Open Peer Review on Qeios

RESEARCH ARTICLE

OpenAD: Open-World Autonomous Driving Benchmark for 3D Object Detection

Zhongyu Xia[1], Jishuo Li[1], Zhiwei Lin[1], Xinhao Wang[1], Yongtao Wang[1], Ming-Hsuan Yang[2]

1 Wangxuan Institute of Computer Technology, Peking University, China
2 University of California, Merced, United States

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.

Abstract

Open-world autonomous driving encompasses domain generalization and open-vocabulary. Domain generalization refers to the capabilities of autonomous driving systems across different scenarios and sensor parameter configurations. Open vocabulary pertains to the ability to recognize various semantic categories not encountered during training. In this paper, we introduce OpenAD, the first real-world open-world autonomous driving benchmark for 3D object detection. OpenAD is built on a corner case discovery and annotation pipeline integrating with a multimodal large language model (MLLM). The proposed pipeline annotates corner case objects in a unified format for five autonomous driving perception datasets with 2000 scenarios. In addition, we devise evaluation methodologies and evaluate various 2D and 3D open-world and specialized models. Moreover, we propose a vision-centric 3D open-world object detection baseline and further introduce an ensemble method by fusing general and specialized models to address the issue of lower precision in existing open-world methods for the OpenAD benchmark. Data, toolkit codes, and evaluation codes are released at https://github.com/VDIGPKU/OpenAD.

Corresponding author: Zhongyu Xia, xiazhongyu@pku.edu.cn

1. Introduction

Table 1. Open-world autonomous driving datasets or benchmarks. "*" means rough estimates. OpenAD is the first real-world open-world benchmark for autonomous driving 3D perception. Compared to other real-world datasets, OpenAD boasts a richer variety of categories and more instances.

Datasets	Sensors	Real	Temporal	Scenes	Classes	Instances	GroundTruth
GTACrash[1]	Cam.	✗	✔	7,720	1	24.0K*	Bbox(2D)
StreetHazards[2]	Cam.	✗	✔	1,500	1	1.5K*	Sem. mask(2D)
Synthetic Fire Hydrants[3]	Cam.	✗	✗	30,000	1	30.0K*	Bbox(2D)
Synthetic Crosswalks[3]	Cam.	✗	✗	20,000	1	20.0K*	Bbox(2D)
CARLA-WildLife[4]	Cam. Depth	✗	✔	26	18	65*	Inst. mask(2D)
MUAD[5]	Cam. Depth	✗	✗	4,641	9	30.0K	Sem. mask(2D)
AnoVox[6]	Cam. Lidar	✗	✔	1,368	35	1.4K	Inst.mask(2D,3D)
YouTubeCrash[1]	Cam.	✔	✔	2,400	1	12.0K*	Bbox(2D)
RoadAnomaly21[7]	Cam.	✔	✗	110	1	0.1K*	Sem. mask(2D)
Street Obstacle Sequences[4]	Cam. Depth	✔	✔	20	13	30*	Inst. mask(2D)
Vistas-NP[8]	Cam.	✔	✗	11,167	4	11.2K*	Sem. mask(2D)
Lost and Found[9]	Cam.	✔	✔	112	42	0.2K*	Sem. mask(2D)
Fishyscapes[10]	Cam.	✔	✗	375	1	0.5K*	Sem. mask(2D)
RoadObstacle21[7]	Cam.	✔	✔	412	1	1.5K*	Sem. mask(2D)
BDD-Anomaly[2]	Cam.	✔	✗	810	3	4.5K	Sem. mask(2D)
CODA[11]	Cam. Lidar	✔	✔	1,500	34	5.9K	Bbox(2D)
OpenAD (ours)	Cam. Lidar	✔	✔	2,000	206	19.8K	Bbox(2D,3D)

With the rapid development of autonomous driving systems, open-world perception has garnered significant and growing attention from the research community. Open-world perception endeavors to develop a model that exhibits robust performance across novel domains, diverse sensor configurations, and various corner case objects. The two most pivotal factors in open-world perception are domain generalization and open-vocabulary.

Domain generalization refers to the performance of a model when confronted with new scenarios outside the training domain. It is a crucial issue that must be addressed to achieve Level 4 autonomous driving. Within autonomous driving 3D perception, the current methodologies[12][13] for evaluating scenario generalization entail training on a specific dataset and then transferring the trained model to a distinct dataset for subsequent testing.

Open-vocabulary denotes the recognition capability of perception models toward semantic categories that are not present or unlabeled within the training domain. Open-vocabulary perception serves as the foundation for subsequent inference and planning in autonomous driving systems. For instance, determining whether an object is collidable, whether it might suddenly move, or whether it signifies that certain surrounding areas are not traversable, necessitates an accurate semantic description of the object in the first place.

Many works are proposed to address these two issues. However, researchers meet three challenges when developing open-world perception models. The first challenge in 3D open-world perception for autonomous driving lies in the scarcity of evaluation benchmarks. Specifically, a unified benchmark for domain transfer evaluation is currently absent, and due to

the varying formats of individual datasets, researchers must expend considerable effort on the engineering aspect of format alignment. Besides, the current 3D perception datasets possess a limited number of semantic categories, lacking effective evaluation for current open-vocabulary 3D perception models.

The second challenge is the difficulty in training open-world perception models due to the limited scales of publicly available 3D perception datasets. Though some open-world natural language models and 2D perception models have recently leveraged large-scale Internet data for training. How to transfer these models' capabilities or 2D data to 3D open-world perception is an important and timely research problem.

The last challenge is the relatively low precision of existing open-world perception models. While specialized models trained on autonomous driving perception datasets lack the capability to generalize to the open world, they exhibit stronger predictive power for seen categories and achieve good performance. This indicates that, as the specialized models, the low precision of open-world perception models limits their real-world application. Consequently, current open-world perception models cannot yet replace specialized models in practice.

To address the aforementioned challenges, we propose OpenAD, an Open-World Autonomous Driving Benchmark for 3D Object Detection. We align the format of five existing autonomous driving perception datasets, select 2,000 scenes, annotate thousands of corner case objects with MLLMs, and develop open-world evaluation metrics to overcome the first challenge of scarcity of evaluation benchmarks. Then, we introduce a vision-centric 3D open-world object detection baseline by utilizing existing 2D open-world perception models to resolve the second challenge. Finally, we further design a fusion method to address the last challenge by leveraging the strengths of open-world perception models (or general models) and specialized models to improve the 3D open-world perception results.

The main contributions of this work are:

- We propose an open-world benchmark that simultaneously evaluates object detectors' domain generalization and open-vocabulary capabilities. To our knowledge, this is the first real-world autonomous driving benchmark for 3D open-world object detection.
- We design a labeling pipeline integrated with MLLM, which is utilized to automatically identify corner case scenarios and provide semantic annotations for abnormal objects.
- We propose a baseline method for 3D open-world perception by combining 2D open-world models. Besides, we analyze the strengths and weaknesses of open-world and specialized models, and further introduce a fusion approach to leverage both advantages.

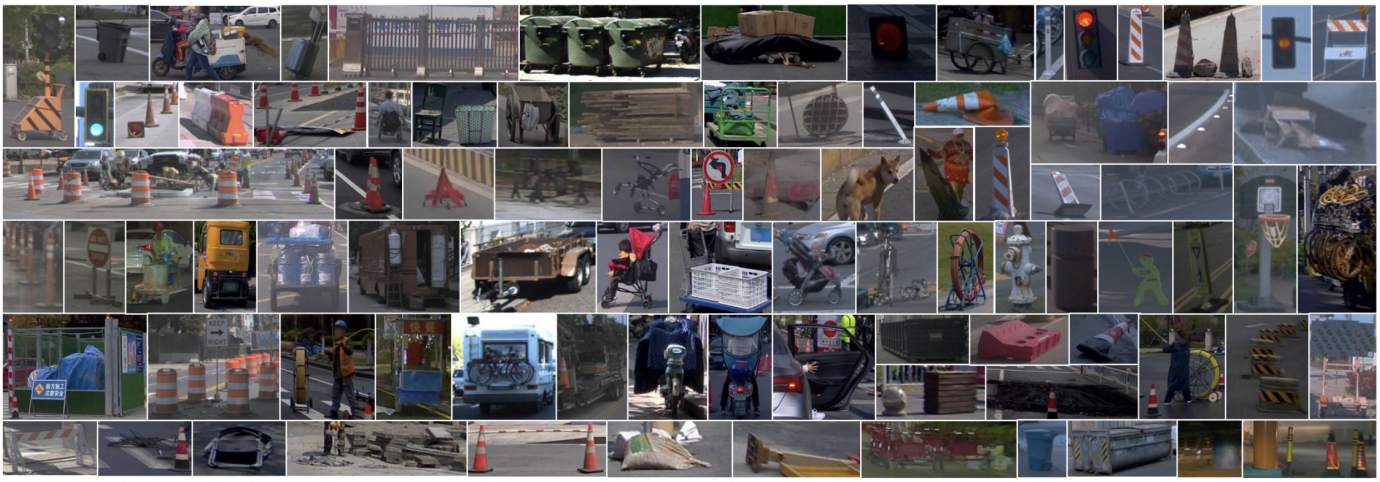

Figure 1. Examples of corner case objects in OpenAD. These object categories have not been encountered by models trained on common 3D perception datasets during their training phase.

2. Related Work

2.1. Benchmark for Open-world Object Detection

2D Benchmark.

Various datasets[14][15][16][17][18] has been used for 2D open-vocabulary object detection evaluation. The most commonly used one is LVIS dataset[15], which contains 1,203 categories.

In the autonomous driving area, as shown in Table 1, many datasets[2][3][4][5][7][8][9][10][2][11] has been proposed too. Among them, CODA[11] is a road corner case dataset for 2D object detection in autonomous driving with 1,500 road driving scenes containing bounding box annotations for 34 categories. However, some datasets only provide semantic segmentation annotations without specific instances or annotate objects as abnormal but lack semantic tags. Moreover, datasets collected from real-world driving data are on a small scale, while synthetic data from simulation platforms such as CARLA[19] lacks realism, making it difficult to conduct effective evaluations. In contrast, our OpenAD offers large-scale 2D and 3D bounding box annotations from real-world data for a more comprehensive open-world object detection evaluation.

3D Benchmark.

The 3D open-world benchmarks can be divided into two categories: indoor and outdoor scenarios. For indoor scenarios, SUN-RGBD[20] and ScanNet[21] are two real-world datasets often used for open-world evaluation, containing about 700 and 21 categories, respectively. For outdoor or autonomous driving scenarios, AnoVox[6] is a synthetic dataset containing instance masks of 35 categories for open-world evaluation. However, due to limited simulation assets, the quality and instance diversity of the synthetic data are inferior to real-world data. In addition to AnoVox, existing real-data 3D object detection datasets for autonomous driving[22][23][24][25][18] only contain a few object categories, which can hardly be used to evaluate open-world models. To address this issue, we propose OpenAD, which is constructed from real-world data and

contains 206 different corner-case object categories that appeared in autonomous driving scenarios.

2.2. 2D Open-world Object Detection Methods

To address the out-of-distribution (OOD) or anomaly detection, earlier approaches[26] typically employed decision boundary, clustering, and so forth, to discover OOD objects. Recently methods[27][28][29][30][31][32][33][17][34][35][36][37][38] employ text encoders, i.e. CLIP[39], to align text features of corresponding category labels with the box features. Specifically, OVR-CNN[33] aligns the image features with caption embeddings. GLIP[17] unifies object detection and phrase grounding for pre-training. OWL-ViT v2[40] uses a pretrained detector to generate pseudo labels on image-text pairs to scale up detection data for self-training. YOLO-World[36] adopts a YOLO-type architecture for open-vocabulary detection and achieves good efficiency. However, all these methods require predefined object categories during inference.

More recently, some open-ended methods[41][42][43] propose to utilize natural language decoders to provide language descriptions, which enables them to generate category labels from RoI features directly. More specifically, GenerateU[41] introduces a language model to generate class labels directly from regions of interest. DetClipv3[42] introduced an object captioner to generate class labels during inference and image-level descriptions for training. VL-SAM[43] introduces a training-free framework with the attention map as prompts.

2.3. 3D Open-world Object Detection Methods

In contrast to 2D open-world object detection tasks, 3D open-world object detection tasks are more challenging due to the limited training datasets and complex 3D environments. To alleviate this issue, most existing 3D open-world models bring power from pretrained 2D open-world models or utilize abundant 2D training datasets.

For instance, some indoor 3D open-world detection methods like OV-3DET[44] and INHA[45] use a pretrained 2D object detector to guide the 3D detector to find novel objects. Similarly, Coda[46] utilizes 3D box geometry priors and 2D semantic open-vocabulary priors to generate pseudo 3D box labels of novel categories. FM-OV3D[47] utilizes stable diffusion to generate data containing OOD objects. As for outdoor methods, FnP[48] uses region VLMs and a Greedy Box Seeker to generate annotations for novel classes during training. OV-Uni3DETR[49] utilizes images from other 2D datasets and 2D bounding boxes or instance masks generated by an open-vocabulary detector.

However, these existing 3D open-vocabulary detection models require predefined object categories during inference. To address this issue, we introduce a vision-centric open-ended 3D object detection method, which can directly generate unlimited category labels during inference.

3. Properties of OpenAD

3.1. Scenes and Annotation

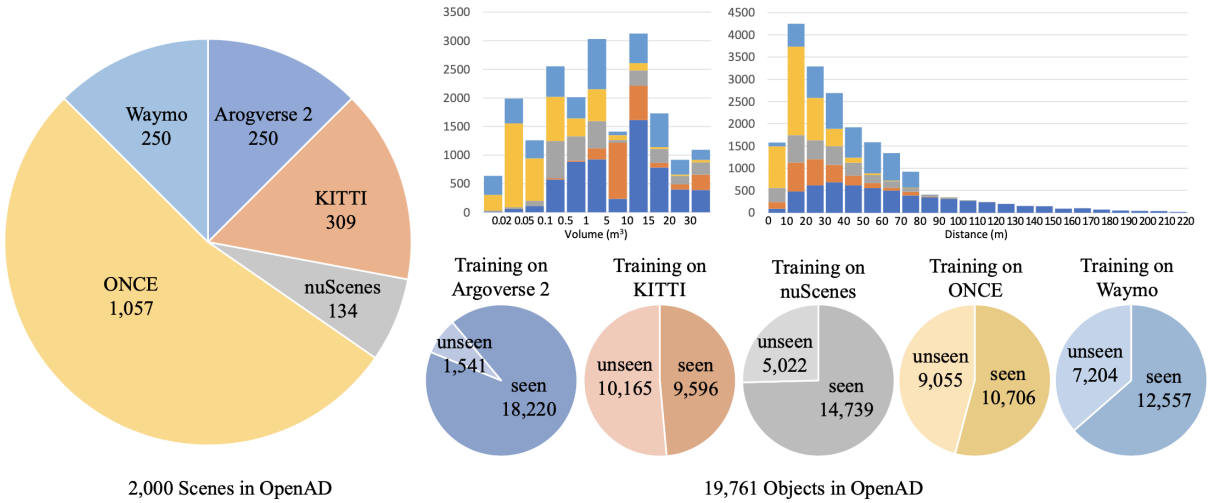

Figure 2. Data composition of OpenAD. We utilized a greater number of scenes from the ONCE dataset since the other four datasets, sampled from major cities in the United States, share certain similarities, and the Once dataset contains a richer variety of corner case scenarios. Additionally, we annotated each object with an indication of whether its category was observed in the training set of each dataset, allowing for separate evaluations of the model's specialized performance and open-vocabulary performance.

The 2,000 scenes in OpenAD are carefully selected from five large-scale autonomous driving perception datasets: Argoverse 2[24], KITTI[18], nuScenes[22], ONCE[23] and Waymo[25], as illustrated in Figure 2. These scenes are collected from different countries and regions, and have different sensor configurations. Each scene has the temporal camera and LiDAR inputs and contains at least one corner case object that the original dataset has not annotated.

For 3D bounding box labels, we annotate 6,597 corner case objects across these 2,000 scenarios, combined with the annotations of 13,164 common objects in the original dataset, resulting in 19,761 objects in total. The location and size of all objects are manually annotated using 3D and 2D bounding boxes, while their semantics categories are labeled with natural language tags, which can be divided into 206 classes. We illustrate some corner case objects in Figure 1. OpenAD encompasses both abnormal forms of common objects, such as bicycles hanging from the rear of cars, cars with doors open, and motorcycles with rain covers, as well as uncommon objects, including open manholes cover, cement blocks, and tangled wires scattered on the ground.

Concurrently, we have annotated each object with a "seen/unseen" label, indicating whether the categories of the objects have appeared in the training set of each dataset. This label is intended to facilitate the evaluation process by enabling a straightforward separation of objects that the model has encountered (seen) and those it has not (unseen), once the training dataset is specified. Moreover, we offer a toolkit code that consolidates scenes from five original datasets into a unified format, converts them into OpenAD data, and facilitates the loading and visualization process.

3.2. Evaluation Metrics

OpenAD provides evaluations for both 2D and 3D open-world object detection.

Average Precision (AP) and Average Recall (AR).

The calculation of AP and AR depends on True Positive (TP). In OpenAD, the threshold of TP incorporates both positional and semantic scores. An object prediction is considered a TP only if it simultaneously meets both the positional and semantic thresholds. For 2D object detection, in line with COCO, Intersection over Union (IoU) is used as the positional score. We use the cosine similarity of features from the CLIP model as the semantic score. When calculating AP, IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05 are used, along with semantic similarity thresholds of 0.5, 0.7, and 0.9.

For 3D object detection, the center distance is adopted as the positional score following nuScenes, and we use the same semantic score as the 2D detection task. Similar to nuScenes, we adopt a multi-threshold averaging method for AP calculation. Specifically, we compute AP across 12 thresholds, combining positional thresholds of 0.5m, 1m, 2m, and 4m with semantic similarity thresholds of 0.5, 0.7, and 0.9, and then average these AP values.

The same principle applies to calculating Average Recall (AR) for 2D and 3D object detection tasks. Both AP and AR are calculated only for the top 300 predictions.

Average Translation Error (ATE) and Average Scale Error (ASE).

Following nuScenes, we also evaluate the prediction quality of TP objects using regression metrics. The Average Translation Error (ATE) refers to the Euclidean center distance, measured in pixels for 2D or meters for 3D. The Average Scale Error (ASE) is calculated as 1 – IoU after aligning the centers and orientations of the predicted and ground truth objects.

In/Out Domain & Seen/Unseen AR.

To evaluate the model's domain generalization ability and open-vocabulary capability separately, we calculate the AR based on whether the scene is within the training domain and whether the object semantics have been seen during training. The positional thresholds for this metric are defined as above, whereas the semantic similarity thresholds are fixed at 0.9.

4. Construction of OpenAD

OpenAD is inspired by the CODA[11] dataset, which focuses on 2D corner cases in autonomous driving. However, certain objects, such as cables or nails close to the road surface, and signboards hanging on walls, cannot be detected solely by LiDAR. Therefore, unlike CODA's LiDAR-based pipeline, we propose a vision-centric semi-automated annotation pipeline, as shown in Figure 3.

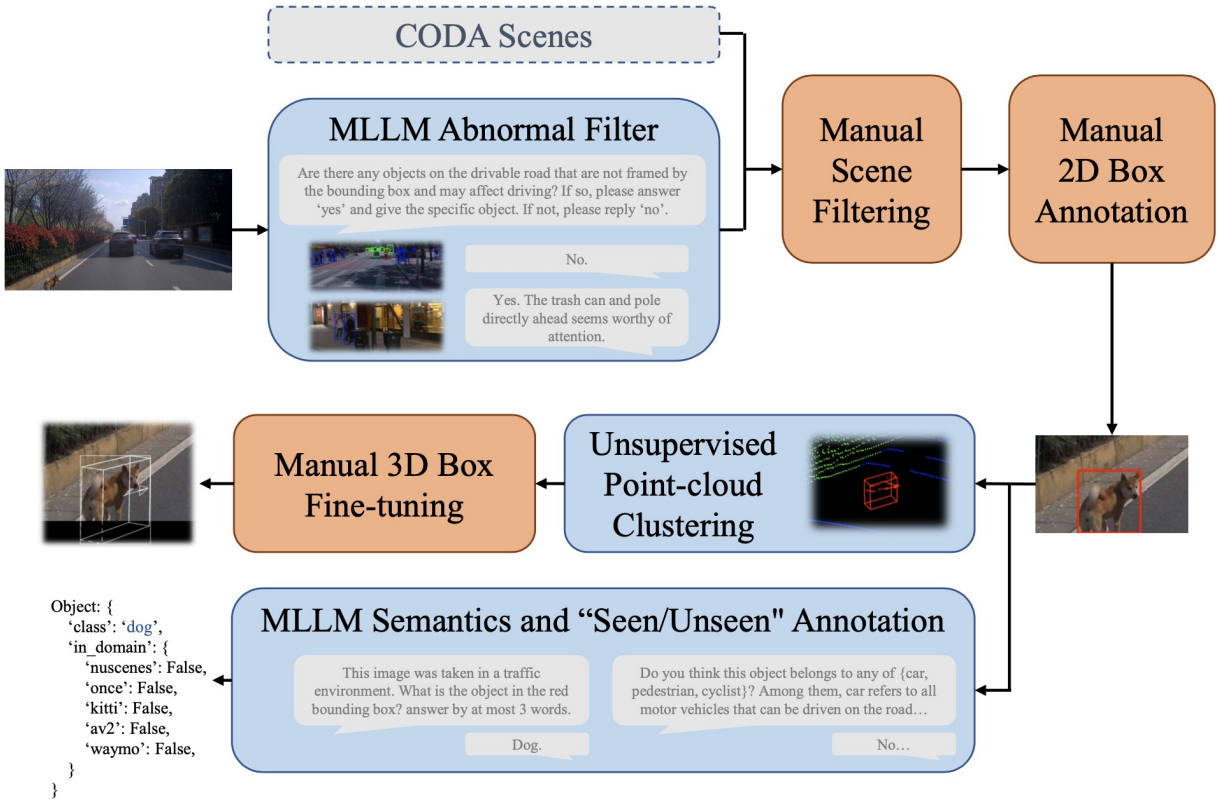

Figure 3. Annotation pipeline. OpenAD is built on a corner case discovery and annotation pipeline that integrates with a multimodal large language model (MLLM).

We use an MLLM Abnormal Filter to identify scenes containing corner cases within the validation and test sets of five autonomous driving datasets, followed by manual filtering. After that, we annotated the corner case objects with 2D bounding boxes.

For objects with relatively complete 3D geometry formed by point clouds, we adopt a methodology similar to CODA by employing point-cloud clustering algorithms[50]. We then utilize camera parameters to project 2D bounding boxes into the point cloud space and identify the corresponding clusters. Finally, the bounding boxes are manually corrected. For objects that are difficult to detect through point-cloud clustering, we manually annotate 3D bounding boxes by referencing multi-view images.

For category labels, we send images with 2D bounding boxes to an MLLM for semantic annotation and indicate for each object whether its category has been seen in each dataset. To select the best MLLM and prompts for object recognition, we manually select 30 challenging annotated image samples and evaluate the accuracy of each MLLM and prompt. We use GPT-4V[51], Claude 3 Opus[52], and InternVL 1.5[53], with InternVL exhibiting the best performance. Our experiments also reveal that closed image prompts, such as 2D bounding boxes or circles, yield the best results, whereas marking the object of inquiry on the image with arrows yields slightly inferior results. The final MLLM and prompt achieve an accuracy rate of approximately 65% on the 30 challenging samples and around 90% on the entire data. Objects like open manholes and wires falling on the road are difficult to identify for existing MLLMs.

Note that though we have utilized tools such as MLLM to automate some stages as much as possible to reduce manual workload, we have also incorporated manual verification into each stage to ensure the accuracy of annotations.

5. Baseline Methods of OpenAD

5.1. Vision-Centric 3D Open-ended Object Detection

Due to the limited scale of existing 3D perception data, it is challenging to directly train a vision-based 3D open-world perception model. We utilize existing 2D models with strong generalization capabilities to address this issue and propose a vision-centric baseline for 3D open-world perception.

As illustrated in Figure 4, an arbitrary existing 2D open-world object detection method is initially employed to obtain 2D bounding boxes and their corresponding semantic labels. Simultaneously, the image feature maps generated by the image encoder of the 2D model are cached. Subsequently, a 2D-to-3D Bbox Converter, which combines multiple features and a few trainable parameters, is introduced to transform 2D boxes into 3D boxes.

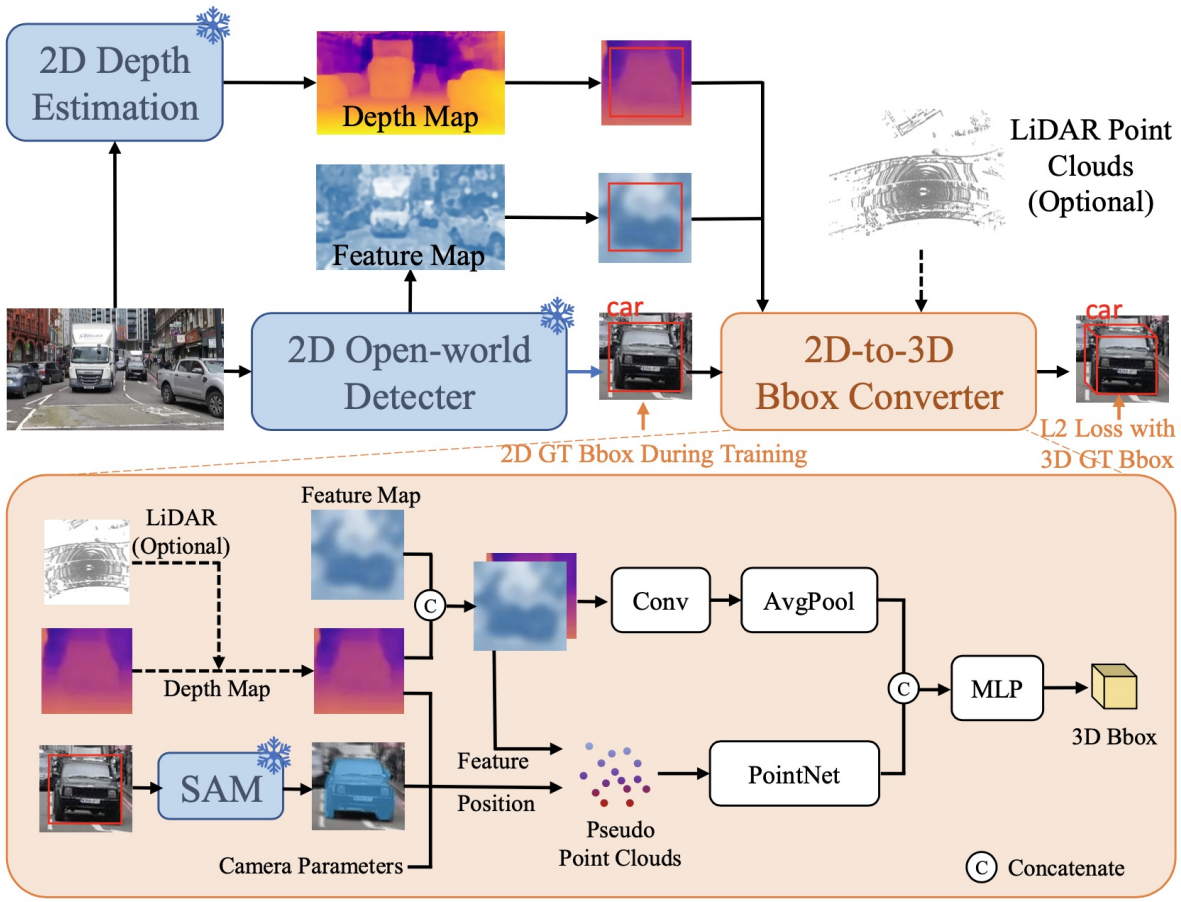

Figure 4. The 3D open-world object detection baseline we proposed. Based on the existing 2D open-world models and depth estimation models, we train a 2D-to-3D Bbox Converter. This module extracts object features through pseudo-point clouds and convolutional dual-branch architecture to predict the 3D bounding boxes of objects.

Specifically, we use existing depth estimation models, such as ZoeDepth[54], DepthAnything[55], and UniDepth[56], to obtain the depth map of the cropped image by the 2D box. We also include an optional branch that utilizes LiDAR point clouds and a linear fitting function to refine the depth map by projecting point clouds onto the image. Simultaneously, to eliminate regions within the 2D bounding box that do not belong to the foreground object, we utilize Segment Anything Model[57] (SAM) to segment the object with the 2D box as the prompt, yielding a segmentation mask. After that, we can construct pseudo point clouds for the segmentation mask with its pixel coordinates, depth map, and camera parameters. We project the pseudo point cloud onto the feature map and depth map, and features are assigned to each point through interpolation. Then, we adopt PointNet[58] to extract the feature f_p of the pseudo point clouds. Meanwhile, the depth map and feature map within the 2D bounding box are concatenated along the channel dimension, and its feature f_c is derived through convolution and global pooling. Finally, we utilize an MLP to predict the object's 3D bounding box with the concatenated features of f_p and f_c.

In this baseline, only a few parameters in the 2D-to-3D Bbox Converter are trainable. Thus, the training cost is low. In addition, during the training, each 3D object serves as a data point for this baseline, allowing for the straightforward construction of multi-domain dataset training.

5.2. General and Specialized Models Fusion

In experiments, we have found that existing open-world methods or general models are inferior to close-set methods or specialized models in handling objects belonging to common categories, but they exhibit stronger domain generalization capabilities and the ability to deal with corner cases. That is to say, existing general and specialized models complement each other. Hence, we leverage their strengths and propose a fusion baseline by combining the prediction results from the two types of models. Specifically, we align the confidence scores of the two types of models and perform non-maximum suppression (NMS) with dual thresholds, *i.e.*, IoU and semantic similarity, to filter duplicates.

6. Experiments

Table 2. Evaluation of 2D open-world methods (top), specialized methods (middle), and ensemble methods (bottom) on OpenAD benchmark. AR^{nusc} refers to scenes derived from nuScenes in OpenAD, with AR_{seen} denoting object categories observed in the nuScenes training set. For 2D open-world methods, we utilize open-source models for zero-shot inference, but for comparison purposes, classification AR against nuScenes is also presented. All specialized methods are trained on nuScenes.

Method	Backbone/Base-model	AP ↑	AR ↑	ATE ↓	ASE ↓	AR^{nusc}_{seen}	AR^{nusc}_{unseen}	AR^{others}_{seen}	AR^{others}_{unseen}
GLIP[17]	Swin-L	7.14	16.01	6.581	0.1352	1.83	1.28	2.33	1.05
VL-SAM[43]	ViT-H	8.46	17.50	6.630	0.1355	9.66	5.41	9.13	3.43
OWL-ViT v2[40]	ViT-L	9.70	21.17	6.284	0.1461	21.42	4.66	18.97	8.01
GenerateU[41]	Swin-L	9.77	21.75	6.743	0.1360	12.74	7.18	18.79	7.31
YOLO-World v2[36]	YOLOv8-X	10.20	23.46	7.489	0.1397	18.68	10.16	20.61	7.27
GroundingDino[30]	Swin-L	8.52	26.67	6.499	0.1432	20.53	4.21	21.26	7.36
MaskRCNN[59]	ResNet50	12.76	20.07	6.126	0.1359	27.77	0.00	23.41	0.07
MaskRCNN[59]	VovNetv2-99	12.32	21.09	5.746	0.1338	30.21	0.00	21.74	0.09
DETR[60]	ResNet50	12.46	20.35	6.066	0.1346	28.27	0.00	21.35	0.03
DINO[61]	ResNet50	15.24	23.41	5.679	0.1258	35.49	0.00	26.39	0.02
Co-DETR[62]	ResNet50	15.65	24.63	5.421	0.1270	38.82	0.00	27.96	0.03
Co-DETR[62]	Swin-L	16.21	27.76	5.386	0.1287	45.41	0.00	26.14	0.01
OpenAD-Ens	YOLO-world + MaskRCNN(V2-99)	13.28	29.74	6.726	0.1409	33.30	10.05	26.92	7.20
OpenAD-Ens	YOLO-world + Co-DETR(Swin-L)	16.94	34.38	6.457	0.1368	46.65	10.06	30.39	7.20

Table 3. Evaluation of 3D open-world methods (top), specialized methods (middle), and ensemble methods (bottom) on OpenAD benchmark. AR^{nusc} refers to scenes derived from nuScenes in OpenAD, with AR_{seen} denoting object categories observed in the nuScenes training set. All methods are trained on nuScenes.

Method	Modality	Backbone/Base-model	AP ↑	AR ↑	ATE ↓	ASE ↓	AR^{nusc}_{seen}	AR^{nusc}_{unseen}	AR^{others}_{seen}	AR^{others}_{unseen}
OpenAD-G	C	GenerateU	6.01	12.90	1.342	0.504	11.35	3.64	15.18	3.71
OpenAD-Y	C	YOLOWorld	6.26	13.89	1.338	0.487	14.64	7.18	18.79	3.53
FnP[48]	L	SECOND	8.85	18.97	0.848	0.493	18.49	10.82	23.42	7.47
OpenAD-G	LC	GenerateU	9.02	23.32	0.970	0.521	19.79	7.14	25.78	10.15
OpenAD-Y	LC	YOLOWorld	9.43	25.17	0.872	0.535	25.54	13.83	31.31	9.84
BEVDet[63]	C	ResNet50	9.42	15.63	1.183	0.438	36.46	0.00	14.11	0.00
BEVFormer[64]	C	ResNet50	10.08	19.36	1.125	0.440	39.38	0.00	15.85	0.00
BEVFormer[64]	C	ResNet101-DCN	14.43	22.73	0.978	0.444	51.86	0.00	16.59	0.03
BEVDepth4D[65]	C	ResNet50	12.33	20.70	1.118	0.480	39.75	0.00	17.94	0.02
BEVStereo[66]	C	ResNet50	11.12	18.27	1.133	0.431	36.73	0.00	16.21	0.00
BEVStereo[66]	C	VovNetv2-99	10.58	16.03	1.118	0.388	51.69	0.00	13.05	0.01
HENet[67]	C	Vov2-99 + R50	11.58	17.48	1.070	0.386	52.02	0.00	14.65	0.01
SparseBEV[68]	C	ResNet50	7.61	16.97	1.142	0.435	60.04	0.00	7.48	0.02
SparseBEV[68]	C	VovNetv2-99	7.64	16.93	1.103	0.431	61.36	0.00	7.09	0.01
BEVFormer v2[69]	C	ResNet50	14.64	33.13	1.064	0.554	56.63	0.00	27.16	0.08
Centerpoint[70]	L	SECOND	13.79	26.79	0.667	0.499	44.23	0.00	11.42	0.04
TransFusion-L[71]	L	SECOND	14.64	34.02	0.653	0.655	52.18	0.00	24.02	0.00
BEVFusion[72]	LC	SECOND + Dual-Swin-T	15.57	33.50	0.730	0.449	59.93	0.00	20.64	0.00
OpenAD-Ens	C	OpenAD-Y + HENet	12.36	24.32	1.176	0.420	54.16	7.18	23.37	3.53
OpenAD-Ens	LC	FnP + BEVFusion	16.19	42.08	0.776	0.458	61.74	10.82	28.40	7.47
OpenAD-Ens	LC	OpenAD-Y + BEVFusion	16.34	44.16	0.792	0.469	62.14	13.83	35.41	9.84

6.1. Evaluation Details

For specialized models that can only predict common categories, we directly match their prediction results with the corresponding categories and sort them according to their confidence scores.

For 2D open-vocabulary methods, which need a predefined object category list from users as additional inputs to detect corresponding objects, we take the union of the categories from five datasets and incorporate two additional open-vocabulary queries, *i.e.*, “object that affects traffic” and “others”, into it. We adopt OWLv2-CLIP-L/14-ST+FT, YOLO-Worldv2-XL, GLIP-L, and GroundingDINO-B for OWL-ViT v2[40], YOLO-World v2[36], GLIP[17], and GroundingDino[30], respectively.

2D open-ended methods can directly provide bounding boxes and corresponding natural language descriptions, enabling direct evaluation for OpenAD. We employ the “vg-grit5m” version for GenerateU[41].

For 3D Open-vocabulary methods, the original version of Find n’Propagate[48] utilizes a 2D detector trained on the full nuScenes dataset to provide pseudo-labels. For a fair comparison, we employ YOLO-world v2 to provide the pseudo-labels instead.

For the 3D open-ended baselines we proposed, the 2D-to-3D Bbox Converter is trained on nuScenes. We use GenerateU[41] and YOLO-World[36] as the 2D detector, Depth Anything[55] as the depth estimation model, and SAM[57] as the segmentation model. All these 2D models are frozen without any fine-tuning.

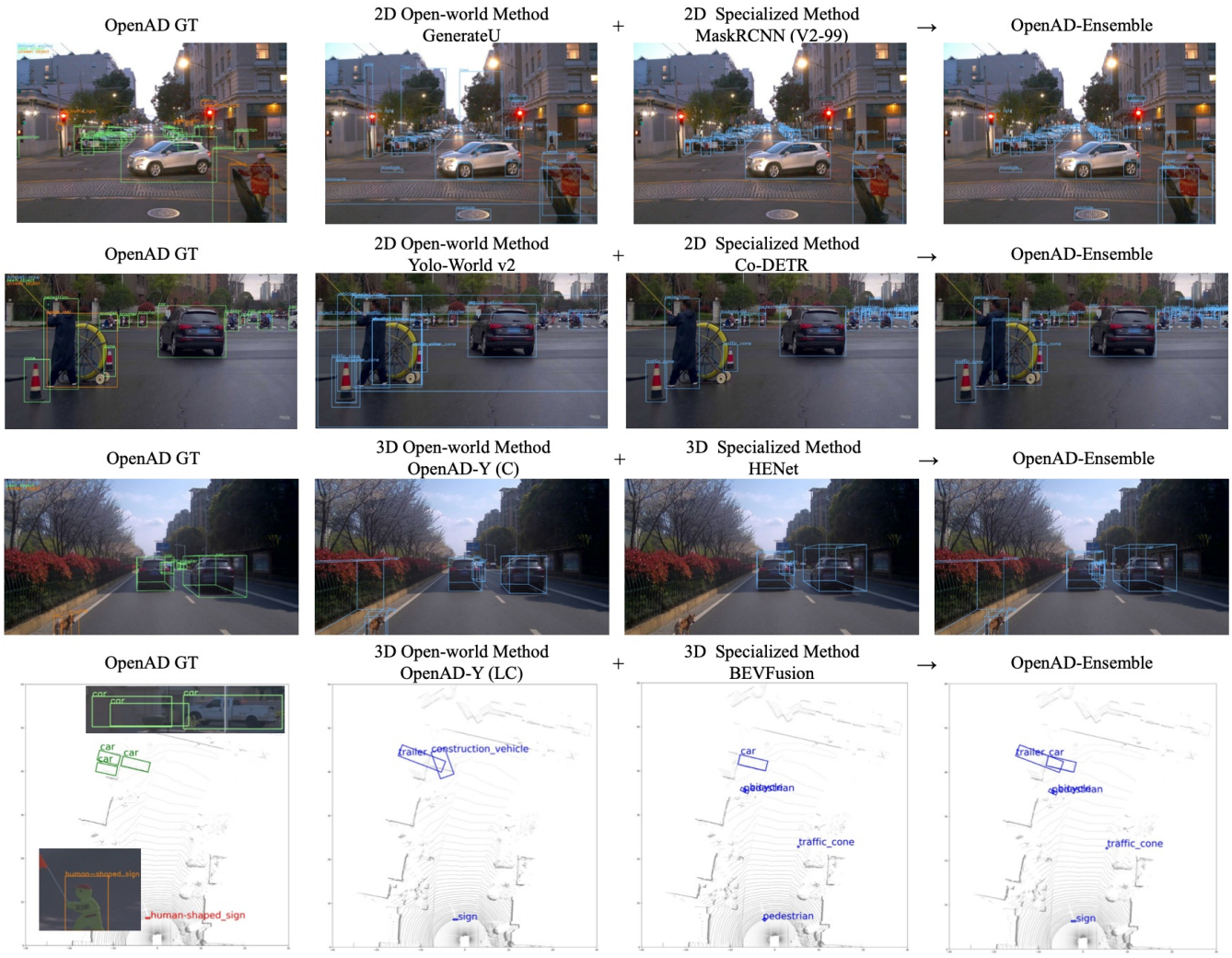

Figure 5. Example results of open-world models, specialized models, and our proposed ensemble method.

6.2. Main Results

As shown in Tables 2 and 3, we conduct evaluations on various 2D and 3D object detection models, including 2D and 3D open-world models, specialized models, and our baselines.

The results show that current open-world models, irrespective of being 2D or 3D detectors, tend to predict objects unrelated to driving (such as the sky) or to make repeated predictions for different parts of the same object, resulting in low precision and AP. Nevertheless, these models demonstrate good domain generalization and open-vocabulary capabilities, which are lacking in current specialized models. Note that our proposed ensemble baselines can effectively combine the advantages of open-world and specialized models, achieving favorable performance in both seen and unseen domains and categories. In addition, in Table 3, our proposed vision-centric baseline for 3D open-world object

detection leverages the capabilities of 2D open-world models. Specifically, by harnessing the open-world capabilities of Yolo-world v2, our method obtains 0.58 AP and 6.2 AR improvement compared to Find n' Propagate.

Moreover, we observed that the issue of overfitting is more pronounced for 3D object detection models on datasets such as nuScenes. Some models perform superior in-domain benchmarks but show worse domain generalization ability. For instance, SparseBEV, compared to methods based on Lift-Splat-Shot, achieves impressive in-domain results, with its in-domain AR even surpassing those of LiDAR-based methods. However, SparseBEV's domain generalization capability is relatively poor. Models with increased parameters by enlarging the backbone, including BEVStereo and SparseBEV, show more severe overfitting issues. These results reveal the limitations of in-domain benchmarks like nuScenes. In contrast, augmenting the parameter count through utilizing BEVFormer v2 or HENet simultaneously enhances both in-domain and out-domain Recall, indicating an inherent improvement in the methodology. Therefore, even for specialized models trained on a single domain, evaluating them on OpenAD benchmarks remains meaningful.

Furthermore, as shown in Figure 5, we provide visualization samples for some methods. Objects enclosed by orange bounding boxes belong to unseen categories in nuScenes. Recognition of these objects relies on open-world models. In contrast, specialized models exhibit significant advantages for common objects, especially for distant objects.

6.3. Ablations of Proposed Baselines

We conduct ablation studies for the proposed baselines, as shown in Table 4. We find that additional Pseudo Point Cloud inputs bring 9.9 mAR. In addition, replacing MLP with unlearnable PCA methods decreases the performance by a large margin, from 45.1 mAR to 27.3 mAR. These results show that the simple MLP can learn to complete the boundaries of objects from the datasets and predict more accurate 3D boxes.

Table 4. Ablation of 2D-to-3D Bbox Converter. This module is trained using the 2D-3D annotation pairs from the nuScenes training set and tested on the 2D-3D annotation pairs from OpenAD.

Conv	Pseudo Point Cloud	Bbox Decoding	mAR
✔	✗	MLP	36.8
✗	✔	PCA for Oriented Bounding Box	27.3
✗	✔	MLP	45.1
✔	✔	MLP	46.7

7. Conclusion

In this paper, we introduce OpenAD, the first open-world autonomous driving benchmark for 3D object detection. OpenAD is built on a corner case discovery and annotation pipeline that integrates with a multimodal large language model. The pipeline aligns five autonomous driving perception datasets in format and annotates corner case objects for 2000

scenarios. In addition, we devise evaluation methodologies and analyze the strengths and weaknesses of existing open-world perception models and autonomous driving specialized models. Moreover, addressing the challenge of training 3D open-world models, we proposed a baseline method for 3D open-world perception by combining 2D open-world models. Furthermore, we introduce a fusion baseline approach to leverage the advantages of open-world models and specialized models.

Through evaluations conducted on OpenAD, we have observed that existing open-world models are still inferior to specialized models within the in-domain context, yet they exhibit stronger domain generalization and open-vocabulary abilities. It is worth noting that the improvement of certain models on in-domain benchmarks comes at the expense of their open-world capabilities, while this is not the case for other models. This distinction cannot be revealed solely by testing on in-domain benchmarks.

We hope that OpenAD can help develop open-world perception models that surpass specialized models, whether in the same domain or across domains, and whether for semantic categories that have been seen or unseen.

References

1. [a, b] *Kim H, Lee K, Hwang G, Suh C. Crash to not crash: Learn to identify dangerous vehicles using a simulator. In: AAAI; 2019.*
2. [a, b, c, d] *Hendrycks D, Basart S, Mazeika M, Zou A, Mostajabi M, Steinhardt J, Song DX. "Scaling Out-of-Distribution Detection for Real-World Settings." In: ICML; 2022.*
3. [a, b, c] *Bu T, Zhang X, Mertz C, Dolan JM. "Carla simulated data for rare road object detection". In1EEE International Intelligent Transportation Systems Conference; 2021.*
4. [a, b, c] *Maag K, Chan R, Uhlemeyer S, Kowol K, Gottschalk H. Two video data sets for tracking and retrieval of out of distribution objects. In: ACCV; 2022.*
5. [a, b] *Franchi G, Yu X, Bursuc A, Tena A, Kazmierczak R, Dubuisson S, Aldea E, Filliat D (2022). "Muad: Multiple uncertainties for autonomous driving, a benchmark for multiple uncertainty types and tasks". arXiv preprint arXiv:2203.01437. arXiv:2203.01437.*
6. [a, b] *Bogdoll D, Hamdard I, Rößler LN, Geisler F, Bayram M, Wang F, Imhof J, de Campos M, Tabarov A, Yang Y, Gottschalk H, Zöllner JM. AnoVox: A Benchmark for Multimodal Anomaly Detection in Autonomous Driving. arXiv preprint arXiv:2405.07865. 2024.*
7. [a, b, c] *Chan R, Lis K, Uhlemeyer S, Blum H, Honari S, Siegwart R, Fua P, Salzmann M, Rottmann M (2021). "SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation". In: NeurIPS Datasets and Benchmarks Track.*
8. [a, b] *Grci{\'c} M, Bevandi{\'c} P, {\v{S}}egvi{\'c} S (2020). "Dense open-set recognition with synthetic outliers generated by real NVP". arXiv preprint arXiv:2011.11094. Available from: https://arxiv.org/abs/2011.11094.*
9. [a, b] *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.*

10. [a, b]*Blum H, Sarlin PE, Nieto JI, Siegwart RY, Cadena C (2019). "The Fishyscapes Benchmark: Measuring Blind Spots in Semantic Segmentation". IJCV.*
11. [a, b, c, d]*Li K, Chen K, Wang H, Hong L, Ye C, Han J, et al. Coda: A real-world road corner case dataset for object detection in autonomous driving. In: ECCV; 2022.*
12. [^]*Jiang K, Huang J, Xie W, Lei J, Li Y, Shao L, Lu S. "Da-bev: Unsupervised domain adaptation for bird's eye view perception." In: ECCV, 2024.*
13. [^]*Acuna D, Philion J, Fidler S (2021). "Towards optimal strategies for training self-driving perception models in simulation". In: NeurIPS.*
14. [^]*Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL. Microsoft coco: Common objects in context. In: ECCV; 2014.*
15. [a, b]*Gupta A, Dollar P, Girshick R. "Lvis: A dataset for large vocabulary instance segmentation". In:CVPR; 2019.*
16. [^]*Shao S, Li Z, Zhang T, Peng C, Yu G, Zhang X, Li J, Sun J. "Objects365: A large-scale, high-quality dataset for object detection." In: ICCV; 2019.*
17. [a, b, c, d, e]*Li LH, Zhang P, Zhang H, Yang J, Li C, Zhong Y, Wang L, Yuan L, Zhang L, Hwang JN, Chang KW, Gao J. "Grounded Language-Image Pre-training." In: CVPR; 2022.*
18. [a, b, c]*Geiger A, Lenz P, Urtasun R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: CVPR; 2012.*
19. [^]*Dosovitskiy A, Ros G, Codevilla F, Lopez A, Koltun V. "CARLA: An Open Urban Driving Simulator." In:Annual Conference on Robot Learning; 2017.*
20. [^]*Song S, Lichtenberg SP, Xiao J (2015). "Sun rgb-d: A rgb-d scene understanding benchmark suite". In:CVPR.*
21. [^]*Dai A, Chang AX, Savva M, Halber M, Funkhouser T, Nießner M. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR; 2017.*
22. [a, b]*Caesar H, Bankiti V, Lang AH, Vora S, Liong VE, Xu Q, Krishnan A, Pan Y, Baldan G, Beijbom O. "nuscenes: A multimodal dataset for autonomous driving." In: CVPR; 2020.*
23. [a, b]*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.*
24. [a, b]*Wilson B, Qi W, Agarwal T, Lambert J, Singh J, Khandelwal S, Pan B, Kumar R, Hartnett A, Pontes JK, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. arXiv preprint arXiv:2301.00493. 2023.*
25. [a, b]*Sun P, Kretzschmar H, Dotiwalla X, Chouard A, Patnaik V, Tsui P, Guo J, Zhou Y, Chai Y, Caine B, et al. Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR; 2020.*
26. [^]*Yang J, Zhou K, Li Y, Liu Z (2024). "Generalized out-of-distribution detection: A survey".IJCV. 2024.*
27. [^]*Kaul P, Xie W, Zisserman A. Multi-modal classifiers for open-vocabulary object detection. In:ICML; 2023.*
28. [^]*Zhou X, Girdhar R, Joulin A, Kr\u00e4henb\u00fchl P, Misra I. Detecting twenty-thousand classes using image-level supervision. In: ECCV; 2022.*

29. ^Ma C, Jiang Y, Wen X, Yuan Z, Qi X (2024). "Codet: Co-occurrence guided region-word alignment for open-vocabulary object detection". NeurIPS.
30. a, b, cLiu S, Zeng Z, Ren T, Li F, Zhang H, Yang J, Li C, Yang J, Su H, Zhu J, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499. 2023.
31. ^Wu S, Zhang W, Xu L, Jin S, Liu W, Loy CC. "Clim: Contrastive language-image mosaic for region representation". In: AAAI, 2024.
32. ^Xu Y, Zhang M, Fu C, Chen P, Yang X, Li K, Xu C (2023). "Multi-modal queried object detection in the wild". In: NeurIPS, 2023.
33. a, bZareian A, Dela Rosa K, Hu DH, Chang SF (2021). "Open-vocabulary object detection using captions". In:CVPR, 2021.
34. ^Wang Z, Li Y, Chen X, Lim SN, Torralba A, Zhao H, Wang S. Detecting everything in the open world: Towards universal object detection. In: CVPR; 2023.
35. ^Zhang H, Li F, Zou X, Liu S, Li C, Yang J, Zhang L. A simple framework for open-vocabulary segmentation and detection. In: ICCV; 2023.
36. a, b, c, d, eCheng T, Song L, Ge Y, Liu W, Wang X, Shan Y. YOLO-World: Real-Time Open-Vocabulary Object Detection. In: CVPR; 2024.
37. ^Wu S, Zhang W, Jin S, Liu W, Loy CC. "Aligning bag of regions for open-vocabulary object detection". In:CVPR, 2023.
38. ^Gu X, Lin TY, Kuo W, Cui Y (2021). "Open-vocabulary object detection via vision and language knowledge distillation". arXiv preprint arXiv:2104.13921.
39. ^Radford A, Kim JW, 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.
40. a, b, cMinderer M, Gritsenko A, Houlsby N (2023). "Scaling Open-Vocabulary Object Detection". In:NeurIPS, 2023.
41. a, b, c, d, eLin C, Yi J, Qu L, Yuan Z, Cai J. Generative region-language pretraining for open-ended object detection. In: CVPR; 2024.
42. a, bYao L, Pi R, Han J, Liang X, Xu H, Zhang W, Li Z, Xu D. DetCLIPv3: Towards Versatile Generative Open-Vocabulary Object Detection. In: CVPR; 2024.
43. a, b, cLin Z, Wang Y, Tang Z (2024). "Training-Free Open-Ended Object Detection and Segmentation via Attention as Prompts". In: NeurIPS, 2024.
44. ^Lu Y, Xu C, Wei X, Xie X, Tomizuka M, Keutzer K, Zhang S. "Open-Vocabulary Point-Cloud Object Detection without 3D Annotation." In: CVPR; 2023.
45. ^Jiao P, Zhao N, Chen J, Jiang YG (2024). "Unlocking Textual and Visual Wisdom: Open-Vocabulary 3D Object Detection Enhanced by Comprehensive Guidance from Text and Image". arXiv preprint arXiv:2407.05256.
46. ^Cao Y, Zeng Y, Xu H, Xu D. "CoDA: Collaborative Novel Box Discovery and Cross-modal Alignment for Open-vocabulary 3D Object Detection". In: NeurIPS; 2023.

47. ^Zhang D, Li C, Zhang R, Xie S, Xue W, Xie X, Zhang S (2023). "FM-OV3D: Foundation Model-based Cross-modal Knowledge Blending for Open-Vocabulary 3D Detection". In: AAAI.
48. a, b, cEtchegaray D, Huang Z, Harada T, Luo Y. Find n' Propagate: Open-Vocabulary 3D Object Detection in Urban Environments. In: CVPR; 2024.
49. ^Wang Z, Li Y, Liu T, Zhao H, Wang S. "OV-Uni3DETR: Towards Unified Open-Vocabulary 3D Object Detection via Cycle-Modality Propagation". In: ECCV, 2024.
50. ^Bogoslavskyi I, Stachniss C (2016). "Fast range image-based segmentation of sparse 3D laser scans for online operation". In: IROS.
51. ^**OpenAI** (2023). "GPT-4V(vision) system card". Available from: https://cdn.openai.com/papers/GPTV_System_Card.pdf.
52. ^Anthropic (2024). "Introducing the next generation of Claude."www.anthropic.com/news/claude-3-family.
53. ^Chen Z, Wu J, Wang W, Su W, Chen G, Xing S, Zhong M, Zhang Q, Zhu X, Lu L, Li B, Luo P, Lu T, Qiao Y, Dai J (2023). "InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks". arXiv preprint arXiv:2312.14238. Available from:https://arxiv.org/abs/2312.14238.
54. ^Bhat SF, Birkl R, Wofk D, Wonka P, Müller M (2023). "Zoedepth: Zero-shot transfer by combining relative and metric depth". arXiv preprint arXiv:2302.12288.
55. a, bYang L, Kang B, Huang Z, Xu X, Feng J, Zhao H. "Depth anything: Unleashing the power of large-scale unlabeled data." In: CVPR; 2024.
56. ^Piccinelli L, Yang YH, Sakaridis C, Segu M, Li S, Van Gool L, Yu F (2024). "UniDepth: Universal Monocular Metric Depth Estimation". In: CVPR.
57. a, bKirillov A, Mintun E, Ravi N, Mao H, Rolland C, Gustafson L, Xiao T, Whitehead S, Berg AC, Lo WY, et al. Segment anything. In: ICCV; 2023.
58. ^Qi CR, Su H, Mo K, Guibas LJ. "Pointnet: Deep learning on point sets for 3d classification and segmentation." In: CVPR; 2017.
59. a, bHe K, Gkioxari G, Dollár P, Girshick R. "Mask r-cnn". In:ICCV; 2017.
60. ^Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. In: ECCV; 2020.
61. ^Caron M, Touvron H, Misra I, Jégou H, Mairal J, Bojanowski P, Joulin A. Emerging properties in self-supervised vision transformers. In: ICCV; 2021.
62. a, bZong Z, Song G, Liu Y (2023). "Detrs with collaborative hybrid assignments training". In:ICCV.
63. ^Huang J, Huang G, Zhu Z, Du D (2021). "BEVDet: High-performance Multi-camera 3D Object Detection in Bird-Eye-View". arXiv preprint arXiv:2112.11790.
64. a, bLi Z, Wang W, Li H, Xie E, Sima C, Lu T, Qiao Y, Dai J (2022). "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers". In: ECCV.
65. ^Huang J, Huang G (2022). "Bevdet4d: Exploit temporal cues in multi-camera 3d object detection".arXiv preprint

arXiv:2203.17054. Available from: https://arxiv.org/abs/2203.17054.

66. [a, b] *Li Y, Bao H, Ge Z, Yang J, Sun J, Li Z (2023). "Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo". In: AAAI.*
67. [^] *Xia Z, Lin Z, Wang X, Wang Y, Xing Y, Qi S, Dong N, Yang M-H. "Henet: Hybrid encoding for end-to-end multi-task 3d perception from multi-view cameras". In: ECCV, 2024.*
68. [a, b] *Liu H, Teng Y, Lu T, Wang H, Wang L. "SparseBEV: High-Performance Sparse 3D Object Detection from Multi-Camera Videos." In: ICCV; 2023.*
69. [^] *Yang C, Chen Y, Tian H, Tao C, Zhu X, Zhang Z, Huang G, Li H, Qiao Y, Lu L, et al. BEVFormer v2: Adapting Modern Image Backbones to Bird's-Eye-View Recognition via Perspective Supervision. In: CVPR; 2023.*
70. [^] *Yin T, Zhou X, Krahenbuhl P. "Center-based 3d object detection and tracking". In:CVPR, 2021.*
71. [^] *Bai X, Hu Z, Zhu X, Huang Q, Chen Y, Fu H, Tai C. "Transfusion: Robust lidar-camera fusion for 3d object detection with transformers." In: CVPR; 2022.*
72. [^] *Liang T, Xie H, Yu K, Xia Z, Lin Z, Wang Y, Tang T, Wang B, Tang Z (2022). "BEVFusion: A Simple and Robust LiDAR-Camera Fusion Framework". In: NeurIPS, 2022.*