Mixed Signals: A Diverse Point Cloud Dataset for Heterogeneous LiDAR V2X Collaboration

Appendix

In this appendix material, we include: 1) extra details about the Mixed Signals dataset and the provided code devkit, 2) annotation details and instructions given to annotators, and 3) additional sensor details. We include an additional dataset teaser video in the dataset website[*] that we encourage readers to watch.

A. Data and Devkit

Please see https://mixedsignalsdataset.cs.cornell.edu/ for the dataset download instructions and the provided devkit. Below, we add a brief description of the devkit and visualize a dataset sample.

A.1. Devkit Description

We provide a separate devkit and additionally integrate our dataset into the framework OpenCOOD [42], which offers the implementation of various state-of-the-art collaborative perception methods. As OpenCOOD only provides single-class models, we adapt its implementation of Early, Intermediate, and Late Fusion models to detect three classes, including vehicles, bikes, and pedestrians. We added detection heads of 1-by-1 convolution layers to existing architectures to achieve this. In addition, we add the recent *Laly* fusion [7] to this framework. Every model in our benchmark uses PointPillar [20] as the backbone. Interested readers can refer to our devkit[†] and code release[‡] and extended OpenCOOD integration for further details on architectures and training settings.

A.2. Sample Data

Figure A1 shows an example of the collected data, where the points are colour-coded to represent the different LiDARs. The dataset aims to replicate realistic urban scenarios that reflect the complexities of real-world implementations by using multiple vehicles with diverse sensor configurations and a roadside unit. Real-world deployments of autonomous vehicles on streets incorporate LiDARs, which are becoming more affordable. Roadside infrastructure, such as roadside units, is also gaining popularity for traffic monitoring and data analytics, now often equipped with LiDAR, traffic light timing information, and communication systems to enhance robustness and applicability.

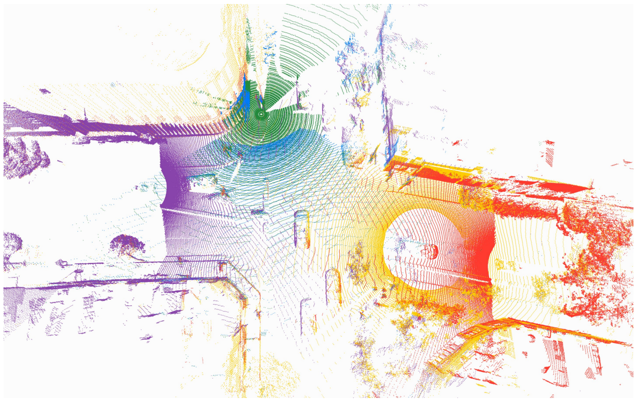


Figure A1. **Top-down view of the data collected at the location.** LiDAR point clouds are colored by the vehicle and RSU that collected them, consisting of the 3 vehicle agents (red, yellow, and purple) and the Top and Dome LiDAR sensors of the RSU (green, blue). Best viewed in colour.

Our dataset consists of LiDAR point clouds, which offer the advantage of not capturing identifiable information like faces or license plates, thus preserving data privacy. This contrasts with camera images, which often require post-processing to anonymize sensitive details, potentially affecting data quality. Our dataset includes tracking IDs for each bounding box, and this information will be released alongside this paper. Benchmarks will be made available at a later date.

Intensity Distributions. Figure A2 shows LiDAR intensity distributions from RSU `TOP`, `DOME`, and `Laser` car sensors. `DOME` and `TOP` sensors record higher intensities because there is a large number of static objects (e.g., buildings, traffic lights) near them. In contrast, the `Laser` car sensor presents a smoother decline in intensity values because of its location on the vehicle, which allows the detection of objects at greater distances. EV-1 and EV-2 sensors do not capture intensity readings. Therefore, a uniform approach to utilizing intensity values across all agent models would be inadequate.

B. Annotation Instructions

We provide the instructions given to the Segments.ai[§] annotators in the attached material, titled "*Spec Document - Multi-sensor labeling*" at the bottom of the appendix. We selected to invest in the quality of the annotations, applying

[*] https://sites.coecis.cornell.edu/mixedsignals/#introvid

[†] https://github.com/quan-dao/mixed-signals-devkit

[‡] https://github.com/acfr/Mixed-Signals-Dataset

[§] https://segments.ai/

Category	Definition
Car	Includes passenger vehicles such as sedans, hatchbacks, SUVs, and coupes that are designed primarily for the transportation of passengers.
Truck	Encompasses larger vehicles primarily used for transporting goods and materials. This category includes pickup trucks, delivery trucks, and heavy-duty trucks.
Emergency Vehicle	Vehicles designated for emergency response, including ambulances, fire trucks, police cars, and other vehicles equipped with sirens and emergency lights.
Bus	Large motor vehicles designed to carry numerous passengers. Buses include city transit buses, school buses, and intercity coaches. They usually have designated routes and schedules.
Motorcycle Motorized Bike	Two-wheeled motor vehicles, including motorcycles and motorized bikes. This category also covers scooters and mopeds.
Portable Personal Mobility Vehicle	Small, lightweight vehicles designed for personal mobility, including electric scooters, hoverboards, and segways.
Bicycle	Human-powered, pedal-driven vehicles with two wheels. Bicycles include standard bikes, mountain bikes, and road bikes. This category include motorized bicycles or electric bikes.
Electric Vehicle	Refers to small, golf car-like vehicles used for data collection purposes.
Trailer	Non-motorized vehicles designed to be towed by a motor vehicle.
Pedestrian	Individuals traveling on foot. This category includes people walking or running.

Table A1. **Definitions of the annotation classes.**

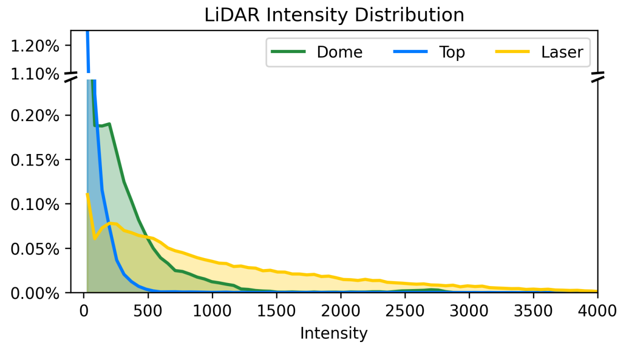


Figure A2. **Distribution of LiDAR intensities from RSU `TOP`, `DOME`, and `Laser` car sensors.** Each sensor shows different intensity ranges and distributions. `EV-1` and `EV-2` LiDAR sensors do not have intensity readings.

rigorous quality control measures to guarantee accurate and consistent labeled data, minimizing errors, and maintaining high standards.

B.1. Definitions of the Annotation Classes

The Mixed Signals dataset categories road agents in different vehicle types and pedestrians. Categories such as “Car” and “Truck” encompass common passenger and large transport vehicles, while “Emergency Vehicle” covers ambulances, fire trucks, and police cars, highlighting their importance in urban scenarios. “Bus” labels are designated for large passenger vehicles typically used in public transportation. The dataset also distinguishes between “Motorcycle” and “Motorized Bike,” and “Portable Personal Mobility Vehicle,” which includes modern personal transport devices like electric scooters and hoverboards. Traditional “Bicycle” labels account for both standard and electric bikes. Labels for “Electric Vehicle” and “Trailer” ensure that smaller, often data-collection vehicles and towable units are accurately represented. Finally, we labeled humans as “Pedestrians”. In Table A1, we provide the definitions of the 10 fine-grained annotation classes in the Mixed Signals dataset. The breakdown of the fine-grain classes into the benchmarked classes can be found in the main text.

B.2. Track Annotations and Multi-agent Tracking

We benchmark the performance of the planned tracking task for our dataset. The Mixed Signals dataset has labels for track ID’s, as seen in Figure 10 of the main text. We hope to include and benchmark tracking methods as an additional task which is supported by our dataset. We report some initial benchmarking results on the AB3DMOT tracking method [37] in Table A2. For further details about track labels, please explore the data itself; a distribution of the tracks are in Figure 9 of the main text.

B.3. Annotation Details

The annotation process for this multi-sensor dataset involves handling joint scenes and synchronization discrep-

Category	sAMOTA	AMOTA	AMOTP
Vehicle	89.6	43.1	63.3
Pedestrian	76.6	32.6	42.8

Table A2. Tracking performance for AB3DMOT with V2X-ViT detections on the Mixed Signals validation split.

ancies between sensors. Due to time synchronization, fast-moving objects might appear slightly offset across the data collected from different sensors. To address these discrepancies, annotators were instructed to prioritize the roadside unit point cloud for bounding box creation, following a set hierarchy. When there is a mismatch, bounding boxes should be aligned with the point cloud in the following order: roadside unit, EVs, and the urban vehicle. For example, if there is a difference between the roadside unit and the vehicles' point cloud, the bounding box should only be fitted around the roadside unit points. This ensures consistency in object localization across frames despite synchronization lags.

C. Sensor Details

C.1. Hardware and Synchronization Details

Sensor	Agent	Range*	Channels	Vertical FOV
Ouster OS1-128	Vehicles	170 m	128	45
Ouster OS1-64	RSU	100 m	64	45
Ouster OS Dome	RSU	45 m	128	180

*Based on 80% Lambertian reflectivity in the sensors' official datasheets.

Table A3. **Hardware specifications.**

Synchronization is especially important in dynamic environments, as any introduced time shifts can lead to positional inconsistencies, resulting in multiple detections of the same object. The sensors in our multi-agent system were timestamped using GPS time as a common reference, and sensor details are provided in Table A3. Rotational LiDARs continuously scan the environment in 360 degrees, thus, different portions of the surroundings are captured at slightly different times during a full rotation. When vehicles are in motion, their positions and orientations change dynamically between LiDARs sweeps. The maximum time gap for matching sensor readings between 10 Hz rotational sensors is 50 ms. Since sensors rotate fully in 100 ms, angular positions differ by at most 180 degrees. If the time difference between readings were larger than 50 ms, it would be matched with the next or previous rotation instead. As shown in the original manuscript, precise sensor synchronization, robust multi-agent localization, and clearly defined annotation protocols produced high-quality data association across all sensors.

C.2. Localization

Localization is one of the most critical tasks for CAV, estimating their position relative to a global reference frame. One of the most commonly used sensors for localization is the Global Navigation Satellite System (GNSS). GNSS offers access to a satellite constellation that provides global positioning via triangulation. However, despite its widespread use, GNSS has several drawbacks, particularly in urban environments. Its accuracy can be reduced in urban canyons, where tall buildings block or reflect signals, leading to degraded positioning accuracy. To overcome this problem, we use dense and accurate point cloud maps [31] as references for our localization algorithm.

C.3. Definition of Heterogeneity in Sensor Suite

Heterogeneity in our context refers to the variability between LiDAR sensors and platform geometry within a single dataset. Heterogeneity can appear in multiple forms [17]; our dataset represents it in five LiDARs that span three models, each mounted in four configurations. In line with the feedback, Tab. 1 of the original manuscript has been updated accordingly. Our dataset demonstrates a realistic setting where collaborative agents have different LiDAR models and position them in different configurations.

Segments.ai

Spec Document - Multi-sensor labeling

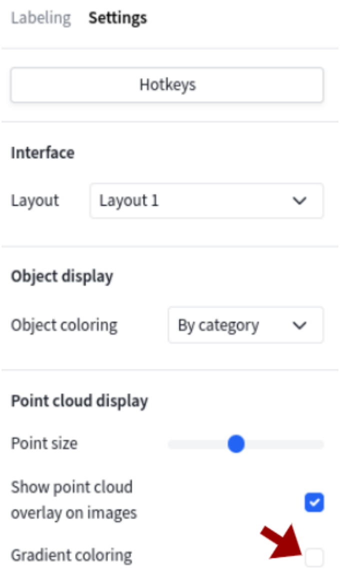
Additional resources

Getting Started

Begin by ensuring the point clouds are ***not*** visualized with the default gradient coloring (and to disable the color-by-gradient). The data is colored by the sensors they are collected from, and should be colored green, blue, orange, yellow, and pink. To do so, follow the instructions:

1. Click on the `Settings` tab of the control panel
2. Scroll to `Point cloud display` and un-check the `Gradient coloring` tab

Everything is correct if there are only 5 colors (green, blue, orange, yellow, and pink) displayed.



Interface

Layout: Layout 1

Object display

Object coloring: By category

Point cloud display

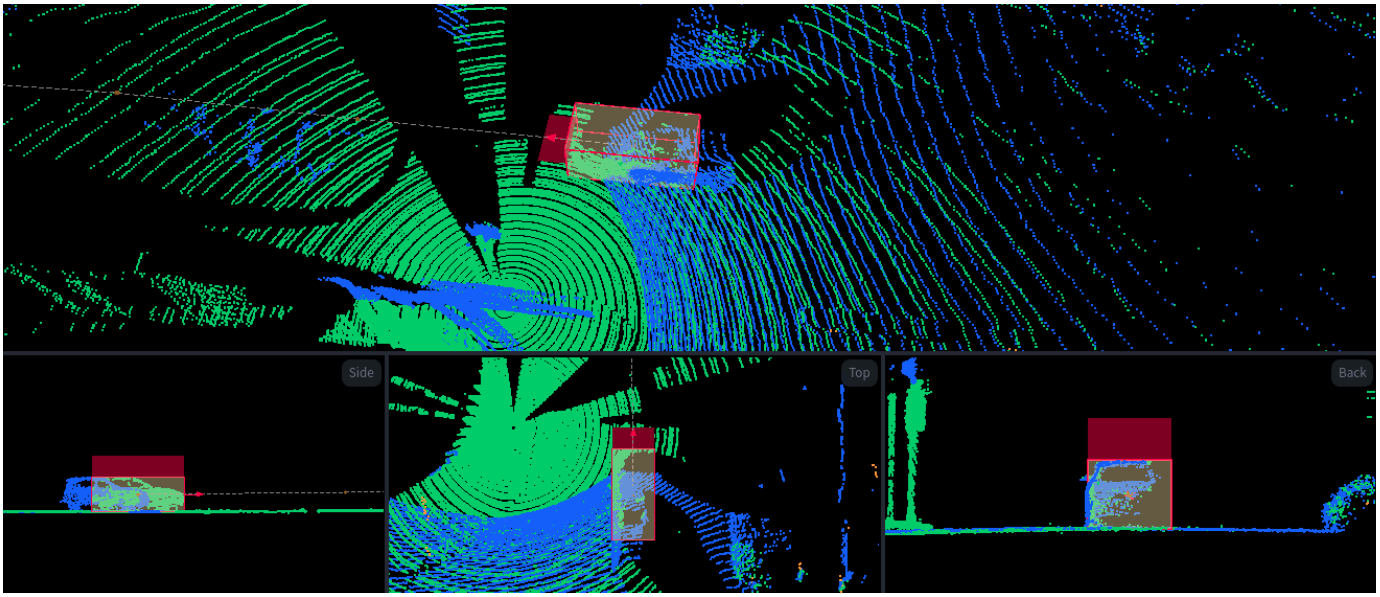
Point size

Show point cloud overlay on images ☑

Gradient coloring □

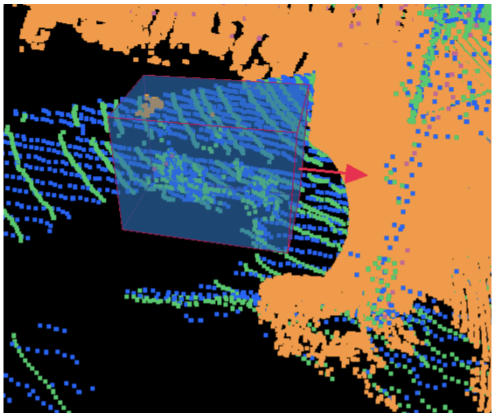
Discrepancy Resolution

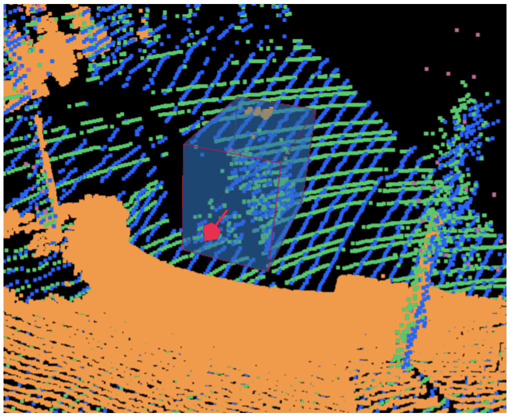
Due to time synchronization issues, a single object may have different scans (discrepancy between sensors). This may occur for fast moving vehicles. When this happens, fit a bounding box ***onto a single sensor***, prioritizing the point cloud colors in the order of: green, yellow, orange, pink, and (lastly) blue. For example, if there is a discrepancy between the blue points and green points, ignore the blue points and fit a bounding box only on the green points.



Special Vehicle Cases

We have 2 electric data-collection vehicles (EV). If they show up in other sensors, label them as ***Electric Vehicle (EV)*** category. They look like small golf cars.





Link to external labeling guidelines	N/A (see above)
Link to reference labeled dataset	https://mobility-lab.seas.ucla.edu/v2v4real/#dataannotation
Link to initial unlabeled dataset	TBD

Sensor information

3D sensors

Type of 3D sensors *E.g. Velodyne, ...* *(Add lines where needed)*	**How many sensors?**	**File format?** *More info*	**Point clouds in local or world reference system?**	**In case of local reference system, ego poses available?**	**Number of frames per sequence + Sampling rate?**	**Typical file size / point quantity per frame?**	**Comments**
One single point cloud composed of multiple Ouster Lidar point clouds	3-6	PCD	Word reference	TBD	40 - 41 frames per sequence (20 s/seq): sampling rate 2 hz		
			TBD	TBD			

Type of task

3D labeling

	Type of labeling	**Comments**
☑	3D cuboids	3D bounding box on point cloud data
☐	3D segmentation	
☐	3D polylines	
☐	3D polygons	
☐	3D keypoints	

Labeling rules

General labeling rules

Rule	**Answer**	**Default assumption going forward without explicit answer from customer**	**Comments**
Maximum distance to label objects?	Everything	Everything	See ability to set a visual radius for taskers here
Other zones that do not require labeling?	Label everything	Label everything *including e.g. car parks*	
What to do with unclear objects/areas?	Do not label	Do not label	
What to do with reflections?	Do not label	Do not label	
How to cope with groups of individual	Label each individual	Label each individual	

instances?			
Are there any specific rules to adverse weather conditions / nighttime / etc.?	Same as day time	Same as day time	

Specific rules for cuboid labeling of instance classes

Type	Rule	Answer	Default assumption going forward without explicit answer from customer	Comments
Position	**Can there be some overlap between cuboid & ground plane, or should cuboids be leveled with the ground plane where applicable?**	Slight overlap is OK	Slight overlap is OK	
Rotation & Headi...	**Label only yaw, or yaw & pitch & roll?**	Yaw, roll & pitch	Yaw, roll & pitch	*Please note that labeling also pitch & roll can reduce throughput with up to a factor 2*
Rotation & Headi...	**What should the yaw direction/heading be of an object?**	Main face of object	Main face of object	
Rotation & Headi...	**What should the yaw direction/heading be of a faceless object, e.g. a cone?**	TBD Do Not Label Faceless Categories	TBD	Do not label "cones" or other faceless object categories
Dimensions	**Is there a minimum size of cuboid?**	No	No	
Dimensions	**Should cuboids be labeled with default dimensions depending on their category?**	No	No	
Dimensions	**Can the dimensions of a cuboid change throughout a sequence?**	Yes No for: vehicles and rigid objects	Yes *when needed*	
Dimensions	**What should the dimension of a cuboid be based on?**	Realistic size, namely ... *[copied over] In order of importance: default dimensions / on the available 3D points elsewhere / on the reference images*	Realistic size, namely ... *In order of importance: default dimensions / on the available 3D points elsewhere / on the reference images*	*Visible 3D points: only using the visible 3D points in the current frame, disregarding the realistic size* *Realistic size: e.g. based on the available 3D points elsewhere in the sequence, on the available reference images or on the provided default dimensions*
Dimensions	**How tight should a cuboid be?**	Loose As tight as possible, but can be a bit larger than the object itself. Ensure all parts are within the cuboid, see *"Extremities" sections*	Loose	*Very tight: in each frame, there should not be any space between the outer points and the cuboid* *Loose: the cuboid can be a bit larger than the object itself*
Occlusion	**Should an object be labeled if it is only visible on 3D sensors and not on 2D sensors?**	Yes	Yes	

Occlusion	Should an object be labeled if it is only visible on 2D sensors and not on 3D sensors?	Minimum of 10 points	Minimum of 10 points	
Occlusion	Should an object be labeled with the same track ID if it re-enters a scene or becomes unoccluded again?	Yes	Yes	
Extremities	Should fixed extremities / protruding parts be included in the cuboid?	Custom ... ***Include***: *side mirrors, larger protruding parts such as bonnet* ***Exclude***: *small protruding parts such as antenna*	Custom ... ***Include***: *side mirrors, larger protruding parts such as bonnet* ***Exclude***: *small protruding parts such as antenna*	
Extremities	Should variable/articulating extremities be included in the current cuboid, or be annotated with separate cuboids?	Custom ... **Include:** rider, people in/on vehicles, non-vehicle objects on trucks **Exclude:** vehicles on trucks	Custom ... ***Include***: *rider, cars on trailers, people in/on vehicles* ***Exclude*** *(only when a relevant category is available): person trolley, car trailer, ...*	
Extremities	Include relational tracker for variable extremities?	No	No	*If yes: if a car has object ID 10, and a trailer is attached to this car, the trailer will be annotated separately and receive a relational attribute with value 10*
Issues/exceptions	Can cuboids overlap?	Only in reasonable cases	Only in reasonable cases	*Reasonable cases: articulating objects (turning truck with trailer attached, ...), bicycles standing very close to each other, ...*
Issues/exceptions	What to do with unclear objects?	Use 'unsure' category	Use 'unsure' category	
Issues/exceptions	In case of bad calibration, how should the cuboid be fitted? should the cuboid be fitted to the 3D point cloud or rather to the most confident reference images?	Fit to 3D	Fit to 3D	
Issues/exceptions	In case of bad ego poses and data drift, what should happen?	Fit on each frame	Fit on each frame	

Specific rules for 3D polygon/polyline labeling

Type	Rule	Answer	Default assumption going forward without explicit answer from customer	Comments

Temporality	**Can nodes be added/removed from polygons/polylines after initialization?**	No	No	*Note: currently not yet compatible with interpolation (below); on short-term roadmap*
Temporality	**Should interpolation be enabled?**	Yes	Yes	*Note: currently not yet compatible with adding/removing nodes (above); on short-term roadmap*

Categories

Category	Label instances?	To be labeled across all sensors & tasks?	What does it include?	What does it exclude?	Comments
Car	Yes	No, only 3D	• Sedan • SUV • Minivan • All personal/recreational vehicles	• *Speed bumps* • *Raised crosswalks*	
Truck	Yes	No, only 3D			
Emergency Vehicle	Yes	No, only 3D	• Police Cars • Ambulance		
Bus	Yes	No, only 3D			
Motorcycle/Motorized Bike	Yes	No, only 3D	• Motorcycles • Electric/Motorized Bike • Scooters		
Portable Personal Mobility Vehicle	Yes	No, only 3D	• Segway • Moped		
Bicycle	Yes	No, only 3D			
Pedestrian	Yes	No, only 3D	• Pedestrians crossing • Pedestrians with strollers (label a single box for both)		
Electric Vehicle (EV)	Yes	No, only 3D	• Data collection golf car vehicle		
Trailer	Yes	No, only 3D	• Trailer attached to vehicle • Wagon • Other vehicles attached to a pickup-truck/utility vehicles		

Attributes

Frame-level attributes

N/A

Attribute	Type	Options	Description	Comments

	TBD			

Object-level attributes

N/A

Attribute	Type	Applicable categories	Options	Description	Comments
Motion State	Select box	All categories (Car, Truck, Emergency Vehicle, Bus, Motorcycle, Bike, Pedestrian, Portable Vehicle...)	In-Motion / Static	Select if the object is in motion during the frame. Either select that it is in-motion (currently moving) or static (waiting, parked, stopped, etc)	

Edge cases

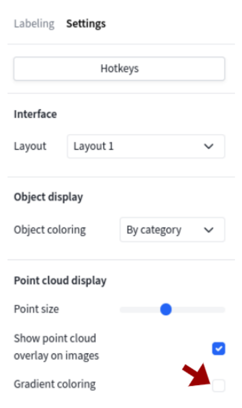
Edge Case	How to handle it	Example
Extreme		

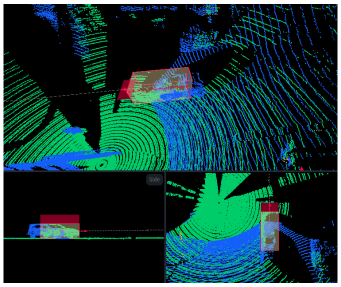
Version history of labeling specification document

N/A

Date	Version	Changes

Additional Q&A

Date	Question status	Question & assumptions	Reference Image/ Link	Answer
03/02/2024	Assumption	I do not see only 4 colors, and instead see the default gradient point color scheme. How do I toggle off the gradient coloring?	 Labeling Settings Hotkeys Interface Layout: Layout 1 Object display Object coloring: By category Point cloud display Point size Show point cloud overlay on images ☑ Gradient coloring □	Click on the "Settings" tab, and make sure "Gradient coloring" is **not** selected. See *Additional Resources*: Getting Started instructions.

03/02/2024	Assumption	There are time synchronization issues, where fast moving objects are captured differently by different colored point clouds. How do I label these?		Fit a tight bounding box around the point cloud of a single color, prioritizing in the order of: 1. Green 2. Yellow 3. Orange 4. Pink 5. Blue In this example, only draw a bounding box around the green points. See *Additional Resources*: Discrepancy Resolution.