant improvements and is much more comparable to human than the baselines.

Moreover, comparing the baseline performances on CODA (Tab. 2) with those on CODA-ONCE (Tab. 8 in Appendix D), we notice all detectors generally achieve higher AR on CODA than CODA-ONCE (*e.g.*, 12.8% vs 10.2% AR for BDD100K-trained RetinaNet), suggesting that CODA-ONCE constructed based on the proposals of COPG is much harder than the corner cases of the other two subsets whose construction does not involve COPG.

Comparison between closed-world and open-world object detection. We visualize and compare the detection results of Faster R-CNN, ORE and CODA ground truth in Fig. 7. Considering that *unknown* objects are usually trained as background for object detection, ORE utilizes the SODA10M validation set to estimate the known and unknown energy functions based on EBM [19]. As shown in Fig. 7, by using an extra data source, ORE can successfully deal with the corner cases of both common and novel classes, which is consistent with the experiment results in Tab. 2. The usage of an extra data source might put ORE at advantage, but the improvement is still impressive since the extra data is only used for the energy function estimation without updating the parameters of the detector at training time.

The performance of ORE does remind us that it is possible to build a more robust perception system by utilizing an additional data source to separate background and *unknown* objects. However, for ORE, the extra data source is required to be labeled. Considering the annotation cost, it is more desirable to build a system requiring unlabeled data only (*e.g.*, SODA10M large-scale unlabeled set, which has demonstrated to improve cross-domain performance [13]), of which CODA would be a great help in the evaluation.

Evaluation of few-shot object detection (FSOD). The main goal of CODA is to evaluate the generalization ability of object detectors in self-driving systems *without* model adaptation. Nevertheless, it can also be used to evaluate adaptation methods like FSOD. So, apart from the typical baselines included in Tab. 2,

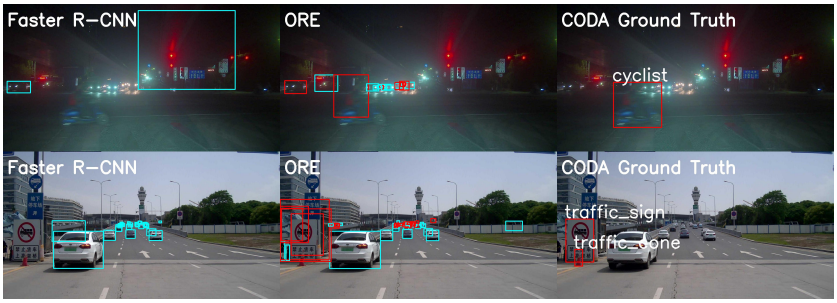

Fig. 7. Visualization of Faster R-CNN (left), ORE (middle) detection results and corner case ground truth (right) on CODA. We annotate the *unknown* predictions of ORE and CODA ground truth with *red* boxes, while the common-class predictions are annotated by *blue* boxes. ORE solves the corner cases of both common (top, cyclist) and novel (bottom, traffic cone & sign) classes.

Table 4. Evaluation of FSOD on CODA. "††" suggests that the reported values are evaluated in a class-agnostic manner, same as the CORNER setting adopted in Tab. 2.

Method	34-way (class-wise)			1-way†† (class-agnostic)		
	AR	AR_{50}	AR_{75}	AR	AR_{50}	AR_{75}
FsDet [40]	$4.9_{\pm 0.8}$	$9.4_{\pm 1.9}$	$4.4_{\pm 0.8}$	$4.2_{\pm 0.4}$	$7.7_{\pm 0.7}$	$4.0_{\pm 0.3}$
DeFRCN [30]	$6.7_{\pm 1.2}$	$12.1_{\pm 1.6}$	$6.6_{\pm 1.6}$	$4.5_{\pm 0.5}$	$8.9_{\pm 0.9}$	$4.2_{\pm 0.5}$

we have also evaluated two of the state-of-the-art FSOD methods, FsDet [40] and DeFRCN [30], on CODA in a 34-way-1-shot setting with **5-time** repeated experiments (see Tab. 4). Neither method demonstrates satisfying performance.

Limitation and potential negative societal impact. We would continue to enlarge CODA by exploring: 1) Use COPG on more real-world road scenes. 2) Since CODA is collected in the real world with high-quality annotation, we can generate more synthesized images following [1,16], or mine large-scale unlabeled road scene images in a semi-supervised manner [30,34,36,25]. Further discussion about potential negative societal impact of CODA are provided in Appendix C.

7 Conclusion

In this paper, we propose CODA, a real-world road corner case dataset for object detection in autonomous driving, constructed by ground truth class separation and automatic proposal. We observe a significant performance drop for state-of-the-art detectors when deployed on CODA. We further provide a thorough comparison of different methods and shed light on potential solutions to a more robust perception system. We hope that CODA can motivate further research in reliable detection for real-world autonomous driving.

References

1. Blum, H., Sarlin, P.E., Nieto, J., Siegwart, R., Cadena, C.: The fishyscapes benchmark: Measuring blind spots in semantic segmentation. arXiv preprint arXiv:1904.03215 (2019) 3, 5, 14
2. Bogoslavskyi, I., Stachniss, C.: Fast range image-based segmentation of sparse 3d laser scans for online operation. In: IROS (2016) 7
3. Breitenstein, J., Termöhlen, J.A., Lipinski, D., Fingscheidt, T.: Corner cases for visual perception in automated driving: Some guidance on detection approaches. arXiv preprint arXiv:2102.05897 (2021) 2
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. arXiv preprint arXiv:1903.11027 (2019) 2, 4, 7, 9
5. Cai, Z., Vasconcelos, N.: Cascade R-CNN: delving into high quality object detection. In: CVPR (2018) 1, 3, 4, 8, 9, 10
6. Chen, K., Hong, L., Xu, H., Li, Z., Yeung, D.Y.: Multisiam: Self-supervised multi-instance siamese representation learning for autonomous driving. In: ICCV (2021) 2, 4
7. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019) 10
8. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018) 8
9. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016) 3, 8
10. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981) 7
11. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR (2012) 2, 4, 7, 9, 12
12. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., van den Hengel, A.: Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: ICCV (2019) 3, 9, 13
13. Han, J., Liang, X., Xu, H., Chen, K., Hong, L., Mao, J., Ye, C., Zhang, W., Li, Z., Liang, X., Xu, C.: SODA10M: A large-scale 2d self/semi-supervised object detection dataset for autonomous driving. arXiv preprint arXiv:2106.11118 (2021) 2, 6, 9, 10, 13
14. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020) 4
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 10
16. Hendrycks, D., Basart, S., Mazeika, M., Mostajabi, M., Steinhardt, J., Song, D.: A benchmark for anomaly segmentation. arXiv preprint arXiv:1911.11132 (2019) 3, 5, 6, 14
17. Jiang, C., Xu, H., Zhang, W., Liang, X., Li, Z.: SP-NAS: serial-to-parallel backbone search for object detection. In: CVPR (2020) 1, 8

18. Joseph, K., Khan, S., Khan, F.S., Balasubramanian, V.N.: Towards open world object detection. In: CVPR (2021) 2, 4, 9, 10
19. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F.: A tutorial on energy-based learning. Predicting Structured Data **1**(0) (2006) 13
20. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017) 4
21. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017) 4, 9, 10
22. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: ECCV (2014) 11
23. Lis, K., Nakka, K., Fua, P., Salzmann, M.: Detecting the unexpected via image resynthesis. In: ICCV (2019) 3, 5
24. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV (2016) 4
25. Liu, Y.C., Ma, C.Y., He, Z., Kuo, C.W., Chen, K., Zhang, P., Wu, B., Kira, Z., Vajda, P.: Unbiased teacher for semi-supervised object detection. In: ICLR (2021) 14
26. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021) 10
27. Liu, Z., Han, J., Chen, K., Hong, L., Xu, H., Xu, C., Li, Z.: Task-customized self-supervised pre-training with scalable dynamic routing. In: AAAI (2022) 4
28. Mao, J., Niu, M., Jiang, C., Liang, H., Chen, J., Liang, X., Li, Y., Ye, C., Zhang, W., Li, Z., et al.: One million scenes for autonomous driving: ONCE dataset. arXiv preprint arXiv:2106.11037 (2021) 2, 4, 7, 9
29. Pinggera, P., Ramos, S., Gehrig, S., Franke, U., Rother, C., Mester, R.: Lost and found: detecting small road hazards for self-driving vehicles. In: IROS (2016) 3
30. Qiao, L., Zhao, Y., Li, Z., Qiu, X., Wu, J., Zhang, C.: DeFRCN: Decoupled faster R-CNN for few-shot object detection. In: ICCV (2021) 14
31. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021) 9
32. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR (2016) 4
33. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NeurIPS (2015) 2, 4, 9, 10, 13
34. Reza, M.A., Naik, A.U., Chen, K., Crandall, D.J.: Automatic annotation for semantic segmentation in indoor scenes. In: IROS (2019) 14
35. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. IJCV **77**(1-3), 157–173 (2008) 9
36. Sohn, K., Zhang, Z., Li, C.L., Zhang, H., Lee, C.Y., Pfister, T.: A simple semi-supervised learning framework for object detection. arXiv:2005.04757 (2020) 14
37. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR (2020) 9, 10
38. Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., et al.: Sparse r-cnn: End-to-end object detection with learnable proposals. In: CVPR (2021) 1, 3, 10

39. Wada, K.: labelme: Image Polygonal Annotation with Python. https://github.com/wkentaro/labelme (2016) 9
40. Wang, X., Huang, T., Gonzalez, J., Darrell, T., Yu, F.: Frustratingly simple few-shot object detection. In: ICML (2020) 14
41. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019) 10
42. Xia, Y., Zhang, Y., Liu, F., Shen, W., Yuille, A.L.: Synthesize then compare: Detecting failures and anomalies for semantic segmentation. In: ECCV (2020) 3, 9, 13
43. Ye, N., Li, K., Bai, H., Yu, R., Hong, L., Zhou, F., Li, Z., Zhu, J.: Ood-bench: Quantifying and understanding two dimensions of out-of-distribution generalization. In: CVPR (2022) 2
44. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: CVPR (2020) 3, 9, 10
45. Zhou, X., Lin, Y., Pi, R., Zhang, W., Xu, R., Cui, P., Zhang, T.: Model agnostic sample reweighting for out-of-distribution learning. In: ICML (2022) 2
46. Zhou, X., Lin, Y., Zhang, W., Zhang, T.: Sparse invariant risk minimization. In: ICML (2022) 2
47. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020) 1, 3, 10