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

Appendix

The official website of CODA is at https://coda-dataset.github.io, where we have released 1000 of the 1500 annotated scenes of CODA. The remaining 500 scenes are reserved for the corner case challenge in the SSLAD workshop[5] of ECCV 2022, and will be released after the challenge.

As a continued effort, CODA is further extended by 8711 additional scenes with more than 28k new corner cases during the reviewing and publishing process of this paper. The extension will also be made fully available after the ECCV challenge. Please stay tuned!

A Supplementary implementation details

A.1 Closed-world detectors

We re-implement the closed-world detectors in Tab. 2 of the main paper following the default configurations in MMDetection [2] if the corresponding official checkpoints[6] are not publicly available. Detectors are by default trained for 12 epochs (1x) on BDD100K and Waymo, while for 24 epochs (2x) on SODA10M, except Deformable DETR and Sparse R-CNN, which are trained for 100 epochs on SODA10M training set, due to the limited labeled data size of SODA10M and data-hungry property of the Transformer layers.

⋆ Equal contribution.
† Corresponding author at honglanqing@huawei.com.
[5] https://ssladcompetition.github.io/
[6] SODA10M: https://soda-2d.github.io/ & BDD100K: https://github.com/SysCV/bdd100k-models/

A.2 ORE

The experiments are conducted using the released code[7] of the original paper [7], without incremental learning. We use the training set of SODA10M [4] for training; the validation set of SODA10M and the whole CODA for fitting the energy distributions; the test set of SODA10M and the whole CODA for testing. The backbone of the network is ResNet-50 [6], and the whole network is trained for 24 epochs with a batch size of 8 under a learning rate of 0.02. More efficient training methods (*e.g.*, sparse training [18]) are potential solutions for better generalization, which will be explored in the future.

A.3 Anomaly detection

We utilize the segmentation model and GAN pre-trained on Cityscapes dataset [3] in method *synthesize then compare* during the test procedure. We crop the generated fake images and the original images according to the top-ranking bounding box proposals obtained from RPN, then we compare the cropped images by pixel-wise cosine similarity as illustrated in the paper and get the detection result according to the similarity rankings. As to *memory-based OOD detection*, we utilize the feature map of common ground truth and some conventional background (vegetation, sky, and pure black) extracted from ResNet-152 [6] as the memory bank. Likewise, We then compare the feature map extracted from off-the-shelf ResNet-152 of the top-ranking bounding boxes from RPN with the memory bank to filter out the anomaly detection results. More advanced OOD detection methods [16,17] will be explored in the future.

B Supplementary ablation studies on COPG

We conduct ablation studies on COPG by tuning its components one at a time. The experiment is conducted on CODA-ONCE whose final ground truths are utilized to compute the AP and AR (under COCO [9] protocol) of the output proposals of the modified COPGs. The results are shown in Tab. 1-3 below, where the **bold** hyperparameters are the ones in effect. The number of proposals (#Proposals) and the number of scenes (#Scenes) containing at least one proposal are also reported.

C Supplementary discussion on potential negative societal impact

CODA has no potential negative societal impact since it is constructed totally based on publicly available datasets with delicate privacy protection. For example, all objects containing personal information (*e.g.*, human faces, license plates) are blurred in ONCE [11].

[7] https://github.com/JosephKJ/OWOD

Table 1. Ablation study on point-cloud clustering. The min. θ controls the minimum tolerance of the angle between two points being assigned to the same object (see Fig. 6 of the main paper). The cluster size controls the minimum tolerance to the number of points of a cluster being considered as a non-trivial object. The max. dist. controls the maximum tolerance to the distance from the nearest point of a cluster to the lidar sensor.

Min. θ	AP	AR	#Proposals	#Scenes
4°	1.3	3.1	1039	763
8°	2.0	5.1	1522	1040
12°	1.9	5.0	1620	1000
16°	1.7	4.7	1652	965

Cluster size	AP	AR	#Proposals	#Scenes
5	2.0	5.1	1522	1040
10	2.0	5.1	1522	1040
20	2.0	5.1	1513	1038

Max. dist.	AP	AR	#Proposals	#Scenes
25	1.8	4.5	1336	935
50	2.0	5.1	1522	1040
100	1.9	5.1	1548	1040

Table 2. Ablation study on background removal. The background (BG) ratio controls the maximum tolerance to the ratio of the proposal area overlapped with the background regions.

BG ratio	AP	AR	#Proposals	#Scenes
0.15	0.4	1.1	412	308
0.30	1.0	2.6	834	614
0.45	2.0	5.1	1522	1040
0.60	2.0	5.6	1834	1055
0.75	1.8	6.1	2315	1056

Table 3. Ablation study on common-class suppression. The IoU threshold controls the maximum tolerance to the IoU between every proposal and each detected common-class object.

IoU	AP	AR	#Proposals	#Scenes
0.00	1.2	2.7	834	643
0.25	2.0	5.1	1522	1040
0.50	1.8	5.2	1660	1041

D Supplementary benchmark results

Table 4. Detection results (%) on our CODA-ONCE dataset. The dramatic performance decrease still maintains, but compared with Tab. 2 of the main paper, we observe a decrease for the reported AR values of all detectors, suggesting that CODA-ONCE constructed with our automatic corner case proposal generation COPG in Sec. 4.2 of the main paper is the most challenging subset of CODA.

CODA-ONCE		CORNER				COMMON				NOVEL			
Method	Dataset	AR^*	AR_{50}	AR_{75}	AR^{10}	AR^*	AR_{50}	AR_{75}	AR^{10}	AR^*	AR_{50}	AR_{75}	AR^{10}
RetinaNet [8]		**9.6**	21.8	7.0	3.0	29.8	59.8	24.5	24.8	-	-	-	-
Faster R-CNN [12]		4.0	8.5	3.4	2.3	23.3	44.3	19.6	22.5	-	-	-	-
Cascade R-CNN [1]		5.6	11.4	4.9	2.9	27.2	46.0	30.7	25.3	-	-	-	-
Deformable DETR [19]		6.1	13.0	5.3	2.7	**36.8**	63.9	40.0	31.2	-	-	-	-
Sparse R-CNN [14]	SODA10M	4.4	9.2	3.7	2.1	26.3	44.5	25.5	22.7	-	-	-	-
Cascade Swin [10]	[4]	5.5	11.3	4.8	2.9	31.2	49.1	35.5	30.0	-	-	-	-
RPN (Faster) [12]		5.1	11.3	4.0	1.3	-	-	-	-	-	-	-	-
RPN (Cascade) [1]		4.8	11.4	3.5	1.3	-	-	-	-	-	-	-	-
ORE [7]		5.3	11.5	4.4	2.8	19.2	36.8	17.1	18.6	**2.0**	4.8	1.6	1.7
RetinaNet [8]		**10.2**	19.4	9.1	2.6	30.1	63.7	23.3	25.8	**7.3**	14.1	6.6	3.8
Faster R-CNN [12]		7.8	15.1	7.4	2.3	26.8	51.8	23.2	24.3	5.5	10.7	5.2	4.1
Cascade R-CNN [1]		7.7	14.6	6.9	2.4	26.7	48.8	24.0	24.6	5.7	10.5	5.1	4.3
Deformable DETR [19]	BDD100K	7.8	18.4	5.6	1.9	30.9	67.8	24.4	28.5	5.8	13.4	4.2	3.0
Sparse R-CNN [14]	[15]	7.6	15.4	6.7	2.6	30.3	56.0	28.8	26.9	5.9	12.0	5.3	3.4
Cascade Swin [10]		7.2	13.3	6.8	2.9	**32.9**	57.0	33.0	31.3	5.3	9.9	5.0	4.5
RPN (Faster) [12]		7.9	16.1	7.0	1.6	-	-	-	-	-	-	-	-
RPN (Cascade) [1]		7.7	15.8	6.9	1.8	-	-	-	-	-	-	-	-
RetinaNet [8]		5.4	11.1	4.5	2.2	26.1	45.0	27.0	23.9	3.0	6.1	2.7	1.4
Faster R-CNN [12]		4.1	8.3	3.5	2.1	22.9	38.9	21.9	20.8	2.3	4.9	2.0	1.4
Cascade R-CNN [1]		3.8	7.1	3.6	2.2	20.0	34.1	21.8	18.7	2.2	3.8	2.2	1.5
Deformable DETR [19]	Waymo	5.5	11.4	4.6	1.9	30.8	52.9	26.8	24.0	3.1	6.5	2.8	1.1
Sparse R-CNN [14]	[13]	**7.3**	14.6	6.4	2.1	**32.4**	55.4	30.0	23.9	**4.2**	8.6	3.8	1.4
Cascade Swin [10]		2.5	4.4	2.4	1.8	23.5	41.1	20.7	23.0	1.2	2.0	1.2	0.8
RPN (Faster) [12]		4.6	9.4	4.3	1.5	-	-	-	-	-	-	-	-
RPN (Cascade) [1]		4.3	9.2	3.6	1.6	-	-	-	-	-	-	-	-

Table 5. More detection results(%) on our CODA-ONCE dataset. AR^s, AR^m and AR^l are the average recall for small, medium and large objects respectively, following COCO definition [9], while AR^1 represents the average recall when only one object prediction is allowed for each image. Here AR^s is also marked as "-", since no small corner cases of common classes are collected in CODA-ONCE subset.

CODA-ONCE		CORNER				COMMON				NOVEL			
Method	Dataset	AR^s	AR^m	AR^l	AR^1	AR^s	AR^m	AR^l	AR^1	AR^s	AR^m	AR^l	AR^1
RetinaNet [8]		3.8	2.9	19.4	0.2	-	0.0	33.4	4.7	-	-	-	-
Faster R-CNN [12]		0.6	1.0	8.5	0.2	-	0.0	26.0	3.7	-	-	-	-
Cascade R-CNN [1]		1.0	1.5	11.9	0.2	-	0.0	30.8	4.9	-	-	-	-
Deformable DETR [19]		0.7	1.9	12.6	0.2	-	1.1	42.3	6.1	-	-	-	-
Sparse R-CNN [14]	SODA10M	1.3	1.6	8.6	0.3	-	3.3	28.6	7.2	-	-	-	-
Cascade Swin [10]	[5]	1.1	1.5	11.5	0.2	-	0.0	35.8	3.9	-	-	-	-
RPN (Faster) [12]		2.2	1.5	10.3	0.3	-	-	-	-	-	-	-	-
RPN (Cascade) [1]		3.6	1.5	9.3	0.3	-	-	-	-	-	-	-	-
ORE [7]		1.6	1.9	10.4	0.3	-	1.7	21.1	5.3	0.5	0.6	4.4	0.4
RetinaNet [8]		2.2	4.4	19.2	0.2	-	9.2	33.1	5.9	1.1	3.4	13.9	0.6
Faster R-CNN [12]		1.3	2.6	15.9	0.1	-	17.2	29.3	4.7	0.4	2.2	11.2	0.5
Cascade R-CNN [1]		1.7	2.6	15.5	0.1	-	12.5	30.4	5.4	0.8	2.2	11.5	0.6
Deformable DETR [19]	BDD100K	1.3	3.6	14.5	0.1	-	18.9	33.1	7.2	1.0	3.0	10.8	0.4
Sparse R-CNN [14]	[15]	1.7	3.3	14.2	0.2	-	4.4	34.3	5.4	1.3	2.9	11.0	0.5
Cascade Swin [10]		1.3	2.4	14.5	0.3	-	27.5	37.3	9.6	0.7	2.0	10.8	0.8
RPN (Faster) [12]		2.2	3.0	15.3	0.2	-	-	-	-	-	-	-	-
RPN (Cascade) [1]		1.9	2.9	14.9	0.2	-	-	-	-	-	-	-	-
RetinaNet [8]		5.4	11.1	4.5	2.2	26.1	45.0	27.0	23.9	3.0	6.1	2.7	1.4
Faster R-CNN [12]		4.1	8.3	3.5	2.1	22.9	38.9	21.9	20.8	2.3	4.9	2.0	1.4
Cascade R-CNN [1]		3.8	7.1	3.6	2.2	20.0	34.1	21.8	18.7	2.2	3.8	2.2	1.5
Deformable DETR [19]	Waymo	5.5	11.4	4.6	1.9	30.8	52.9	26.8	24.0	3.1	6.5	2.8	1.1
Sparse R-CNN [14]	[13]	**7.3**	14.6	6.4	2.1	**32.4**	55.4	30.0	23.9	**4.2**	8.6	3.8	1.4
Cascade Swin [10]		2.5	4.4	2.4	1.8	23.5	41.1	20.7	23.0	1.2	2.0	1.2	0.8
RPN (Faster) [12]		4.6	9.4	4.3	1.5	-	-	-	-	-	-	-	-
RPN (Cascade) [1]		4.3	9.2	3.6	1.6	-	-	-	-	-	-	-	-

Table 6. More detection results(%) on our CODA dataset. AR^s, AR^m and AR^l are the average recall for small, medium and large objects respectively, following COCO definition [9], while AR^1 represents the average recall when only one object prediction is allowed for each image.

CODA		CORNER				COMMON				NOVEL			
Method	Dataset	AR^s	AR^m	AR^l	AR^1	AR^s	AR^m	AR^l	AR^1	AR^s	AR^m	AR^l	AR^1
RetinaNet [8]		5.6	6.5	20.6	0.2	40.0	1.3	31.8	5.0	-	-	-	-
Faster R-CNN [12]		2.7	4.6	10.8	0.2	60.0	4.7	24.8	5.6	-	-	-	-
Cascade R-CNN [1]		3.3	5.1	13.7	0.2	60.0	4.7	28.6	6.2	-	-	-	-
Deformable DETR [19]		1.9	3.5	13.6	0.2	0.0	6.7	38.8	6.5	-	-	-	-
Sparse R-CNN [14]	SODA10M	1.7	4.4	10.2	0.5	70.0	5.3	27.2	8.3	-	-	-	-
Cascade Swin [10]	[4]	3.3	5.3	13.2	0.3	50.0	4.0	33.7	4.7	-	-	-	-
RPN (Faster) [12]		5.1	5.7	11.9	0.6	-	-	-	-	-	-	-	-
RPN (Cascade) [1]		4.8	5.3	11.6	0.5	-	-	-	-	-	-	-	-
ORE [7]		4.0	5.5	13.0	0.3	0.0	1.3	20.2	5.5	1.4	1.9	5.9	2.9
RetinaNet [8]		4.1	8.3	21.0	0.1	0.0	7.7	30.1	5.4	3.1	6.5	15.9	1.2
Faster R-CNN [12]		3.4	6.6	17.8	0.1	10.0	15.7	26.3	4.3	2.2	4.4	12.3	0.9
Cascade R-CNN [1]		3.3	6.6	17.2	0.1	60.0	15.0	27.5	5.8	1.7	4.1	12.3	1.1
Deformable DETR [19]	BDD100K	2.1	5.4	15.5	0.1	0.0	17.7	29.9	7.1	1.7	4.4	12.0	0.5
Sparse R-CNN [14]	[15]	3.2	6.6	15.6	0.2	0.0	4.0	30.7	4.9	2.8	5.7	12.7	1.1
Cascade Swin [10]		3.4	6.3	16.4	0.2	0.0	27.0	35.5	9.8	2.0	3.9	11.4	1.6
RPN (Faster) [12]		4.3	6.9	17.2	0.5	-	-	-	-	-	-	-	-
RPN (Cascade) [1]		3.8	6.9	17.1	0.4	-	-	-	-	-	-	-	-
RetinaNet [8]		1.2	5.6	14.0	0.2	30.0	4.0	27.1	4.5	0.6	4.8	10.9	0.0
Faster R-CNN [12]		1.1	4.5	11.3	0.2	0.0	50.0	23.5	4.2	0.7	4.1	8.6	0.1
Cascade R-CNN [1]		1.2	4.5	10.8	0.1	0.0	10.0	20.5	3.7	0.6	4.2	8.2	0.1
Deformable DETR [19]	Waymo	2.1	4.9	11.7	0.2	0.0	8.0	30.8	2.6	1.2	3.8	8.3	0.1
Sparse R-CNN [14]	[13]	3.3	7.5	15.2	0.4	0.0	4.0	33.2	4.7	2.4	5.8	11.6	0.3
Cascade Swin [10]		0.3	3.7	8.8	0.3	0.0	8.3	24.5	2.8	0.2	3.5	6.4	0.2
RPN (Faster) [12]		1.3	5.1	12.3	0.5	-	-	-	-	-	-	-	-
RPN (Cascade) [1]		1.0	5.0	12.3	0.4	-	-	-	-	-	-	-	-

E COPG examples

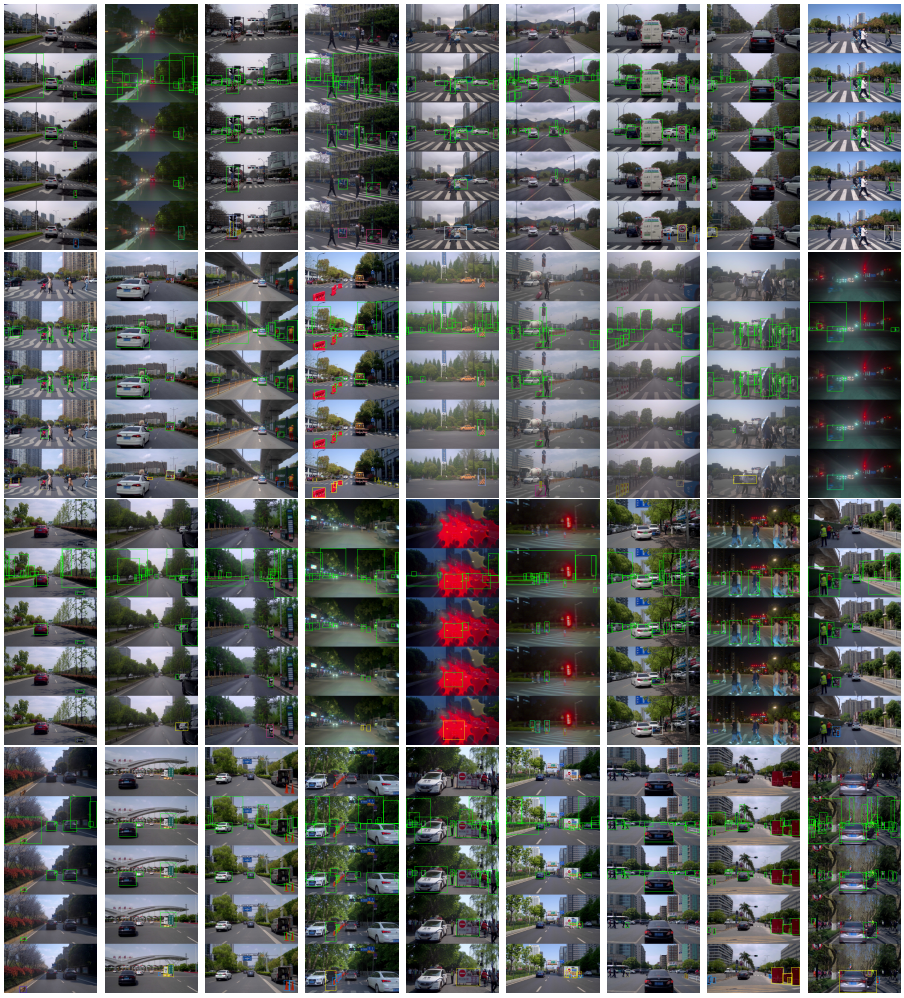

Fig. 1. Examples of COPG step-by-step proposals and final annotations. In each stack, the images from top to bottom correspond to the steps indicated in Fig. 5 of the main paper: **(b)** camera image, **(e)** initial proposal, **(f)** intermediate proposal, and **(g)** final proposal. The last row is the result after manual labeling. Best viewed with color and zoom in.

References

1. Cai, Z., Vasconcelos, N.: Cascade R-CNN: delving into high quality object detection. In: CVPR (2018) 4, 5, 6
2. 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) 1
3. 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) 2
4. 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, 4, 6
5. Han, J., Liang, X., Xu, H., Chen, K., Hong, L., Ye, C., Zhang, W., Li, Z., Liang, X., Xu, C.: Soda10m: Towards large-scale object detection benchmark for autonomous driving. arXiv:2106.11118 (2021) 5
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 2
7. Joseph, K., Khan, S., Khan, F.S., Balasubramanian, V.N.: Towards open world object detection. In: CVPR (2021) 2, 4, 5, 6
8. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017) 4, 5, 6
9. 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) 2, 5, 6
10. 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) 4, 5, 6
11. 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
12. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NeurIPS (2015) 4, 5, 6
13. 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) 4, 5, 6
14. 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) 4, 5, 6
15. 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) 4, 5, 6
16. 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

17. Zhou, X., Lin, Y., Zhang, W., Zhang, T.: Sparse invariant risk minimization. In: ICML (2022) 2
18. Zhou, X., Zhang, W., Xu, H., Zhang, T.: Effective sparsification of neural networks with global sparsity constraint. In: CVPR (2021) 2
19. 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) 4, 5, 6