Towards Robust Robot 3D Perception in Urban Environments: The UT Campus Object Dataset

Arthur Zhang, Chaitanya Eranki*, Christina Zhang*, Ji-Hwan Park*, Raymond Hong*, Pranav Kalyani*, Lochana Kalyanaraman*, Arsh Gamare*, Arnav Bagad, Maria Esteva, Joydeep Biswas

Abstract—**We introduce the UT Campus Object Dataset (CODa), a mobile robot egocentric perception dataset collected on the University of Texas Austin Campus. Our dataset contains 8.5 hours of multimodal sensor data from 3D LiDAR, stereo RGB and RGBD cameras, and a 9-DOF IMU. CODa contains 58 minutes of ground-truth annotations containing 1.3 million 3D bounding boxes with instance IDs for 53 semantic classes, 5000 frames of 3D semantic annotations for urban terrain, and pseudo-ground truth localization. We repeatedly traverse identical geographic regions for diverse indoor and outdoor areas, weather conditions, and times of the day. Using CODa, we empirically demonstrate that: 1) 3D object detection performance improves in urban settings when trained using CODa compared to existing datasets, 2) sensor-specific fine-tuning increases 3D object detection accuracy and 3) pretraining on CODa improves cross-dataset 3D object detection performance in urban settings compared to pretraining on AV datasets. We release benchmarks for 3D object detection and 3D semantic segmentation, with future plans for additional tasks. We publicly release CODa on the Texas Data Repository [1], pre-trained models, dataset development package, and interactive dataset viewer[1]. We expect CODa to be a valuable dataset for egocentric perception and planning for navigation in urban environments.**

Index Terms—**Data Sets for Robotic Vision; Object Detection, Segmentation and Categorization; Performance Evaluation and Benchmarking; Service Robots**

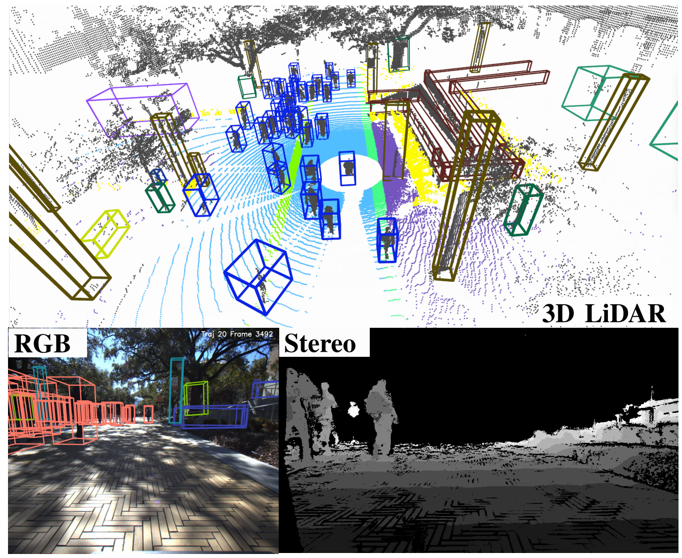


Fig. 1: Three of the five modalities available in CODa. **RGB** image with **3D object** labels (bottom left), **3D point cloud** (middle), **stereo depth** image (bottom right).

I. Introduction

Accurate and robust perception of objects and scenes is crucial for autonomous mobile robots performing tasks in urban environments. To this end, the computer vision and robotics communities have proposed datasets and benchmarks [2]–[7] to serve as training data for the development and fair evaluation of modern data-driven approaches. However, perception models trained on existing datasets do not perform well in urban environments for the following reasons: 1) they exhibit significant sensor and viewpoint differences from urban robots, 2) they focus exclusively on RGB images, 3) they lack sufficient object or terrain annotation diversity. These characteristics limit egocentric robot capabilities [8]–[10], which are important for navigation and planning tasks.

Many egocentric 3D perception datasets are collected from urban robots or autonomous vehicles (AVs). Existing urban robotics datasets [6], [7], [11] in human-centric environments possess similar sensors and viewpoints but lack semantic annotation diversity. In contrast, autonomous vehicle (AV) datasets [5], [12], [13] contain semantic annotations but are collected from cars on streets, roads, or highways. They operate higher fidelity sensor suites, encounter different geometric and semantic entities, and have different sensor viewpoints compared to urban robots. This causes perception models trained on AV datasets to perform poorly on robots in urban settings — Section VII-B presents quantitative analyses demonstrating this significant performance gap.

To address this gap, we contribute the **UT Campus Object Dataset (CODa)**, a large-scale annotated multimodal dataset for training and benchmarking egocentric 3D perception for robots in urban environments. Our dataset is comprised of 23 sequences in indoor and outdoor settings on a university campus and contains repeated traversals from different viewpoints, weather conditions (**sunny, rainy, cloudy, low-light**), and scene densities.

The sensor data includes 1) 3D point clouds from a

Manuscript received September 6, 2023; revised March 5, 2024; Accepted March 19, 2024; date of current version March 25, 2024. This paper was recommended for publication by Editor W. Burgard and B. Sven upon evaluation of the reviewers' comments.

The authors, A. Zhang, C. Eranki, C. Zhang, J. Park, R. Hong, P. Kalyani, L. Kalyanaraman, A. Gamare, A. Bagad, M. Esteva, and J. Biswas, are with the University of Texas, Austin, TX 78705, USA. {arthurz, chai77, yymzhang, raymond22, lochanak, arnavb, joydeepb}@cs.utexas.edu {jihwanpark98, pranavkalyani, arsh.gamare}@utexas.edu maria@tacc.utexas.edu

*Equal Contribution.

[1]Interactive dataset viewer available on the CODa website

128-channel 3D LiDAR, 2) RGB images from a stereo camera pair synchronized with the 3D LiDAR, 3) RGB-D images from an active depth camera, 4) RGB-D images from a passive depth camera, and 5) 9 DoF inertial measurements. The dataset includes sensor intrinsic and extrinsic calibrations for each sequence and pseudo ground truth global poses.

CODa contains **1.3 million** ground truth 3D bounding box annotations, instance IDs, and occlusion values for objects in the 3D point cloud. Furthermore, it includes **5000** frames of 3D terrain segmentation annotations for 3D point clouds. All annotations are provided by human annotators, and labeled at 10Hz for 3D bounding boxes, and 2-10Hz for terrain semantic segmentation. Compared to similar 3D perception datasets, CODa has far more class diversity, containing **53** object classes and **23** urban terrain types. This includes classes that are useful to urban navigation, such as doors, railings, stairs, emergency phones, and signs. Using our annotations, we release benchmarks using established metrics [5], [14] for 3D object detection and 3D semantic segmentation with plans for perception and planning tasks relevant to autonomous navigation.

In the rest of the manuscript, we review existing datasets and relate CODa to them (Section II), describe the sensor setup (Section III), data collection procedure (Section IV), dataset contents (Section VI), and annotation details. We characterize the semantic composition of our dataset, proposed train/validation/test splits, and provide qualitative sensor data visualizations. Finally, in Section VII we empirically analyze how: using CODa improves object detection performance for robots in urban settings, different 3D LiDAR resolutions affect pre-trained object detector performance, and pre-training on CODa outperforms AV datasets in cross-dataset object detection on JRDB [7].

II. Related Work

In this section, we review existing egocentric 3D LiDAR datasets for urban and AV domains and summarize these datasets in Table I and Table II. We limit the discussion to real-world datasets, as there still exists a significant domain gap between simulation and real-world [36], [37].

A. Urban Datasets

Urban datasets are collected in human-centric environments, such as college campuses, city streets, and shopping malls. Similar to our work, these datasets are used to benchmark robot performance in human-centric environments, often emphasizing long-term SLAM, object detection, and semantic segmentation. While there exist computer vision benchmarks for 3D object detection [3] and semantic segmentation [24], we focus on datasets collected from mobile robots due to differences in perspective shift and sensor suite.

Long-term SLAM datasets [11], [15], [21], [22], [25], [26] contain globally consistent absolute ground truth poses and multimodal sensor data. They are repeatedly collected over multiple times of day to fairly evaluate long-term SLAM methods that rely on geometric, visual, or proprioceptive sensor information.SLAM datasets like BotanicGarden [27] contain 2D semantic segmentation annotations and globally consistent poses in a static urban scene. Likewise, CODa contains 3D semantic annotations and globally consistent pseudo-ground truth poses in dynamic scenes. This uniquely positions it for evaluating semantic and long-term SLAM methods across traversals in urban environments.

SCAND [6] is another large-scale dataset with multimodal sensor data collected over multiple times of day in a campus environment. Instead of ground truth poses, it contains socially compliant navigation demonstrations and operator commands to support social navigation research. Similarly, CODa contains multimodal sensor data with repeated trials over multiple times of day, but distinguishes itself by providing object and terrain annotations to support methods that rely on semantic information.

Besides CODa, there does not exist a large-scale urban robot dataset that contains 3D object and terrain annotations. RUGD [18] and Rellis-3D [19] are robot datasets with 2D and 3D semantic segmentation annotations respectively, but are collected on off-road terrains. These environments contain distinct semantic entities from those found in urban environments. The closest work to ours is JRDB [7], a mobile robot dataset with 1) 1.8 million 3D bounding box annotations 2) indoor and outdoor sequences 3) egocentric sensor data. However, JRDB [7] is intended for pedestrian understanding research as it only contains pedestrian semantic annotations. In contrast, CODa contains object and terrain level annotations for a wide range of semantic classes to support general-purpose egocentric perception and navigation in urban environments.

B. AV Perception Datasets

Unlike urban robot datasets, AV datasets are collected from car-mounted, high-fidelity sensor suites and operate exclusively on roads, parking lots, and highways. Despite these differences, their large size and scene diversity may be leveraged to train 3D perception algorithms for urban settings.

Among AV datasets, the Oxford RobotCar dataset [33] contains the most repeated traversals over different weather, object density, and lighting conditions. It provides ground truth poses for evaluating long-term SLAM methods that only rely on visual and geometric information. For 2D multitask learning problems, Berkeley DeepDrive [32] provides semantic annotations at both the object and pixel level for a wide range of semantic classes and weather conditions.

Lyft L5 [35], CityScapes3D [31], and KITTI-360 [38] contain labeled 2D images or 3D bounding boxes with more non-overlapping semantic classes than other AV datasets. They contain vehicle-centric semantic classes to support multi-class object detection research in AV domains. Conversely, large-scale datasets like Waymo Open [12] and nuScenes [13] have fewer unique semantic classes but have more 3D semantic annotations per class and greater scene diversity. These characteristics establish them as de facto benchmarks for 3D object detection and semantic segmentation tasks, while also being valuable for pre-training 3D object detectors to recognize similar objects across domains.

Other works like Argoverse2 [28] and ONCE [29] support self-supervised point cloud learning by providing more unan-

Dataset	Pose	#Cls	#3D Bbx	#3D/2D Seg	Inst. Labels	#3D Ann. Frames	3D Frames	3D Pts/ Frame	2D Frames	In/Out	Time of Day	Night/Rain
MIT Stata[15]	S	0	0	0	Y	0	∼5.1M	1.4K	∼5.1M	I	N/A	N/N
TUM RGB-D[16]	MC	0	0	0	N	0	47K	0	47K	I	N/A	N/N
Newer College[17]	S	0	0	0	N	0	23K	131K	23K	I	M, A	N/N
JRDB[7]	None	1	**1.8M***	0	Y	28K	28K	130K	28K	I+O	M, A	N/N
SCAND[6]	None	0	0	0	N	0	313K	15K†	626K	I+O	M, A	N/N
RUGD[18]	None	24	0	7.4K$_{2D}$	N	0	0	0	37K	O	M, A	N/N
Rellis-3D[19]	S+G	20	0	13K	N	13K	13K	**161K**	6K	O	M, A	N/N
NCLT[11]	S+G+R	0	0	0	N	0	1.2M	69.5K†	628K	I+O	M, A, E	Y/N
ALITA[20]	S	0	0	0	N	0	**7.2M**	15K†	**7.2M**	O	M, A	N/N
FusionPortable[21]	S+MC	0	0	0	N	0	1.4M	131K	2.9M	I+O	M, A	N/N
OpenLORIS[22]	MC	**40**	0	0	N	0	497K	N/A	497K	I+O	M, A	N/N
Pascal VOC3D+[23]	None	12	36K	0	Y	30K	0	N/A	22K	I+O	M, A, E	N/N
NYU Depthv2[24]	None	26	0	1.45K	Y	1.45K	407K	N/A	407K	I	M, A, E	N/N
M2DGR[25]	S+R+MC	0	0	0	N	0	107K	34K	160K	I+O	M, A, E	Y/N
S3E[26]	S+R+MC	0	0	0	N	0	55K	30K	55K	I+O	M, A	N/N
BotanicGarden[27]	S	0	0	1.1K$_{2D}$	N	0	288K	54K	2.3M	O	M, A	N/N
CODa (Ours)	**S+G**	**53**	**1.3M**	**6K**	**Y**	**32K**	**324K**	**131K**	**324K**	**I+O**	**M, A, E**	**Y/Y**

TABLE I: Comparison between CODa (ours) and similar campus scale robot datasets. The most significant entry for each column in CODa is highlighted in blue. CODa provides the largest number of object classes, 3D bounding box annotations, and annotated 3D frames under the widest range of environmental and weather conditions. Pose annotations: G - GPS, R - GPS-RTK, S - SLAM, MC - Motion Capture. Indoor/Outdoor: I - Indoor, O - Outdoor. Time of Day: M - Morning, A - Afternoon, E - Evening. * only contains pedestrian annotations. † estimated from the manufacturer datasheet.

Dataset	Pose	#Cls	#3D/2D Bbx	#3D/2D Seg	Inst. Labels	#3D Ann. Frames	3D Frames	3D Pts/ Frame	2D Frames	Time of Day	Night/Rain
KITTI[5]	G+R	3	80K	43K	Y	15K	15K	120K	13K	M, A	N/N
nuScenes[13]	G+R	23	1.4M	40K	Y	40K	400K	**34K**	1.4M	M, A, E	Y/Y
Argoverse2[28]	G+R	30	12M	0	Y	150K	6M	107K	300K	M, A, E	Y/Y
Waymo Open[12]	G+R	4	12M	230K	Y	192K	192K	**169K**	1M	M, A, E	Y/Y
ONCE[29]	G	5	417K	0	Y	21K	1M	70	7M	M, A, E	Y/Y
KITTI-360[30]	G+R	14	68K	156K	Y	100K	100K	200K	150K	Not Given	Not Given
CityScapes3D[31]	G+R	8	Not Given	20K	Y	20K	0	N/A	25K	M, A, E	Y/Y
BDD100K[32]	G+R	10	1.8M$_{2D}$	10K$_{2D}$	Y	0	0	N/A	120M	M, A, E	Y/Y
Oxford RobotCar[33]	G+R	0	0	0	N	0	0	N/A	20M	M, A, E	Y/Y
ApolloCar3D[34]	G	6	60K	120K$_{2D}$	Y	0	Not Given	N/A	5.27K	M, A, E	Y/Y
Lyft L5[35]	G+R	9	15K	15K	Y	Not Given	Not Given	Not Given	323K	M, A	N/N
CODa (Ours)	S+G	53	1.1M	6K	Y	32K	324K	**131K**	131K	M, A, E	Y/Y

TABLE II: Comparing dataset statistics between CODa (ours) and existing AV datasets. We use the following abbreviations: M - Morning, A - Afternoon, E - Evening, G - GPS, R - RTK, S - SLAM. We bold the number of points per frame for the nuScenes [13], Waymo [12], and CODa datasets to highlight the difference in point cloud density.

notated 3D point clouds than any other AV dataset. Additionally, both contain 3D object labels, but Argoverse2's [28] labels are limited to five meters within the drivable area and ONCE [29] is limited to five object classes. For robots operating in urban environments, it is important to identify a diverse set of objects in non-drivable areas, reinforcing the need for a dataset like CODa.

III. Sensor Setup

CODa was collected using a Clearpath Husky robot [39] equipped with a custom sensor suite with the following sensors, illustrated in Fig. 2:

- 1 × Ouster OS1-128 3D LIDAR, 128 beams - 0.35° vertical angular resolution, 2048 beams - 0.17° horizontal angular resolution, up to 2.6 million points/second, field of view: 360° horizontal, 45° vertical, range: 128 m. Point clouds captured in 128x1024 channels @ 10 Hz.
- 2 × Teledyne FLIR Blackfly S RGB cameras (BFS-U3-51S5C-C) up to 5 Megapixels, 75 Hz, global shutter. Paired with KOWA F2.8/5mm lenses. Field of view (H x W): 70°x79°. RGB images captured in 1.25 Megapixels @ 10 Hz, hardware synchronized with 3D LIDAR.

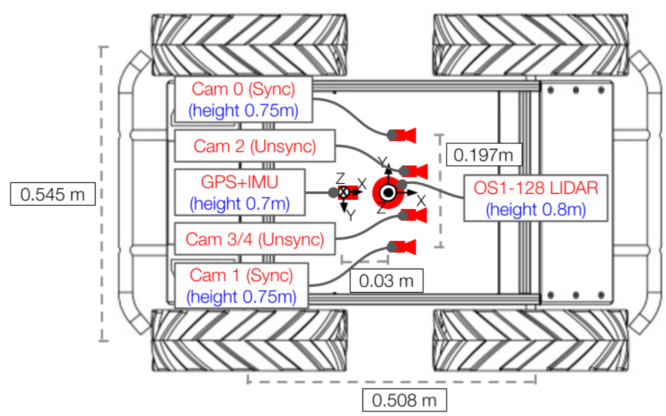


Fig. 2: **Sensor setup** including mounting positions. All heights are relative to the ground plane.

- 1 × Microsoft Azure Kinect active RGBD camera up to 12 and 1 MP (RGB and Depth) @ 15 Hz, rolling shutter. 7 microphone circular array. RGB images captured in 2.0 MP @ 5Hz
- 1 × Stereolabs ZED 2i passive stereo camera up to 4

Megapixels @ 15 Hz, rolling shutter. RGB and depth images captured in 0.5MP @ 5Hz
- 1 × Vectornav VN-310 Dual GNSS/INS, up to 800 Hz IMU Data. Inertial and GPS data captured @ 20Hz

An onboard computer with an Intel I7-8700 3.2GHz CPU and 32 GB RAM is securely mounted inside the robot and records all sensor streams to a high-speed Intel 760P SS-DPEKKW512G8 512GB SSD. A GPU-equipped laptop is mounted on the robot to process the Azure Kinect and ZED 2i camera data before transmitting both RGBD streams to the computer via 10 Gig Ethernet. The coordinate system definitions are described in the CODa documentation [1].

The Ouster LIDAR and FLIR cameras are synchronized by hardware using the Ouster LIDAR 10Hz sync pulse to trigger the FLIR cameras. This ensures that the start of the LIDAR scan is synchronized with the start of the exposure of the FLIR cameras. All other sensors have timestamps, but their capture times are not synchronized.

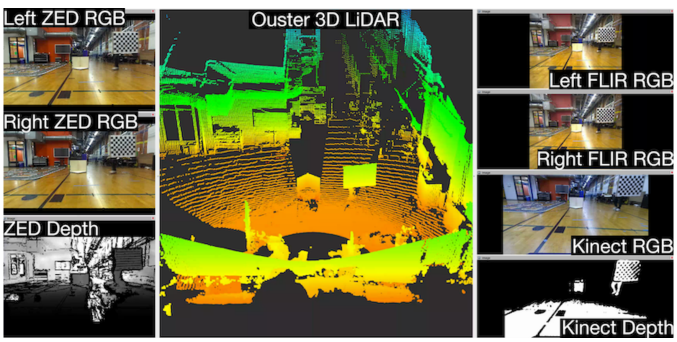


Fig. 3: Sample frame from calibration file. Calibration half-cube and checkerboard are simultaneously visible in all RGB, depth, and 3D LIDAR frames.

IV. Data Collection Procedure

In this section, we describe the sensor calibration procedure and data collection routes for CODa.

A. Per Sequence Calibration Procedure

Route	Setting	Region	Traversals	Dist. (m)	Dur. (hr)
Gates-Dell	Both	GDC, SWY	7	5139	1.95
Guad24	Out	Guad, SWY	6	8799	3.07
WCPowers	Both	WCP, SWY	7	8005	2.79
Union	In	UNB, Guad	3	2450	0.93

TABLE III: Summary of the four routes in CODa. We traverse each route multiple times to capture diverse viewpoints, weather, and lighting conditions. Each route passes through a set of geographic regions defined in Fig. 4. The setting column describes whether the route is indoors, outdoors, or both.

Each sequence in CODa includes a pre or post-run calibration log file containing the raw sensor data for each modality and calibration targets in the field of view. Fig. 3 shows a sample frame from the calibration log file. We calibrate the camera and IMU intrinsics once and use the same intrinsics for all sequences. We recalibrate the camera, LiDAR, and IMU extrinsics for each sequence to account for any changes in the sensor setup. We describe the calibration procedure for each sensor modality below.

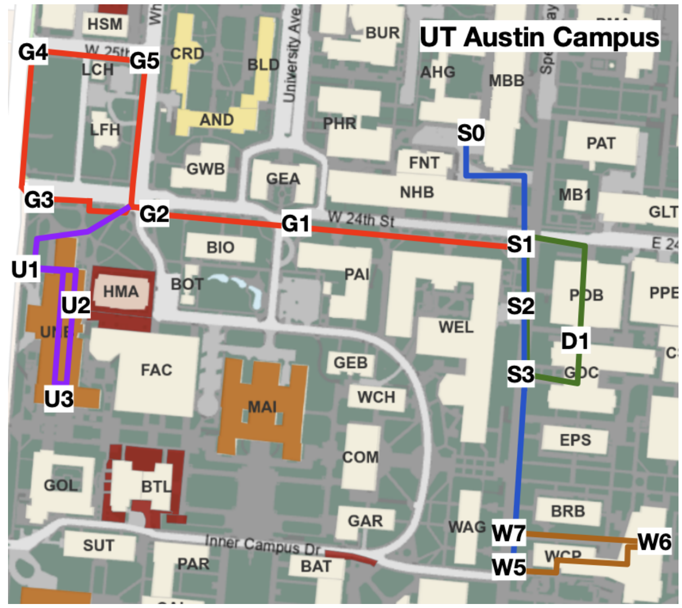


Fig. 4: Spatial map of the five geographic regions in CODa. Each region spans the area surrounding its respective colored line. Operators pause the robot at each waypoint denoted on the map to correct global pseudo-ground truth pose estimates. We refer to the regions surrounding the blue, green, brown, red, and purple lines as SWY, GDC, WCP, Guad, and UNB in Fig. 6

We calibrate the stereo RGB cameras with a checkerboard calibration pattern [40] using multiple images of the checkerboard at different positions. We obtain the LiDAR camera extrinsics using checkerboard images and an approach [41] that optimizes the sensor pose with respect to the checkerboard target and the entire scene. To obtain the LiDAR-IMU extrinsic, we use a target-free extrinsic calibration algorithm [42] that exploits vehicle motion. A calibration half-cube is used to ensure that the LiDAR depth camera extrinsic is accurate.

B. Operator Roles and Data Privacy

After calibration, pairs of operators drove the robot along one of four predetermined routes on UT campus. The primary operator drove the robot along predefined routes, including stopping at waypoints defined in Fig. 4, which are used to ensure global pose consistency between sequences. The second operator addressed questions from the crowd about CODa and handed out research information sheets containing a data privacy disclaimer and contact information. This operator logged all individuals' requests to opt-out from participating in CODa. We increase transparency by mounting a sign on the robot to indicate when it is recording data. While no individuals opt-ed out from our experiments, we protect the privacy of those who do with our user data removal procedure. We describe the user data removal procedure and release the research information sheet on TDR [1]. In the next section, we explain the routes in detail.

C. Data Collection Routes

The four navigation routes along UT campus are: Gates-Dell, WCPowers, Guad24, and Union. Fig. 5 shows the reconstructed map for the first three routes in red, blue, and green. We summarize the characteristics for each route in Table III, including the total distance traversed, total duration, number of traversals, and geographic regions visited for each route. These geographic regions are shown in Fig. 4 as SWY, GDC, WCP, GUAD, and UNB. We choose each region for the following attributes: SWY has a large open area shared by vehicles and pedestrians, GDC has large open areas with classrooms, WCP has scenes from a cafeteria, GUAD has scenes from sidewalks and vehicle-only roads, and UNB has scenes from a library and study area.

Each region is observed multiple times from various viewpoints, weather, and lighting conditions. Each route is traversed from two directions to provide opposing views of all regions. Additional information on the waypoint traversal order for each route can be found in the data report[2] We quantify the observation diversity in Fig. 6 by counting the number of observed frames in CODa for each region under four weather/lighting conditions (CLOUDY, DARK, SUNNY, RAINY) during three times of day (MORNING, AFTERNOON, EVENING). While we are unable to deploy the robot when it is actively raining, we collect data immediately after rainfall and label frames that satisfy these conditions as rainy. Across all sequences, CODa contains 3 rainy, 7 cloudy, 4 dark, and 9 sunny sequences. Fig. 17 qualitatively showcases the data diversity in CODa using sampled images from each sequence.

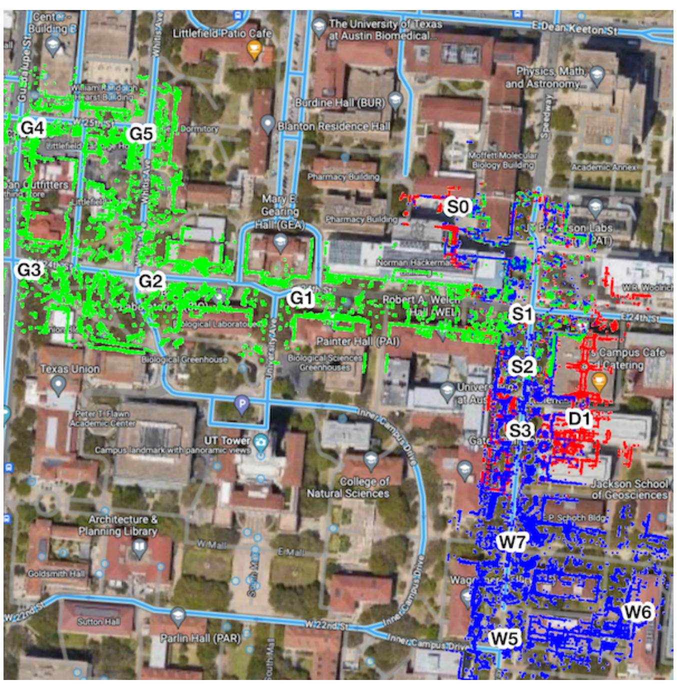


Fig. 5: Satellite image of UT campus with the transformed point clouds and waypoints overlaid. The blue, green, and blue points correspond to the WCP, GDC, and Guad routes respectively. Operators pause the robot at each waypoint to establish global correspondences for pseudo-ground truth pose estimates. Most sequences exhibit poor GPS reception, thus requiring poses to be estimated from LiDAR, inertial, and waypoint data.

[2]The official CODa data report is available at https://doi.org/10.18738/T8/BBOQMV, Texas Data Repository, V2; baa_DATA_REPORT.pdf [fileName].

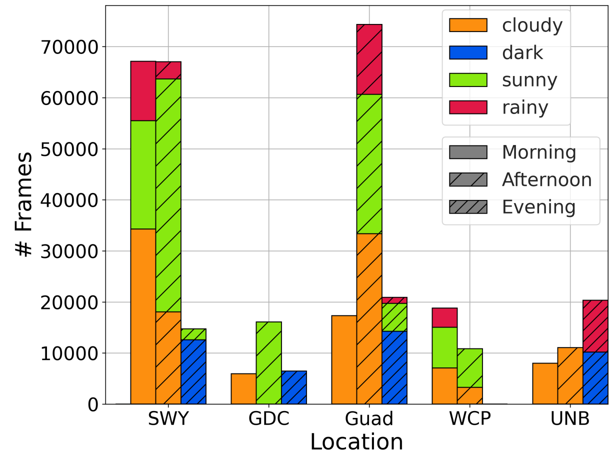


Fig. 6: Number of frames in CODa by geographic region and weather condition. Regions with temporally diverse observations contain frames during multiple times of the day. The coverage areas for regions SWY, GDC, WCP, Guad, and UNB are marked by blue, green, brown, red, and purple lines respectively in Fig. 4.

V. ANNOTATION AND LABELS

We utilized Deepen AI [3], a 3rd party annotation company, to annotate point clouds from our 3D LiDAR with 3D bounding box and semantic segmentation annotations. We instructed Deepen annotators using our annotation guide, which we provide in the data report [1]. The annotation guide contains visual examples for each object and terrain class described in Fig. 15 and Fig. 13, quantitative occlusion level definitions, and operating procedures to determine object instance IDs. Following these instructions, Deepen annotators manually labeled 58 minutes of frames, followed by manual quality assurance checks to ensure that at least 95% of the bounding boxes and 90% of the terrain segmentation annotations were valid on the 3D point clouds. Our internal team then inspected each frame for additional issues. We now describe each annotation type in CODa in detail.

A. 3D Bounding Boxes

Each 3D bounding box has 9 degrees of freedom, instance ID, object class, and occlusion level attributes. We maintain the same instance ID for each object as long as it is observable from the LiDAR or camera sensor or if it does not leave view for longer than 3 seconds. There are six occlusion types, ranging from NONE, LIGHT, MEDIUM, HEAVY, FULL, and UNKNOWN occlusion. The first five occlusion types are used if the object is observable by the cameras or can be identified fully in the 3D point cloud. Objects that never enter the camera view or are geometrically ambiguous are given the unknown occlusion status. This label definition makes CODa useful for evaluating the 3D object tracking task under occlusion. Fig. 15 defines the object ontology for CODa. Because the full list of object classes is large, we refer the reader to the data report [1] for visual examples of each class.

[3]Company Website (Deepen): https://www.deepen.ai/

B. 3D Semantic Segmentation

We annotate each point on the surrounding terrain with a semantic class label. We differentiate terrain classes by their visual appearance and geometric shape. For instance, red and yellow bricks are similar geometrically but are treated as different terrains because they are visually distinct. This makes 3D semantic segmentation challenging with just a single 3D LiDAR and encourages multi-modal methods that fuse 2D images and 3D LiDAR to infer terrain-level semantic labels. We label ambiguous points as unknown and points not associated with terrain as unlabeled. The full terrain ontology and examples for each class can be found in Fig. 13 and Fig. 14 respectively.

C. Pseudo Ground Truth Poses

Due to the unreliability of GPS in urban environments, we use Lego-LOAM [43] to obtain initial robot poses and automatic SLAM tools [44], [45] to refine these pose estimates globally. We manually impose constraints using known global map correspondences to minimize the relative trajectory error across sequences. Prior work [44] using similarly sized campus SLAM datasets demonstrate that pose optimization using these constraints can reduce the relative trajectory error to less than 1m. While it is difficult to provide a bound on the absolute localization error due to the lack of GPS information, we estimate the relative localization error across traversals to be less than 1m based on this prior work. This makes CODa well-suited for evaluating relative localization drift across traversals for long-term SLAM algorithms. In Fig. 4, we qualitatively assess our method's accuracy by visualizing the global pose estimate on a satellite image of UT campus and 3D map reconstruction.

VI. Analysis of CODa Annotations and Statistics

In this section, we analyze the distribution of data in CODa by geographic region, weather, and lighting conditions.

Fig. 6 shows that all geographic regions in CODa (besides WCP) contain data in the morning, afternoon, and evening. All routes with outdoor observations contain at least one sequence captured under rainy conditions. While the full dataset is biased toward sunny and cloudy weather, Fig. 9 shows that the annotated dataset contains 20 object classes that have at least 100 labels under all conditions. With this number going up to 36 classes if we only consider 3 of the 4 conditions. Aside from ATM, most classes contain 100 to 1000 labels each, with Fig. 9 showing the top five classes: pedestrian, tree, pole, railing, and chair. This class and weather imbalance is common in real-world datasets [13], [29] and is a challenging aspect that perception algorithms deployed in urban environments need to be resilient to.

Fig. 10 shows the proportion of each terrain class among the annotated points in CODa, organized by the parent class defined in the terrain ontology in Fig. 13. Among the 23 terrain classes, 21 have more than 200,000 annotated points each, with outdoor classes dominating the majority of the annotations. The two classes that do not satisfy this are dome mat and metal floor. This is because these terrains are small in size and uncommon in environments where they are found. This class imbalance is present in other real-world semantic segmentation datasets [14], [31] as well.

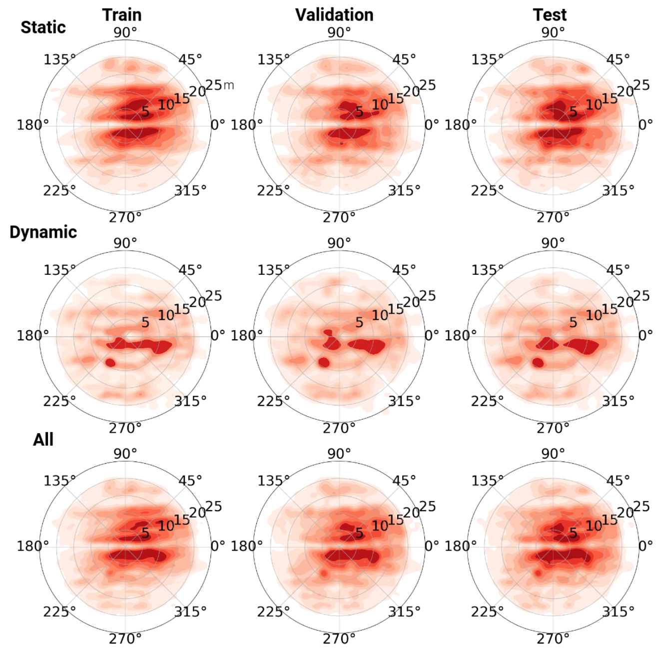


Fig. 7: Spatial distribution of static (top), dynamic (middle), and all (bottom) objects around the robot for the train (left), validation(center), and test (right) splits. Angles (in degrees) are with respect to the forward heading of the robot, range values in meters.

We propose train, validation, and test splits for our 3D object detection and 3D terrain segmentation benchmarks, with each split containing 70%, 15%, and 15% of each annotated sequence respectively. We visualize the spatial distribution of objects around the robot in Fig. 7 for static, dynamic, and all objects for each proposed split in a Kernel Density Estimate (KDE) plot. This demonstrates that both the density and relative position of objects around the robot are similar between our proposed splits.

VII. Experiments and Analysis

We leveraged the unique characteristics of CODa to conduct experiments that answer the following questions:

- Question 1: How well do 3D object detectors trained on large-scale AV datasets perform on CODa?
- Question 2: How well does unsupervised domain adaptation from AV datasets perform on CODa?
- Question 3: Can we improve object detection performance for low-resolution single 3D LiDAR setups on robots by fine-tuning on downsampled LiDAR point clouds?
- Question 4: Does pre-training on CODa improve cross-dataset object detection on existing urban robotics datasets?

A. Experimental Setup — Selecting a 3D Object Detection Algorithm

We choose a 3D object detector by evaluating the performance of three 3D object detectors: PointPillars [46], Cen-

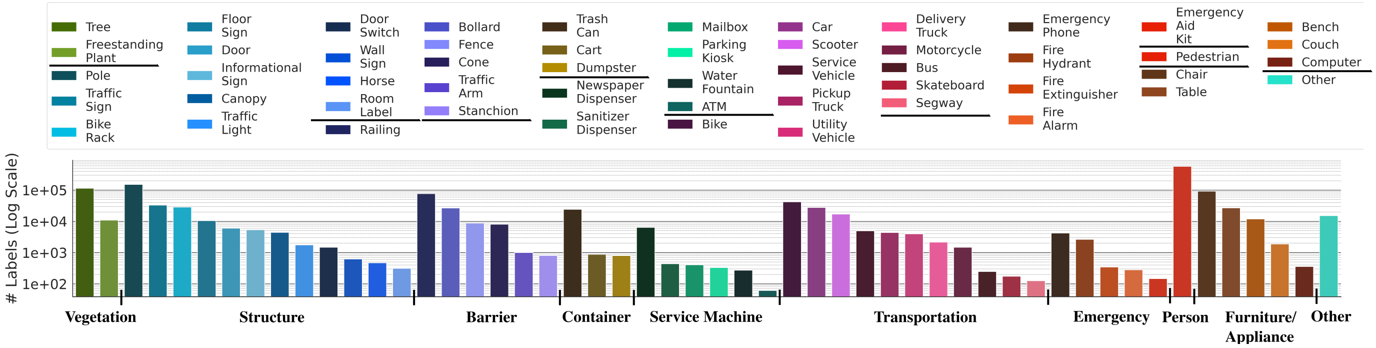


Fig. 8: Number of object labels per class organized by topological category. Objects in each topological category are sorted in order of most to least common.

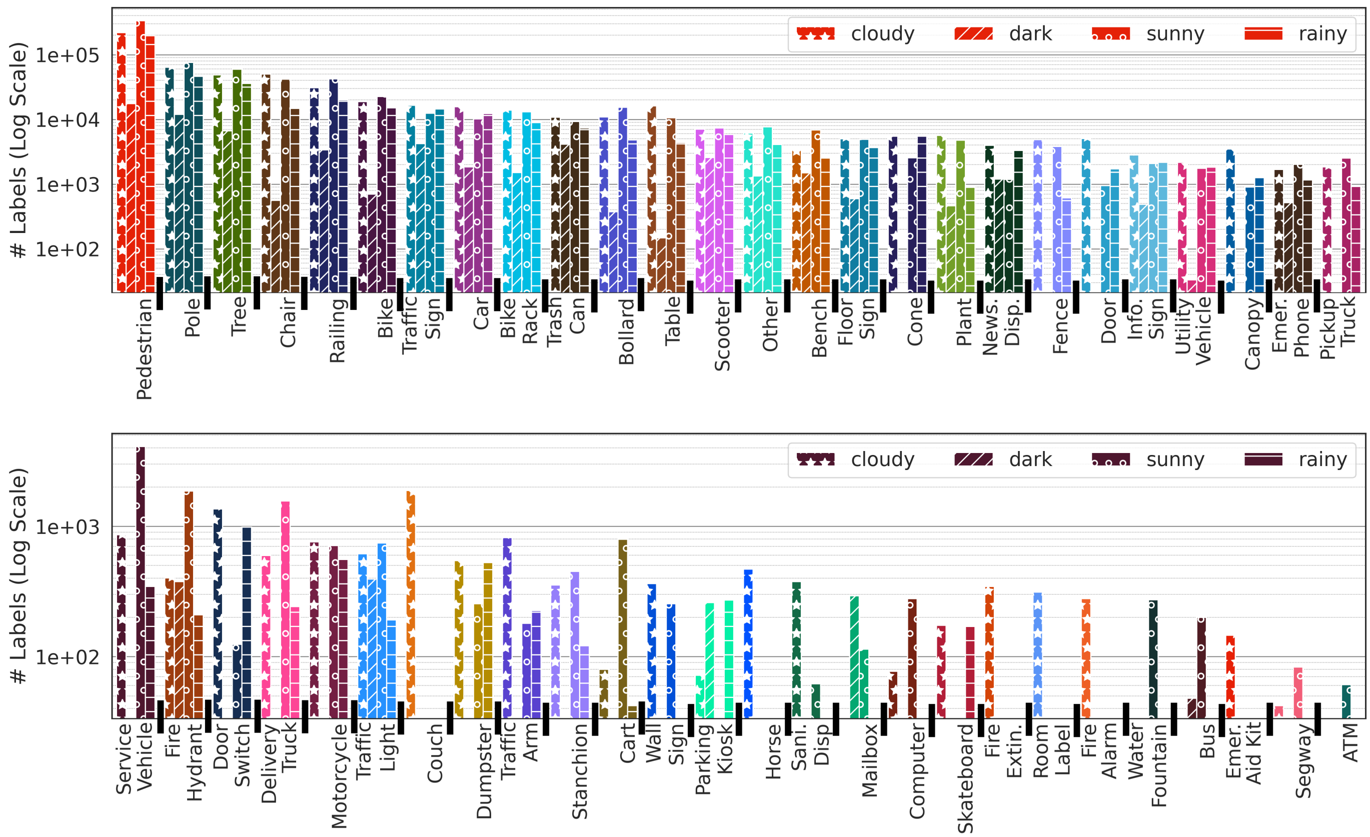


Fig. 9: Histogram of the number of annotations per object class under four weather conditions (sunny, rainy, cloudy, dark). Object classes are organized by most to least frequent from left to right. Bars with stars are cloudy, diagonal lines are dark, circles are sunny, and horizontal lines are rainy. We consider labels for objects in rainy conditions to count towards cloudy as well.

terPoint [47]. PVRCNN [48] on KITTI, Waymo, nuScenes, and CODa. These datasets are among the most widely used in 3D object detection benchmarks and for cross-dataset domain adaptation analysis [29], [49]. We evaluate the preceding models because they are LiDAR-only approaches, easy to reproduce, and achieve state-of-the-art detection performance on AV datasets. Both Centerpoint and PointPillars are top-performing open-source methods on Waymo and nuScenes leaderboards, and the OpenPCDet [50] implementation of PVRCNN unofficially outperforms the former models on Waymo. We use the OpenPCDet implementation of each model because it provides the model configuration files, making results more reproducible.

We use the default model configurations provided in OpenPCDet and train each model for 30 epochs or until the performance saturates. For models that OpenPCDet does not provide configurations for, we benchmark various model architectures in Table IV and select the most favorable one.

All experiments involving CODa in Tables IV, V, and VI are conducted using the medium train, validation, and test split for computational reasons. We use the full CODa split for Table VII experiments to better match the scene diversity in AV datasets. For all metrics, we use the 3D object detection and bird's eye view evaluation metric proposed in the KITTI Vision Benchmark Suite [5] with an IOU of 0.7, 0.5, and 0.5 for the car, pedestrian, and cyclist classes respectively. This class list is consistent across CODa and AV datasets. For completeness, we report model performance on the full list of

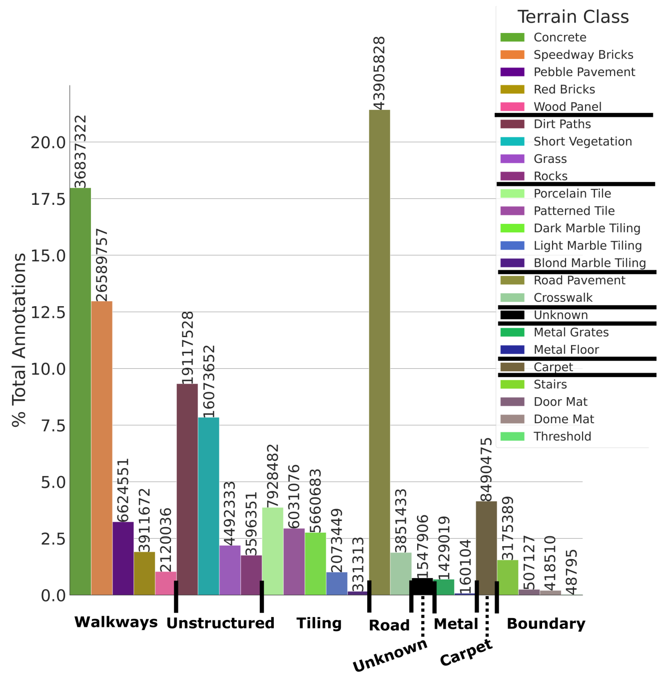


Fig. 10: Histogram of 3D semantic segmentation annotations labels for outdoor, indoor, and both environments in CODa. Vertical numbers above each bar indicate the total number of points annotated for that semantic class. The semantic classes in the legend map to the bars from left to right.

object classes for multiple LiDAR resolutions in the appendix.

Mod. \ Data.	nuScenes	Waymo	KITTI	CODa
PointPillars [46]	28.42 17.94	55.11 47.55	70.27 63.32	49.78 48.86
CenterPoint [47]	36.91 23.86	62.66 54.86	69.34 63.87	82.08 76.92
PVRCNN [48]	33.85 **25.41**	**62.73 56.40**	70.22 **65.28**	**92.08 91.11**

TABLE IV: Evaluation of several 3D object detectors on AV Datasets and CODa, We report mean average precision for the car, pedestrian, and cyclist categories in bird's eye view (AP_{BEV}) and 3D (AP_{3D}) with IOU 0.7, 0.5, and 0.5 respectively. We average the results at the easy, medium, and hard difficulties (following the KITTI Vision Benchmark). The blue and red indicate the highest-performing training method for BEV and 3D detection for each dataset. Mod. - Model Data. - Dataset

We observe in Table IV that PVRCNN generally performs the best for 3D bounding box detection on large-scale AV datasets and CODa. As such, we select this model architecture to use in all of our later experiments. For a full summary of all models evaluated for this experiment, please refer to Table X in Appendix Section XI-D.

B. AV Dataset to CODa Adaptation

We apply several domain adaptation strategies to evaluate 3D object detector performance on CODa with and without domain-specific labels. In our experiment setup, we chose the object class list to be car, pedestrian, and cyclist so that it is consistent with the standard class list evaluated for nuScenes and Waymo. We perform the standard 3D data augmentation techniques (scaling, rotation, flipping) and align the point cloud ground plane heights for our experiments.

Direct Transfer (Direct). In this experiment, the campus dataset is not accessible and the pre-trained model is evaluated

PT \ DA	Direct	ST	FT	ST + FT
nuScenes [13]	21.30 15.53	14.07 10.76	91.39 90.16	92.38 91.02
Waymo [12]	46.20 43.11	38.27 34.36	**93.12 92.07**	92.36 91.18
CODa	**92.08 91.11**	-	-	-

TABLE V: Evaluation of PV-RCNN pretrained (**PT**) on AV Datasets and evaluated on the CODa test split after undergoing different domain adaptation (**DA**) methods. DA methods include: 1) **Direct** — train on the source dataset and evaluate directly on CODa; 2) *ST3D++* **ST** [51]— for unsupervised adaptation; 3) Fine-tuning (**FT**)— with CODa after pre-training on the source dataset; and 4) Both ST3D++ and Fine-tuning (**ST + FT**). The results demonstrate that even state-of-the-art unsupervised domain adaptation methods for 3D object detectors are not competitive with approaches that use domain-specific training labels. All models are evaluated on the medium test split of CODa.

directly on the test split. This is our baseline for the expected performance when deploying on a campus scale without CODa.

ST3D++ (ST). In this scenario, the campus dataset is accessible but the ground truth labels are not available. This is typical for robot deployments as domain-specific raw sensor data is readily available. We used ST3D++ [51] to adapt to the campus domain as the authors demonstrated state-of-the-art unsupervised domain adaptation performance improvement between different AV datasets when using their method.

We perform a coarse hyperparameter tuning sweep across the positive and negative thresholds for each object class and use the same model weights for each class. After performing the self-training process for 25 epochs, we evaluate the highest-performing epoch directly on CODa. We present the highest performing models in Table X and include the full experiment list in Appendix Section XI-D, Table XI. This is the best performance we can achieve without labels when deploying on a campus scale with raw sensor data available.

Domain Specific Finetuning (FT). We assume that domain-specific ground truth labels are available. We pre-train the model backbone on nuScenes or Waymo before fine-tuning on the train split of CODa, hypothesizing that learning features on other datasets benefit domain-specific performance.

We pre-train PVRCNN from scratch on each AV dataset for 30 epochs and evaluate the model on the CODa test split. After pre-training, we freeze the encoder and backbone weights and randomly initialize the detection, classification, and dense heads. We finetune the heads for 25 epochs, unfreeze the encoder and backbone weights, and train the entire model for another 25 epochs. Our experiments X show that a learning rate of 0.01 and the Adam 1cycle optimizer [52] provide the best empirical performance.

ST3D++ with Domain Specific Finetuning (ST + FT). This experiment combines the ST and FT methods described earlier. We follow the same procedure for self-training using the ST method with the same hyperparameters. After self-training, we apply the training procedure in FT. For this approach, we find that a learning rate of 0.01 and the Adam 1cycle optimizer provides the best empirical performance.

Domain Adaptation Discussion. Table V demonstrates a significant performance gap between unsupervised domain adaptation and fully supervised methods. The highest-performing pre-trained model is about 40 percent lower than the same model trained from scratch on CODa. This is expected due to the large domain and sensor-specific differences between nuScenes, Waymo, and CODa in Table II, such as the number of LiDAR points per frame. For ST, the model pre-trained on AV datasets decreases in performance after self-training on CODa. These results are consistent with findings from the ONCE [29] AV dataset. They show that performing unsupervised domain adaptation with ST3D from nuScenes to ONCE decreases performance and hypothesize that this is due to differences in LiDAR beam resolution. We believe that differences in sensor viewpoint and resolution cause ST3D to produce poor quality pseudo labels on CODa and support this hypothesis in Section VII-C.

Our experiments show that pre-training on Waymo improves performance on 3D bounding box and BEV tasks by about 1-2 percent compared to training from scratch. However, performance does not consistently improve between FT and ST+FT adaptation techniques. We speculate that when trained to performance saturation, FT improvements dominate the effects of ST pre-training. We conclude from these studies that downstream tasks like 3D object detection benefit from better initial 3D representations. In addition, we hope that our empirical analysis of methods like ST3D spurs future work on how to continue improving self-training methods between domains with significant changes in sensor resolution, viewpoint, and geometric features.

C. Impact of Sensor Resolution Differences on Object Detection Performance

Train \ Test	CODa-16		CODa-32		CODa-64		CODa-128	
CODa-16	**75.15**	**73.29**	64.99	63.24	49.17	47.36	21.93	18.94
CODa-32	50.79	47.95	**78.30**	**76.90**	70.49	69.37	59.95	56.59
CODa-64	21.10	22.05	67.27	64.77	**86.20**	**84.48**	77.63	77.53
CODa-128	12.58	12.16	48.05	45.76	76.51	75.38	**92.61**	**91.34**

TABLE VI: Evaluating the impact of point cloud resolution differences between the source and target domain on 3D object detector performance. All experiments are conducted with a PV-RCNN detector first pre-trained on Waymo using the pedestrian, car, and cyclist classes. We fine-tune the pre-trained model on CODa downsampled to 16, 32, 64, and 128 vertical channels (CODa-#channels) for 50 epochs. We then evaluate the model performance on different point cloud resolutions for the pedestrian, car, and cyclist classes using the same evaluation metric as Table IV. All models are evaluated on the medium test split of CODa.

While most robots benefit from having high-quality object detections, their wide range of sensor setups presents a challenge for object detectors. Therefore, it is important to understand how object detection performance is affected by sensor resolution differences between the train and test domains.

For our experiments, we train PV-RCNN from scratch on 20% of the Waymo train dataset and fine-tune the model on the CODa medium train split at four LiDAR resolutions (16, 32, 64, 128) on the car, pedestrian, and cyclist classes for 30 epochs or until performance saturates. Our LiDAR is originally 128 channels, so we subsample the original point cloud to obtain the lower resolutions. For fine-tuning, we follow the same two-stage process used in the prior experiments: train a randomly initialized detection head for 15 epochs while keeping the model backbone frozen and then train the full model for another 30 epochs. After training, we evaluate the model directly on the CODa medium test split at all four LiDAR resolutions.

Sensor Resolution Discussion. Table VI shows that 3D object detectors trained on a specific sensor resolution perform best on the same sensor resolution during test time. Furthermore, the larger the resolution difference between the train and test domains, the more performance is affected. This vindicates our hypothesis that large differences in LiDAR resolutions negatively affect object detection performance. Thus, we release pre-trained models on all classes in CODa for the (16, 32, 64, and 128) channel LiDAR resolutions and encourage users to select the pre-trained model that is most similar to the target dataset's resolution. Table IX in Appendix Section XI-A reproduces this experiment for all classes in CODa.

D. JRDB Adaptation

Train \ Test	JRDB (15m) Prec.	JRDB (15m) Rec.	JRDB (15m) F1	JRDB (25m) Prec.	JRDB (25m) Rec.	JRDB (25m) F1
nuScenes	45.15	12.10	20.31	42.12	11.83	18.48
Waymo	55.39	18.70	27.96	52.76	17.19	25.94
CODa	60.29 $^{+4.90}$	25.32 $^{+6.62}$	35.66 $^{+7.7}$	57.38 $^{+4.62}$	25.31 $^{+8.12}$	35.13 $^{+9.19}$
JRDB	65.64	27.14	38.39	64.15	27.15	38.15

TABLE VII: Cross-dataset 3D object detection performance comparison on JRDB [7] after training on CODa, nuScenes [13], and Waymo [12]. We train a PV-RCNN detector on only pedestrians for all datasets. We evaluate the average precision, recall, and F1 score for objects within 15 meters (15m) and 25 meters (25m) of the ego vehicle. We report the performance difference between the highest performing AV and CODa models in red and blue superscripts.

Aside from sensor variations, viewpoint and scene differences between train and test domains also present a challenge for LiDAR-based object detectors. To understand the impact of these differences, we evaluate the performance of 3D object detectors trained on CODa and AV datasets on JRDB, a large-scale urban robot dataset with LiDAR point clouds and 3D bounding box annotations.

For our experiments, we train three PV-RCNN models from scratch on 20% of the Waymo train split, full nuScenes train split, full CODa train split, and full JRDB train split for 30 epochs or until performance saturates. For consistency, all models are only trained on pedestrians and evaluated on the proposed JRDB validation split using their 3D detection benchmark metrics (average precision, recall, and F1 score). We repeat our evaluation on two variations of the validation split: one containing ground truth annotations exclusively within 15 meters of the ego vehicle and the other within 25 meters of the ego vehicle.

JRDB Performance Discussion. Table VII shows that CODa models consistently outperform AV models in all metrics for both the 15m and 25m range. Furthermore, pretraining on CODa offers similar performance to training with labels on JRDB, corroborating our claim that pre-trained CODa models generalize to other urban settings. We believe this can be

Approach	mIoU	Concrete	Grass	Rocks	Speedway Bricks	Red Bricks	Pebble Pavement	Light Marble Tiling	Dark Marble Tiling	Dirt Paths	Road Pavement	Short Vegetation	Porcelain Tile	Metal Grates	Blond Marble Tiling	Wood Panel	Patterned Tile	Carpet	Crosswalk	Dome Mat	Stairs	Door Mat	Threshold	Metal Floor
Cylinder3D [53]	49.9	**68.7**	**54.7**	0.2	**69.7**	57.4	38.2	48.2	45.3	**70.9**	**80.6**	**81.8**	81.2	**39.2**	7.9	**78.8**	3.5	79.0	**52.8**	**24.9**	**93.9**	10.7	**5.3**	**59.6**
2DPass [54]	**51.5**	51.2	36.3	**68.1**	67.0	**63.6**	**39.4**	**62.9**	**68.6**	61.4	64.4	70.5	**83.2**	13.2	**34.2**	73.9	**81.5**	**84.1**	34.5	0.1	87.1	**29.0**	0.0	25.9

TABLE VIII: Evaluation of two 3D semantic segmentation models on the full CODa test split. We report mean intersection over union and accuracy for each semantic class. Bold numbers indicate the highest-performing method for each category.

explained by CODa's similarity to JRDB in terms of sensor resolution, viewpoint, and scene diversity. By utilizing prior knowledge of similar environments in JRDB, CODa models are more robust to point cloud sparsity than Waymo models. Fig. 16 in Section VII-D vindicates our claim with several examples where CODa models detect sparse pedestrians that the highest performing AV model misses.

To assess how variations in sensor resolution affect model performance across datasets, we evaluated models trained on different resolutions of CODa on JRDB in Table XII in Appendix Section VII-D. Our findings indicate that detection performance decreases as the sensor resolution difference increases between the train and test datasets. This aligns with the insights we presented in Section VII-C, demonstrating that cross-dataset performance is maximized when the train and test resolutions closely match. Thus, we recommend that users select the pre-trained model that is most similar to their target dataset's resolution for optimal performance. Our findings should motivate future work to leverage scene context and develop density invariant models to improve 3D object detection performance.

VIII. Benchmarks

In this section, we define the 3D object detection and 3D semantic segmentation for this dataset. We plan on adding additional tasks in the future for robot perception and planning, such as long-term SLAM, cross-domain information retrieval, and preference-aware navigation.

A. 3D Object Detection

The 3D object detection task involves predicting 7 degrees of freedom boxes for all object classes. We use the 3D object detection metric proposed in the KITTI Vision Benchmark Suite. For the car, pedestrian, and cyclist classes, we require a minimum bounding box overlap of 70%, 50%, and 50% to determine if detection is correct. For all other object classes, we use a minimum overlap of 50% with the ground truth bounding box. All methods are limited to using up to 10 prior LiDAR frames for predictions. All sensor modalities and pseudo-ground truth poses can be used, and we will evaluate all predictions on the 3D point cloud annotations.

B. 3D Semantic Segmentation

For the 3D semantic segmentation benchmark, we use the same evaluation metric proposed in SemanticKITTI [14]. This is the mean intersection-over-union (mIoU) metric [3] over all classes. All sensor modalities can be used, but we will evaluate all predictions using the 3D point cloud annotations. Table VIII benchmarks Cylinder3D [53] and 2DPass [54], two state-of-the-art LiDAR only and LiDAR camera approaches respectively. For our benchmarks, we train both models from scratch for 30 epochs or until performance saturates and take the highest-performing model.

IX. Conclusion and Future Work

In this work, we presented the UT Campus Object Dataset (CODa), a multi-modal dataset that contains greater object and scene-level annotation diversity than any other similar existing dataset. CODa contains 1.3 million human-annotated 3D bounding boxes and 5000 semantic segmentation annotations over 8.5 hours of data collected from the perspective of a mobile robot across UT campus. We publicly release CODa on the Texas Data Repository [1], pre-trained models for various LiDAR resolutions (16, 32, 64, 128 channels), and dataset development package.

We conducted extensive experiments to select a high-performing model architecture for urban environments. We demonstrated a performance gap for 3D object detectors in urban environments by comparing the performance on CODa's test split after training on CODa versus AV datasets. We empirically demonstrated that 3D object detection performance is significantly affected by differences in LiDAR sensor resolution during test time. Finally, we conducted various ablation studies to show that pre-training on CODa instead of AV datasets improves cross-dataset object detection performance on existing urban robotics datasets. This constitutes motivation for future work to improve 3D object detector invariance to point cloud density and highlights the importance of selecting a pre-trained model that closely resembles the target domain during robot deployments. We expect that this work will spur future research toward learning sensor-invariant 3D feature representations, object-centric localization approaches, and terrain-aware navigation planners. In the future, we plan on releasing additional benchmarks on CODa to facilitate fair comparison for methods in these research areas.

X. Acknowledgement

This work was conducted with the Autonomous Mobile Robotics Laboratory (AMRL) at UT Austin. This project is partially supported by NSF Awards CAREER-2046955 and

IIS-1954778. We would like to thank Roberto Martín-Martín, Zhangyang Wang, and Philipp Krähenbühl for their fruitful comments, suggestions, and inspiration. The authors acknowledge the Texas Advanced Computing Center (TACC)[4] at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper. URL: http://www.tacc.utexas.edu. The CODa study is covered under the University of Texas at Austin IRB Number STUDY00003493.

References

[1] A. Zhang, C. Eranki, C. Zhang, R. Hong, P. Kalyani, L. Kalyanaraman, A. Gamare, M. Esteva, and J. Biswas, "UT Campus Object Dataset (CODa)," 2023. [Online]. Available: https://doi.org/10.18738/T8/BBOQMV

[2] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[3] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, p. 303–338, jun 2010. [Online]. Available: https://doi.org/10.1007/s11263-009-0275-4

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[5] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[6] H. Karnan, A. Nair, X. Xiao, G. Warnell, S. Pirk, A. Toshev, J. Hart, J. Biswas, and P. Stone, "Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation," 2022.

[7] R. Martín-Martín, H. Rezatofighi, A. Shenoi, M. Patel, J. Gwak, N. Dass, A. Federman, P. Goebel, and S. Savarese, "JRDB: A dataset and benchmark for visual perception for navigation in human environments," *CoRR*, vol. abs/1910.11792, 2019. [Online]. Available: http://arxiv.org/abs/1910.11792

[8] A. Kirillov, K. He, R. B. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," *CoRR*, vol. abs/1801.00868, 2018. [Online]. Available: http://arxiv.org/abs/1801.00868

[9] T. Guan, D. Kothandaraman, R. Chandra, and D. Manocha, "Ganav: Group-wise attention network for classifying navigable regions in unstructured outdoor environments," *CoRR*, vol. abs/2103.04233, 2021. [Online]. Available: https://arxiv.org/abs/2103.04233

[10] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, 2006, pp. 519–528.

[11] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of Michigan North Campus long-term vision and lidar dataset," *International Journal of Robotics Research*, vol. 35, no. 9, pp. 1023–1035, 2015.

[12] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," *CoRR*, vol. abs/1912.04838, 2019. [Online]. Available: http://arxiv.org/abs/1912.04838

[13] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *CVPR*, 2020.

[14] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "A dataset for semantic segmentation of point cloud sequences," *CoRR*, vol. abs/1904.01416, 2019. [Online]. Available: http://arxiv.org/abs/1904.01416

[15] M. Fallon, H. Johannsson, M. Kaess, and J. Leonard, "The mit stata center dataset," *International Journal of Robotics Research*, vol. 32, pp. 1695–1699, 12 2013.

[16] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[17] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. F. Fallon, "The newer college dataset: Handheld lidar, inertial and vision with ground truth," *CoRR*, vol. abs/2003.05691, 2020. [Online]. Available: https://arxiv.org/abs/2003.05691

[18] M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon, "A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments," in *International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[19] P. Jiang, P. R. Osteen, M. B. Wigness, and S. Saripalli, "RELLIS-3D dataset: Data, benchmarks and analysis," *CoRR*, vol. abs/2011.12954, 2020. [Online]. Available: https://arxiv.org/abs/2011.12954

[20] P. Yin, S. Zhao, R. Ge, I. Cisneros, R. Fu, J. Zhang, H. Choset, and S. Scherer, "Alita: A large-scale incremental dataset for long-term autonomy," 2022.

[21] J. Jiao, H. Wei, T. Hu, X. Hu, Y. Zhu, Z. He, J. Wu, J. Yu, X. Xie, H. Huang, R. Geng, L. Wang, and M. Liu, "Fusionportable: A multi-sensor campus-scene dataset for evaluation of localization and mapping accuracy on diverse platforms," 2022.

[22] X. Shi, D. Li, P. Zhao, Q. Tian, Y. Tian, Q. Long, C. Zhu, J. Song, F. Qiao, L. Song, Y. Guo, Z. Wang, Y. Zhang, B. Qin, W. Yang, F. Wang, R. H. M. Chan, and Q. She, "Are we ready for service robots? the OpenLORIS-Scene datasets for lifelong SLAM," in *2020 International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3139–3145.

[23] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond pascal: A benchmark for 3d object detection in the wild," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014.

[24] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[25] J. Yin, A. Li, T. Li, W. Yu, and D. Zou, "M2DGR: A multi-sensor and multi-scenario SLAM dataset for ground robots," *CoRR*, vol. abs/2112.13659, 2021. [Online]. Available: https://arxiv.org/abs/2112.13659

[26] D. Feng, Y. Qi, S. Zhong, Z. Chen, Y. Jiao, Q. Chen, T. Jiang, and H. Chen, "S3e: A large-scale multimodal dataset for collaborative slam," 2023.

[27] Y. Liu, Y. Fu, M. Qin, Y. Xu, B. Xu, F. Chen, B. Goossens, P. Z. Sun, H. Yu, C. Liu, L. Chen, W. Tao, and H. Zhao, "Botanicgarden: A high-quality dataset for robot navigation in unstructured natural environments," *arXiv preprint arXiv:2306.14137*, 2023.

[28] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

[29] J. Mao, M. Niu, C. Jiang, H. Liang, X. Liang, Y. Li, C. Ye, W. Zhang, Z. Li, J. Yu, H. Xu, and C. Xu, "One million scenes for autonomous driving: ONCE dataset," *CoRR*, vol. abs/2106.11037, 2021. [Online]. Available: https://arxiv.org/abs/2106.11037

[30] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *CoRR*, vol. abs/2109.13410, 2021. [Online]. Available: https://arxiv.org/abs/2109.13410

[31] N. Gählert, N. Jourdan, M. Cordts, U. Franke, and J. Denzler, "Cityscapes 3d: Dataset and benchmark for 9 dof vehicle detection," *CoRR*, vol. abs/2006.07864, 2020. [Online]. Available: https://arxiv.org/abs/2006.07864

[32] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving video database with scalable annotation tooling," *CoRR*, vol. abs/1805.04687, 2018. [Online]. Available: http://arxiv.org/abs/1805.04687

[33] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017. [Online]. Available: http://dx.doi.org/10.1177/0278364916679498

[34] X. Song, P. Wang, D. Zhou, R. Zhu, C. Guan, Y. Dai, H. Su, H. Li, and R. Yang, "Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving," *CoRR*, vol. abs/1811.12222, 2018. [Online]. Available: http://arxiv.org/abs/1811.12222

[35] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska, "One thousand and one hours: Self-driving motion prediction dataset," *CoRR*, vol. abs/2006.14480, 2020. [Online]. Available: https://arxiv.org/abs/2006.14480

[4]TACC website: http://www.tacc.utexas.edu

[36] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "Fcns in the wild: Pixel-level adversarial and constraint-based adaptation," 2016.

[37] Y. Tsai, W. Hung, S. Schulter, K. Sohn, M. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," *CoRR*, vol. abs/1802.10349, 2018. [Online]. Available: http://arxiv.org/abs/1802.10349

[38] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *Pattern Analysis and Machine Intelligence (PAMI)*, 2022.

[39] C. Robotics, Online, , retreived 22-July-2023.

[40] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[41] D. Tsai, S. Worrall, M. Shan, A. Lohr, and E. Nebot, "Optimising the selection of samples for robust lidar camera calibration," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 2631–2638.

[42] S. Mishra, G. Pandey, and S. Saripalli, "Motion based extrinsic calibration of a 3d lidar and an IMU," *CoRR*, vol. abs/2104.12280, 2021. [Online]. Available: https://arxiv.org/abs/2104.12280

[43] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.

[44] K. Koide, J. Miura, M. Yokozuka, S. Oishi, and A. Banno, "Interactive 3d graph slam for map correction," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 40–47, 2020.

[45] S. B. Nashed and J. Biswas, "Human-in-the-loop SLAM," *CoRR*, vol. abs/1711.08566, 2017. [Online]. Available: http://arxiv.org/abs/1711.08566

[46] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," 2019.

[47] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3d object detection and tracking," *CoRR*, vol. abs/2006.11275, 2020. [Online]. Available: https://arxiv.org/abs/2006.11275

[48] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "PV-RCNN: point-voxel feature set abstraction for 3d object detection," *CoRR*, vol. abs/1912.13192, 2019. [Online]. Available: http://arxiv.org/abs/1912.13192

[49] Y. Wang, X. Chen, Y. You, L. E. Li, B. Hariharan, M. E. Campbell, K. Q. Weinberger, and W. Chao, "Train in germany, test in the USA: making 3d object detectors generalize," *CoRR*, vol. abs/2005.08139, 2020. [Online]. Available: https://arxiv.org/abs/2005.08139

[50] O. D. Team, "Openpcdet: An open-source toolbox for 3d object detection from point clouds," https://github.com/open-mmlab/OpenPCDet, 2020.

[51] J. Yang, S. Shi, Z. Wang, H. Li, and X. Qi, "St3d++: Denoised self-training for unsupervised domain adaptation on 3d object detection," 2021.

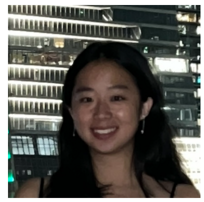
[52] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay," *CoRR*, vol. abs/1803.09820, 2018. [Online]. Available: http://arxiv.org/abs/1803.09820

[53] H. Zhou, X. Zhu, X. Song, Y. Ma, Z. Wang, H. Li, and D. Lin, "Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation," *CoRR*, vol. abs/2008.01550, 2020. [Online]. Available: https://arxiv.org/abs/2008.01550

[54] X. Yan, J. Gao, C. Zheng, C. Zheng, R. Zhang, S. Cui, and Z. Li, "2dpass: 2d priors assisted semantic segmentation on lidar point clouds," in *European Conference on Computer Vision*. Springer, 2022, pp. 677–695.

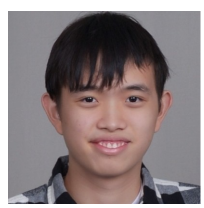
Arthur Zhang received a Bachelor's Degree in Computer Engineering from the University of Michigan, Ann Arbor in 2022. He is currently pursuing a Ph.D. in Computer Science at the University of Texas at Austin with the Autonomous Mobile Robotics Laboratory. His research interests include robot perception, visual representation learning for navigation, and 3D computer vision.

Chaitanya Eranki is an undergraduate student in his final year at the University of Texas at Austin. He is majoring in Computer Science and is interested in pursuing machine learning and software engineering. He is currently a part of the Autonomous Mobile Robotics Laboratory.



Christina Zhang is an undergraduate student in her final year at the University of Texas at Austin. She is majoring in Computer Science and Mathematics and is interested in software engineering. She has worked in the Autonomous Mobile Robotics Laboratory.

Ji-Hwan Park received a Bachelor's Degree in Mechanical Engineering from the University of Texas at Austin in 2023. He is currently pursuing a Master of Science in Mechanical Engineering at the University of Texas at Austin. His main research interests are robotics, SLAM, and computer vision.

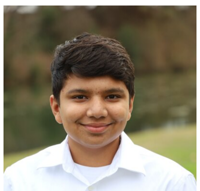


Raymong Hong received a Bachelor's Degree in Computer Science with Honors from the University of Texas at Austin in 2023. His main research interests are 3D computer vision, robotics, and mathematics.

Pranav Kalyani received a Bachelor's and Master's degree in mechanical engineering from the University of Texas (2019 and 2023, respectively). He is currently pursuing a Ph.D. in mechanical engineering at the University of Texas at Austin. His main research interests are controls, dynamics, and machine learning.

Lochana Kalyanaraman is an undergraduate student at the University of Texas at Austin. She is majoring in Computer Science and has an interest in data science and big data. She is working with the Autonomous Mobile Robotics Laboratory as part of the Perception Autonomy EcoCar Team.

Arsh Gamare is an undergraduate student in his final year at the University of Texas at Austin. He is majoring in Electrical and Computer Engineering and is interested in 3D computer vision and Large Language Models. He is a member of the Autonomous Mobile Robotics Laboratory.



Arnav Bagad is an undergraduate student at the University of Texas at Austin. He is in his first year majoring in Computer Science in the Turing Scholars program and is interested in software engineering and robotics. He is a member of the Autonomous Mobile Robotics Laboratory.

Dr. Maria Esteva is a Research Scientist at the Texas Advanced Computing Center at the University of Texas at Austin. Her work focuses on the design, implementation, and evaluation of large-scale data curation applications in High Performance Computing environments. She works with the Texas Robotics researchers to curate and publish their datasets in accordance with FAIR data principles.

Prof. Joydeep Biswas is an Associate Professor in the Department of Computer Science at the University of Texas at Austin. He leads the Autonomous Mobile Robotics Laboratory, which undertakes research along several directions in support of building service mobile robots capable of performing human-assistive tasks in everyday environments for extended periods of time. He earned his PhD and MS in Robotics from Carnegie Mellon University in 2014 and 2010 respectively, and a B.Tech degree in Engineering Physics from the Indian Institute of Technology Bombay in 2008. He is a recipient of the NSF CAREER award, an Amazon Research award, two JP Morgan Faculty Research awards, and is a Siebel Scholar, Class of 2015.

XI. Appendix

We organize the appendix into the following sections: extended model analysis (Section XI-A), CODa organization structure (Section XI-B), model training experiments (Section XI-D), annotation ontology (Section XI-E), qualitative 3D object detection results (Section XI-F), and ground truth annotation visualizations (Section XI-G).

A. Extended Model Analysis

Table IX repeats the analysis in Table VI for all annotated classes in CODa. We conduct these experiments using the same experimental setup and conclude that our findings in Section VII-C hold for all classes. As such, users should select the pre-trained model with the closest LiDAR resolution to their target domain for optimal performance.

Train \ Test	CODa-16		CODa-32		CODa-64		CODa-128	
CODa-16	**23.36**	**21.15**	19.39	17.03	9.65	8.37	2.99	2.38
CODa-32	14.52	12.23	**26.19**	**23.86**	23.38	21.28	17.10	15.08
CODa-64	5.70	4.93	20.49	18.29	**27.38**	**25.51**	25.13	23.30
CODa-128	3.14	2.53	12.85	11.31	25.40	23.34	**28.18**	**26.14**

TABLE IX: Evaluating the impact of point cloud resolution differences between the source and target domain on 3D object detector performance. All experiments are conducted with a PV-RCNN [48] detector first pre-trained on Waymo [12] using the pedestrian, car, and cyclist classes. We fine-tune the pre-trained model on the full CODa train split downsampled to 16, 32, 64, and 128 vertical channels (CODa-#channels) for 30 epochs. Models are finetuned on all classes in CODa and evaluated using the same evaluation metric as Table IV.

B. CODa Organization Structure

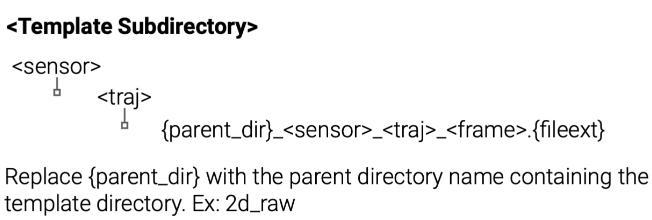


Fig. 11: Full 3D bounding box object list. Bolding represents the topological category for the semantic classes below.

We describe the organization structure of CODa at a high level and refer the reader to the data report [1] for a detailed breakdown of the file contents. The primary non-sensor subdirectories in CODa contain sensor calibrations, metadata, timestamps, and poses. We provide these files for each sequence. The remaining directories contain 2D/3D sensor data and annotations. Each of these directories has an identical subdirectory structure to Fig. 11. The metadata files contain information on each sequence, such as the list of semantic objects present and the dataset splits by task.

C. CODa Dataset Development Kit

We release a dataset development kit, which provides download and visualization tools for the raw data and annotations. We include sample code snippets and tutorials for getting started on the official dataset development kit code release.

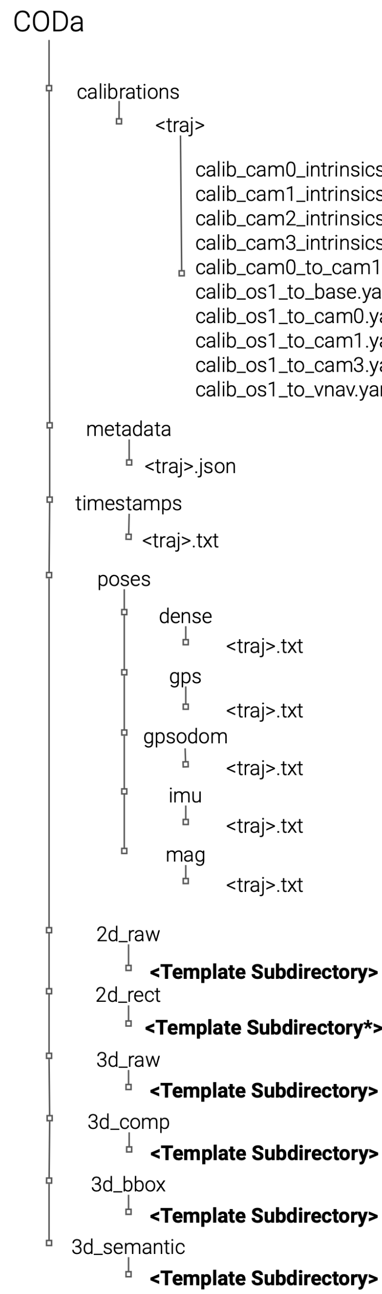


Fig. 12: Directory structure of CODa. All sensor and annotation directories use the subdirectory structure found in Fig. 11. The full dataset contains point cloud, image, inertial, sensor calibrations, timestamps, and annotations. More details can be found on the data report [1].

D. Model Training Experiments

In this section, we present all model experiments for the object detector architecture selection found in Table IV, AV to CODa domain adaptation in Table V, cross sensor resolution experiments in Table VI, and adaptation to JRDB in Table VII.

Architecture Selection Experiments. Object detection detection performance is highly dependent on the architecture and preprocessing hyperparameters. We limit our evaluation to PointPillars, CenterPoint, and PVRCNN for reasons discussed in Section VII. Table X summarizes the model configurations considered. Broadly speaking, we find that performance is highly dependent on the voxel size and points per voxel. While

Backbone	Head	VS	P/V	LR	OPT	Epochs	Train	FT	Eval	**AP$_{BEV}$**	**AP$_{3D}$**
PointPillars	AnchorMulti	0.1	5	3e-3	adam1cycle	30	Waymo	-	Waymo	21.86	16.78
PointPillars	AnchorSingle	0.32	20	3e-3	adam1cycle	30	Waymo	-	Waymo	**55.11**	**47.55**
PointPillars	AnchorMulti	0.1	5	1e-3	adam1cycle	30	nuScenes	-	nuScenes	**28.42**	**17.94**
PointPillars	AnchorMulti	0.2	5	1e-3	adam1cycle	30	nuScenes	-	nuScenes	24.14	14.28
PointPillars	AnchorMulti	0.1	5	3e-3	adam1cycle	80	KITTI	-	KITTI	27.99	25.03
PointPillars	AnchorSingle	0.1	32	3e-3	adam1cycle	80	KITTI	-	KITTI	**70.27**	**63.32**
PointPillars	AnchorSingle	0.16	32	3e-3	adam1cycle	80	KITTI	-	KITTI	70.94	64.49
PointPillars	AnchorMulti	0.1	5	3e-3	adam1cycle	50	CODa	-	CODa	15.96	14.75
PointPillars	AnchorSingle	0.1	5	3e-3	adam1cycle	50	CODa	-	CODa	**49.78**	**48.86**
CenterPoint-Pillar	CenterHead	0.1	5	3e-3	adam1cycle	50	Waymo	-	Waymo	39.38	31.67
CenterPoint-Voxel	CenterHead	0.1	5	3e-3	adam1cycle	30	Waymo	-	Waymo	**62.66**	**54.86**
CenterPoint-Pillar	CenterHead	0.1	10	1e-3	adam1cycle	50	nuScenes	-	nuScenes	21.82	12.08
CenterPoint-Voxel	CenterHead	0.1	10	3e-3	adam1cycle	50	nuScenes	-	nuScenes	29.17	18.79
CenterPoint-Voxel	CenterHead	0.075	10	1e-3	adam1cycle	30	nuScenes	-	nuScenes	**36.91**	**23.86**
CenterPoint-Pillar	CenterHead	0.1	5	3e-3	adam1cycle	80	KITTI	-	KITTI	**69.34**	**63.87**
CenterPoint-Voxel	CenterHead	0.1	5	3e-3	adam1cycle	80	KITTI	-	KITTI	66.83	60.15
CenterPoint-Pillar	CenterHead	0.1	5	1e-2	adam1cycle	50	CODa	-	CODa	61.78	52.46
CenterPoint-Voxel	CenterHead	0.075	10	3e-3	adam1cycle	50	CODa	-	CODa	**82.08**	**76.92**
PVRCNN	AnchorSingle	0.1	5	1e-2	adam1cycle	50	Waymo	-	Waymo	43.60	48.41
PVRCNN	CenterHead	0.1	5	1e-2	adam1cycle	30	Waymo	-	Waymo	**62.73**	**56.40**
PVRCNN	AnchorSingle	0.1	10	1e-2	adam1cycle	30	nuScenes	-	nuScenes	30.89	21.74
PVRCNN	CenterHead	0.1	5	1e-2	adam1cycle	50	nuScenes	-	nuScenes	**33.85**	**25.41**
PVRCNN	AnchorSingle	0.1	5	1e-2	adam1cycle	80	KITTI	-	KITTI	**70.22**	**65.28**
PVRCNN	CenterHead	0.1	20	1e-2	adam1cycle	50	CODa	-	CODa	**92.08**	**91.11**

TABLE X: Full model training summary for Table IV. We evaluate several 3D object detection architectures on AV Datasets and CODa. We separate the table columns by model architecture, data processing parameters, training hyperparameters, dataset, and performance. The column abbreviations are as follows: VS - voxel length and width, P/V - points per voxel, LR - learning rate, OPT - optimizer, FT - finetune dataset, Eval - evaluation dataset. We train the the PointPillars [46], Centerpoint [47], and PVRCNN [48] detectors on 20% of Waymo [12], all of nuScenes [13], all of KITTI [5], and the medium split of CODa. We report mean average precision for the car, pedestrian, and cyclist categories in bird's eye view (AP$_{BEV}$) and 3D (AP$_{3D}$) with IOU 0.7, 0.5, and 0.5 respectively. We average the results at the easy, medium, and hard difficulties. Results included in the main paper are bolded.

this is not true for all models, we find that a voxel size and points per voxel of 0.1 and 5 often improve performance. We do not perform a hyperparameter search over learning rates and optimizers for AV datasets because the OpenPCDet contributors already provide highly performant learning rates and optimizers. For CODa, we find that using the same learning rate as AV datasets offers good performance.

AV to CODa Experiments. We conduct the following experiments to optimize unsupervised and finetuning performance on CODa after pretraining on AV datasets. We tuned the hyperparameters for the ST3D (ST) and finetuning (FT) domain adaptation strategies. Section VII describes the experimental setup in detail. Table XI demonstrates that a high positive score threshold is beneficial for unsupervised domain adaptation. We speculate this is because a high positive threshold filters out more low-confidence detections, reducing the amount of erroneous pseudo-ground truth detections from ST3D. We perform a coarse hyperparameter sweep over the learning rate on the medium CODa split (CODa-md) and find that higher learning rates significantly improve finetuning performance.

JRDB Adaptation Experiments. We conduct the following experiments to quantitatively evaluate cross-dataset object detection performance on JRDB. We use the experiment setup and evaluation metrics described in Section VII, training a PVRCNN detector on Waymo, multiple LiDAR resolutions of the full CODa split, and JRDB before evaluating on JRDB. We empirically assess data augmentation effects using two sets: the default suite (random world flipping, rotation, and scaling) and the complete suite from previous cross-dataset object detection research [49] (random object scaling, rotation). Table XII presents these results, showing that using the full suite of data augmentation techniques benefits cross-dataset performance. Lastly, we quantify how sensor resolution differences affect cross-dataset detection performance in Table XII, showing that performance is maximized when CODa's resolution matches JRDB's LiDAR resolution (32 channels).

E. Annotation Ontology

CODa is annotated with object classes from Fig. 15 and terrain classes from Fig. 13. We include RGB examples of each terrain class in Fig. 14 by projecting the 3D point cloud annotation to the corresponding 2D image. For 2D examples of annotated objects, we refer the reader to the annotation instruction document in the data report [1].

Backbone	Head	DA	MV	+ST	VS	P/V	LR	Epochs	Train	FT	Eval	AP_{BEV}	AP_{3D}
PVRCNN	CenterHead	Direct	Y	-	0.1	5	1e-2	30	nuScenes	-	CODa-md	**21.30**	**15.53**
PVRCNN	CenterHead	Direct	Y	-	0.1	5	1e-2	30	Waymo	-	CODa-md	**46.20**	**43.11**
PVRCNN	AnchorSingle	ST	Y	0.4, 0.5, 0.5	0.1	5	1e-3	30	nuScenes	-	CODa-sm	7.65	5.39
PVRCNN	AnchorSingle	ST	Y	0.4, 0.3, 0.3	0.1	5	1e-2	30	nuScenes	-	CODa-sm	16.91	15.34
PVRCNN	CenterHead	ST	Y	0.4, 0.3, 0.3	0.1	5	1e-2	30	nuScenes	-	CODa-sm	23.20	19.02
PVRCNN	CenterHead	ST	Y	0.6, 0.55, 0.55	0.1	5	1e-2	30	nuScenes	-	CODa-sm	40.25	35.36
PVRCNN	CenterHead	ST	Y	0.6, 0.55, 0.55	0.1	5	1e-2	30	nuScenes	-	CODa-md	**14.07**	**10.76**
PVRCNN	CenterHead	ST	Y	0.6, 0.55, 0.55	0.1	5	1e-2	30	Waymo	-	CODa-md	41.55	37.32
PVRCNN	CenterHead	ST	Y	0.4, 0.3, 0.3	0.1	5	1.5e-3	30	Waymo	-	CODa-md	35.70	32.51
PVRCNN	CenterHead	ST	Y	0.6, 0.55, 0.55	0.1	5	1.5e-3	30	Waymo	-	CODa-md	**38.27**	**34.36**
PVRCNN	CenterHead	FT	Y	0.6, 0.55, 0.55	0.1	5	1e-2	30	nuScenes	CODa-md	CODa-md	**91.39**	**90.16**
PVRCNN	CenterHead	FT	Y	0.6, 0.55, 0.55	0.1	5	1e-4	30	Waymo	CODa-md	CODa-md	52.22	48.07
PVRCNN	CenterHead	FT	Y	0.6, 0.55, 0.55	0.1	5	1e-3	30	Waymo	CODa-md	CODa-md	57.51	56.01
PVRCNN	CenterHead	FT	Y	0.6, 0.55, 0.55	0.1	5	1e-2	30	Waymo	CODa-md	CODa-md	**93.12**	**92.07**
PVRCNN	CenterHead	ST+FT	Y	0.6, 0.55, 0.55	0.1	5	1.5e-3	15	nuScenes	CODa-md	CODa-md	**91.87**	**89.89**
PVRCNN	CenterHead	ST+FT	Y	0.6, 0.55, 0.55	0.1	5	1e-2	30	Waymo	CODa-md	CODa-md	**92.36**	**91.18**

TABLE XI: Full model training summary for Table V. Results reported in Table V are bolded. We train the PVRCNN [48] detector on 20% of Waymo [12] and all of nuScenes [13]. CODa is divided into small (25%) and medium (50%) splits of the full dataset. The column abbreviations are as follows: DA - domain adaptation method (refer to Section VII-B for definitions of these methods), MV - memory voting hyperparameter, +ST - positive score threshold hyperparameter, VS - voxel size, P/V - points per voxel LR - learning rate, FT - finetuning dataset, Eval - evaluation dataset. Results included in the main paper are bolded.

Range	CT	Augs	Train	Test	AP	AR	AF
15m	0.1	Def.	nuScenes	JRDB	**45.15**	**12.10**	**20.31**
25m	0.1	Def.	nuScenes	JRDB	**42.12**	**11.83**	**18.48**
15m	0.1	Def.	Waymo	JRDB	**55.39**	**18.7**	**27.96**
25m	0.1	Def.	Waymo	JRDB	**52.76**	**17.19**	**25.94**
15m	0.1	Def.	JRDB	JRDB	**65.64**	**27.13**	**38.39**
25m	0.1	Def.	JRDB	JRDB	**64.15**	**27.15**	**38.15**
15m	0.1	All	CODa-32	JRDB	**60.29**	**25.32**	**35.66**
15m	0.1	Def.	CODa-64	JRDB	57.30	23.56	33.39
25m	0.1	Def.	CODa-16	JRDB	52.22	23.57	32.49
25m	0.1	All	CODa-16	JRDB	54.32	25.32	34.54
25m	0.1	Def.	CODa-32	JRDB	55.42	21.89	31.39
25m	0.1	All	CODa-32	JRDB	**57.38**	**25.31**	**35.13**
25m	0.1	Def.	CODa-64	JRDB	56.49	20.26	29.83
25m	0.1	All	CODa-64	JRDB	56.27	20.25	29.78
25m	0.1	Def.	CODa-128	JRDB	54.31	18.69	27.81

TABLE XII: Full model training summary for Table VII. Results reported in Table VII are bolded. We train the PVRCNN [48] detector on 20% of Waymo [12] and all of CODa. For JRDB, we separate the train/validation splits for JRDB [7] into two datasets: one excluding labels farther than 15 meters from the ego-vehicle (15m) and the other with all labels within 25 meters (25m). We indicate the data augmentations used by default (Def.) and all (All). The confidence threshold (CT) is the threshold required to be considered a detection. We report the mean average precision (AP), recall (AR), and F1 score (AF) using an IOU of 0.3 on the JRDB validation split. Results included in the main paper are bolded.

Unstructured
Grass
Rocks
Dirt Paths
Short Vegetation

Tiling
Light Marble Tiling
Dark Marble Tiling
Blond Marble Tiling
Patterned Tile
Porcelain Tile

Road
Road Pavement
Crosswalk

Walkways
Concrete
Speedway
Bricks
Pebble
Pavement
Red Bricks
Wood Panel

Boundary
Dome Mat
Door Mat
Stairs
Threshold

Indoor Floor
Carpet

Metal
Metal Floor
Metal Grates

Other
Unlabeled
Unknown

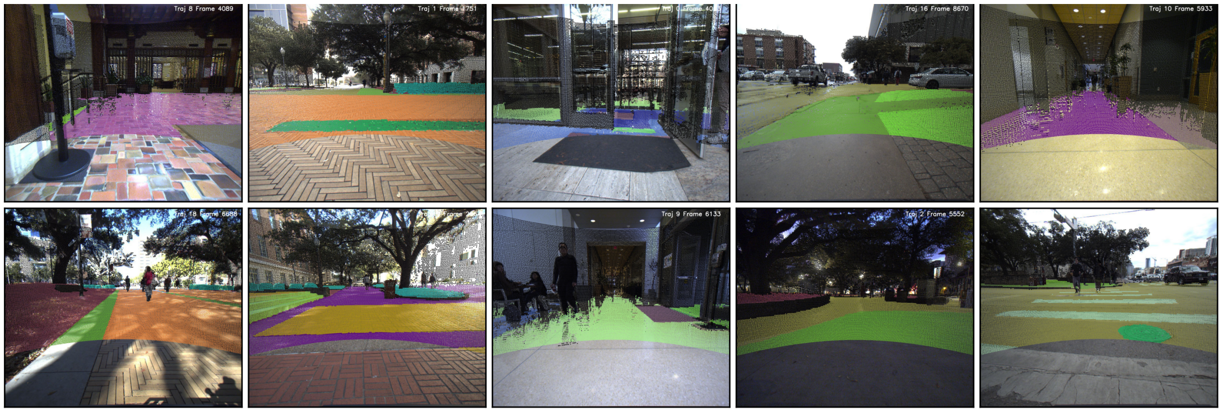
Fig. 13: Full 3D semantic segmentation class list. Bolding represents the topological category for the semantic classes below.

F. Qualitative Object Detection Results

We supplement the quantitative cross-dataset object detection evaluation in Table XII with qualitative results to vindicate our claim that pretraining on CODa provides better sparse detection performance. Fig. 16 contains examples of false negative, true positive, and false positive detections from our pre-trained models on JRDB. The first two examples demonstrate that detecting non-ground plane pedestrians and pedestrians on bicycles is challenging. This is because our pretraining datasets annotate pedestrians on cyclists differently than JRDB and do not contain examples of pedestrians below the ground plane. The last four examples show that CODa pre-training outperforms Waymo at detecting pedestrians that are sparse in LiDAR point cloud data.

G. Ground Truth Annotation Visualizations

We conclude with visualizations of the ground truth 3D bounding box annotations in CODa. Fig. 17 and Fig. 18



■	Grass	■	Pebble Pavement	■	Patterned Tile	■	Road Pavement
■	Rocks	■	Red Bricks	■	Porcelain Tile	■	Crosswalk
■	Dirt Paths	■	Wood Panel	■	Dome Mat	■	Metal Floor
■	Short Vegetation	■	Light Marble Tiling	■	Door Mat	■	Metal Grates
■	Concrete	■	Dark Mable Tiling	■	Stairs	■	Carpet
■	Speedway Bricks	■	Blond Marble Tiling	■	Threshold	■	Unlabeled

Fig. 14: Terrain segmentation colormap for CODa. Segmentation labels are verified on the 3D point clouds by human annotators and projected onto 2D images for visualization purposes.

showcase images from each annotated sequence in CODa. We characterize the dataset and annotation diversity in Fig. 6, Fig. 8, Fig. 7 and Fig. 9, Fig. 10. The data diversity, large semantic class list, and real-world nature of CODa make it a comprehensive dataset and benchmark for egocentric perception algorithms.

Vegetation	**Service Machine**	**Structure**
Tree	Parking Kiosk	Traffic Sign
Freestanding Plant	Mailbox	Traffic Light
	Newspaper Dispenser	Canopy
Person	Sanitizer Dispenser	Bike Rack
Pedestrian	Condiment Dispenser	Pole
	Water Dispenser	Room Label
Furniture/ Appliance	Vending Machine	Informational Sign
Couch	ATM	Floor Sign
Chair		Wall Sign
Bench	**Emergency Device**	Door
Table	Emergency Aid Kit	Door Switch
Computer	Emergency Phone	Horse
Television*	Fire Extinguisher	
Misc	Fire Hydrant	**Transportation**
	Fire Alarm	Scooter
Barrier		Motorcycle
Bollard	**Container**	Segway
Traffic Arm	Dumpster	Skateboard
Construction Barrier*	Trash Can	Bike
Fence	Cart	Car
Railing		Bus
Cone	**Misc**	Pickup Truck
Stanchion	Other	Utility Vehicle
		Service Vehicle
		Delivery Truck

Fig. 15: Full 3D bounding box object list. Bolding represents the topological category for the semantic classes below. Classes with an asterisk (*) do not have annotations in CODa.

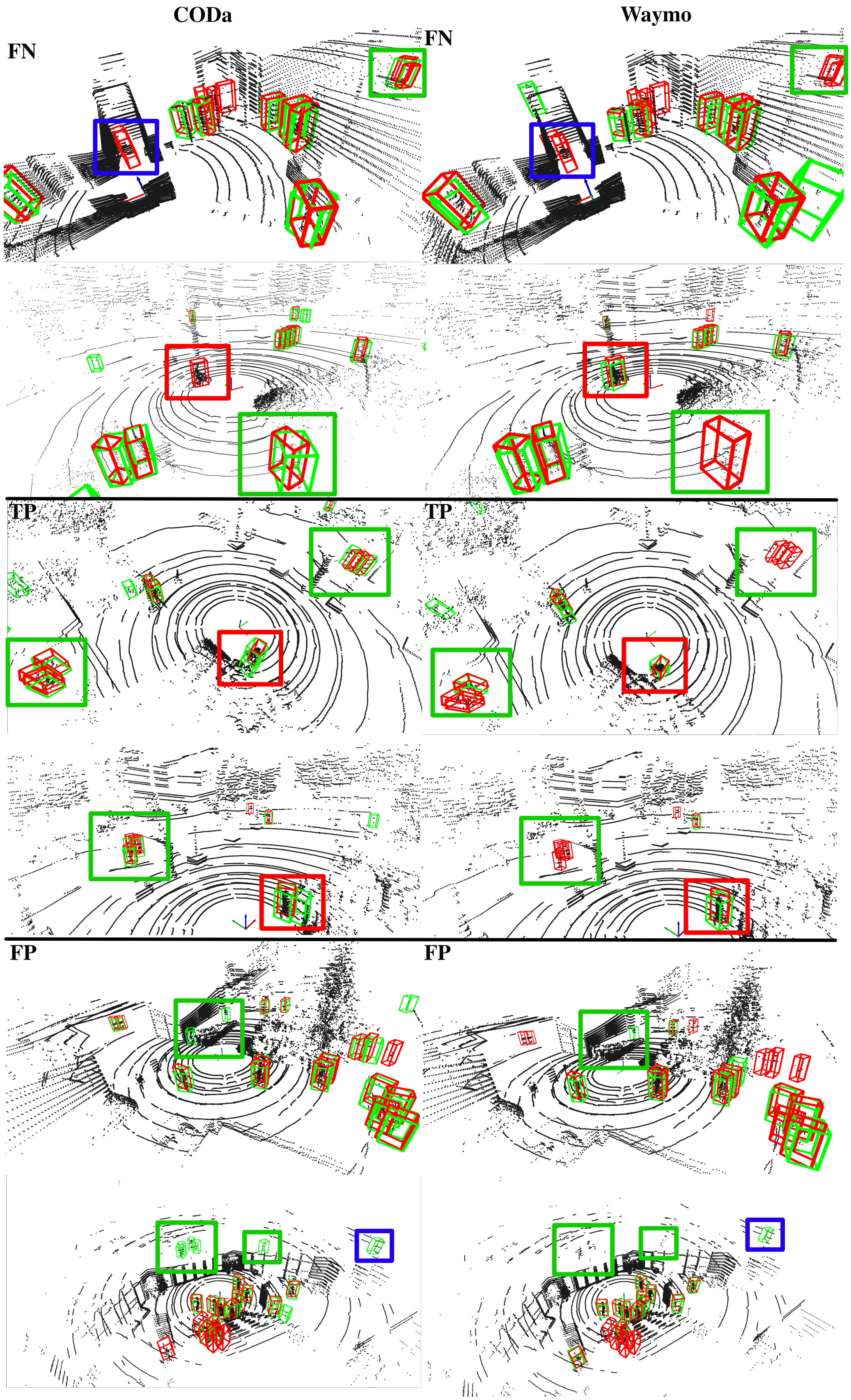


Fig. 16: Qualitative 3D object detection comparison on JRDB [7] after pretraining on CODa (left) and Waymo [12] (right). We provide examples for false negative (FN), true positive (TP), and false positive (FP) and show predictions in green and ground truth annotations in red. Areas where CODa models perform favorably are shown in green and areas where Waymo models perform favorably are shown in red. Areas where both detectors perform similarly are shown in blue.

Fig. 17: Scenes from all annotated sequences in CODa. Each row contains images from the same sequence. These sequences are collected from the perspective of an urban robot in indoor and outdoor environments: cafeterias, public workspaces, sidewalks, and libraries. We quantify the dataset's weather, lighting, and viewpoint diversity in Fig. 6 and Fig. 9. The data diversity, large semantic class list, and real-world nature of CODa make it a comprehensive dataset and benchmark for egocentric perception algorithms.

Fig. 18: Scenes from all annotated sequences in CODa (cont.)