

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

Arthur Zhang^{ID}, Graduate Student Member, IEEE, Chaitanya Eranki, Christina Zhang, Ji-Hwan Park^{ID}, Raymond Hong, Pranav Kalyani, Lochana Kalyanaraman^{ID}, Arsh Gamare, Arnav Bagad^{ID}, Maria Esteva^{ID}, and Joydeep Biswas^{ID}, Member, IEEE

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 h of multimodal sensor data from 3-D light detection and ranging (LiDAR), stereo RGB and rgbd and depth (RGBD) cameras, and a 9-DoF inertial measurement unit (IMU). CODa contains 58 min of ground truth annotations containing 1.3 million 3-D bounding boxes with instance identifiers (ID) for 53 semantic classes, 5000 frames of 3-D semantic annotations for urban terrain, and pseudoground 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) 3-D object detection performance improves in urban settings when trained using CODa compared with existing datasets, 2) sensor-specific fine-tuning increases 3-D object detection accuracy, and 3) pretraining on CODa improves cross-dataset 3-D object detection performance in urban settings compared with pretraining on AV datasets. We release benchmarks for 3-D object detection and 3-D semantic segmentation, with future plans for additional tasks. We publicly release CODa on the Texas Data Repository (Zhang et al., 2023), pretrained models, dataset development package, and interactive dataset viewer. 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, performance evaluation and benchmarking, segmentation and categorization, service robots.

I. INTRODUCTION

ACCURATE and robust perception of objects and scenes is crucial for autonomous mobile robots performing

Manuscript received 6 September 2023; revised 5 March 2024; accepted 19 March 2024. Date of publication 14 May 2024; date of current version 1 July 2024. This paper was recommended for publication by Associate Editor R. Tron and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported by NSF under Grant CAREER-2046955 and Grant IIS-1954778. (Chaitanya Eranki, Christina Zhang, Ji-Hwan Park, Raymond Hong, Pranav Kalyani, Lochana Kalyanaraman, and Arsh Gamare contributed equally to this work.) (Corresponding author: Arthur Zhang.)

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the University of Texas at Austin Institutional Review Board under Application No. STUDY00003493.

The authors are with the University of Texas, Austin, TX 78705 USA (e-mail: arthurz@cs.utexas.edu; chai77@cs.utexas.edu; yymzhang@cs.utexas.edu; jihwanpark98@utexas.edu; raymond22@cs.utexas.edu; pranavkalyani@utexas.edu; lochanak@cs.utexas.edu; arsh.gamare@utexas.edu; arnavb@cs.utexas.edu; maria@tacc.utexas.edu; joydeepb@cs.utexas.edu).

Interactive dataset viewer, development package, dataset, and pre-trained models available on the CODa website: <https://amrl.cs.utexas.edu/coda>

Digital Object Identifier 10.1109/TRO.2024.3400831

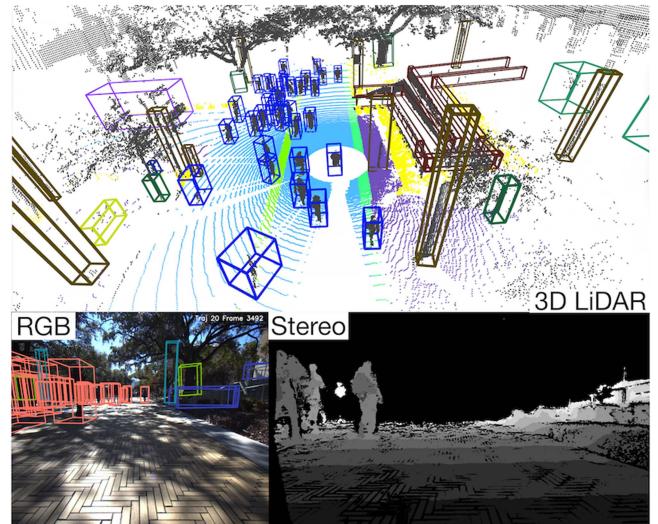


Fig. 1. Three of the five modalities available in CODa. RGB image with 3-D object labels (bottom left), 3-D point cloud (middle), and stereo depth image (bottom right).

tasks in urban environments. To this end, the computer vision and robotics communities have proposed datasets and benchmarks [2], [3], [4], [5], [6], [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 [7], [8], [9], which are important for navigation and planning tasks.

Many egocentric 3-D perception datasets are collected from urban robots or autonomous vehicles (AVs). Existing urban robotics datasets [5], [6], [10] in human-centric environments possess similar sensors and viewpoints but lack semantic annotation diversity. In contrast, AV datasets [4], [11], [12] 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 with urban robots. This

TABLE I
COMPARISON BETWEEN CODA (OURS) AND SIMILAR CAMPUS SCALE ROBOT DATASETS

Dataset	Pose	#Cls	#3-D Bbx	#3-D/2-D Seg	Inst. Labels	#3-D Ann. Frames	3-D Frames	3-D Pts/Frame	2-D 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

The most significant entry for each column in CODA is highlighted in blue. CODA provides the largest number of object classes, 3-D bounding box annotations, and annotated 3-D 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.

TABLE II
COMPARING DATASET STATISTICS BETWEEN CODA (OURS) AND EXISTING AV DATASETS

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

We use the following abbreviations: M—morning, A—afternoon, E—evening, G—GPS, R—RTK, and 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.

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 3-D 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, and low-light), and scene densities.

The sensor data includes

- 1) 3-D point clouds from a 128-channel 3-D LiDAR,
- 2) RGB images from a stereo camera pair synchronized with the 3-D 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 pseudoground truth global poses.

CODA contains 1.3 million ground truth 3-D bounding box annotations, instance IDs, and occlusion values for objects in the 3-D point cloud. Furthermore, it includes 5000 frames of 3-D terrain segmentation annotations for 3-D point clouds. All annotations are provided by human annotators, and labeled at 10 Hz for 3-D bounding boxes, and 2–10 Hz for terrain semantic segmentation. Compared with similar 3-D 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. We visualize these annotations and various sensor modalities in Fig. 1. Using our annotations, we release benchmarks using established metrics [4], [13] for 3-D object detection and 3-D semantic segmentation with plans for perception and planning tasks relevant to autonomous navigation.

The rest of this article is organized as follows. We review existing datasets and relate CODa to them in Section II, describe the sensor setup in Section III, data collection procedure in Section IV, annotation and labels in Section V, and analysis of coda annotations and statistics in Section VI. We characterize the semantic composition of our dataset, proposed train/validation/test splits, and provide qualitative sensor data visualizations. In Section VII we empirically analyze how: using CODa improves object detection performance for robots in urban settings, different 3-D LiDAR resolutions affect pre-trained object detector performance, and pretraining on CODa outperforms AV datasets in cross-dataset object detection on jackrabbot dataset and benchmark (JRDB) [6]. Section VIII presents the benchmarks. Finally, Section IX concludes this article.

II. RELATED WORK

In this section, we review existing egocentric 3-D LiDAR datasets for urban and AV domains and summarize these datasets in Tables I and II. We limit the discussion to real-world datasets, as there still exists a significant domain gap between simulation and real-world [35], [36].

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 simultaneous localization and mapping (SLAM), object detection, and semantic segmentation. While there exist computer vision benchmarks for 3-D object detection [2] and semantic segmentation [23], we focus on datasets collected from mobile robots due to differences in perspective shift and sensor suite.

Long-term SLAM datasets [10], [14], [20], [21], [24], [25] 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, such as BotanicGarden [26], contain 2-D semantic segmentation annotations and globally consistent poses in a static urban scene. Likewise, CODa contains 3-D semantic annotations and globally consistent pseudoground truth poses in dynamic scenes. This uniquely positions it for evaluating semantic and long-term SLAM methods across traversals in urban environments.

SCAND [5] 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 3-D object and terrain annotations. RUGD

[17] and Rellis-3D [18] are robot datasets with 2-D and 3-D 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 [6], a mobile robot dataset with 1) 1.8 million 3-D bounding box annotations, 2) indoor and outdoor sequences, and 3) egocentric sensor data. However, JRDB [6] 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 3-D perception algorithms for urban settings.

Among AV datasets, the Oxford RobotCar dataset [32] 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 2-D multitask learning problems, Berkeley DeepDrive [31] provides semantic annotations at both the object and pixel level for a wide range of semantic classes and weather conditions.

Lyft L5 [34], CityScapes3D [30], and KITTI-360 [37] contain labeled 2-D images or 3-D bounding boxes with more nonoverlapping 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, such as Waymo Open [11] and nuScenes [12], have fewer unique semantic classes but have more 3-D semantic annotations per class and greater scene diversity. These characteristics establish them as de facto benchmarks for 3-D object detection and semantic segmentation tasks, while also being valuable for pretraining 3-D object detectors to recognize similar objects across domains.

Other works like Argoverse2 [27] and ONCE [28] support self-supervised point cloud learning by providing more unannotated 3-D point clouds than any other AV dataset. In addition, both contain 3-D object labels, but Argoverse2's [27] labels are limited to 5 m within the drivable area and ONCE [28] is limited to five object classes. For robots operating in urban environments, it is important to identify a diverse set of objects in nondrivable areas, reinforcing the need for a dataset like CODa.

III. SENSOR SETUP

CODa was collected using a Clearpath Husky robot [38] equipped with a custom sensor suite with the following sensors, illustrated in Fig. 2.

- 1) 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

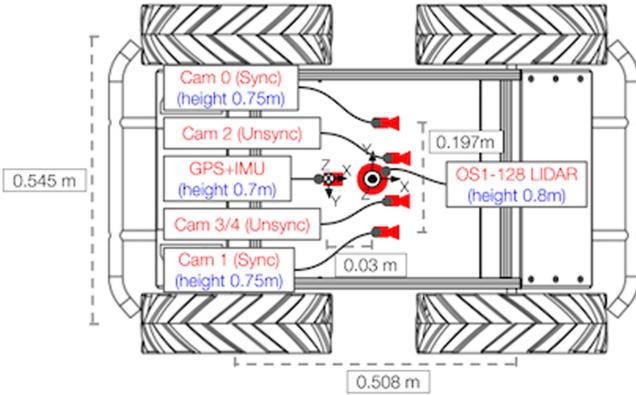


Fig. 2. Sensor setup including mounting positions. All heights are relative to the ground plane.

of view: 360° horizontal, 45° vertical, range: 128 m. Point clouds captured in 128 × 1024 channels @ 10 Hz.

- 2) 2 × Teledyne FLIR Blackfly S RGB cameras (BFS-U3-51S5C-C) up to 5 Megapixels, 75 Hz, global shutter. Paired with KOWA F2.8/5 mm lenses. Field of view (H × W): 70° × 79°. RGB images captured in 1.25 Megapixels @ 10 Hz, hardware synchronized with 3-D LiDAR.
- 3) 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
- 4) 1 × Stereolabs ZED 2i passive stereo camera up to 4 Megapixels @ 15 Hz, rolling shutter. RGB and depth images captured in 0.5 MP @ 5Hz
- 5) 1 × Vectorsnav VN-310 Dual GNSS/INS, up to 800 Hz IMU Data. Inertial and GPS data captured @ 20 Hz.

An onboard computer with an Intel i7-8700 3.2 GHz CPU and 32 GB RAM is securely mounted inside the robot and records all sensor streams to a high-speed Intel 760P SSDPEKKW512G8 512 GB 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 teledyne forward-looking infrared (FLIR) cameras are synchronized by hardware using the Ouster LiDAR 10 Hz 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.

IV. DATA COLLECTION PROCEDURE

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

A. Per Sequence Calibration Procedure

Each sequence in CODa includes a pre- or postrun 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

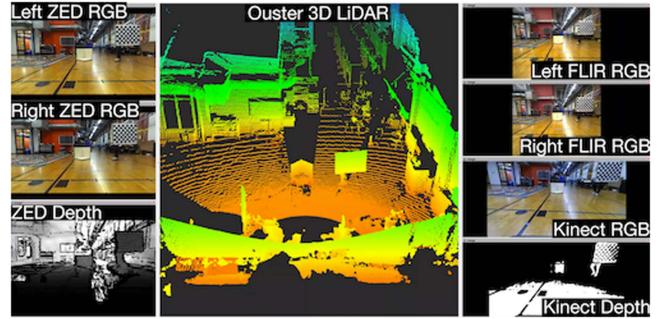


Fig. 3. Sample frame from calibration file. Calibration half-cube and checkerboard are simultaneously visible in all RGB, depth, and 3-D LiDAR frames.

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.

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 opted 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 [39]. 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 speedway (SWY), gates dell complex (GDC), william c. powers (WCP), guadalupe (GUAD), and union building (UNB). We choose each region for the following attributes: SWY has a large open area shared

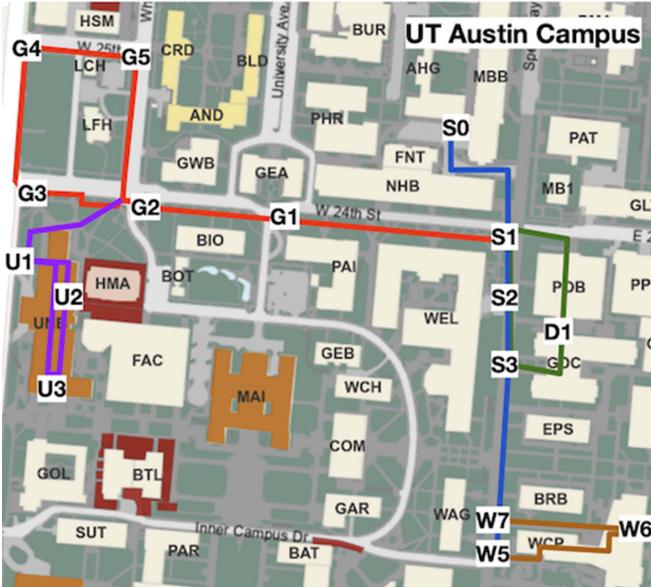


Fig. 4. Spatial map of the five geographic regions in CODa. Each region spans the area surrounding its respective colored line. Operators pause the robot at each waypoint denoted on the map to correct global pseudoground truth pose estimates. We refer to the regions surrounding the blue, green, brown,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

TABLE III
SUMMARY OF THE FOUR ROUTES IN CODA

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
WCOPowers	Both	WCP, SWY	7	8005	2.79
Union	In	UNB, Guad	3	2450	0.93

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.

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.¹ 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, and rainy) during three times of day (morning, afternoon, and 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 three rainy, seven cloudy, four dark, and nine sunny sequences. Fig. 17



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.

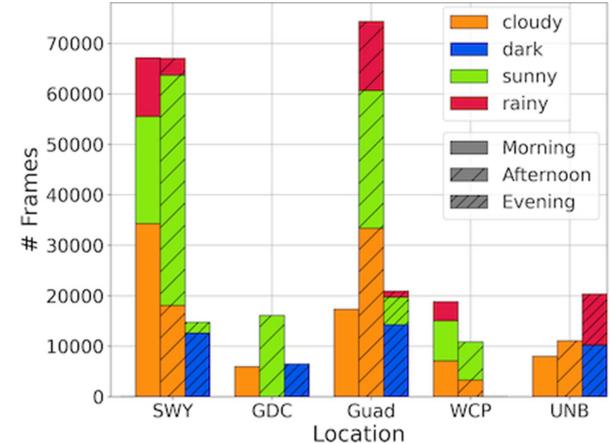


Fig. 6. Number of frames in CODa by geographic region and weather condition. Regions with temporally diverse observations contain frames during multiple times of the day. The coverage areas for regions SWY, GDC, WCP, Guad, and UNB are marked by blue, green, brown, red, and purple lines respectively in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

qualitatively showcases the data diversity in CODa using sampled images from each sequence.

V. ANNOTATION AND LABELS

We utilized Deepen AI,² a third party annotation company, to annotate point clouds from our 3-D LiDAR with 3-D bounding box and semantic segmentation annotations. We instructed

¹The official CODa data report is available at <https://doi.org/10.18738/T8/BQOQMV>, Texas Data Repository, V2; baa_DATA_REPORT.pdf [fileName].

²Company Website (Deepen): <https://www.deepen.ai/>

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 Figs. 15 and 13, quantitative occlusion level definitions, and operating procedures to determine object instance IDs. Following these instructions, Deepen annotators manually labeled 58 min 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 3-D point clouds. Our internal team then inspected each frame for additional issues. We now describe each annotation type in CODa in detail.

A. 3-D Bounding Boxes

Each 3-D bounding box has 9-DoF, 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 s. 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 3-D 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 3-D 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.

B. 3-D 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 3-D semantic segmentation challenging with just a single 3-D LiDAR and encourages multimodal methods that fuse 2-D images and 3-D 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 Figs. 13 and 14, respectively.

C. Pseudoground Truth Poses

Due to the unreliability of GPS in urban environments, we use lightweight and ground-optimized lidar odometry and mapping (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 1 m. 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

1 m 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 3-D 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 automated teller machine (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 [12], [28] 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 [13], [30] as well.

We propose train, validation, and test splits for our 3-D object detection and 3-D 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 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.

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

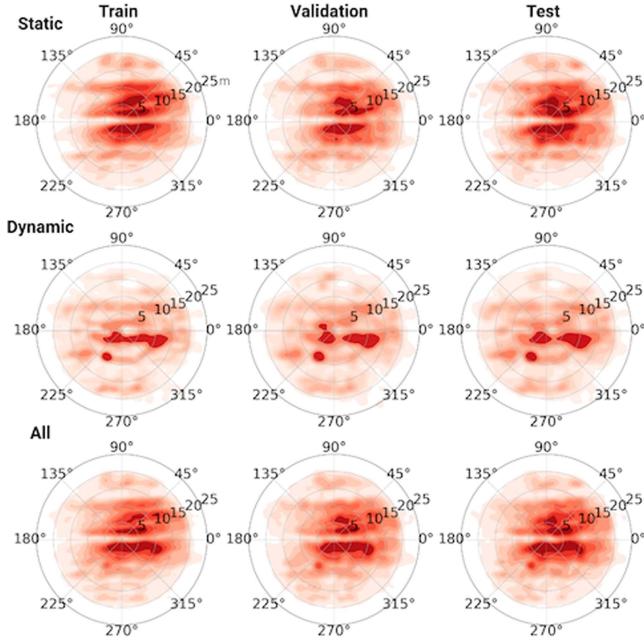


Fig. 7. Spatial distribution of static (top), dynamic (middle), and all (bottom) objects around the robot for the train (left), validation (center), and test (right) splits. Angles (in degrees) are with respect to the forward heading of the robot, range values in meters.

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

We choose a 3-D object detector by evaluating the performance of three 3-D object detectors: PointPillars [46], CenterPoint [47], pointvoxel RCNN (PVRCNN) [48] on KITTI, Waymo, nuScenes, and CODa. These datasets are among the most widely used in 3-D object detection benchmarks and for cross-dataset domain adaptation analysis [28], [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–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 3-D object detection and bird's eye view evaluation metric proposed in the KITTI Vision Benchmark Suite [5] with an intersection over union (IOU) of 0.7, 0.5, and 0.5 for the car, pedestrian, and cyclist classes, respectively.

TABLE IV

EVALUATION OF SEVERAL 3-D 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

Data. Mod.	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

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 3-D detection for each dataset. Mod.—Model and Data.—dataset.

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 PT \	Direct	ST	FT	ST + FT
nuScenes [12]	21.30 15.53	14.07 10.76	91.39 90.16	92.38 91.02
Waymo [11]	46.20 43.11	38.27 34.36	93.12 92.07	92.36 91.18
CODa	92.08 91.11	-	-	-

DA methods include: 1) Direct—Train on the Source Dataset and Evaluate Directly on CODa; 2) ST3D++ ST [51]—for unsupervised adaptation; 3) FT—with CODa after pretraining on the source dataset; and 4) both ST3D++ and FT (ST + FT). The results demonstrate that even state-of-the-art unsupervised domain adaptation methods for 3-D object detectors are not competitive with approaches that use domain-specific training labels. All models are evaluated on the medium test split of CODa.

TABLE VI

EVALUATING THE IMPACT OF POINT CLOUD RESOLUTION DIFFERENCES BETWEEN THE SOURCE AND TARGET DOMAIN ON 3D OBJECT DETECTOR PERFORMANCE

Test Train \	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

All experiments are conducted with a PV-RCNN detector first pretrained on Waymo using the pedestrian, car, and cyclist classes. We fine-tune the pretrained 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.

TABLE VII

CROSS-DATASET 3D OBJECT DETECTION PERFORMANCE COMPARISON ON JRDB [6] AFTER TRAINING ON CODA, NUSCENES [12], AND WAYMO [11]

Test Train \	JRDB (15m)			JRDB (25m)		
	Prec.	Rec.	F1	Prec.	Rec.	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

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 and 25 m of the ego vehicle. We report the performance difference between the highest performing AV and CODa models in red and blue superscripts.

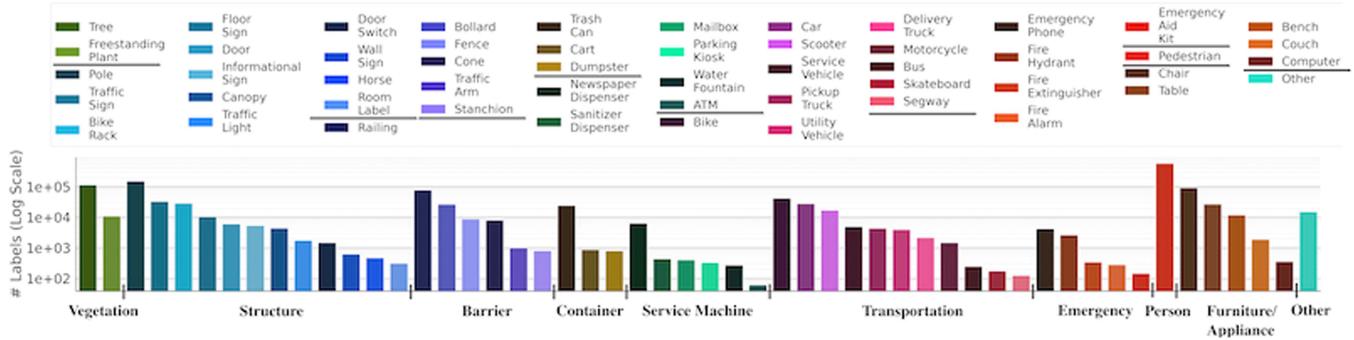


Fig. 8. Number of object labels per class organized by topological category. Objects in each topological category are sorted in order of most to least common.

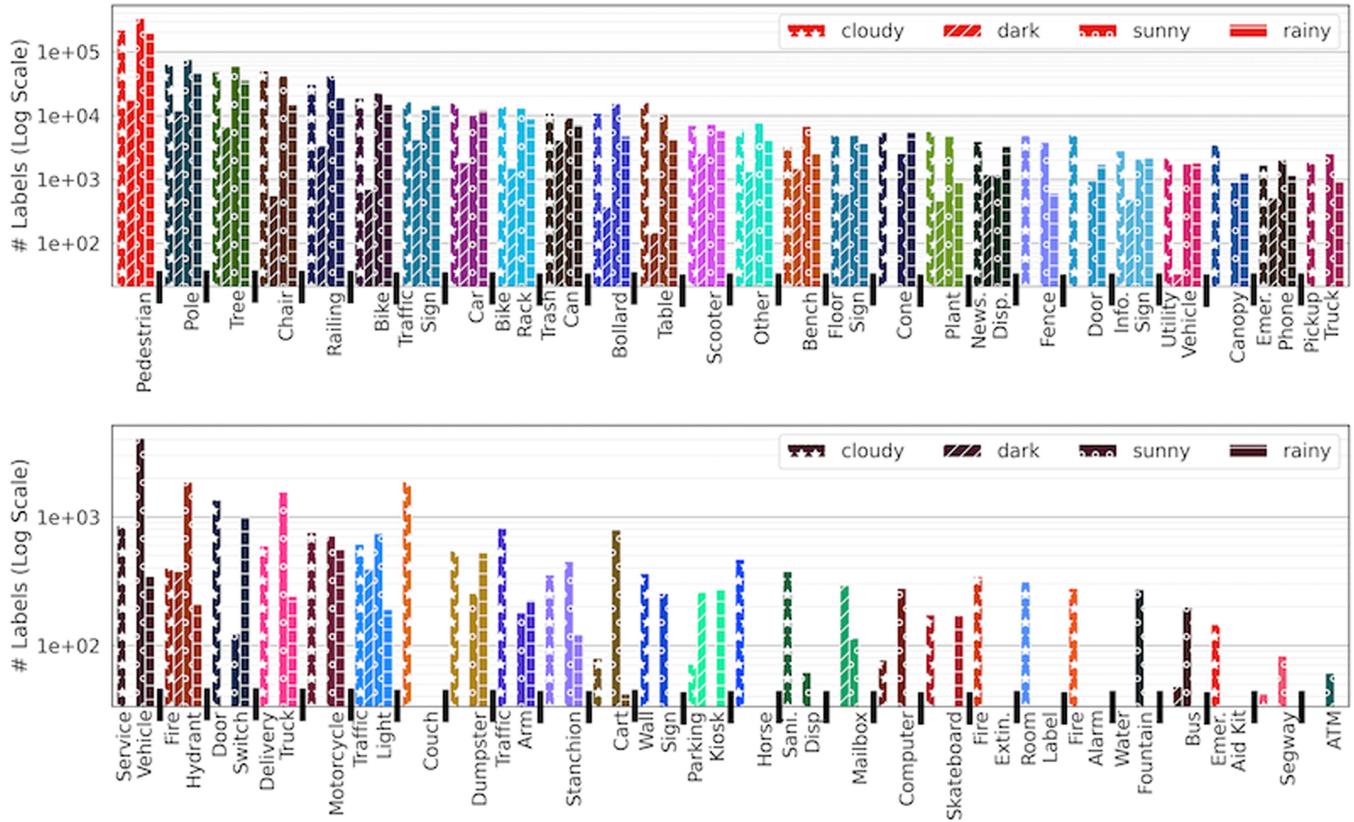


Fig. 9. Histogram of the number of annotations per object class under four weather conditions (sunny, rainy, cloudy, and dark). Object classes are organized by most to least frequent from left to right. Bars with stars are cloudy, diagonal lines are dark, circles are sunny, and horizontal lines are rainy.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

This class list is consistent across CODa and AV datasets. For completeness, we report model performance on the full list of object classes for multiple LiDAR resolutions in the Appendix.

We observe in Table IV that PVRCNN generally performs the best for 3-D 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 A-D.

B. AV Dataset to CODa Adaptation

We apply several domain adaptation strategies to evaluate 3-D 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 3-D data augmentation techniques (scaling, rotation, and 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 pretrained model is evaluated 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

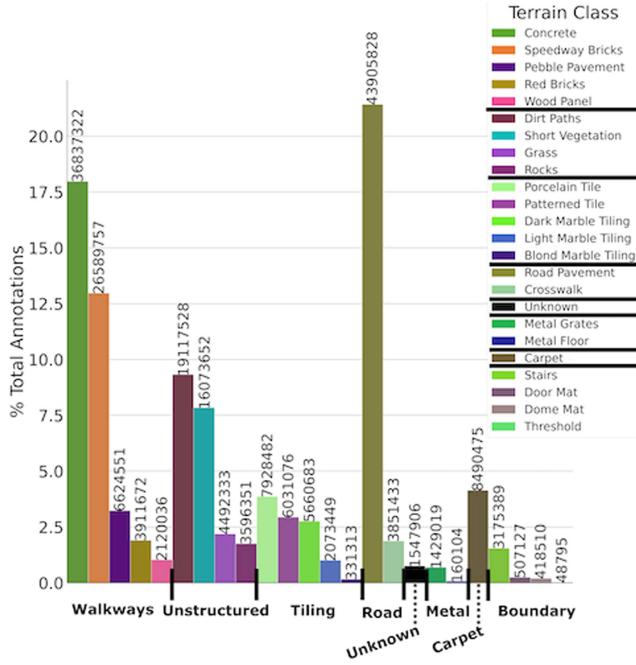


Fig. 10. Histogram of 3-D semantic segmentation annotations labels for outdoor, indoor, and both environments in CODa. Vertical numbers above each bar indicate the total number of points annotated for that semantic class. The semantic classes in the legend map to the bars from left to right.

data are 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 A-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 FT: We assume that domain-specific ground truth labels are available. We pretrain the model backbone on nuScenes or Waymo before FT on the train split of CODa, hypothesizing that learning features on other datasets benefit domain-specific performance.

We pretrain PVRCNN from scratch on each AV dataset for 30 epochs and evaluate the model on the CODa test split. After pretraining, 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 Table X show that a learning rate of 0.01 and the Adam 1cycle optimizer [52] provide the best empirical performance.

ST3D++ with domain specific FT (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% 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 pretrained 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 pseudolabels on CODa and support this hypothesis in Section VII-C.

Our experiments show that pretraining on Waymo improves performance on 3-D bounding box and BEV tasks by about 1%-2% compared with the 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 pretraining. We conclude from these studies that downstream tasks like 3-D object detection benefit from better initial 3-D 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

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 PVRCNN 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 FT, 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 3-D object detectors trained on a specific sensor resolution perform best on the same sensor resolution during test time. Furthermore,

TABLE VIII
EVALUATION OF TWO 3-D SEMANTIC SEGMENTATION MODELS ON THE FULL CODA TEST SPLIT

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	

We report mean intersection over union and accuracy for each semantic class. Bold numbers indicate the highest-performing method for each category.

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 pretrained models on all classes in CODa for the (16, 32, 64, and 128) channel LiDAR resolutions and encourage users to select the pretrained model that is most similar to the target dataset’s resolution. Table IX in Appendix A reproduces this experiment for all classes in CODa.

D. JRDB Adaptation

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 3-D object detectors trained on CODa and AV datasets on JRDB, a large-scale urban robot dataset with LiDAR point clouds and 3-D 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 3-D 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 m of the ego vehicle and the other within 25 m of the ego vehicle.

JRDB performance discussion: Table VII tabulates that CODa models consistently outperform AV models in all metrics for both the 15 and 25 m range. Furthermore, pretraining on CODa offers similar performance to training with labels on JRDB, corroborating our claim that pretrained CODa models generalize to other urban settings. We believe this can be 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 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 3-D object detection performance.

VIII. BENCHMARKS

In this section, we define the 3-D object detection and 3-D 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. 3-D Object Detection

The 3-D object detection task involves predicting 7-DoF boxes for all object classes. We use the 3-D 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 ten prior LiDAR frames for predictions. All sensor modalities and pseudoground truth poses can be used, and we will evaluate all predictions on the 3-D point cloud annotations.

B. 3-D Semantic Segmentation

For the 3-D semantic segmentation benchmark, we use the same evaluation metric proposed in SemanticKITTI [13]. This is the mean intersection-over-union (mIoU) metric [2] over all classes. All sensor modalities can be used, but we will evaluate all predictions using the 3-D 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.

TABLE IX

EVALUATING THE IMPACT OF POINT CLOUD RESOLUTION DIFFERENCES
BETWEEN THE SOURCE AND TARGET DOMAIN ON 3-D OBJECT DETECTOR
PERFORMANCE

Train \ Test	CODa-16	CODa-32	CODa-64	CODa-128
CODa-16	23.36 21.15	19.39	17.03	9.65
CODa-32	14.52	12.23	26.19 23.86	23.38
CODa-64	5.70	4.93	20.49	18.29
CODa-128	3.14	2.53	12.85	11.31
				25.40
				23.34
				28.18 26.14

All experiments are conducted with a PV-RCNN [48] detector first pretrained on Waymo [12]. Using the pedestrian, car, and cyclist classes. We fine-tune the pretrained 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.

IX. CONCLUSION AND FUTURE WORK

In this work, we presented the UT CODa, a multimodal dataset that contains greater object and scene-level annotation diversity than any other similar existing dataset. CODa contains 1.3 million human-annotated 3-D bounding boxes and 5000 semantic segmentation annotations over 8.5 h of data collected from the perspective of a mobile robot across UT campus. We publicly release CODa on the Texas Data Repository [39], pretrained models for various LiDAR resolutions (16, 32, 64, and 128 channels), and dataset development package. We refer readers to the abstract for accessing these resources.

We conducted extensive experiments to select a high-performing model architecture for urban environments. We demonstrated a performance gap for 3-D 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 3-D object detection performance is significantly affected by differences in LiDAR sensor resolution during test time. Finally, we conducted various ablation studies to show that pretraining 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 3-D object detector invariance to point cloud density and highlights the importance of selecting a pretrained model that closely resembles the target domain during robot deployments. We expect that this work will spur future research toward learning sensor-invariant 3-D 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.

APPENDIX

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

<Template Subdirectory>

```

<sensor>
  ↓
  <traj>
    ↓
    {parent_dir}_<sensor>_<traj>_<frame>.{fileext}
  
```

Replace {parent_dir} with the parent directory name containing the template directory. Ex: 2d_raw

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

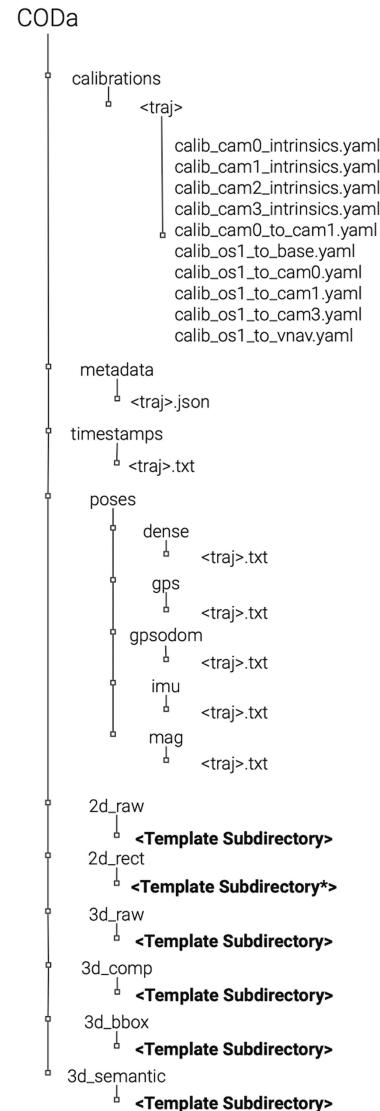


Fig. 12. Directory structure of CODa. All sensor and annotation directories use the subdirectory structure found in Fig. 11. The full dataset contains point cloud, image, inertial, sensor calibrations, timestamps, and annotations. More details can be found on the data report [1].

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 pretrained model with the closest LiDAR resolution to their target domain for optimal performance.

TABLE X
FULL MODEL TRAINING SUMMARY FOR TABLE IV

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

We evaluate several 3-D 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 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 3-D (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 bold.

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

Indoor Floor	Carpet
Metal	
	Metal Floor
	Metal Grates

Boundary	
	Dome Mat
	Door Mat
	Stairs
	Threshold

detailed breakdown of the file contents. The primary nonsensor subdirectories in CODa contain sensor calibrations, metadata, timestamps, and poses. We provide these files for each sequence. The remaining directories contain 2-D/3-D 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.

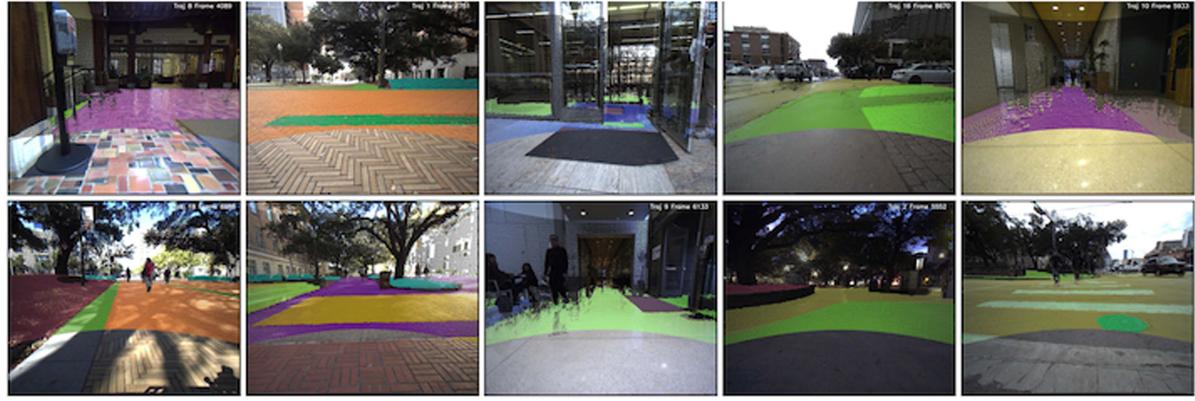
D. Model Training Experiments

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.

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

B. CODa Organization Structure

We describe the organization structure of CODa at a high level in Fig. 12 and refer the reader to the data report [1] for a



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 Marble Tiling	Stairs	Carpet
Speedway Bricks	Blond Marble Tiling	Threshold	Unlabeled

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

TABLE XI
FULL MODEL TRAINING SUMMARY FOR TABLE V

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

Results reported in Table V are in bold. 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, and Eval—evaluation dataset. Results included in the main paper are in bold.

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 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

TABLE XII
FULL MODEL TRAINING SUMMARY FOR TABLE VII

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

Results reported in Table VII are in bold. 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 m from the ego-vehicle (15 m) and the other with all labels within 25 m (25 m). 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 in bold.

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 FT performance on CODa after pretraining on AV datasets. We tuned the hyperparameters for the ST3D (ST) and 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 pseudoground 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 FT 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

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

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

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 3-D point cloud annotation to the corresponding 2-D image. For 2-D examples of annotated objects, we refer the reader to the annotation instruction document in the data report [1].

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 (FN), true positive (TP), and false positive (Fp) detections from our pretrained models on JRDB. The first two examples demonstrate that detecting nonground 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.

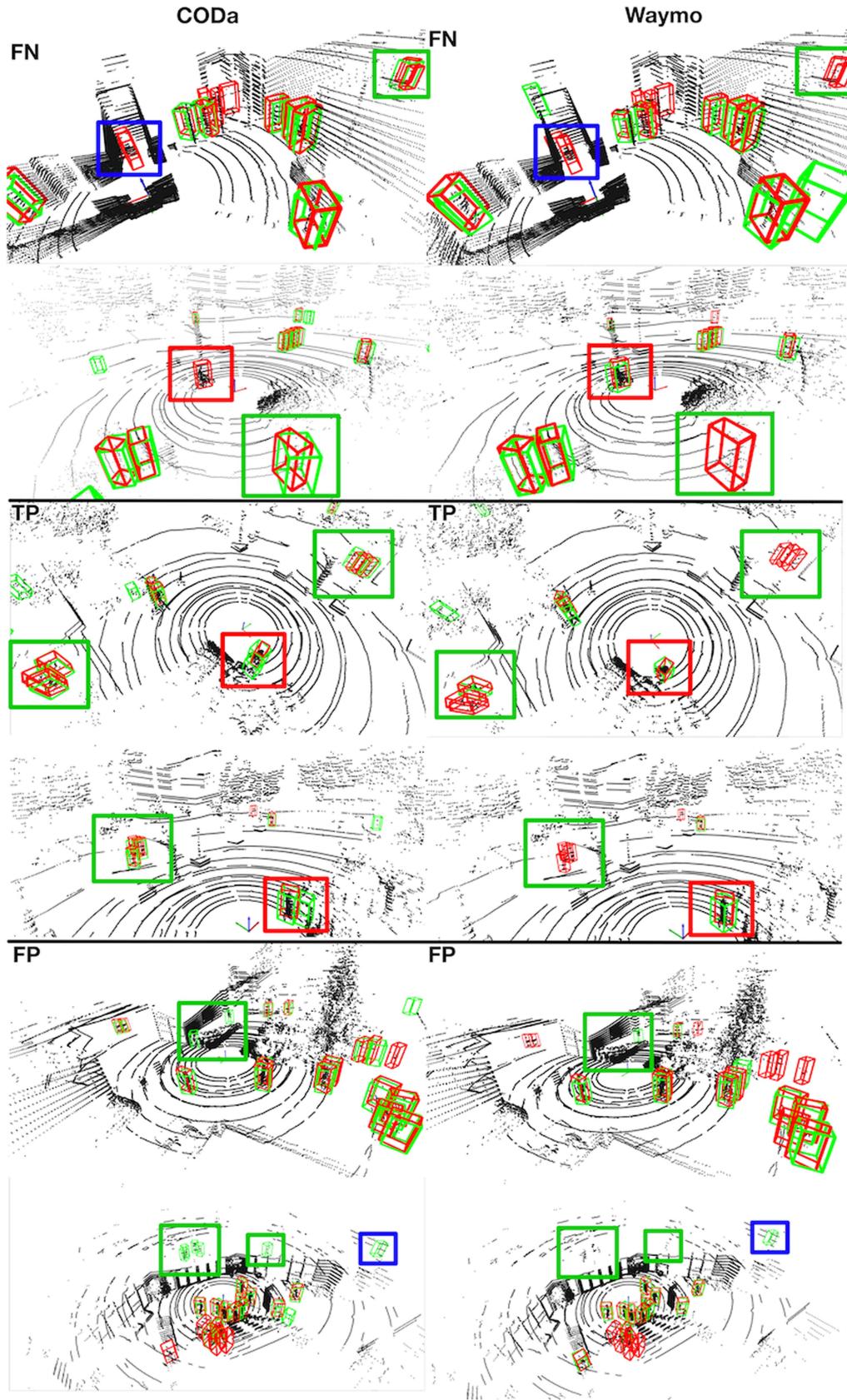


Fig. 16. Qualitative 3-D object detection comparison on JRDB [7] after pretraining on CODa (left) and Waymo [12] (right). We provide examples for FN, TP, and FP and show predictions in green and ground truth annotations in red. Areas where CODa models perform favorably are shown in green and areas where 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 Figs. 6 and 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.).

The last four examples show that CODa pretraining 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 3-D bounding box annotations in CODa. Figs. 17 and 18 showcase images from each annotated sequence in CODa. We characterize the dataset and annotation diversity in Figs. 6–10. The data diversity, large semantic class list, and real-world nature of CODa make it a comprehensive dataset and benchmark for egocentric perception algorithms.

ACKNOWLEDGMENT

This work was conducted with the Autonomous Mobile Robotics Laboratory (AMRL) at UT Austin. The authors 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)³ at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this article. The CODa study is covered under the University of Texas at Austin IRB Number STUDY00003493.

³TACC website: <http://www.tacc.utexas.edu>

REFERENCES

- [1] A. Zhang et al., "UT campus object dataset (CODa)" 2023. Accessed: Jan. 1, 2024. [Online]. Available: <https://doi.org/10.18738/T8/BBOQMV>
- [2] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Compu. Vis. ECCV 2014*, 2014, pp. 740–755. [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. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: [10.1007/s11263-009-0275-4](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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
- [5] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.
- [6] H. Karman et al., "Socially compliant navigation dataset (SCAND): A large-scale dataset of demonstrations for social navigation," in *Proc. IEEE Robot. Automat. Lett.*, vol. 7, no. 4, 2022, pp. 11807–11814.
- [7] R. Martín-Martín et al., "JRDB: A dataset and benchmark for visual perception for navigation in human environments," 2019, *arXiv:1910.11792*.
- [8] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9404–9413.
- [9] T. Guan, D. Kothandaraman, R. Chandra, and D. Manocha, "Ganav: Group-wise attention network for classifying navigable regions in unstructured outdoor environments," 2021, *arXiv: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 *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 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," *Int. J. Robot. Res.*, vol. 35, no. 9, pp. 1023–1035, 2015.
- [12] P. Sun et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2446–2454.
- [13] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. CVPR*, 2020, pp. 11618–11628.
- [14] J. Behley et al., "A dataset for semantic segmentation of point cloud sequences," 2019, *arXiv:1904.01416*.
- [15] M. Fallon, H. Johannsson, M. Kaess, and J. Leonard, "The mit stata center dataset," *Int. J. Robot. Res.*, vol. 32, pp. 1695–1699, 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. Int. Conf. Intell. Robot Syst.*, 2012, pp. 573–580.
- [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," 2020, *arXiv: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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 5000–5007.
- [19] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "RELLIS-3D dataset: Data, benchmarks and analysis," in *Proc. IEEE Int. Conf. Robot. Autom.* 2020, pp. 1110–1116.
- [20] P. Yin et al., "Alita: A large-scale incremental dataset for long-term autonomy," 2022.
- [21] J. Jiao et al., "Fusionportable: A multi-sensor campus-scene dataset for evaluation of localization and mapping accuracy on diverse platforms," 2022.
- [22] X. Shi et al., "Are we ready for service robots? the OpenLORIS-Scene datasets for lifelong SLAM," in *Proc. Int. Conf. Robot. Autom.*, 2020, pp. 3139–3145.
- [23] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond PASCAL: A benchmark for 3 D object detection in the wild," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2014, pp. 75–82.
- [24] P. K. Nathan Silberman, D. Hoiem, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. ECCV*, 2012, pp. 746–760.
- [25] J. Yin, A. Li, T. Li, W. Yu, and D. Zou, "M2DGR: A multi-sensor and multi-scenario SLAM dataset for ground robots," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2266–2273, 2021.
- [26] D. Feng et al., "S3e: A large-scale multimodal dataset for collaborative SLAM," 2023.
- [27] Y. Liu et al., "Botanicgarden: A high-quality dataset for robot navigation in unstructured natural environments," *IEEE Robot. Automat. Lett.*, 2024.
- [28] B. Wilson et al., "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Proc. Neural Inf. Process. Syst. Track Datasets Benchmarks*, 2021.
- [29] J. Mao et al., "One million scenes for autonomous driving: ONCE dataset," 2021, *arXiv:2106.11037*.
- [30] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2 D and 3 D," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3292–3310 2022.
- [31] N. G. hlert, N. Jourdan, M. Cordts, U. Franke, and J. Denzler, "Cityscapes 3 D: Dataset and benchmark for 9 DoF vehicle detection," 2020, *arXiv:2006.07864*.
- [32] F. Yu et al., "BDD100 K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2636–2645.
- [33] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The Oxford RobotCar dataset," *Int. J. Robot. Res. (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017, doi: [10.1177/0278364916679498](https://doi.org/10.1177/0278364916679498).
- [34] X. Song et al., "ApolloCar3D: A large 3D car instance understanding benchmark for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5452–5462.
- [35] J. Houston et al., "One thousand and one hours: Self-driving motion prediction dataset," in *Proc. Conf. Robot. Learn.*, 2020, pp. 409–418.
- [36] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "Fcns in the wild: Pixel-level adversarial and constraint-based adaptation," 2016.
- [37] Y. H. Tsai, W. C. Hung, S. Schulter, K. Sohn, M. H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7472–7481.
- [38] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2 D and 3 D," *Pattern Anal. Mach. Intell.*, 2022.
- [39] C. Robotics, Online, retrieved Jul. 22, 2023.
- [40] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.
- [41] D. Tsai, S. Worrall, M. Shan, A. Lohr, and E. Nebot, "Optimising the selection of samples for robust LiDAR camera calibration," in *Proc. IEEE Int. Intell. Transp. Syst. Conf.*, 2021, pp. 2631–2638.
- [42] S. Mishra, G. Pandey, and S. Saripalli, "Motion based extrinsic calibration of a 3 D LiDAR and an IMU," in *Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst.*, Sep. 2021, pp. 1–7.
- [43] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4758–4765.
- [44] K. Koide, J. Miura, M. Yokozuka, S. Oishi, and A. Banno, "Interactive 3 D graph SLAM for map correction," *IEEE Robot. Autom. Lett.*, vol. 6, no. 1, pp. 40–47, Jan. 2020.
- [45] S. Nashed and J. Biswas, "Human-in-the-loop SLAM," *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, Apr. 2018.
- [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 3 D object detection and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11784–11793.
- [48] S. Shi et al., "PV-RCNN: Point-voxel feature set abstraction for 3 D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* 2020, pp. 10529–10538.
- [49] Y. Wang et al., "Train in Germany, test in the USA: Making 3 D object detectors generalize," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* 2020, pp. 11713–11723.
- [50] O. D. Team, "Openpcdet: An open-source toolbox for 3 D object detection from point clouds," 2020. Accessed: Aug. 30, 2023. [Online]. Available: <https://github.com/open-mmlab/OpenPCDet>, 2020.
- [51] J. Yang, S. Shi, Z. Wang, H. Li, and X. Qi, "Si3D : Denoised self-training for unsupervised domain adaptation on 3 D 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," 2018, *arXiv:1803.09820*.
- [53] H. Zhou et al., "Cylinder3D: An effective 3 D framework for driving-scene LiDAR semantic segmentation," 2020, *arXiv:2008.01550*.
- [54] X. Yan et al., "2Dpass: 2 D priors assisted semantic segmentation on LiDAR point clouds," in *Proc. Eur. Conf. Comput. Vis.*. Cham: Springer Nature Switzerland, 2022, pp. 677–695.



Arthur Zhang (Graduate Student Member, IEEE) received the bachelor's degree in computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2022. He is currently working toward the Ph.D. degree in computer science with the Autonomous Mobile Robotics Laboratory, The University of Texas at Austin, Austin, TX, USA.

His research interests include robot perception, visual representation learning for navigation, and 3-D computer vision.



Lochana Kalyanaraman is currently working toward the undergraduate degree in computer science with The University of Texas at Austin, Austin, TX, USA.

She is working with the Autonomous Mobile Robotics Laboratory, The University of Texas at Austin, as part of the Perception Autonomy EcoCar Team. Her research interests include data science and Big Data.



Chaitanya Eranki is currently working toward the undergraduate degree in computer science with The University of Texas at Austin, Austin, TX, USA.

He is currently with the Autonomous Mobile Robotics Laboratory, The University of Texas at Austin. His research interests include machine learning and software engineering.



Arsh Gamare is currently working toward the undergraduate degree in electrical and computer engineering with The University of Texas at Austin, Austin, TX, USA.

He is a Member of the Autonomous Mobile Robotics Laboratory, The University of Texas at Austin. His research interests include 3-D computer vision and large language models.



Christina Zhang is currently working toward the undergraduate degree in computer science and mathematics with the University of Texas at Austin, Austin, TX, USA.

She was with the Autonomous Mobile Robotics Laboratory, The University of Texas at Austin. Her research focuses on software engineering.



Arnav Bagad is currently working toward the undergraduate degree in computer science in Turing Scholars program with The University of Texas at Austin, Austin, TX, USA.

He is a Member of the Autonomous Mobile Robotics Laboratory, The University of Texas at Austin. His research interests include 3-D computer vision and large language models software engineering and robotics.



Ji-Hwan Park received the bachelor's degree in mechanical engineering in 2023 from The University of Texas at Austin, Austin, TX, USA, where he is currently working toward the Master of Science in mechanical engineering.

His main research interests include robotics, SLAM, and computer vision.



Maria Esteva received the Ph.D. degree in information science from The University of Texas, Austin, TX, USA, in 1985.

She is a Research Scientist with the Texas Advanced Computing Center, The University of Texas at Austin, Austin, TX, USA. She works with the Texas Robotics researchers to curate and publish their datasets in accordance with FAIR data principles. Her research interests include the design, implementation, and evaluation of large-scale data curation applications in high performance computing environments.



Raymond Hong received the bachelor's degree in computer science with Honors from The University of Texas at Austin, Austin, TX, USA, in 2023.

His main research interests include 3-D computer vision, robotics, and mathematics.



Joydeep Biswas (Member, IEEE) received the B.Tech degree in engineering physics from the Indian Institute of Technology Bombay, Mumbai, India, in 2008, and the M.S. and Ph.D. degrees in robotics from Carnegie Mellon University, Pittsburgh, PA, USA, in 2010 and 2014, respectively.

He is an Associate Professor with the Department of Computer Science, The University of Texas at Austin, Austin, TX, USA. 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.

Dr. Biswas was the recipient of the NSF CAREER award, an Amazon Research award, two JP Morgan Faculty Research awards, and is a Siebel Scholar, Class of 2015.



Pranav Kalyani received the bachelor's and master's degrees in mechanical engineering in 2019 and 2023, respectively, from The University of Texas, Austin, TX, USA, where he is currently working toward the Ph.D. degree in mechanical engineering.

His main research interests include controls, dynamics, and machine learning.