Karlsruhe Institute of Technology

Anomaly Detection with Model Contradictions for Autonomous Driving

Bachelor Thesis

Vincent Geppert

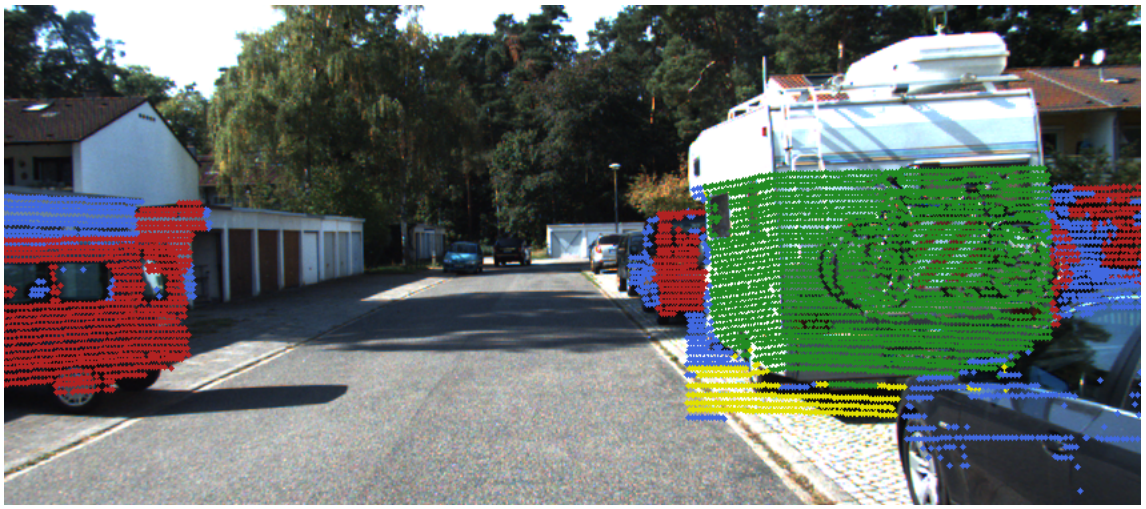
Department of Economics and Management
Institute of Applied Informatics and Formal Description Methods
and
FZI Research Center for Information Technology

Reviewer:	Prof. Dr.–Ing. J. M. Zöllner
Second reviewer:	Prof. Dr. Andreas Oberweis
Advisor:	M.Sc. Daniel Bogdoll

Research Period: 01. April 2023 – 01. August 2023

Anomaly Detection with Model Contradictions for Autonomous Driving

by
Vincent Geppert



Bachelor Thesis
August 2023

FZI

Bachelor Thesis, FZI
Department of Economics and Management, 2023
Reviewers: Prof. Dr.–Ing. J. M. Zöllner, Prof. Dr. Andreas Oberweis

Department Technical Cognitive Systems
FZI Research Center for Information Technology

Affirmation

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

Karlsruhe, *Vincent Geppert*
August 2023

Abstract

Anomaly detection is a critical aspect of safe autonomous driving systems, where detecting and understanding uncommon and unpredictable scenarios, often referred to as corner cases or anomalies, is crucial for ensuring the safety of passengers and pedestrians. In this bachelor's thesis, I quantitatively evaluate an anomaly detection method proposed by Sartoris [38] that utilizes Light Detection and Ranging (lidar) data for detecting anomalies. The method combines a supervised (SV) and a self-supervised (SSV) part to detect motion anomalies in the environment. By analyzing the discrepancies between the two parts, the detection method identifies points that deviate from the expected behavior, indicating potential anomalies.

This evaluation utilizes the Common Anomaly Detection in Autonomous Driving (CODA) dataset [25], which provides the only anomaly dataset including lidar data. However, the corner cases are only labeled in the form of 2D bounding boxes. To address this limitation, I convert the 2D bounding boxes in the CODA dataset into 3D point-wise labels. The CODA dataset is then translated into the KITTI-odometry data format suitable for evaluating Sartoris' method. Additionally, improvements are proposed for the clustering algorithms used to create 3D point-wise labels, aiming to reduce the need for manual verification.

Given the lack of suitable metrics for semantic segmentation, except for Mean Intersection over Union (mIoU), I propose two novel approaches that utilize the metrics Average Precision (AP), Average Recall (AR), and F1 Score (F1) to quantitatively evaluate Sartoris' anomaly detection approach on the CODA dataset. The results demonstrate the potential of this method in detecting anomalies compared to standard object detection techniques carried out by Li et al. [25], despite the challenge of comparing my metrics with theirs.

In the outlook of this thesis, the potential extensions and improvements to the detection approach are discussed, such as fine-tuning the models for the original datasets and addressing challenging scenarios like fast turns or speed bumps. Furthermore, the need for appropriate metrics to effectively evaluate Sartoris' anomaly detection methods is addressed.

Contents

1 Introduction **1**

2 Background **3**
2.1 Frustum 3
2.2 Clustering 3
2.2.1 Mean shift 4
2.2.2 Density-Based Spatial Clustering of Applications with Noise (DBSCAN) 5

3 State of the Art **7**
3.1 Corner Cases in Autonomous Driving 7
3.2 Anomaly Detection on 3D Data 8
3.3 Research Gap 10

4 Evaluation Data **11**
4.1 Data Selection 11
4.2 CODA Dataset 13

5 Method **17**
5.1 Data Processing 17
5.2 Data Labeling 20
5.3 Anomaly Detection 22
5.4 Evaluation Method 24

6 Evaluation **29**
6.1 CODA Evaluation 29
6.2 3D CODA Groundtruth 29
6.3 Quantitative Anomaly Detection 33

7 Conclusion and Outlook **39**
7.1 Outlook 40

A Appendix **41**

B List of Figures **49**

C List of Tables **53**

D Bibliography 55

Acronyms

SV supervised

SSV self-supervised

lidar Light Detection and Ranging

AP Average Precision

AR Average Recall

F1 F1 Score

mIoU Mean Intersection over Union

CODA Common Anomaly Detection in Autonomous Driving

nuScenes CODA-nuScenes

KITTI CODA-KITTI

ONCE CODA-ONCE

DBSCAN Density-Based Spatial Clustering of Applications with Noise

mean shift Mean shift

1 Introduction

Critical situations in road traffic occur on a daily basis, presenting challenges that autonomous vehicles must overcome to ensure safe and reliable driving. These situations can arise due to various factors, such as adverse weather conditions, sudden appearance of pedestrians from behind obstacles, or abrupt lane changes. Detecting and addressing these critical situations before they escalate is crucial for establishing a robust autonomous driving system. Literature refers to such situations as corner cases or anomalies [18]. This thesis, in line with previous works [18, 7], adopts the definition of corner cases proposed by Bolte et al. [5], where corner cases are identified as situations involving non-predictable relevant objects or classes in specific locations. Consequently, corner cases encompass situations that deviate significantly from the learned ones and cannot be easily anticipated by the autonomous driving system. To effectively address these corner cases, it is necessary to detect and classify them accurately.

One of the essential tasks for self-driving cars is to perceive the movement of other road users, enabling the distinction between dynamic and static objects. This task becomes particularly challenging in urban areas where pedestrians, cyclists, and various other road users interact with traffic. Reliable prediction of an object's movement is crucial in such environments. Autonomous vehicles employ multiple sensors to perceive the surrounding environment, leveraging the strengths of each sensor to compensate for their individual limitations. Cameras provide high-resolution colored images, facilitating the identification of road signs, traffic lights, and other visual cues. Cameras are susceptible to low light conditions and reduced visibility [28, 46]. Radar sensors, on the other hand, utilize high-frequency electromagnetic waves to detect obstacles and are not affected by lighting or weather conditions. However, they suffer from coarse resolutions and sensitivity to target reflectivity [28, 46]. Lidar sensors generate 3D point clouds by measuring the time taken for a laser beam to return after reflecting on objects. While lidar sensors are robust in capturing depth information, they can be affected by adverse weather conditions and exhibit reduced detection rates for dark or specular objects. Furthermore, distant objects may result in sparse point clouds [13].

Motivated by the need to detect anomalies in road traffic, this thesis aims to re-evaluate the approach of Sartoris [38] on the new CODA dataset. His approach is based on identifying anomalies in lidar data by examining the consistency of point-wise motion predictions from SV and SSV models. In this context, an anomaly is defined as an object whose motion label cannot be accurately predicted. Inconsistencies between the supervised and self-supervised models serve as indicators of incorrect predictions, shedding light on the limitations of each model. This quantitative evaluation enhances the comparability of the proposed method to state-of-the-art object detection methods.

To accomplish this goal, the first step involves converting the 2D bounding boxes of corner cases of the CODA dataset into 3D point-wise labels. Subsequently, the CODA data format is transformed into the KITTI odometry format, as utilized by Sartoris [38]. In the next step, Sartoris' approach is applied to the CODA dataset, enabling a quantitative evaluation of the method against the ground truth.

Chapter 2 provides a concise overview of frustums and clustering methods, essential for converting labels from 2D to 3D. In Chapter 3, the state of the art in anomaly detection is discussed, highlighting the existing research gap. Chapter 4 explains the principle of utilizing the CODA dataset for this research. Chapter 5 states the methods employed to prepare the CODA dataset, as well as the reimplementation and evaluation of Sartoris's work [38] against the ground truth. Lastly, Chapter 6 presents the evaluation of the generated 3D ground truth data for the CODA dataset and assesses the quantitative performance of Sartoris' detection method against the ground truth.

This research contributes to the advancement of anomaly detection in autonomous driving systems, offering valuable insights into the strengths and limitations of combining SV and SSV models for anomaly detection in lidar data. By addressing corner cases in road traffic, this thesis makes strides toward developing safer and more reliable autonomous driving systems.

2 Background

This background chapter presents a comprehensive overview of the methodology used to generate the ground truth data for the CODA dataset, which involves the utilization of frustums and clustering techniques. Given that the existing labeling in CODA is limited to 2D bounding boxes for corner cases, my objective was to accurately label the corresponding 3D point clouds to facilitate the re-evaluation of Sartoris' work [38]. To achieve this, frustums are employed to restrict the analysis on specific regions in the point cloud corresponding to the 2D bounding boxes. Subsequently, clustering techniques are applied to precisely determine the 3D points of the corner cases.

2.1 Frustum

Frustums play a fundamental role in the translation of 2D to 3D data in autonomous driving. A frustum is the portion of a body that lies between two parallel planes cutting this body. In this case, the body is a pyramid, defined by the viewing point and the 2D bounding box, see Figure 2.1. In perception systems, frustums provide an upper bound of potentially related points in 3D in relation to a bounding box in 2D.

The importance of frustums arises from their ability to geometrically relate the 2D bounding box to the three-dimensional position of objects in 3D point clouds [33, 35, 40, 43]. By projecting a 2D bounding box onto the corresponding point cloud, I obtain a frustum that encompasses a subset of the point cloud data potentially associated with the object of interest. This localized region enables focused analysis and facilitates robust estimation of the object's spatial properties.

This restrictive step saves computational resources and improves real-time performance. Secondly, they allow me to integrate pixel-accurate object estimations from image-based object detection by utilizing 2D bounding boxes [35]. Combining these two modalities achieves more accurate and comprehensive 3D bounding box estimation [27]. Frustums also help handle occlusions better, as I can mitigate the impact of occluded or partially visible objects through localized analysis within the point cloud.

To achieve a more precise localization of the 3D objects within the point cloud, clustering algorithms were employed to provide point-wise labeling of the point cloud.

2.2 Clustering

To cluster point clouds in my study, I employed the Mean shift (mean shift) and DBSCAN algorithms, as proposed in [34]. In the following subsection, I provide a brief overview of both algorithms.

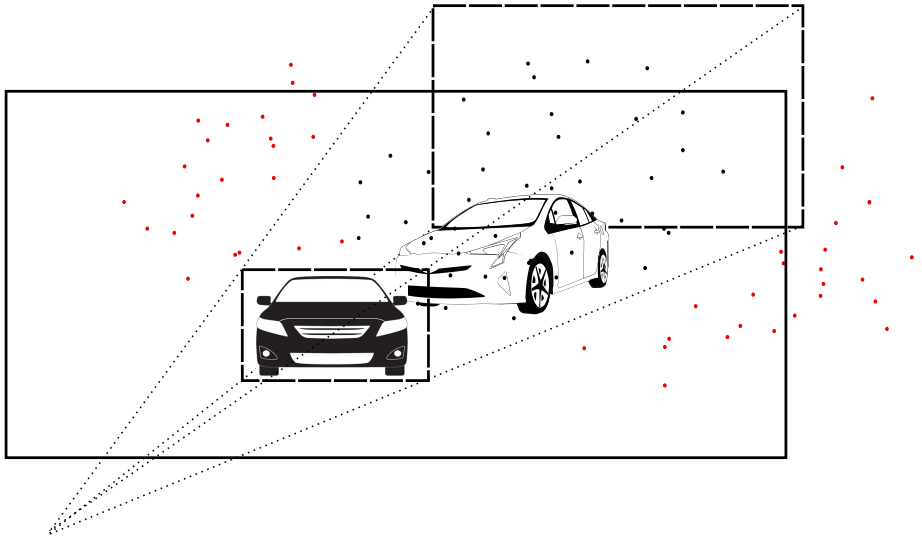


Figure 2.1: Frustum for extracting a part of the point cloud based on the 2D bounding box.

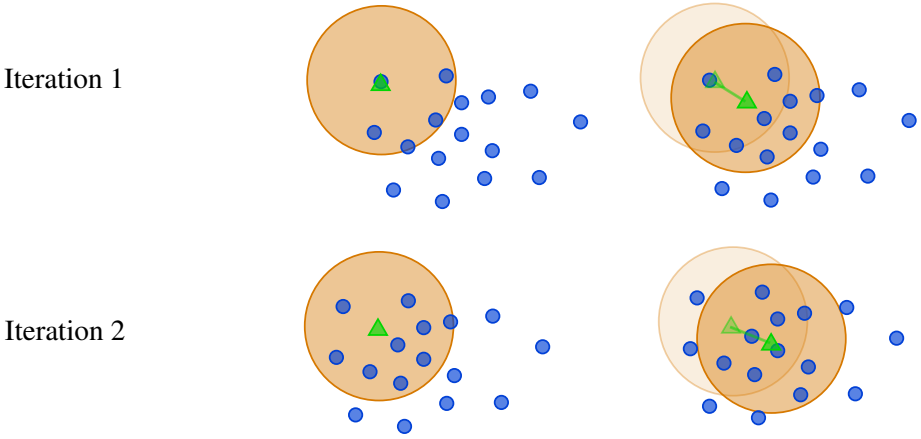


Figure 2.2: Visualization of the mean shift algorithm, adapted from [39].

2.2.1 Mean shift

The mean shift algorithm outlined in [34] employs a one-parameter clustering approach to identify regions of high density in the data distribution. This is achieved by iteratively updating the position of each data point, shifting it towards the mean of the neighboring points within a predefined radius, see Figure 2.2. The iterative process continues until convergence, causing the points to settle in the densest regions and form clusters. It is important to note that the effectiveness of the mean shift algorithm heavily relies on the bandwidth parameter, which determines the radius.

One advantage of this algorithm is its ability to cluster data without requiring a predefined number of clusters, making it suitable for situations with unknown cluster counts. Moreover, the mean shift algorithm can cluster data points of various shapes. However, a significant drawback is the sensitivity of the results to the bandwidth parameter. If this parameter is not appropriately set, there is a risk of excessive clustering or missing certain clusters.

The mean shift algorithm is particularly well-suited for clustering larger objects that encompass clusters with varying shapes.

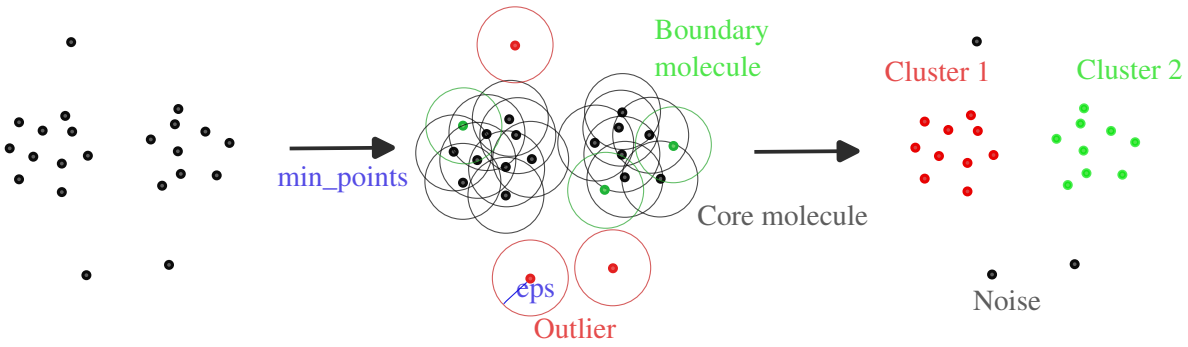


Figure 2.3: Visualization of the DBSCAN algorithm, adapted from [22].

2.2.2 DBSCAN

The DBSCAN algorithm, an acronym for Density-Based Spatial Clustering of Applications with Noise, characterizes clusters as dense regions separated by sparser areas. The algorithm initiates by selecting a random point and extends the cluster by linking neighboring points within a specified distance threshold, see Figure 2.3. Moreover, it identifies points situated in sparser regions as noise or outliers. In this context, the size of the neighborhood is defined by the *eps* parameter, while the *min_points* parameter specifies the minimum number of neighboring points required within the *eps* distance for a point to be considered part of a cluster.

The adaptability of parameter selection is deemed unnecessary, as mentioned in the study by Yabroudi et al. [45] when a single set of suitable parameters exists for the given dataset. The DBSCAN algorithm obviates the need for a fixed number of clusters. It should be noted, however, that the algorithm tends to generate an excessive number of clusters for complex structures, yet it is particularly well-suited for smaller and simpler objects.

3 State of the Art

This chapter establishes the contextual background for the presented method. It discusses the different types of anomalies and shows a categorization approach for them. It also provides an overview of the existing research conducted in anomaly detection on lidar data, highlighting the current research gap. Finally, it presents the contribution of this work in addressing the identified research gap.

3.1 Corner Cases in Autonomous Driving

Outlier, *Anomaly*, and *Novelty* are widely utilized terms in scientific literature, and they bear a strong relationship to corner cases, often exhibiting overlapping meanings [6, 18]. An *outlier* is an observation that deviates significantly from other observations, arousing suspicions of a distinct generating mechanism [17]. For instance, in corner cases, such deviations occur when a lidar beam experiences maximum reflection or absorption. *Anomalies* do not conform to learned patterns or general normal behavior [9]. In corner cases, the consideration of *anomalies* needs to be combined with their relevance to driving behavior [5], often arising in complex scenarios that exhibit overall anomalous characteristics but do not consist of individual anomalous objects. Conversely, *novelties* represent instances or objects that have not previously been encountered [9]. They are correlated with corner cases in that the appearance of new situations, objects, and movement patterns is an essential characteristic of corner cases [18]. In this thesis, the terms *corner cases* and *anomalies* are used interchangeably, following Heidecker et al. [18]. Generally, machine learning (ML) systems often exhibit poor performance on corner case data, as such data encompasses novel situations or effects absent from the system's trained data. Identifying these corner cases poses a significant challenge [18], as they are crucial for validating and retraining the system.

To effectively investigate and develop suitable detection methods for corner cases, it is crucial to address their inherent ambiguity and provide precise specifications [18, 7]. Heidecker et al. propose a comprehensive framework that classifies sensor-specific corner cases into different *levels*, demonstrating their diverse nature [18, 7]. These *levels*, ordered by theoretical complexity, span three *layers*: *sensor*, *content*, and *temporal*. The *sensor layer* encompasses the *hardware level*, which involves corner cases arising from a faulty sensor setup, and the *physical level*, which encompasses corner cases resulting from specific limitations of the sensor technology itself, such as the occurrence of absorbing solid surfaces in the case of lidar sensors. Moving to the *content layer*, corner cases manifest at the *domain level*, wherein anomalies arise due to disparities between the observed world and the model's ability to explain it. An example is the presence of unfamiliar road markings in a different country. At the *object level*, corner cases revolve around single-point clouds and the objects contained within them. The *scene level* describes whether these unknown

	Sensor Layer		Content Layer			Temporal Layer
	Hardware Level	**Physical Level**	**Domain Level**	**Object Level**	**Scene Level**	**Scenario Level**
LiDAR-based corner cases	Laser Error • *Broken mirror* • *Misaligned actuator*	Beam-Based Corner Case • *Black cars disappear* • ...	Domain Shift on Single Point Cloud • *Shape of Road markings*	Single-Point Anomaly on Single Point Cloud • *Dust cloud* • ...	Contextual/Collective Anomaly on Single Point Cloud • *Sweeper cleaning the sidewalk*	Corner Cases on Multiple Point Clouds and Frames • *Person breaks traffic rule* • *Overtaking a cyclist* • *Car accident* • ...
Camera-based corner cases	Pixel Error • *Dead pixel* • *Broken lense*	Pixel-Based Corner Case • *Dirt on lense* • *Overexposure*	Domain Shift on Single Frame • *Location (EU-U.S.A.)* • ...	Single-Point Anomaly on Single Frame • *Animal* • ...	Contextual/Collective Anomaly on Single Frame • *People on a billboard* • ...	
RADAR-based corner cases	Impulse Error • *Low voltage* • *Low temperature*	Impulse-Based Corner Case • *Interference* • ...	Domain Shift on Single Point Cloud • *Weather, e.g., snow, rain, etc.*	Single-Point Anomaly on Single Point Cloud • *Lost objects* • ...	Contextual/Collective Anomaly on Single Point Cloud • *Demonstration* • *Tree on street*	

Figure 3.1: Categorization of single-source corner cases based on used sensor, reprinted from [18]

objects are in unseen quantities or locations. Finally, the *temporal layer* addresses *scenario level* anomalies that span multiple scenes and involve analyzing patterns across a sequence of point clouds, such as detecting a person violating traffic rules, see Figure 3.1.

Furthermore, Heidecker et al. [18] also introduces a *method level* to complement the sensor-specific corner cases. This *level* pertains to anomalies that arise from applying specific methods, often due to a lack of knowledge or understanding. Distinguishing between a *method level* anomaly and a single-source anomaly, as discussed in [18], can be challenging since they can coexist and influence each other.

3.2 Anomaly Detection on 3D Data

Recently, there has been a surge of research in image-based anomaly detection [7, 12, 30, 21]. However, the field of anomaly detection on lidar data still exhibits significant gaps, as highlighted by Bogdoll [2], see Figure 3.2. In the subsequent discussion, I provide a concise overview of the proposed approaches for anomaly detection specifically tailored to lidar data.

Classical deep learning methods for object detection in lidar data operate under the closed-set assumption, limiting their ability to handle unknown objects during testing. Wong et al. [44], and Cen et al. [8] proposed open-set 3D object detection approaches to address this limitation. Among these, Cen et al.'s method, known as Metric Learning with Unsupervised Clustering (MLUC), employs metric learning techniques to identify regions containing unknown objects and refines bounding boxes using an unsupervised clustering algorithm. The MLUC method utilizes Euclidean distance-based probability to position embeddings of unknown objects at the center of the embedding space, facilitating their differentiation from known objects. In a different approach, Masuda et al. [29] utilize a reconstruction-based method for anomaly detection in lidar data by employing a Variational Autoencoder (VAE). The discrepancy between input and reconstructed point clouds is leveraged to indicate anomalies, with smaller differences indicating normal data and larger differences suggesting deviations from normality. Iqbal et al. [20] propose a technique to detect abnormal motion in point clouds by estimating scene flows and learning motion features. By clustering points based on distance metrics, the method identifies the closest object to the autonomous vehicle and converts it into a 3D grid structure known as Voxel-Carries-Flows (VoxCF). Dynamic features extracted from VoxCF, capturing the direction of an object's motion,

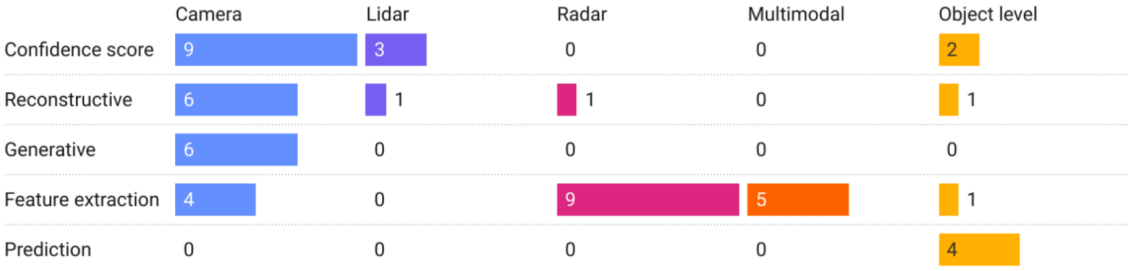


Figure 3.2: Overview of anomaly detection approaches based on camera, lidar, radar, multimodal, and abstract *object level* data, reprinted from [2].

are compared with predictions from a Long Short-Term Memory (LSTM) network to quantify abnormality. Addressing the impact of weather conditions on lidar scans, Zhang et al. [48] tackle the problem of lidar degradation in rainy weather using an anomaly detection model. They transform lidar point clouds into a 2D image representation, where changes in laser beam intensity due to degradation are associated with specific pixels. The Deep Semi-Supervised Anomaly Detection (DeepSAD) [37] model is employed to learn a hypersphere in the latent space, with non-degraded images mapping close to the hypersphere center and degraded images mapping further away. During testing, the distance between the hypersphere center and the mapped image is a degradation score.

All of the methods above exclusively address the *domain*, *object*, or *scenario level*, leaving a gap in anomaly detection on the *scene level* in lidar data. The method proposed by Masuda et al. [29] does not apply to autonomous driving, as it does not operate on complete point clouds. Zhang et al. [48] focus explicitly on differentiating between normal and rainy weather conditions, while Iqbal et al. [20] only detect anomalies in the nearest object to the vehicle.

In contrast to the anomaly detection approaches mentioned above, Sartoris [38] introduces a novel method for detecting anomalies in lidar data. Sartoris exploits the limitations of existing methods to identify anomalies at the *method level*. Anomalies can be detected by identifying inconsistencies in point-wise motion labels between the SV and SSV methods. Sartoris's approach revolves around evaluating the individual motion of each point to determine whether it is static or dynamic rather than focusing on abnormal motion, as emphasized in Iqbal et al.'s method [20]. The terms SV and SSV refer to different training methods used in anomaly detection. SV training involves supervised learning, where a model learns to map input and output data by comparing predictions to manually labeled data. This requires extensive manual labeling efforts. On the other hand, SSV training is unsupervised and relies on the model to generate supervisory signals based on the data's structure. SSV training offers advantages in terms of scalability and not requiring manual labeling. Both SV and SSV models are employed for anomaly detection in lidar data. The approach consists of two parts: the SV part predicts semantic motion labels using a semantic segmentation model and a motion segmentation model. In contrast, the SSV part uses a scene flow model and an odometry model to predict motion labels. Sartoris defines a corner case as a discrepancy in labeling between the two methods, enabling the detection of anomalies regardless

of external conditions. Moreover, this approach considers the significance of corner cases in terms of their impact on driving behavior [18], particularly when static and dynamic objects are taken into account.

3.3 Research Gap

Sartoris [38] acknowledged that, during his work, only one dataset was available for anomaly detection on lidar data, namely CODA [25]. However, since the publication of CODA was relatively recent and close to the time of his study, Sartoris relied on a purely qualitative evaluation for assessing anomaly detection. In this thesis, I will re-evaluate Sartoris's work and conduct a quantitative evaluation using an existing ground truth dataset designed for corner cases. In the next Chapter 4, I will provide a concise overview of existing datasets for anomaly detection and substantiate why CODA currently stands as the most suitable dataset for evaluating Sartoris's approach.

4 Evaluation Data

This chapter concisely presents an overview of datasets for anomaly detection in autonomous driving scenarios. The chapter introduces the tool *ad-datasets* and identifies datasets suitable for anomaly detection. It discusses the rationale for selecting the CODA dataset and provides a brief introduction to the CODA dataset itself, along with the challenges it poses.

4.1 Data Selection

The number of annual publications on autonomous driving datasets has been consistently increasing over the years [3]. However, there is a lack of good tools that can provide researchers with a quick, comprehensive, and up-to-date overview of datasets and their features. To address this issue, Bogdoll et al. introduced an online tool called "ad-datasets" [3]. This tool effectively tackles the problem by offering an extensive overview of more than 150 datasets, allowing users to sort and filter them based on 16 categories. Consequently, researchers gain a comprehensive understanding of the current landscape of autonomous driving datasets.

An in-depth analysis of 40 datasets reveals that all but one utilize camera data, while 23 incorporate lidar data. Radar data, however, is only used in eight datasets, none published before the end of 2018. Further examination focusing on anomaly detection narrows down the dataset selection to only seven. Unfortunately, none of these datasets include annotations for lidar point clouds or any lidar data, for that matter.

Bogdoll et al. [4] comprehensively covers eight anomaly datasets, with five of them overlapping with the seven datasets mentioned earlier. The paper provides a detailed analysis of these datasets, focusing on their value for anomaly detection. All the datasets discussed in the paper are publicly available as of February 1, 2023, and include sensor data captured from the ego perspective, along with pixel- or point-wise anomaly labels. These datasets primarily focused on *object-* and *scene-level* anomalies and were specifically created to evaluate anomaly detection methods. They can be categorized into three groups: datasets with real anomalies in real-world scenarios, datasets with synthetic anomalies augmented into real-world data, and datasets with completely synthetic scenes. Detecting such atypical and hazardous situations is crucial for ensuring the safety of all road users, making anomaly detection a critical factor for scaling autonomous vehicles.

Anomaly datasets can be classified based on six different techniques, *anomaly sources*, used to create them, as Bogdoll et al. [4] outlined. The first technique, called *Automated OOD Proposal*, involves using an automated proposal method to generate anomaly proposals, which are manually reviewed and refined to eliminate false positives. The *Misc Classes* technique involves relabeling all regions labeled as *void* or *misc* by human experts. *Class Exclusion* is performed on

Dataset	Year	Sensors	Size (Test/Val)	Anomaly Source	Temporal	OOD Classes	Groundtruth
Fishyscapes							
FS Lost and Found	2019	Camera	275 / 100	Recording	✗	1	Semantic Mask
FS Static	2019	Camera	1,000 / 30	Data Augmentation	✗	1	Semantic Mask
CAOS							
StreetHazards	2019	Camera	1,500	Simulation	✓	1	Semantic Mask
BDD-Anomaly	2019	Camera	810	Class Exclusion	✗	3	Semantic Mask
SegmentMeIfYouCan							
RoadAnomaly21	2021	Camera	100 / 10	Web Sourcing	✗	1	Semantic Mask
RoadObstacle21	2021	Camera	327 (+55) / 30	Recording	✓	1	Semantic Mask
CODA							
CODA-KITTI	2022	Camera, Lidar	309	Void Classes	✗	6	Bounding Boxes
CODA-nuScenes	2022	Camera, Lidar	134	Void Classes	✗	17	Bounding Boxes
CODA-ONCE	2022	Camera, Lidar	1,057	Automated OOD Proposal	✗	32	Bounding Boxes
CODA2022-ONCE	2022	Camera, Lidar	717	Automated OOD Proposal	✗	29	Bounding Boxes
CODA2022-SODA10M	2022	Camera	4,167	Automated OOD Proposal	✗	29	Bounding Boxes
Wuppertal OOD Tracking							
Street Obstacle Sequences (SOS)	2022	Camera, Depth	1,129	Recording	✓	13	Instance Mask
CARLA-WildLife (CWL)	2022	Camera, Depth	1,210	Simulation	✓	18	Instance Mask
Misc							
Lost and Found	2016	Stereo Cameras	2,104	Recording	✓	42	Semantic Mask
WD-Pascal	2019	Camera	70	Data Augmentation	✗	1	Semantic Mask
Vistas-NP	2020	Camera	11,167	Varying Class Exclusion	✗	4	Semantic Mask

Table 4.1: Overview over all analyzed datasets, clustered by the benchmark in which they were presented, adapted from [4]

Figure 4.1: WD-Pascal: Two examples of synthetically inserted anomalies into real-world scenes, reprinted from [4].

a labeled dataset by excluding frames containing known classes from the training and validation splits, treating the selected classes as anomalies. The *Web Sourcing* approach involves experts actively searching for images with atypical classes. *Recording and Simulation* methods collect anomalies by driving in the real world or generating them in synthetic environments. Lastly, *Data Augmentation* involves augmenting any dataset by synthetically adding anomalies to the original images. A review of the datasets reveals that most of them are based on these techniques, involving introducing additional data, creating artificial anomalies, or excluding specific classes, see Table 4.1. However, many of these anomalies are not highly realistic, as exemplified by the *WD-Pascal* dataset, which unrealistically employs data augmentation, see Figure 4.1.

CODA [25] stands out among all anomaly datasets as the only dataset incorporating both the *Automated OOD Proposal* and *Misc Classes* techniques. This distinction makes CODA unique, as it is based on real-world data encompassing situations where existing detection methods fail to identify objects. Consequently, CODA fulfills the novelty criteria outlined in my corner case definition. Moreover, all anomalies presented in CODA are categorized under the *Misc Classes*, indicating their potential impact on driving behavior since they are not confined to sidewalks or background regions. This characteristic also applies to anomalies generated through the *Automated*

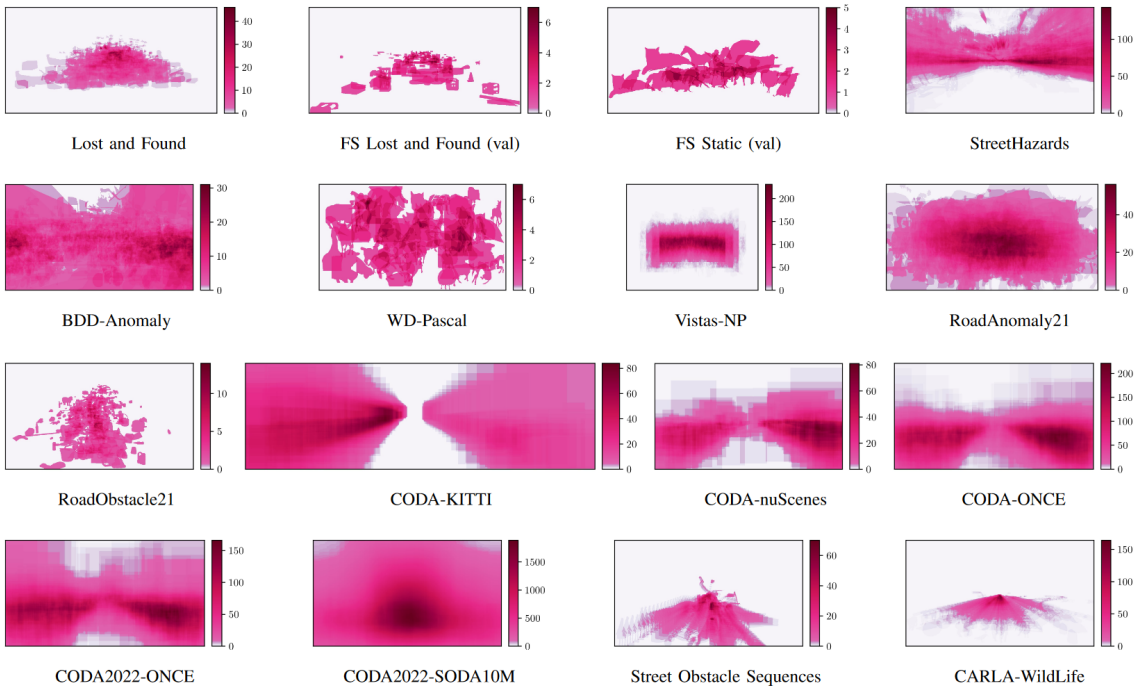


Figure 4.2: Cumulated masks of all contained anomalies within the respective datasets, reprinted from [4].

OOD Proposal, elaborated in Section 4.2 *Common Anomaly Detection in Autonomous Driving Dataset*. Notably, the anomalies in CODA are evenly distributed across the image space, setting it apart from most other datasets, see Figure 4.2. Given that Sartoris's approach relies on flow-based methods that require preceding frames to determine whether an object is static or dynamic, CODA becomes the ideal choice. Despite lacking temporal context, as indicated in Table 4.1, CODA remains an unaltered selection of raw datasets, allowing for the derivation of preceding frames from the original data. As Sartoris's work revolves around detecting corner cases using lidar data, the inclusion of lidar data is crucial. Consequently, the CODA-2022 dataset is excluded due to the absence of lidar data, specifically, the inclusion of SODA10M [16]. This leaves us with the Common Anomaly Detection in Autonomous Driving dataset, which comprises frames from ONCE [19], KITTI [15], and NuScenes [31]. The subsequent section introduces the Common Anomaly Detection in Autonomous Driving dataset and addresses the associated challenges.

4.2 CODA Dataset

As previously stated, there is a scarcity of publicly available datasets suitable for evaluating object detectors on corner cases. Addressing this gap, the Common Anomaly Detection in Autonomous Driving (CODA) dataset [25], referred to as CODA from now on, offers a comprehensive collection of nearly 6,000 *object level* corner cases across 1,500 images, encompassing over 30 object categories. In CODA, classes are denoted as categories, and henceforth, I will utilize the term "categories" to address them. The dataset's characteristics not only diminish the performance of standard object detectors to a mere 12.8% mAR, see Figure 4.3, but also pose significant chal-

CODA		ORIGIN		CORNER				COMMON				NOVEL			
Method	Dataset	AP	AR	AR^*	AR_{50}	AR_{75}	AR^{10}	AR^*	AR_{50}	AR_{75}	AR^{10}	AR^*	AR_{50}	AR_{75}	AR^{10}
RetinaNet† [22]	SODA10M [13]	34.0	50.7	**11.9**	25.2	9.5	5.4	28.7	58.9	23.5	23.9	-	-	-	-
Faster R-CNN† [34]		36.7	46.9	6.8	13.0	6.4	4.9	23.9	46.8	20.1	23.1	-	-	-	-
Cascade R-CNN† [5]		39.4	51.6	8.3	15.5	7.6	5.5	27.2	47.0	29.4	25.3	-	-	-	-
D-DETR [49]		31.8	49.4	7.2	16.7	4.9	3.6	**34.6**	60.2	36.5	29.6	-	-	-	-
Sparse R-CNN [39]		31.2	51.0	6.4	13.2	5.4	3.9	26.4	47.1	25.6	23.0	-	-	-	-
Cascade Swin [27]		41.1	52.9	8.2	15.5	7.6	5.7	30.4	51.3	32.2	29.3	-	-	-	-
RPN (Faster)† [34]		-	59.7	8.1	16.2	7.4	3.1	-	-	-	-	-	-	-	-
RPN (Cascade)† [5]		-	57.1	7.7	16.0	6.8	2.8	-	-	-	-	-	-	-	-
ORE [19]		**49.2**	**59.7**	8.3	16.4	7.4	5.6	18.5	35.5	18.2	18.1	**3.4**	7.6	2.8	2.9
RetinaNet† [22]	BDD100K [45]	28.6	40.4	**12.8**	23.2	11.9	4.8	27.5	58.1	21.5	23.6	**9.7**	17.7	9.1	5.9
Faster R-CNN† [34]		31.0	40.7	10.7	19.2	10.2	4.3	24.4	48.1	20.9	22.0	7.2	13.3	6.8	5.9
Cascade R-CNN† [5]		32.4	41.4	10.4	18.5	9.7	4.5	25.7	48.4	23.3	23.6	6.9	12.5	6.5	5.7
D-DETR [49]		28.5	42.3	9.0	22.2	5.6	2.8	28.5	63.0	22.3	26.2	7.0	17.3	4.3	3.9
Sparse R-CNN† [39]		26.7	40.2	9.8	19.0	8.9	4.5	27.4	51.7	25.8	24.3	8.0	15.4	7.4	5.1
Cascade Swin [27]		**34.5**	43.5	9.9	17.2	9.7	4.9	**31.0**	55.0	29.9	29.4	6.5	11.4	6.4	5.9
RPN (Faster)† [34]		-	50.2	10.6	20.0	10.2	3.7	-	-	-	-	-	-	-	-
RPN (Cascade)† [5]		-	**51.0**	10.6	20.0	10.2	3.9	-	-	-	-	-	-	-	-
RetinaNet [22]	Waymo [38]	39.7	47.7	8.4	15.6	7.7	5.1	24.5	43.2	24.4	22.2	6.7	11.9	6.4	4.6
Faster R-CNN [34]		40.9	47.0	6.8	12.4	6.4	4.8	20.9	36.0	19.6	19.1	5.5	9.6	5.2	4.3
Cascade R-CNN [5]		42.6	48.1	6.6	11.4	6.6	5.0	18.9	32.6	20.1	17.6	5.3	8.7	5.5	4.4
D-DETR [49]		40.4	49.8	7.3	15.8	5.4	3.6	28.5	49.4	24.6	22.5	5.2	11.5	4.0	3.0
Sparse R-CNN [39]		38.8	49.8	**10.1**	19.6	9.0	4.7	**29.5**	51.8	27.0	22.1	**7.6**	14.3	7.1	4.2
Cascade Swin [27]		**44.2**	49.0	5.4	8.7	5.5	4.4	21.8	38.1	18.8	21.3	4.3	6.7	4.6	3.7
RPN (Faster) [34]		-	**53.9**	7.5	13.7	7.5	3.6	-	-	-	-	-	-	-	-
RPN (Cascade) [5]		-	52.8	7.4	13.8	7.3	3.9	-	-	-	-	-	-	-	-

Figure 4.3: Detection results (%) on CODA, reprinted from [25].

lenges for the state-of-the-art open-world object detector in accurately identifying novel objects.

The CODA dataset is composed of selected images from three sources: ONCE [19], the KITTI object [15], and nuScenes [31]. It provides labels for seven supercategories: vehicle, pedestrian, cyclist, animal, traffic facility, obstruction, and misc, encompassing a total of 34 fine-grained categories. These categories can be divided into novel categories, such as dogs and strollers, and common categories, including cars and pedestrians. Notably, over 90% of all instances belong to novel categories. The dataset exhibits diversity at the *object-level*, reflecting real-world representations. Consequently, certain categories have only a limited number of instances due to the inherent scarcity of corner cases, while others, like traffic cones and barriers, are more prevalent in both frequency and quantity. At the *scene-level*, diversity is achieved by incorporating scenes from three distinct countries, each offering unique object variations and contributing to domain shifts. This is evident in the distribution of the top-four common categories from each source dataset, see Figure 4.4. Additionally, the dataset encompasses variations in weather conditions and daytime, further expanding the *scene-level* diversity.

As previously discussed, CODA satisfies the defined criteria for corner cases, as they present risks to the driving vehicle and exhibit novelty by not aligning with any of the common categories in SODA10M [16], making them challenging to detect. A two-stage approach is proposed to address the detection of these corner cases. The first stage involves an automated generation of proposals for potential corner cases, referred to as the *Automated OOD Proposal* by Bogdoll et

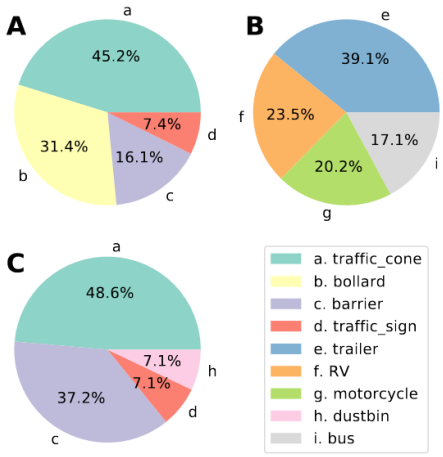


Figure 4.4: Distribution of the top-4 categories in: **A** ONCE, **B** Kitti, and **C** NuScenes, reprinted from [25].

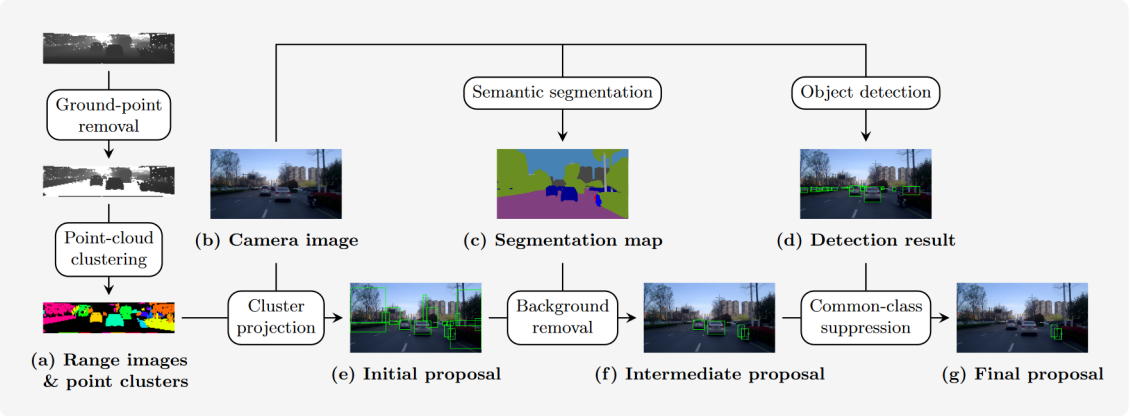


Figure 4.5: The Corner Case Proposal Generation (COPG) pipeline. The input to the pipeline includes the point cloud and the camera image of a given scene. The point cloud is used to compute (a), while the camera image (b) is utilized to produce (c) and (d). The results from (c) and (d) are then used to remove invalid proposals. The final output (g) consists of a set of bounding boxes indicating the proposed corner cases in the camera image, reprinted from [25].

al. [4]. This stage utilizes a novel pipeline called COPG, which leverages raw sensor data from the camera and the lidar sensor to identify potential corner cases, see Figure 4.5. Notably, the automatic proposal generation step is bypassed for the images sourced from KITTI and NuScenes, as these instances were manually selected, with all objects labeled as *misc* in the case of KITTI. For the second stage, CLIP [36] is employed for pre-labeling the objects, followed by the utilization of the toolkit developed by Wada et al. [42] to refine the labels manually.

As a result, the images are annotated with 2D bounding boxes for each anomaly. However, since Sartoris's approach [38] focuses on point-wise labeled lidar point clouds, the 2D bounding boxes must be projected into the 3D lidar space to label the lidar point clouds accurately. The dataset used in Sartoris's work, KITTI odometry [15], consists of 21 driving scenes and is organized into scenes with poses and calibration files in the KITTI odometry format. As described in Chapter 5.1, it is necessary to translate not only the KITTI object [15] dataset but also the ONCE and nuScenes datasets, which have distinct data formats compared to KITTI, into the KITTI odometry data

format.

5 Method

In this chapter, I will present the data processing steps involved in converting the CODA dataset into the KITTI odometry data format. I will explain the process of translating the 2D annotations of corner cases into 3D to generate accurate ground truth annotations for the lidar point clouds. Additionally, I will examine the methods proposed by Sartoris [38] and outline my approach for evaluating them using the ground truth data from CODA.

5.1 Data Processing

Certain adaptations were necessary to align the CODA dataset with the KITTI odometry dataset used by Sartoris [38]. The KITTI odometry dataset consists of 21 scenes, including images from *cam2* and *cam3*, as well as lidar point clouds. Each scene is accompanied by a calibration, timestamp, and poses file. The calibration file contains projection matrices for each camera and a transformation matrix for converting coordinates from lidar to camera space, see Figure 5.2. The timestamp files offer synchronized timestamps for the images and lidar point clouds, while the poses files supply the odometry data for the scenes. This odometry data is utilized by the SV motion segmentation component, as discussed in Section 5.3.

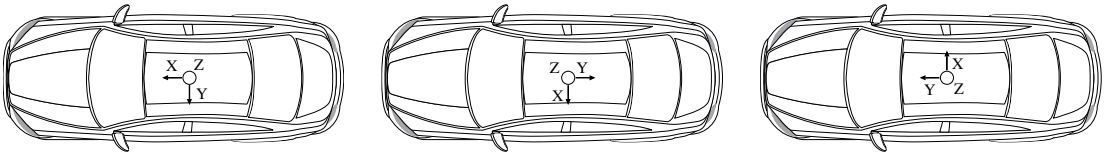
The downloaded CODA dataset includes ONCE-sourced images, a JSON file defining the images and classes used, and two JSON files for mapping images from the KITTI object and nuScenes to their original datasets. Since Sartoris' methods require only the left camera images (*cam2*), the translation process focuses on converting the images from other datasets to this format. Hence, determining the values *P2* and *Tr* is necessary for the calibration file.

To create a scene suitable for flow-based methods, I extracted a selection of eight preceding and eight subsequent images using the mappings to the original raw data of the datasets. For the ONCE dataset, an additional step involves applying the distortion matrix to undistort the images, similar to the originally included CODA-ONCE (ONCE) images in the CODA dataset.

The lidar coordinates of nuScenes and ONCE exhibit noticeable differences, see Figure 5.1. To address this, the lidar point clouds of nuScenes and ONCE needed to be rotated to align with the lidar coordinates of KITTI. This was achieved by simply rotating the point clouds.

The calibration file is crucial in translating lidar coordinates to the image space. Specifically for the *cam2* camera, this translation was achieved by using the *P2* (3x4) and *Tr* (3x4) matrices, as shown in formula 5.1. In the formula, *cam* represents the points in camera coordinates, while *points* refers to the points in lidar coordinates.

$$cam = P2 \times Tr \times points \quad [5.1]$$



(a) KITTI (b) ONCE (c) nuScenes

Figure 5.1: Lidar coordinate for each original dataset [14, 19, 31].

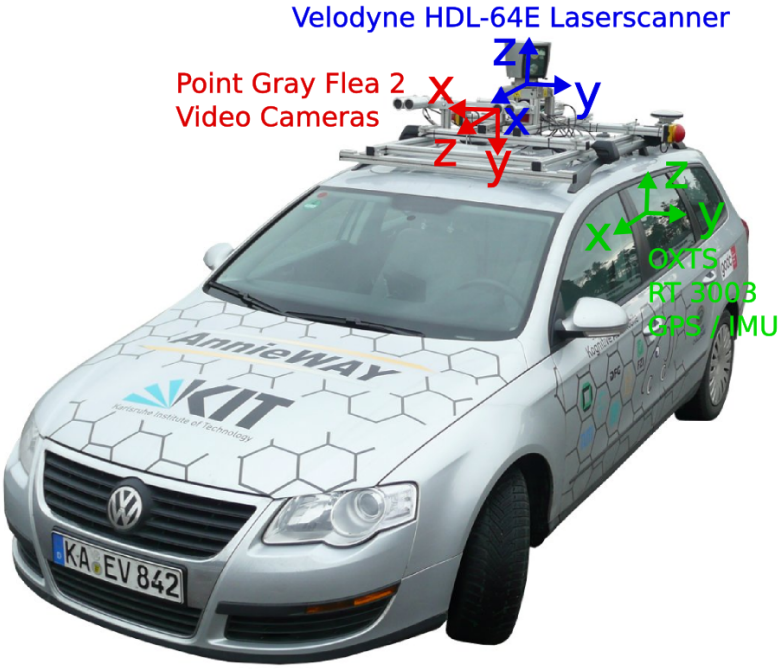


Figure 5.2: KITTI sensor coordinates, reprinted from [14].

For the KITTI object dataset, the *P2* matrix was already provided, and the *Tr* matrix was calculated using the following formula:

$$Tr = R0\_rect \times Tr\_velo\_to\_cam$$

In the case of the nuScenes dataset, the *Tr* matrix was obtained by combining the *lidar_to_ego* and *ego_to_cam* matrices. These matrices were derived by rotating the *translation* matrices of the *calibrated_sensors*. Additionally, the combined matrix was further rotated by an angle of $\pi/2$ around the z-axis. The *P2* matrix for nuScenes was obtained from the *camera_intrinsic* parameter of the raw data.

$$Tr = ego\_to\_cam \times lidar\_to\_ego \times R\left(\frac{\pi}{2}\right)$$

For the ONCE dataset, the *Tr* matrix was obtained by inverting the original *cam_to_velo* matrix and then rotating it by $-\pi/2$ along the z-axis to align with KITTI's lidar coordinates. The *P2* matrix was obtained using the *getOptimalNewCameraMatrix* function from the OpenCV package, using the camera's *cam_intrinsic* and *distortion* parameters.

$$Tr = cam\_to\_velo^{-1} \times R\left(-\frac{\pi}{2}\right)$$

The timestamps in the original data were given in milliseconds in UNIX time format. For ONCE, the timestamps were extracted from the file names, while for KITTI and nuScenes, separate files provided the timestamps. To align the timestamps with the KITTI odometry format, they were converted to be relative to the first timestamp in their respective raw data sequences and expressed in seconds.

The poses files in the KITTI odometry dataset consisted of (3x4) matrices for each frame, representing the translation from the i-th point to the first point of the scene. In the KITTI raw data [14], only the OXTS files were provided, which included GPS data for each frame. These GPS data could be used to calculate the pose for each frame relative to the first frame of the complete scene in the raw data. However, in the case of nuScenes and ONCE datasets, only ego poses were available. To derive the poses for each frame, the ego poses were first converted into GPS data, and then into poses for every frame. It should be noted that the GPS data in ONCE does not correspond to the real world due to the absence of starting GPS positions. However, since the poses were relative to the first point of the sequence, this discrepancy had no effect. Once the poses for every scene in the raw data were created, only the required poses from the original image, as well as the eight preceding and eight succeeding images, were copied to the new scene for CODA.

As a result, a single scene was generated for each image in CODA, consisting of 17 images, 17 lidar point clouds, and their corresponding timestamps and poses. A calibration file was also included to facilitate the translation from lidar to image coordinate space.

5.2 Data Labeling

The methods proposed by Sartoris [38] result in the point-wise labeling of the lidar point cloud for anomaly detection 5.3. However, CODA [25] provides only 2D bounding boxes for corner cases. Therefore, I needed to generate my own ground truth labels in the form of 3D point-wise annotations. This allowed me to evaluate Sartoris's work by comparing the labels generated using Sartoris's detection methods with the ground truth corner case labels.

In the Background section, I introduced the concept of frustums to narrow down the point cloud and propose clustering to accurately determine the 3D points of the corner cases. As discussed in the previous section, frustums are an effective way to restrict the point cloud based on the 2D bounding box by translating it into the 3D point cloud space. However, many approaches in the literature focus on localizing well-known objects such as cars and pedestrians [24, 43, 27]. These objects are relatively easier to locate using image processing [27] and are associated with a higher number of 3D points compared to most corner cases investigated in this thesis. Proposed approaches cannot be utilized in my thesis since most of the cited work relies on highly trained networks that require large datasets with labeled corner cases, which are not available at this time. Therefore, I propose a more labor-intensive approach to create point-wise labels for the corner cases. In this approach, I first used frustums to restrict the point cloud to the space of the 2D bounding box. Afterwards, I used clustering methods to determine the individual point of the actual anomaly.

The frustums were generated by first filtering the point cloud to include only the points in front of the vehicle. Then, the point clouds are translated into the image coordinate space using the transformation methods specified for each dataset. To account for perspective distortion, I divided each x-value and y-value by the corresponding z-value, bringing all the points onto a single 2D plane. From there, I further filtered the remaining point cloud based on the coordinates of the 2D bounding box. The filtered point cloud was preserved in both 2D coordinates and its original lidar coordinate space during the subsequent clustering steps. This preservation was necessary for further visual analysis as well as the ground truth for evaluation purposes.

For clustering, I used two of the three clustering methods proposed by Peng et al. [34], as they are suitable for clustering without a predefined number of clusters. The mean shift clustering algorithm is particularly effective for clustering data points of various shapes, while DBSCAN is better suited for simple and small objects. For the mean shift algorithm, automated bandwidth estimation with a quantile value of 0.3 is employed, which provides the best results across all three datasets contained in CODA. Through experimentation, I determined that for DBSCAN, using parameters $eps = 0.15$ and $min\_samples = 6$ yields the best results for all three datasets. Since DBSCAN is primarily designed to detect smaller objects, the point clouds get compressed by a factor of ten in the x-axis to improve the detection of flat surfaces as a single cluster rather than splitting them into multiple clusters.

The selection of the labeled cluster containing the anomaly involved identifying the five nearest

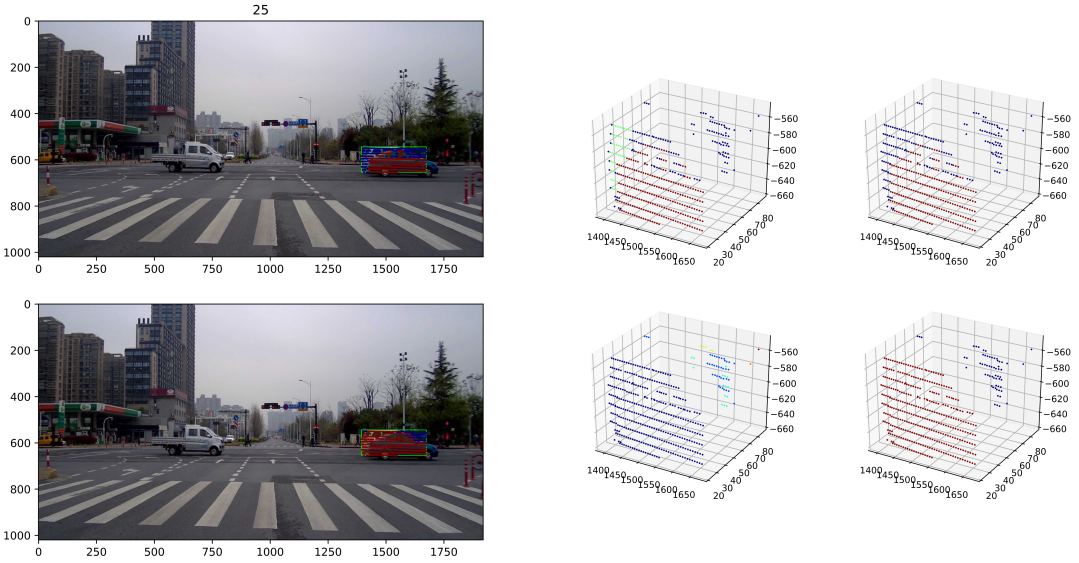


Figure 5.3: The manual evaluation is based on several visual representations. Firstly, the DBSCAN clustering results are depicted in an image, followed by presenting all identified clusters in the 3D scatter plot in the center. In the right 3D scatter plot, the selected cluster is highlighted in red. Similarly, the same process is repeated for the mean shift clustering, with the corresponding results displayed in the bottom row.

points in the 2D image space around the center point of the 2D bounding box. The point closest to the lidar sensor among these five points was chosen, and the cluster containing this selected point was labeled as *1*, while all other clusters received a label of *-1*.

As a result, binary files are obtained containing seven values for each point within the frustum. The first three values represent the original coordinates of the points in their respective data formats, corresponding to their lidar coordinate space. The fourth value denotes the label assigned by the DBSCAN clustering algorithm, where each integer value represents a distinct cluster, and outliers are denoted with *-1*. The fifth value indicates the selected cluster from the DBSCAN clustering, as described earlier, with either *-1* or *1*. The sixth and seventh values serve the same purpose as the DBSCAN labels but for the mean shift clustering algorithm.

To assess the suitability of the clustering, I conducted a manual inspection for each individual anomaly in the CODA dataset and assigned each anomaly a number to facilitate evaluating and selecting the appropriate cluster for the ground truth labeling.

1	*DB-SCAN right cluster*	2	*meanshift right cluster*	3	*both clusters are the same*
4	*mix of DB-SCAN and mean-shift*	5	*DB-SCAN wrong cluster*	6	*wrong clustering*
7	*wrong annotation*	8	*meanshift wrong cluster*	9	*too many small clusters*

The ground truth labeled point cloud was generated by combining the clustering results with the manual evaluation conducted earlier. The labels were copied from column five or seven, depending on the clustering algorithm used. For cases where both clusterings were the same, the labels were directly copied from the corresponding clustering. When the selection process resulted in the

wrong cluster being chosen, the negated clustering was copied to rectify the error, as it often indicated the background instead of the object. In situations where a mixture of both clustering results was deemed appropriate, a label of *1* was assigned if at least one of the clusterings classified the point as *1*, while all other points were assigned a label of *-1*. In cases where numerous small clusters were present, all points were labeled as *1*, as these clusters typically represented parts of a larger object incorrectly segmented. The single instance where the original bounding box contained no object was also labeled as *1*. Additionally, the misclassified points, accounting for seven percent of the entire dataset, were labeled as *1* to avoid discarding a significant portion of the data. These misclassifications primarily occurred due to factors such as bicycles on the side of the road, where each bicycle was labeled as a separate anomaly with very few lidar points.

The rotation method described in Section 5.1 was applied to the ONCE and nuScenes point clouds in order to align the lidar points with the labels generated by Sartoris' detection methods. As a result, pixel-wise ground truth annotations were generated for each annotation in the CODA dataset. Furthermore, the annotations within each of the 1500 original images were combined, as Sartoris' labeling is based on entire images, not individual anomalies. These results now facilitate the quantitative evaluation of Sartoris' approach.

5.3 Anomaly Detection

Sartoris's work integrates both SV and SSV methods for analyzing lidar data. The objective is to identify discrepancies between the semantic class assigned to a point using SV models and the motion label assigned to the same point using SSV models. Figure 5.4 depicts the interaction between these models.

The SV part, indicated in red in Figure 5.4, encompasses an SV semantic segmentation model and an SV motion segmentation model in Sartoris's work [38]. The SV semantic segmentation model assigns class labels to each point in the point cloud, distinguishing between static and dynamic classes. However, for certain classes such as *person*, which can exhibit both static and dynamic behavior, further subdivision into *moving person* and *standing person* is performed using the SV motion segmentation model. The resulting labels are referred to as *semantic motion labels*.

The SSV part, highlighted in green in Figure 5.4, comprises an SV ground segmentation, an SSV scene flow model, and an SSV odometry model. Before applying the scene flow model, a common step in SSV methods involves performing ground segmentation to preprocess the data. Many SSV scene flow models tend to discard points that fall below a fixed threshold, typically set at 0.3 m [23, 41, 1]. However, this static threshold does not account for road inclinations or the potential filtering out of small objects. Additionally, parts of larger objects, such as car tires, may be removed. Sartoris trains an SV ground segmentation model to filter out ground points specifically to address these limitations. The SSV scene flow model calculates a 3D displacement vector for each point in the first point cloud by considering two consecutive point clouds. To account for the ego motion of the vehicle, the SSV odometry model estimates a relative rigid body transformation between the two point clouds. This transformation allows for the transformation of the second point cloud into the coordinate frame of the first point cloud. The points are classified

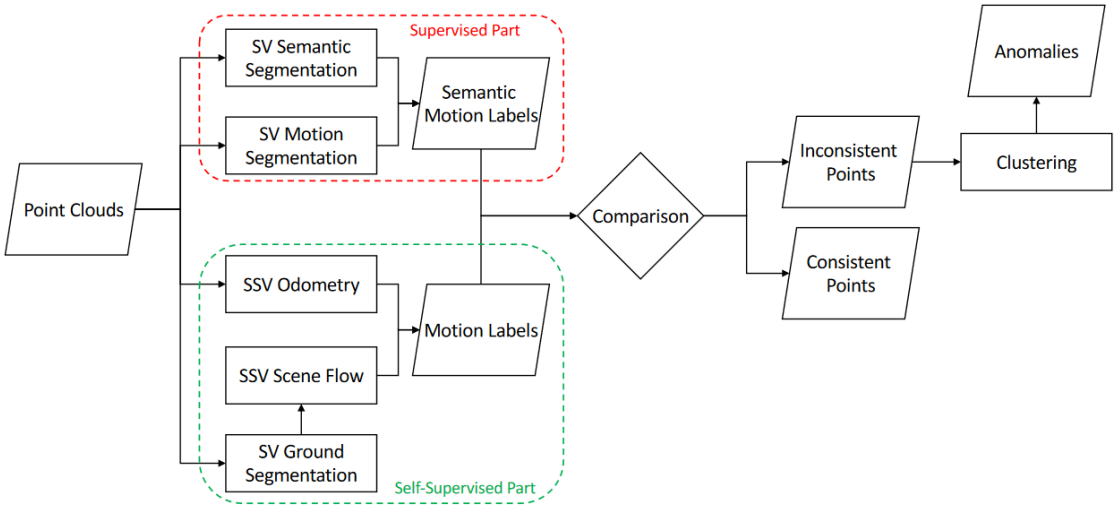


Figure 5.4: Sartoris proposed an approach [38] that consists of two main components: the SV part and the SSV part. In the SV part, each point is assigned a semantic motion class, which represents the motion state of the point. This assignment is achieved by combining an SV semantic segmentation model and an SV motion object segmentation model. These models provide information about the semantic category of the point and its motion characteristics, respectively. In the SSV part, a displacement vector is predicted for each non-ground point using an SSV scene flow model. Additionally, the ego-motion of the vehicle is predicted using an SSV odometry model. By combining the predicted displacement vectors and ego motion, a motion label is assigned to each point. A comparison is then made between the motion labels obtained from the SV part and the SSV part for each point. Points that have inconsistent motion labels are identified as potential anomalies. These inconsistent points are clustered together, forming clusters that serve as indications of anomalies in the scene, reprinted from [38].

	SV Part	SSV Part	Consistent	Color
Scenario 1	static	static	✓	green
Scenario 2	dynamic	dynamic	✓	blue
Scenario 3	static	dynamic	✗	red
Scenario 4	dynamic	static	✗	yellow

Table 5.1: All possible scenarios that can occur when comparing the labels between the SV and the SSV part, reprinted from [38].

as either static or dynamic by applying a threshold to the resulting displacement vectors, yielding the *motion labels*.

The semantic motion labels and the motion labels are then compared point-wise. Inconsistencies arise when a point is labeled static in terms of semantic motion but is labeled as dynamic based on the motion label, and vice versa, see Table 5.1. Individual inconsistent points are clustered using the DB-SCAN algorithm to identify and localize anomalies. This clustering process results in labels for entire inconsistent objects, indicating anomalies within the context of the work. The colors green and blue signify consistency between the two models, while red and yellow represent inconsistencies, suggesting possible anomalies.

In my evaluation, I utilized the pre-trained models from Sartoris's work, which were trained on the KITTI-360 dataset.

5.4 Evaluation Method

In the evaluation of the anomaly detection method, I considered the entire image by evaluating all points in the image space as well as individual anomalies in isolation. The comparison was conducted between the original labels generated by Sartoris' methods and the 3D ground truth data obtained from the CODA dataset. It should be noted that the comparison was performed solely with the original images, disregarding the additional eight preceding and eight subsequent images. By combining the two point clouds and preserving all the original information, new labels were created. The labels were combined based on the presence of points in both point clouds, the ground truth, or the detection method's labels. The resulting labels were represented by combinations of numbers, such as -12 for points present in both with a ground truth label of -1 and a detection method label of 2. For points that exist solely in the ground truth data, their labels were combined with a 0. For instance, if a point was labeled as *1* in the ground truth but does not have a corresponding label in the detection method, it would be assigned a label of *10*. Points solely labeled by the detection method retained their original label. Table 5.2 provides an overview of these label combinations and their corresponding positions in the confusion matrix. Every position within the confusion matrix corresponds to a specific color, which is utilized to visually represent the points in Figure 5.5. This image displays all the scenarios collectively, providing a visual representation of the associated colors for each point. The color coding scheme follows Sartoris' approach 5.1, with blue and green representing positive cases and red and yellow

	GT Label	Detected Label	New Label	Consufion Matrix	Color
Scenario 1	-1	1	-11	TN	blue
Scenario 2	-1	2	-12	TN	blue
Scenario 3	-1	3	-13	FP	red
Scenario 4	-1	4	-14	FP	red
Scenario 5	1	1	11	FN	yellow
Scenario 6	1	2	12	FN	yellow
Scenario 7	1	3	13	TP	green
Scenario 8	1	4	14	TP	green
Scenario 9	-1	-	-10	TN	blue
Scenario 10	1	-	10	FN	yellow
Scenario 11	-	1	1	TN	blue
Scenario 12	-	2	2	TN	blue
Scenario 13	-	3	3	FP	red
Scenario 14	-	4	4	FP	red

Table 5.2: Overview of all possible scenarios encountered during the comparison of labels between the ground truth and the anomaly detection method. The *New Label* represents the combined labels for each point, while the *Confusion Matrix* denotes the corresponding position of each scenario within the matrix. The *Color* column indicates the color assigned to visualize each scenario during the evaluation process.

representing errors.

To further analyze the individual data points, the evaluation was performed by partitioning the overall results based on the original datasets ONCE, KITTI, and nuScenes, as well as by reducing the spatial extent of the analysis. The reduction was achieved by considering only the points within the original 2D bounding box with corresponding ground truth labels (scenarios 1-10: *boxes*). Subsequently, a narrower subset was considered by including only the points that had labels from both the ground truth and the detection method (scenarios 1-8: *overlap*). All the parts based on reduction of the point cloud are shown in Figure 5.6. This approach aimed to assess the performance of the proposed method within smaller areas, as the comparison of the entire point cloud with only the anomalies makes it challenging to draw comparisons with other object detection methods. It is

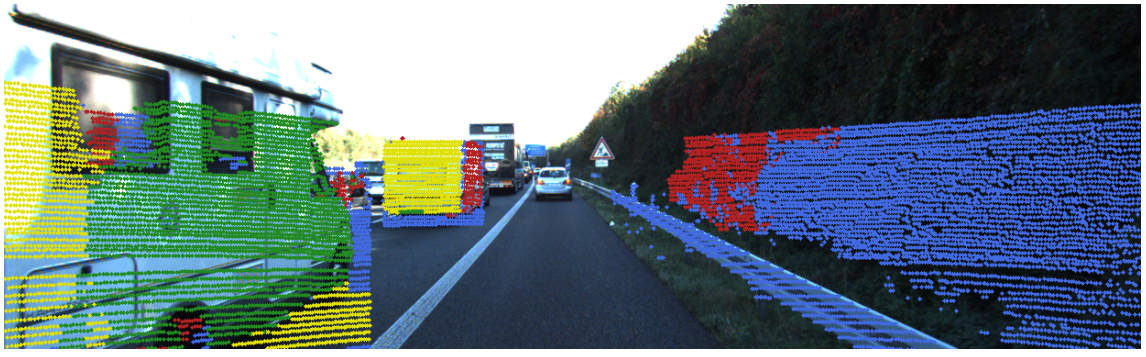


Figure 5.5: Image 1137 from CODA, an example of all possible scenarios in one image.

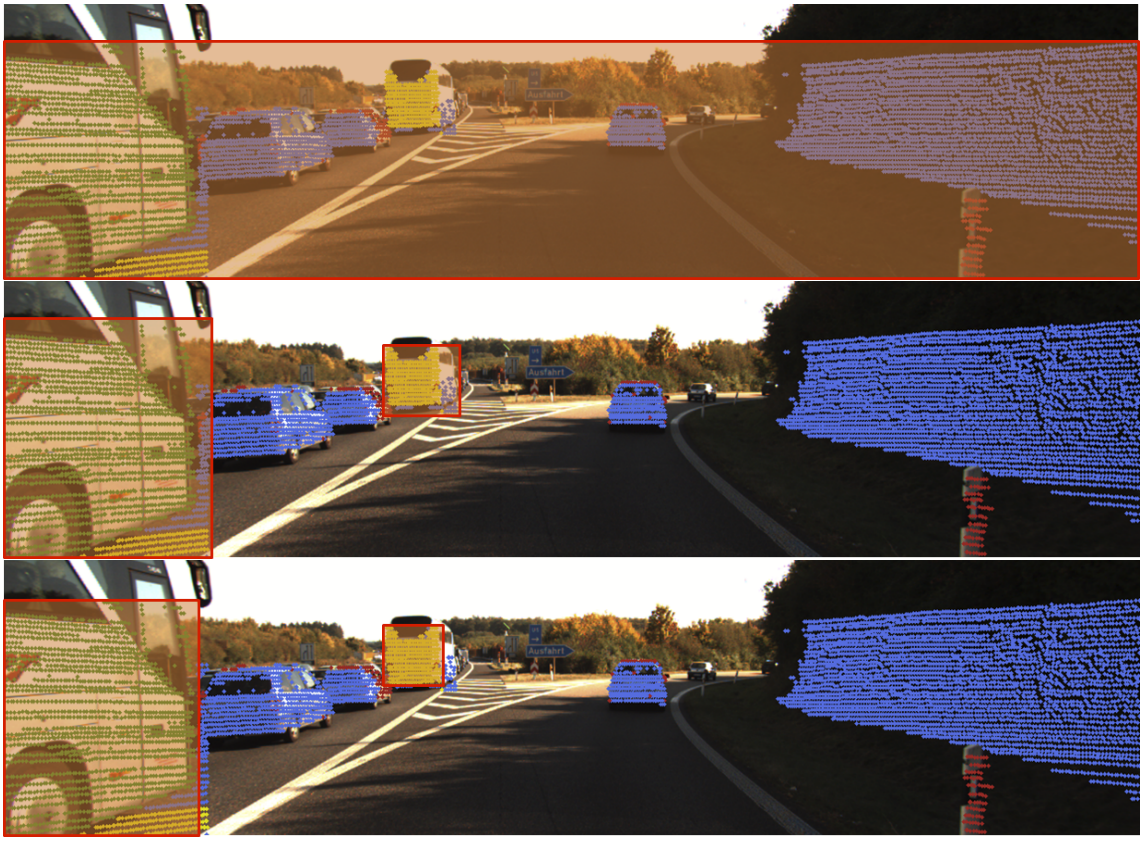


Figure 5.6: The visualized boxes represent the various aspects evaluated at the image level. The first image corresponds to the evaluation of all points from both the detection method and the ground truth combined (scenarios 1-14). The middle image corresponds to the part *boxes* (scenarios 1-10), and the bottom image corresponds to the part *overlap* (scenarios 1-8).

important to note that for the evaluation of individual anomalies, only the parts *boxes*, and *overlap* were calculated. This corresponds to a single box in the middle or bottom image in Figure 5.6. Evaluating all points detected by Sartoris' method would not be meaningful, as a single anomaly can be relatively small compared to the entire image. Therefore, focusing on these specific parts provides a more relevant assessment of the detection performance.

Figure 5.7 displays the resulting confusion matrix, which clearly illustrates the correspondence between each scenario and its corresponding position in the matrix. As previously mentioned, a pixel-wise analysis was conducted to label the points for a confusion matrix.

The evaluation of the confusion matrices involved the utilization of several metrics, namely mIoU, AP, AR, and F1. Notably, the mIoU is the most commonly used metric in evaluating semantic segmentation, as observed in previous studies [11, 10]. The variable N denotes the total number of images or annotations present in the specific partitioning of the CODA dataset.

$$\text{mIoU}_{individual} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i + FN_i}$$

$$\text{mIoU}_{aggregated} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} FP_i}$$

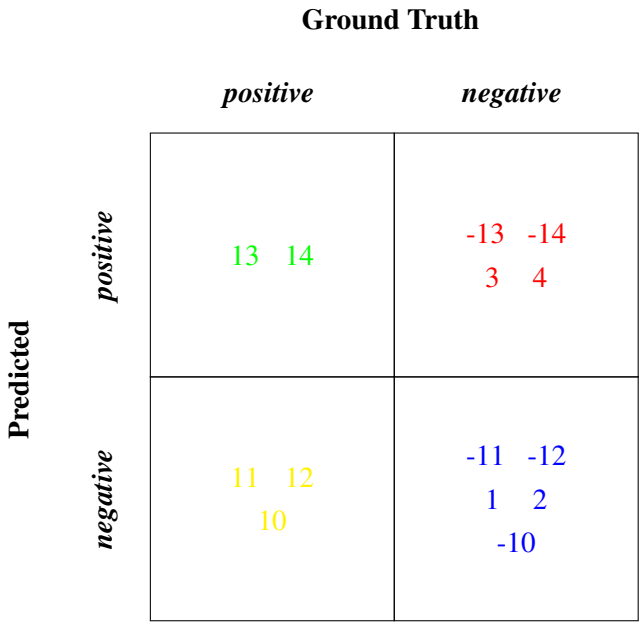


Figure 5.7: The confusion matrix for all scenarios in 5.2. The true positive (TP) labels are 13 and 14. The false positive (FP) labels are -13, -14, 3, and 4. The false negative (FN) labels are 11, 12, and 10. The true negative (TN) labels are -11, -12, 1, 2, and -10.

Considering that Sartoris' approach does not include single object detection and lacks prediction scores for individual objects, it can be considered as a semantic segmentation task and, therefore, evaluated accordingly. Li et al.[25] stated that the AP metric is not employed to evaluate different object detection methods on CODA due to their proficiency in detecting foreground objects regardless of whether they occupy the road, resulting in low AP scores. In contrast, the purpose of my detection method was to identify anomalies, which was why I had included the AP metric in my evaluation.

$$\mathrm{AP}_{individual} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i+FP_i}$$

$$\mathrm{AP}_{aggregated} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i+\sum_{i=1}^{N} FP_i+\sum_{i=1}^{N} FN_i}$$

AR is widely regarded as the most informative metric for anomaly detection methods in autonomous driving, as errors of the second type are of greater significance [25].

$$\mathrm{AR}_{individual} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i+FN_i}$$

$$\mathrm{AR}_{aggregated} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i+\sum_{i=1}^{N} FN_i}$$

The F1 metric was used to provide a condensed and comparable result based on the AP and AR values.

$$\mathrm{F1}_{individual} = 2 \cdot \frac{\mathrm{AP}_{individual} \cdot \mathrm{AR}_{individual}}{\mathrm{AP}_{individual} + \mathrm{AR}_{individual}}$$

$$\mathrm{F1}_{aggregated} = 2 \cdot \frac{\mathrm{AP}_{aggregated} \cdot \mathrm{AR}_{aggregated}}{\mathrm{AP}_{aggregated} + \mathrm{AR}_{aggregated}}$$

Due to the absence of prediction scores in Sartoris' method, this study could not utilize curve-based evaluation metrics like the Area Under the Precision Recall curve (AUPR) or the Area under the Receiver Operating Characteristics (AUROC).

As there is, to the best of my knowledge, currently no standardized method for applying these metrics to semantic segmentation data, I adopted two different approaches to address this issue, providing future researchers with options to evaluate against. Traditional object detection metrics typically classify predictions of objects or images as positive or negative based on a threshold, then compare them to the ground truth, and generate a confusion matrix for the entire dataset. However, since the labels produced by Sartoris' method were on the image level and the ground truth on the object level, I utilized individual points to construct confusion matrices, as shown in Table 5.2. My first approach involved creating a confusion matrix for each annotation or image, computing metrics for each of them, and then calculating the mean average of these metrics across all images, denoted as *individual metrics*. In the second approach, I aggregated points from all images to create a single confusion matrix containing all annotations or images, on which I applied the metrics, referred to as *aggregated metrics*.

To evaluate my own clustering, I additionally used all points contained in the 2D bounding box to calculate the metrics (*no clustering*). This corresponds to setting all *GT Label* to *1* and therefore bringing all scenarios with a negative *New Label* to the left side of the confusion matrix, in Figure 5.7.

The upcoming chapter presents the findings obtained from evaluating the generation of 3D ground truth data for the CODA dataset and assessing the performance of Sartoris' detection method against the ground truth.

6 Evaluation

This Chapter contains the quantitative evaluation of Sartoris' anomaly detection method on the CODA dataset. Firstly, I want to quickly evaluate the CODA dataset and the translation from the 2D bounding boxes to the 3D pixel-wise annotation. Afterwards, I will evaluate Sartoris' anomaly detection method, pointing out difficulties, and presenting some ideas on how to improve the detection method and the dataset to achieve better results.

6.1 CODA Evaluation

Containing data from different countries and varying weather and daylight conditions, the CODA dataset encompasses various *domain-level* anomalies [18]. However, I employed CODA to evaluate Sartoris' approach to detect anomalies at the *method level* [38]. Sartoris trained his models on KITTI360 to reduce the impact of *domain-level* anomalies, as it exhibits similarity to his evaluation dataset, KITTI odometry. For my evaluation, I utilized Sartoris' pre-trained models for all steps. With 309 images of the KITTI object dataset present in CODA, I not only evaluated the entire dataset but also performed separate evaluations for each original dataset, i.e., CODA-KITTI (KITTI), ONCE, and CODA-nuScenes (nuScenes). This allowed me to use KITTI as a baseline and compare its performance to ONCE and nuScenes.

The three parts of the CODA dataset not only differ at the *domain-level* but also in the categories of annotated anomalies. Among the 5937 anomalies in CODA, 4746 correspond to the supercategory *traffic_facility*, 929 to *vehicle*, and 197 to *obstruction*, as shown in Figure 6.1. All other anomaly supercategories together have only 65 instances. Notably, 396 of KITTI's 399 anomalies are *vehicles*, while ONCE and nuScenes mainly consist of *traffic_facility* anomalies.

6.2 3D CODA Groundtruth

As discussed in Chapter 5.2, I opted for simple clustering methods instead of complex machine learning approaches [24, 43, 27] to identify the points corresponding to anomalies in the 3D point cloud. To evaluate the effectiveness of this method, I visually represented the results of manual inspection by showing the percentage of each label in Figure 6.2. The graph on the left displays the overall performance across all anomalies in the CODA dataset, indicating that 79.8% of the clustering results accurately identified the anomalies in the 3D point cloud. In 9.2% of the cases, the DBSCAN clustering proposed incorrect clusters, primarily due to foreground objects situated near the center of the 2D bounding box, see Figure 6.3. Incorrect clustering in both methods occurred only when the anomaly itself was hard to detect. As illustrated in Figure 6.3, this is often related to anomalies with small surface areas stacked with other objects, such as bicycles on the

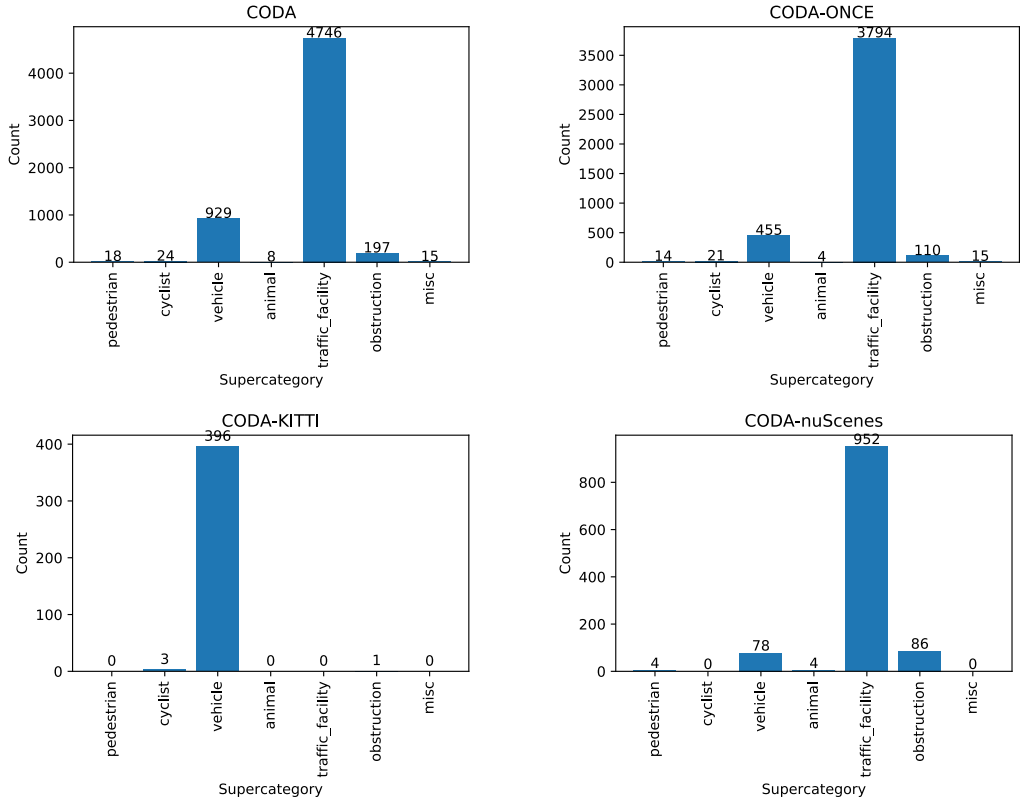


Figure 6.1: Number of occurrences of supercategories in the respective datasets.

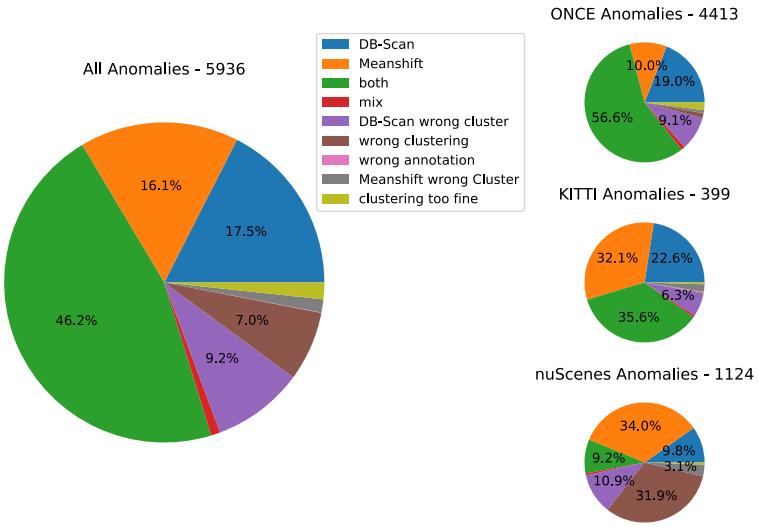


Figure 6.2: Results of the manual inspection of the two different clustering methods mean shift and DBSCAN. The different labels refer to those defined in 9. On the left is the distribution over all CODA anomalies, and on the right the distribution is split into the original datasets.

Figure 6.3: In the left image, an example is presented where a foreground object cluster is selected due to its proximity to the center of the 2D bounding box. In the right image, an illustration is provided of a fence-like structure with a low surface area.

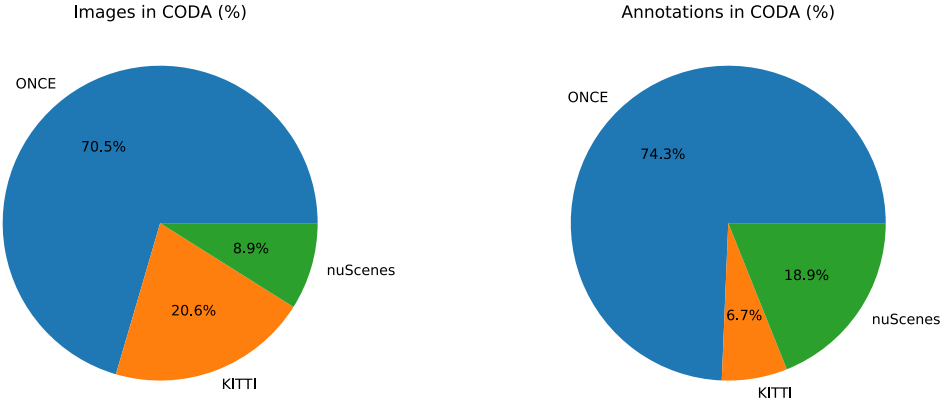


Figure 6.4: Distibution of images and anomalies in CODA across the three original datasets.

side of the road or small fence-like structures.

The differences between the original datasets are presented on the right side of Figure 6.2. Notably, there is an dissimilar distribution of anomalies in the different datasets. Despite CODA containing twice as many images from KITTI as from nuScenes, there are nearly three times as many anomalies in the nuScenes images compared to those from KITTI (Figure 6.4). Consequently, the 90.3% correct annotated clusters in KITTI were not given as much weight as the 31.9% of wrong clustering in nuScenes. Although mean shift clustering performed better on the KITTI and nuScenes datasets, the nearly one-to-two ratio between mean shift and DBSCAN in ONCE contributed to DBSCAN showing superior overall distribution performance. The lower clustering accuracy of 53% in nuScenes can be attributed primarily to the thin point cloud generated by the lidar sensor used in nuScenes, as depicted in Figure 6.5. This resulted in many anomalies being represented by only a few lidar points, especially for small objects like traffic cones situated far away, where some anomalies were not even detected by a single lidar point (Figure 6.7). For further details about individual supercategories see Figure A.1.

In addition to the findings mentioned above, it is worth highlighting the disparity in the anomalies present in CODA concerning the number of lidar points. Figure 6.6 illustrates that KITTI

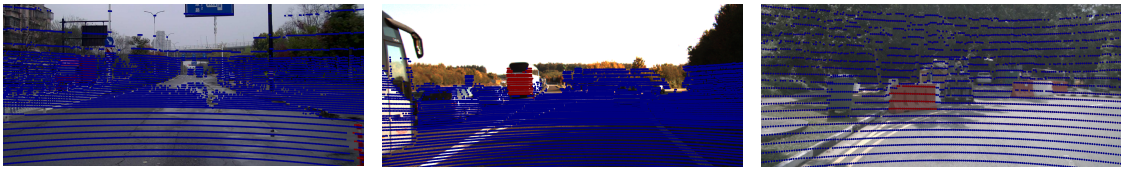


Figure 6.5: Difference in lidar point density between the different datasets: ONCE - left; KITTI - middle; nuScenes - right

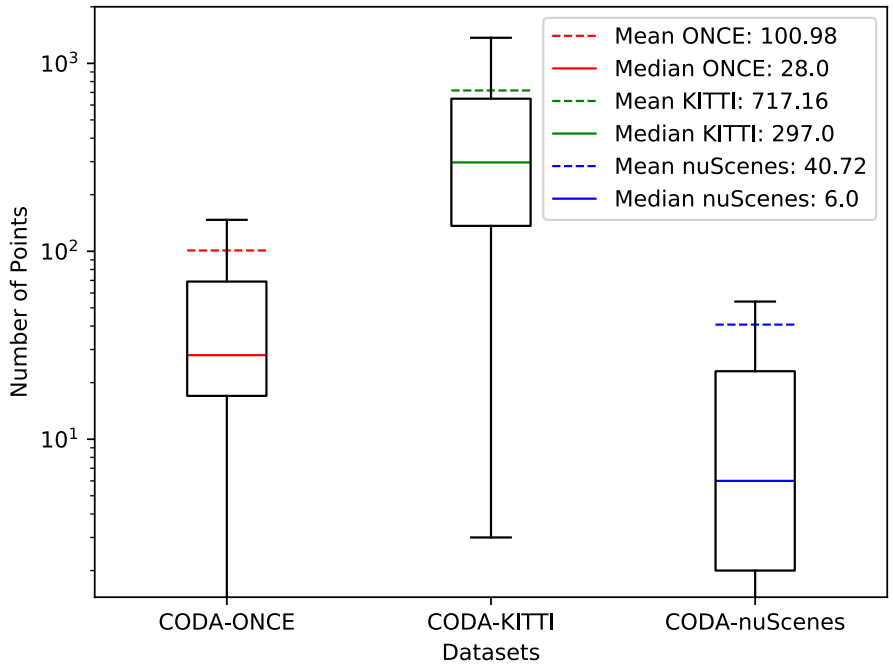


Figure 6.6: The distribution of points labeled as anomalies in the three datasets.

anomalies, as defined in the 3D lidar space, comprise a significantly higher number of lidar points compared to ONCE and nuScenes. All three datasets exhibit a notably lower median compared to the mean number of lidar points per anomaly. Furthermore, the distribution of positive and negative labeled points within the original 2D bounding box varies among the datasets. Specifically, nuScenes contains only 56.89% positive-labeled points, while ONCE and KITTI comprise 64.45% and 69.54% positive-labeled points, respectively.

The original assumptions made for the clustering methods in Chapter 2 regarding their different advantages were validated. Meanshift proved to be more effective in detecting larger and more complex objects with varying shapes, such as two cars from the side, where one is partly obstructed by the other (Figure 6.7). Whereas DBSCAN has its benefits in detecting plain surface areas against the background, as seen in the left image of Figure 6.3, although the wrong cluster was selected, both signs were clustered perfectly against the background.

Figure 6.7: In the left image, an example of a distant object is depicted, with no single lidar point hitting it (near the middle above the white car). The right image displays two cars, with one significantly occluding the other. The mean shift algorithm correctly clustered both cars together as a single cluster.

The primary challenge with my method of translating 2D bounding boxes into 3D point-wise labeling is the significant manual effort it demands to verify the labeling accuracy. In this study, it consumed approximately 12 hours to validate all 5937 annotations, averaging to approximately 500 annotations per hour. However, this level of manual inspection is not feasible for larger datasets due to time constraints and resource limitations.

Several additional steps and improvements can be implemented to enhance the translation process from 2D bounding boxes to 3D point-wise labels and reduce the need for manual verification. Firstly, a ground segmentation method, such as gndnet [32], could be applied to remove all ground points. For anomaly detection, simpler ground segmentation techniques, like segmenting points below a certain threshold, e.g. 30 cm [23, 41, 1] as ground, could not be effective, as many anomalies in datasets like CODA are small objects, such as traffic cones on the road, which would be separated by this method. Semantic segmentation could be utilized to further refine the point cloud to determine the position of the anomaly within the 2D bounding box. This information can then be used to create a frustum that encompasses the region of interest based on the semantic mask. However, this approach requires a semantic segmentation method capable of detecting novel object classes. Regarding clustering, improvements can be achieved by dynamically adjusting the clustering parameters based on the lidar point cloud characteristics of different datasets. Additionally, the selection of the final cluster could be optimized by considering the shapes and sizes of the clusters. Implementing these enhancements could lead to a more accurate and efficient conversion of 2D bounding boxes to 3D point-wise labels, making the process less reliant on manual verification.

6.3 Quantitative Anomaly Detection

For the quantitative analysis of Sartoris' detection approach [38] on the CODA dataset, I only used 1412 out of the 1500 images contained in CODA. This decision was made because 88 images lacked either eight preceding or eight subsequent images in their original dataset sequence. As these additional images are crucial for enabling the motion-based detection methods, they were omitted to maintain consistency with Sartoris' method and avoid potential results variations due to the reduced number of images.

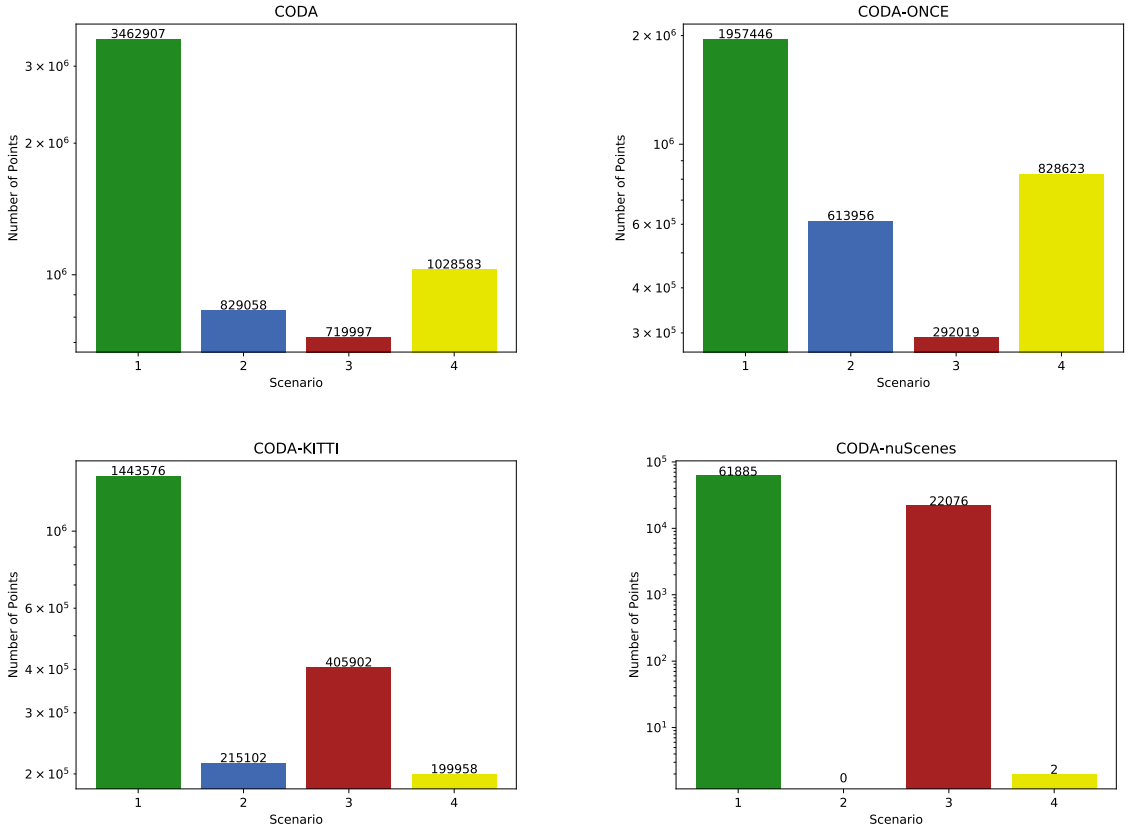


Figure 6.8: The top-left diagram represents the distribution for the CODA dataset, while the other three diagrams pertain to the individual components of the CODA dataset. Notably, all y-scales are logarithmized, and the scenarios correspond to those from Sartoris' original work 5.1.

To begin the analysis, the number of points labeled by the detection method was examined. Figure 6.8 presents the logarithmized number of points per scenario in the entire CODA dataset and splits the data into its individual components. Sartoris' original diagram is included for comparison in Figure 6.9. Notably, CODA contains fewer points labeled as static by the SV part and dynamic by the SSV part but a slightly higher number of points labeled as dynamic by both methods. Significant differences exist among the individual parts of CODA. For instance, nuScenes exhibits a considerable number of *scenario 3* points, while *scenario 4* and *scenario 2* points are relatively scarce. This could indicate potential issues with the nuScenes lidar format and the SV part, as it seems to detect very few dynamic points in nuScenes. The same observations apply to KITTI, where *scenario 4* dominates. It is crucial to acknowledge that nuScenes is underrepresented in the overall CODA dataset due to its lower total number of points compared to the smallest scenario in the other datasets.

Table 6.1 presents the evaluation results (in %) for Sartoris' detection method on the CODA dataset. The table is divided into different sections, namely *all*, *boxes*, and *overlap*, as described in Section 5.4. Additionally, the results are categorized according to the individual datasets within CODA. The metrics are further separated into *individual metrics* and *aggregated metrics*, as also

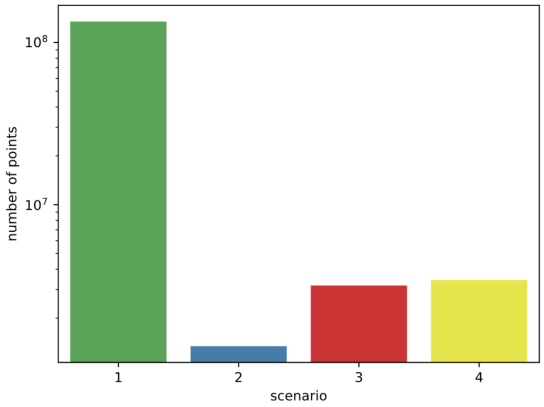


Figure 6.9: The diagram illustrates the original distribution of points across various scenarios for the KITTI odometry dataset, reprinted from [38]. The y-scale is logarithmized, and the colors are adopted from Sartoris' original scenario definition.

explained in Section 5.4. Similarly, Table 6.2 focuses on the examination of each individual anomaly instead of the entire image. In this case, only *boxes* and *overlap* evaluations are performed, as considering the entire point cloud would not be meaningful for assessing individual anomalies. All the metrics are derived from the confusion matrices presented in Figure A.2, and A.3.

The direct comparison of the quantitative evaluation of Sartoris' detection method with the object detection methods assessed by Li et al.[25] is limited due to the constraints discussed in Section 5.4. Nevertheless, it is worth noting that Sartoris' method outperforms all other object detection methods evaluated by Li et al. [25] in terms of the AR metric. While the best overall performance was achieved by RetinaNet [26] trained on BBD100K [47] with an AR of 12.8%, Sartoris' method obtained an AR of 26.2% for *individual metrics* and 33.5% for *aggregated metrics*.

The mIoU scores are relatively low compared to most standard object detection methods [11]. Notably, KITTI exhibits better overall performance across all metrics compared to ONCE. The strong similarities between ONCE and the overall CODA performance indicates the dominance of ONCE, representing 70.5% of the entire CODA dataset. nuScenes shows the poorest performance, with one outlier in the AP of the *aggregated metrics* for *boxes* and *overlap*. This outcome is likely attributed to the presence of numerous small overlapping anomalies in the nuScenes dataset.

Analyzing individual anomaly categories, as shown in Table 6.3, it becomes apparent that objects contained in normal object detection datasets, even when labeled as anomalies, outperform the average anomaly in CODA. Notably, the *pedestrian*, *cyclist*, and *car* categories exhibit superior performance to the average anomaly in CODA. It should be noted that these three categories combined only account for 120 of the 5937 anomalies in CODA. The *motorcycle* category with 89 occurrences in CODA stands out as the largest outlier, with an mIoU of 37.2% for *individual metrics* (38.7% *aggregated metrics*) and an AR of 52.0% for *individual metrics* (51.9% *aggregated metrics*). Detailed metrics for the individual supercategories and categories are provided in Tables A.1, A.2, and A.3 in the appendix.

Part	Dataset	Count	Individual Metrics				Aggregated Metrics			
			mIoU	AP	AR	F1	mIoU	AP	AR	F1
All										
	CODA	1412	**8.9**	**13.2**	**26.2**	**17.5**	**10.8**	**13.7**	**33.5**	**19.4**
	CODA-ONCE	1034	8.9	14.0	27.1	18.4	8.8	11.1	29.4	16.1
	CODA-KITTI	307	10.9	13.3	29.0	18.3	14.9	19.0	40.9	25.9
	CODA-nuScenes	71	0.4	0.7	1.0	0.8	0.4	0.6	1.4	0.9
Boxes										
	CODA	1412	**23.1**	**50.8**	**26.2**	**34.6**	**31.2**	**81.4**	**33.5**	**47.5**
	CODA-ONCE	1034	24.4	57.6	27.1	36.9	27.2	78.2	29.4	42.7
	CODA-KITTI	307	24.0	37.8	29.0	32.8	38.2	85.3	40.9	55.3
	CODA-nuScenes	71	1.0	7.6	1.0	1.8	1.4	69.1	1.4	2.7
Overlap										
	CODA	1304	**32.4**	**55.0**	**37.9**	**44.9**	**40.5**	**81.4**	**44.6**	**57.6**
	CODA-ONCE	985	33.9	60.5	38.8	47.2	36.6	78.2	40.7	53.6
	CODA-KITTI	250	35.0	46.4	44.2	45.3	47.1	85.3	51.3	64.0
	CODA-nuScenes	69	2.4	7.9	2.5	3.8	1.9	69.1	1.9	3.7

Table 6.1: The evaluation results (in %) on CODA are presented for entire images. *Individual Metrics* refer to the computation of the given metric for all points in each image, followed by calculating the mean of these metrics across all images. *Aggregated Metrics* are obtained by consolidating all points from all images into a single confusion matrix and then calculating the metrics based on this unified confusion matrix.

During my manual inspection of the clusters, I observed that the anomalies in CODA are well-defined and suitable for Sartoris' detection method. This observation is supported by the point distribution across different scenarios, as depicted in Figure 6.6. The proposed anomalies in CODA by Sartoris' detection method closely align with the findings in his original work on the KITTI odometry dataset.

Since the models are pre-trained on KITTI360, the anomaly detection results could be improved by retraining the models used in the SV and SSV part on the corresponding original datasets and splitting up the anomaly detection accordingly. To further improve the results, the settings of the individual methods could be adjusted for every dataset, e.g. the field of view for the motion segmentation and the clustering at the end.

The evaluation of the unclustered ground truth is depicted in Figure 6.4. As expected, the AP is considerably higher for the unclustered ground truth compared to the clustered metrics. This increase in AP is primarily due to a higher number of TP and unchanged FP points. Consequently, when considering the overall evaluated data, there is an 18.5% decrease in *individual* AP (18.5% *aggregated*) from the unclustered to the clustered scenario.

Since the FN have also increased, the impact on the mIoU is relatively small, with only a 3.3% decrease in *individual* (8.5% *aggregated*) mIoU. However, the crucial observation is that the increase in FN is more significant than the increase in TP, resulting in a 23.6% improvement in *individual* (18.4% *aggregated*) AR. It is important to note that AR is the most crucial metric for

Part	Dataset	Count	Individual Metrics				Aggregated Metrics			
			mIoU	AP	AR	F1	mIoU	AP	AR	F1
Boxes										
	CODA	5142	**19.6**	**34.4**	**22.7**	**27.3**	**31.2**	**81.2**	**33.6**	**47.5**
	CODA-ONCE	4321	21.1	37.5	24.4	29.6	27.3	78.1	29.6	43.0
	CODA-KITTI	397	22.5	34.3	27.7	30.7	38.3	84.9	41.1	55.4
	CODA-nuScenes	424	0.8	2.4	0.8	1.1	1.8	76.3	1.4	0.9
Overlap										
	CODA	4279	**29.2**	**41.3**	**34.9**	**37.8**	**40.5**	**81.2**	**44.6**	**57.6**
	CODA-ONCE	3693	30.9	43.9	36.6	39.9	36.8	78.1	41.0	53.8
	CODA-KITTI	305	34.6	44.7	45.0	44.8	47.0	84.9	51.3	64.0
	CODA-nuScenes	281	1.5	3.6	1.5	2.1	2.4	76.3	2.4	4.7

Table 6.2: The evaluation results (in %) on CODA are presented for single anomalies. *Individual Metrics* refer to the computation of the given metric for all points in each image, followed by calculating the mean of these metrics across all images. *Aggregated Metrics* are obtained by consolidating all points from all anomalies into a single confusion matrix and then calculating the metrics based on this unified confusion matrix.

anomaly detection [25], so the clustering of the point clouds has significantly enhanced the ground truth for anomaly detection.

Part	Category	Count	Individual Metrics				Aggregated Metrics			
			mIoU	AP	AR	F1	mIoU	AP	AR	F1
Boxes										
	Pedestrian	17	28.3	41.5	31.0	35.5	53.8	79.0	62.7	70.0
	Cyclist	24	32.5	53.4	38.6	44.8	38.0	76.6	42.9	55.0
	Car	79	35.8	64.1	37.3	47.1	36.0	85.8	38.3	52.9
	Motorcycle	89	37.2	52.2	52.0	52.1	38.7	60.3	51.9	55.8
Overlap										
	Pedestrian	16	33.9	44.1	37.3	40.4	57.7	79.0	68.1	73.2
	Cyclist	22	41.6	58.3	49.5	53.5	49.0	76.6	57.6	65.8
	Car	75	44.4	67.5	46.3	54.9	41.2	85.8	44.2	58.4
	Motorcycle	87	47.2	53.4	70.9	61.0	48.0	60.3	70.0	64.8

Table 6.3: The evaluation results (in %) on CODA are presented for selected single anomaly categories. *Individual Metrics* refer to the computation of the given metric for all points in each image, followed by calculating the mean of these metrics across all images. *Aggregated Metrics* are obtained by consolidating all points from all anomalies into a single confusion matrix and then calculating the metrics based on this unified confusion matrix.

Part	Dataset	Count	Individual Metrics$_{noclustering}$				Aggregated Metrics$_{noclustering}$			
			mIoU	AP	AR	F1	mIoU	AP	AR	F1
All										
	CODA	1412	**9.2**	**16.2**	**21.2**	**18.4**	**11.8**	**16.8**	**28.3**	**21.1**
	CODA-ONCE	1034	9.0	16.9	21.5	18.9	9.9	14.2	24.5	18.0
	CODA-KITTI	307	11.8	17.4	25.0	20.6	16.0	22.3	36.3	27.6
	CODA-nuScenes	71	0.3	1.0	0.6	0.7	0.5	0.9	1.1	1.0
Boxes										
	CODA	1412	**21.2**	**73.8**	**21.2**	**33.0**	**28.3**	**100.0**	**28.3**	**44.2**
	CODA-ONCE	1034	21.5	77.9	21.5	33.7	24.5	100.0	24.5	39.4
	CODA-KITTI	307	25.0	74.9	25.0	37.5	36.3	100.0	36.3	53.3
	CODA-nuScenes	71	0.6	8.5	0.6	1.1	1.1	100.0	1.1	2.3
Overlap										
	CODA	1304	**23.0**	**79.9**	**23.0**	**35.7**	**28.3**	**100.0**	**28.3**	**44.2**
	CODA-ONCE	985	22.6	81.8	22.6	35.4	24.5	100.0	24.5	39.4
	CODA-KITTI	250	30.8	92.0	30.8	46.1	36.3	100.0	36.3	53.3
	CODA-nuScenes	69	0.6	8.7	0.6	1.1	1.1	100.0	1.1	2.3

Table 6.4: The evaluation results (in %) on CODA are presented for entire images, with no previous clustering of the 2D bounding boxes. *Individual Metrics* refer to the computation of the given metric for all points in each image, followed by calculating the mean of these metrics across all images. *Aggregated Metrics* are obtained by consolidating all points from all images into a single confusion matrix and then calculating the metrics based on this unified confusion matrix.

7 Conclusion and Outlook

In this thesis the anomaly detection approach proposed by Sartoris [38] using SV and SSV models on the CODA dataset is evaluated quantitatively. To achieve this, I first create suitable 3D point-wise labeling by translating the different data formats of CODA's constituent datasets into a unified format. During this process, I highlight the benefits of adopting consistent data formats for future anomaly datasets merged from multiple sources, as it would streamline research efforts and enable more comprehensive evaluations. Furthermore, I emphasize the importance of considering differences among the datasets included in a fusion, as significant disparities, such as point cloud density, could impact the applicability of certain detection models. While CODA is well-suited for Sartoris' detection method, it is evident that nuScenes is less compatible, but it only makes up a small portion of CODA.

The clustering algorithms employed for translating 2D bounding boxes to 3D point-wise labels exhibit good performance. After manual inspection, approximately 79.8% of all anomalies are correctly clustered. With the correct selection of some wrongly clustered instances (9.2%), the overall percentage of accurately clustered anomalies is about 85%. Moreover, the evaluation of anomaly detection exhibited a noteworthy improvement of 23.6% *individual* AR in comparison to the unclustered dataset. I also propose additional improvements to enhance the detection rate and reduce the labor-intensive manual inspection, discussed at the end of Section 6.2.

The anomaly detection method produces point-wise labels for entire images, resembling a semantic segmentation of the entire scene without distinct anomaly proposals. Due to the lack of defined metrics for evaluating data from semantic segmentation methods other than mIoU, I used two different approaches for calculating AP, AR, and F1 metrics, providing future researchers with options for assessment. According to these metrics, Sartoris' method outperforms standard object detectors by a significant margin. This achievement is noteworthy despite the presence of approximately 15% incorrect clustering in the CODA ground truth and the fact that the detection method is not finely tuned for CODA. To improve performance even further, I suggest adjusting the settings of the different methods and retraining pre-trained models for each of the original datasets, as detailed in Section 6.3.

For future evaluations of anomaly detection methods, including original labels for all objects in anomaly datasets is crucial to enhance comparability between anomalies and non-anomalies. This inclusion would facilitate more accurate assessments and promote advancements in the field of anomaly detection.

7.1 Outlook

Anomalies are characterized by their scarcity in large labeled datasets, rendering them challenging for machine learning approaches to learn effectively. Therefore, a crucial aspect of anomaly labeling lies in clustering 3D point clouds based on straightforward principles. While I have proposed simple enhancements for the clustering method, other methods from different domains of object detection may already exist for detecting previously unknown objects in 3D point clouds.

With the successful completion of the first quantitative evaluation of Sartoris' work, promising avenues for further research have opened up. Fine-tuning Sartoris' method to address errors encountered in his qualitative evaluation, such as handling fast turns or speed bumps [38], is one direction for future exploration. Additionally, retraining the models for CODA or assessing Sartoris' methods on other anomaly datasets could yield valuable insights. Sartoris also suggests exploring the determination of the wrong part during inference, conducting in-depth analyses of individual components, or combining the SSV part with a closed-set object detector to detect anomalies at the *object-layer*.

The evaluation reveals a lack of suitable metrics tailored to this particular detection scenario. To address this issue, new indicators for evaluating the performance of such methods could be sought. For instance, including the labeling of all objects from the original datasets could enhance the assessment. Another approach involves examining the individual clusters formed for all inconsistent points and treating each as an individual anomaly rather than aggregating all points within an image. An alternative perspective to address this issue involves considering it from another angle. The results of standard object detectors on CODA can be transformed into my metrics by extracting the 3D lidar point cloud from the 2D bounding boxes proposed by the detectors. The comparison between the extracted point cloud and the ground truth labels can then be performed on a point-wise level, which follows the approach used in this thesis.

In conclusion, the quantitative evaluation has shed light on the potential of Sartoris' detection approach, and there are numerous avenues for further exploration and refinement to advance the field of anomaly detection in autonomous driving systems.

A Appendix

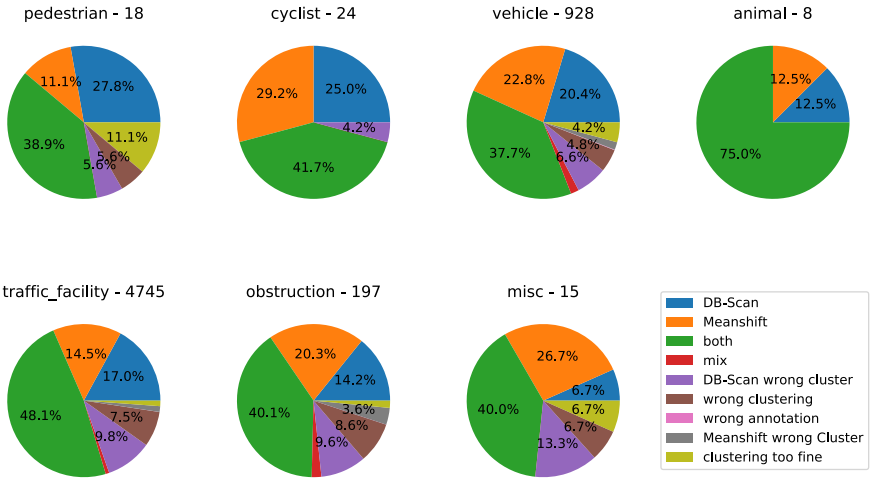


Figure A.1: Results of the manual inspection of the two different clustering methods mean shift and DBSCAN devided according to supercategory. The different labels refer to those defined in 9.

Part	**Supercategory**	**Count**	**Individual Metrics**				**Aggregated Metrics**				**Individual Metrics**$_{noclustering}$				**Aggregated Metrics**$_{noclustering}$			
			mIoU	AP	AR	F1	mIoU	AP	AR	F1	mIoU	AP	AR	F1	mIoU	AP	AR	F1
Boxes																		
	pedestrian	17	28.3	41.5	31.0	35.5	53.8	79.0	62.7	70.0	24.6	64.7	24.6	35.7	38.3	100.0	38.3	55.3
	cyclist	24	32.5	53.4	38.6	44.8	38.0	76.6	42.9	55.0	24.4	75.0	24.4	36.8	28.4	100.0	28.4	44.2
	vehicle	866	22.8	41.0	27.3	32.8	35.2	82.3	38.1	52.1	23.9	68.6	23.9	35.5	33.8	100.0	33.8	50.5
	animal	8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	traffic_facility	4062	19.1	33.0	22.0	26.4	25.3	78.9	27.2	40.4	17.0	46.9	17.0	25.0	21.7	100.0	21.7	35.7
	obstruction	150	11.1	28.3	12.0	16.8	14.2	79.7	14.7	24.9	9.5	38.0	9.5	15.2	12.4	100.0	12.4	22.1
	misc	15	25.2	60.7	25.8	36.2	27.4	88.6	28.4	43.1	18.4	73.3	18.4	29.4	19.2	100.0	19.2	32.2
Overlap																		
	pedestrian	16	33.9	44.1	37.3	40.4	57.7	79.0	68.1	73.2	26.2	68.8	26.2	37.9	38.3	100.0	38.3	55.3
	cyclist	22	41.6	58.3	49.5	53.5	49.0	76.6	57.6	65.8	26.6	81.8	26.6	40.1	28.4	100.0	28.4	44.2
	vehicle	736	33.0	48.3	41.0	44.4	43.3	82.3	47.7	60.4	28.2	80.7	28.2	41.8	33.8	100.0	33.8	50.5
	animal	5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	traffic_facility	3360	28.6	39.9	33.9	36.7	35.8	78.9	39.6	52.7	20.6	56.7	20.6	30.2	21.7	100.0	21.7	35.7
	obstruction	125	20.5	34.0	22.9	27.4	22.8	79.7	24.2	37.1	11.4	45.6	11.4	18.2	12.4	100.0	12.4	22.1
	misc	15	36.7	60.7	37.9	46.7	37.7	88.6	39.6	54.8	18.4	73.3	18.4	29.4	19.2	100.0	19.2	32.2

Table A.1: Evaluation results (in %) on CODA for single anomaly supercategories. *Individual Metrics* correspond to calculating the given metric for all points in every anomaly and calculating the mean of all these metrics. *Aggregated Metrics* are achieved by cumulating all points of all annotations together into one confusion matrix and calculating the metrics on this one confusion matrix. Additionally, the metrics for the unclustered ground truth are shown.

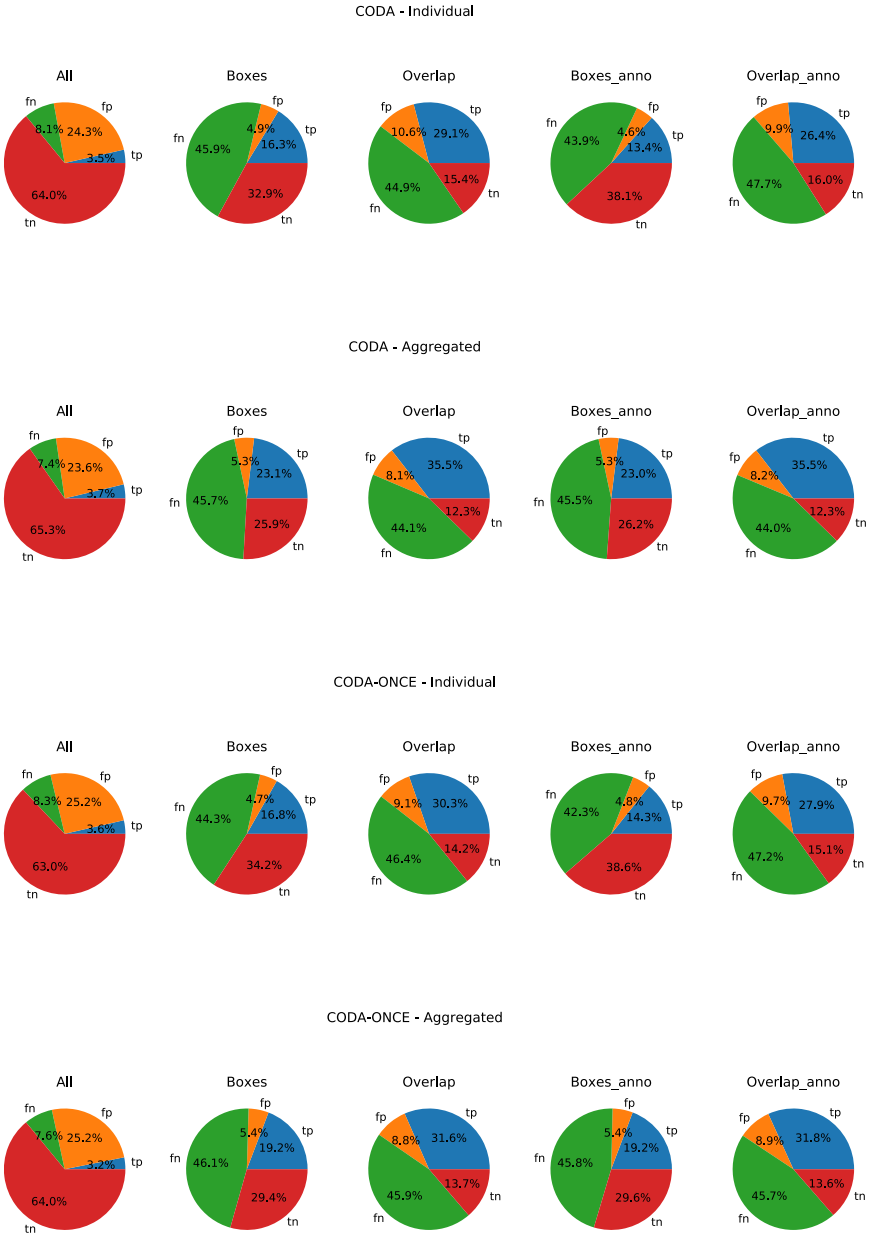


Figure A.2: Confusion matrices as piecharts for CODA and ONCE.

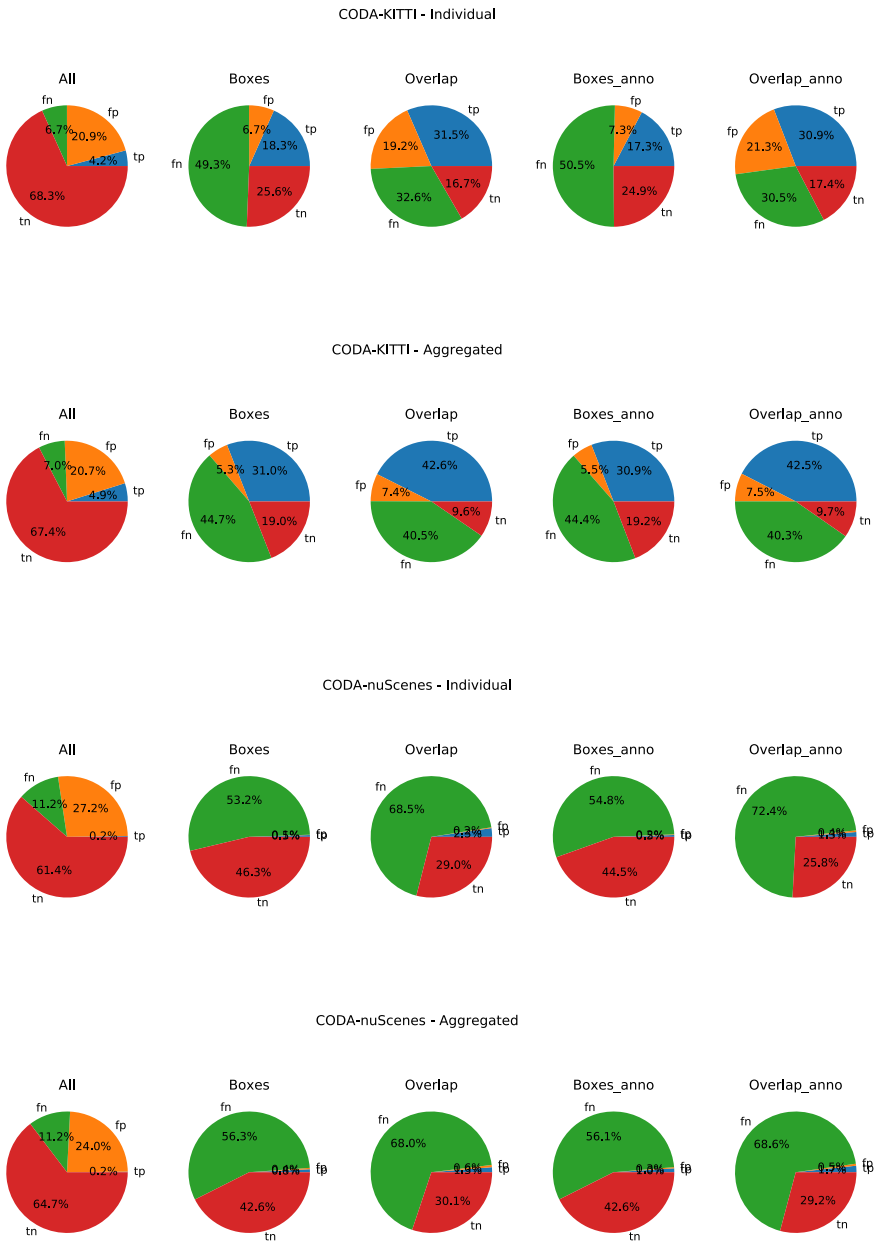


Figure A.3: Confusion matrices as piecharts for KITTI and nuScenes.

Part	Category	Count	Individual Metrics				Aggregated Metrics				Individual Metrics$_{noclustering}$				Aggregated Metrics$_{noclustering}$			
			mIoU	AP	AR	F1	mIoU	AP	AR	F1	mIoU	AP	AR	F1	mIoU	AP	AR	F1
Boxes																		
	pedestrian																	
	pedestrian	17	28.3	41.5	31.0	35.5	53.8	79.0	62.7	70.0	24.6	64.7	24.6	35.7	38.3	100.0	38.3	55.3
	cyclist																	
	cyclist	24	32.5	53.4	38.6	44.8	38.0	76.6	42.9	55.0	24.4	75.0	24.4	36.8	28.4	100.0	28.4	44.2
	vehicle																	
	car	79	35.8	64.1	37.3	47.1	36.0	85.8	38.3	52.9	34.3	84.8	34.3	48.8	33.9	100.0	33.9	50.6
	truck	6	28.4	67.0	30.2	41.6	28.7	81.8	30.7	44.6	29.5	83.3	29.5	43.6	31.0	100.0	31.0	47.3
	tram	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	tricycle	16	24.1	60.4	29.0	39.2	26.4	72.9	29.3	41.8	24.6	81.2	24.6	37.7	27.3	100.0	27.3	42.9
	bus	71	15.5	27.3	16.3	20.4	51.9	95.0	53.4	68.4	14.8	49.3	14.8	22.8	47.1	100.0	47.1	64.1
	bicycle	58	22.4	43.5	28.3	34.3	30.6	64.0	37.0	46.9	24.0	60.3	24.0	34.4	35.3	100.0	35.3	52.2
	moped	224	20.1	42.6	23.4	30.2	22.7	71.2	25.0	37.0	20.6	62.9	20.6	31.1	22.5	100.0	22.5	36.8
	motorcycle	89	37.2	52.2	52.0	52.1	38.7	60.3	51.9	55.8	44.7	88.8	44.7	59.5	43.0	100.0	43.0	60.2
	stroller	17	18.2	64.3	19.4	29.8	15.5	87.7	15.8	26.8	14.2	82.4	14.2	24.2	10.9	100.0	10.9	19.7
	wheelchair	11	6.7	8.7	6.9	7.7	15.6	95.5	15.7	27.0	6.0	9.1	6.0	7.2	9.8	100.0	9.8	17.9
	cart	23	22.8	51.2	30.6	38.3	23.1	61.9	27.0	37.6	24.5	78.3	24.5	37.4	24.4	100.0	24.4	39.2
	trailer	155	20.5	27.9	24.9	26.3	31.9	82.8	34.2	48.4	20.1	60.6	20.1	30.2	28.1	100.0	28.1	43.9
	construction_vehicle	25	13.1	26.0	15.8	19.7	26.7	76.1	29.2	42.2	11.2	36.0	11.2	17.0	24.8	100.0	24.8	39.7
	recreational_vehicle	92	19.2	33.5	21.5	26.2	36.7	90.1	38.3	53.7	23.3	90.2	23.3	37.1	36.6	100.0	36.6	53.6
	animal																	
	dog	8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	traffic_facility																	
	barrier	737	19.3	39.2	21.1	27.4	22.4	80.2	23.7	36.6	17.9	53.1	17.9	26.8	20.9	100.0	20.9	34.5
	bollard	1125	20.6	32.0	24.0	27.4	24.2	74.5	26.4	39.0	18.7	44.8	18.7	26.4	18.9	100.0	18.9	31.8
	warning_sign	3	16.5	66.7	16.5	26.5	14.1	100.0	14.1	24.7	12.1	66.7	12.1	20.5	9.0	100.0	9.0	16.4
	sentry_box	13	27.5	60.7	29.6	39.8	28.1	81.9	29.9	43.9	25.3	76.9	25.3	38.1	26.5	100.0	26.5	41.9
	traffic_box	1	36.3	100.0	36.3	53.3	36.3	100.0	36.3	53.3	26.9	100.0	26.9	42.5	26.9	100.0	26.9	42.5
	traffic_cone	1836	16.8	29.1	20.2	23.8	20.8	67.8	23.0	34.4	14.3	42.7	14.3	21.4	15.3	100.0	15.3	26.5
	traffic_island	33	12.2	48.2	12.3	19.6	16.8	86.6	17.3	28.8	11.8	60.6	11.8	19.8	14.1	100.0	14.1	24.7
	traffic_light	19	28.7	55.9	33.8	42.1	29.4	58.6	37.0	45.4	29.0	89.5	29.0	43.7	31.7	100.0	31.7	48.1
	traffic_sign	295	26.7	41.2	28.3	33.6	36.2	85.7	38.5	53.1	24.7	59.7	24.7	34.9	31.4	100.0	31.4	47.8
	obstruction																	
	debris	58	9.4	27.9	10.0	14.7	12.2	83.4	12.5	21.8	8.1	39.7	8.1	13.5	10.5	100.0	10.5	19.1
	suitcace	19	15.1	60.6	15.3	24.4	18.9	89.0	19.4	31.8	9.0	68.4	9.0	15.9	9.3	100.0	9.3	17.0
	dustbin	47	10.9	18.1	11.6	14.2	15.6	96.1	15.7	27.0	9.1	19.1	9.1	12.3	12.0	100.0	12.0	21.4
	concrete_block	12	7.0	16.1	7.3	10.1	8.5	62.1	9.0	15.7	6.9	25.0	6.9	10.9	11.3	100.0	11.3	20.2
	machinery	4	37.7	74.5	38.1	50.4	29.1	99.0	29.1	45.0	35.2	75.0	35.2	47.9	25.8	100.0	25.8	41.0
	chair	4	6.4	11.6	9.1	10.2	9.3	44.2	10.5	17.0	9.8	50.0	9.8	16.4	13.9	100.0	13.9	24.4
	phone_booth	2	29.6	30.5	47.5	37.2	23.0	42.2	33.5	37.3	36.3	100.0	36.3	53.2	29.2	100.0	29.2	45.2
	basket	4	1.1	8.3	1.2	2.1	0.8	14.3	0.9	1.7	4.1	50.0	4.1	7.5	4.0	100.0	4.0	7.8
	misc																	
	misc	15	25.2	60.7	25.8	36.2	27.4	88.6	28.4	43.1	18.4	73.3	18.4	29.4	19.2	100.0	19.2	32.2

Table A.2: Evaluation results (in %) on CODA for single anomaly categories. *Individual Metrics* correspond to calculating the given metric for all points in every anomaly and calculating the mean of all these metrics. *Aggregated Metrics* are achieved by cumulating all points of all annotations together into one confusion matrix and calculating the metrics on this one confusion matrix. Additionally, the metrics for the unclustered ground truth are shown.

Part	Category	Count	Individual Metrics				Aggregated Metrics				Individual Metrics$_{noclustering}$				Aggregated Metrics$_{noclustering}$			
			mIoU	AP	AR	F1	mIoU	AP	AR	F1	mIoU	AP	AR	F1	mIoU	AP	AR	F1
Overlap																		
	pedestrian																	
	pedestrian	16	33.9	44.1	37.3	40.4	57.7	79.0	68.1	73.2	26.2	68.8	26.2	37.9	38.3	100.0	38.3	55.3
	cyclist																	
	cyclist	22	41.6	58.3	49.5	53.5	49.0	76.6	57.6	65.8	26.6	81.8	26.6	40.1	28.4	100.0	28.4	44.2
	vehicle																	
	car	75	44.4	67.5	46.3	54.9	41.2	85.8	44.2	58.4	36.1	89.3	36.1	51.4	33.9	100.0	33.9	50.6
	truck	5	36.2	80.4	38.7	52.3	31.6	81.8	34.0	48.1	35.4	100.0	35.4	52.3	31.0	100.0	31.0	47.3
	tram	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	tricycle	16	35.9	60.4	42.3	49.8	31.6	72.9	35.8	48.0	24.6	81.2	24.6	37.7	27.3	100.0	27.3	42.9
	bus	35	38.7	55.4	40.6	46.8	73.0	95.0	75.8	84.4	30.0	100.0	30.0	46.2	47.1	100.0	47.1	64.1
	bicycle	52	34.2	48.6	45.6	47.0	41.4	64.0	54.0	58.6	26.8	67.3	26.8	38.4	35.3	100.0	35.3	52.2
	moped	205	29.0	46.5	35.2	40.1	30.9	71.2	35.3	47.2	22.5	68.8	22.5	34.0	22.5	100.0	22.5	36.8
	motorcycle	87	47.2	53.4	70.9	61.0	48.0	60.3	70.0	64.8	45.7	90.8	45.7	60.8	43.0	100.0	43.0	60.2
	stroller	17	23.5	64.3	25.1	36.1	20.7	87.7	21.3	34.2	14.2	82.4	14.2	24.2	10.9	100.0	10.9	19.7
	wheelchair	11	8.4	8.7	8.8	8.7	24.7	95.5	25.0	39.6	6.0	9.1	6.0	7.2	9.8	100.0	9.8	17.9
	cart	23	27.9	51.2	40.0	44.9	29.6	61.9	36.2	45.7	24.5	78.3	24.5	37.4	24.4	100.0	24.4	39.2
	trailer	109	33.5	39.7	42.8	41.2	40.0	82.8	43.6	57.1	28.6	86.2	28.6	43.0	28.1	100.0	28.1	43.9
	construction_vehicle	18	19.4	36.2	23.3	28.3	33.1	76.1	36.9	49.7	15.5	50.0	15.5	23.7	24.8	100.0	24.8	39.7
	recreational_vehicle	83	23.2	37.2	26.0	30.6	42.3	90.1	44.3	59.4	25.9	100.0	25.9	41.1	36.6	100.0	36.6	53.6
	animal																	
	dog	5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	traffic_facility																	
	barrier	547	34.4	52.8	38.2	44.3	34.2	80.2	37.4	51.0	24.2	71.5	24.2	36.1	20.9	100.0	20.9	34.5
	bollard	973	28.3	37.0	33.6	35.2	31.5	74.5	35.3	47.9	21.6	51.8	21.6	30.5	18.9	100.0	18.9	31.8
	warning_sign	3	33.9	66.7	33.9	45.0	37.5	100.0	37.5	54.5	12.1	66.7	12.1	20.5	9.0	100.0	9.0	16.4
	sentry_box	12	46.4	65.8	50.0	56.8	38.5	81.9	42.0	55.6	27.5	83.3	27.5	41.3	26.5	100.0	26.5	41.9
	traffic_box	1	36.3	100.0	36.3	53.3	36.3	100.0	36.3	53.3	26.9	100.0	26.9	42.5	26.9	100.0	26.9	42.5
	traffic_cone	1546	25.3	34.5	31.6	33.0	29.9	67.8	34.9	46.1	17.0	50.7	17.0	25.5	15.3	100.0	15.3	26.5
	traffic_island	28	26.3	56.8	26.8	36.4	34.4	86.6	36.3	51.2	13.9	71.4	13.9	23.3	14.1	100.0	14.1	24.7
	traffic_light	17	35.8	62.5	43.6	51.4	36.0	58.6	48.3	53.0	32.4	100.0	32.4	48.9	31.7	100.0	31.7	48.1
	traffic_sign	233	37.5	52.2	39.9	45.2	43.9	85.7	47.3	61.0	31.3	75.5	31.3	44.2	31.4	100.0	31.4	47.8
	obstruction																	
	debris	46	21.1	35.1	25.1	29.3	25.8	83.4	27.2	41.0	10.2	50.0	10.2	17.0	10.5	100.0	10.5	19.1
	suitcace	19	27.3	60.6	27.9	38.2	34.2	89.0	35.7	50.9	9.0	68.4	9.0	15.9	9.3	100.0	9.3	17.0
	dustbin	39	16.3	21.9	17.5	19.4	20.9	96.1	21.1	34.5	11.0	23.1	11.0	14.9	12.0	100.0	12.0	21.4
	concrete_block	8	15.4	24.1	16.5	19.6	13.5	62.1	14.7	23.8	10.4	37.5	10.4	16.3	11.3	100.0	11.3	20.2
	machinery	4	51.2	74.5	51.7	61.0	45.7	99.0	45.9	62.7	35.2	75.0	35.2	47.9	25.8	100.0	25.8	41.0
	chair	4	7.3	11.6	11.0	11.3	11.4	44.2	13.4	20.5	9.8	50.0	9.8	16.4	13.9	100.0	13.9	24.4
	phone_booth	2	29.6	30.5	47.5	37.2	23.2	42.2	34.1	37.7	36.3	100.0	36.3	53.2	29.2	100.0	29.2	45.2
	basket	3	4.8	11.1	6.7	8.3	2.1	14.3	2.4	4.2	5.4	66.7	5.4	10.0	4.0	100.0	4.0	7.8
	misc																	
	misc	15	36.7	60.7	37.9	46.7	37.7	88.6	39.6	54.8	18.4	73.3	18.4	29.4	19.2	100.0	19.2	32.2

Table A.3: Evaluation results (in %) on CODA for single anomaly categories. *Individual Metrics* correspond to calculating the given metric for all points in every anomaly and calculating the mean of all these metrics. *Aggregated Metrics* are achieved by cumulating all points of all annotations together into one confusion matrix and calculating the metrics on this one confusion matrix. Additionally, the metrics for the unclustered ground truth are shown.

Part	Dataset	Count	Individual Metrics				Aggregated Metrics				Individual Metrics$_{noclustering}$				Aggregated Metrics$_{noclustering}$			
			mIoU	AP	AR	F1	mIoU	AP	AR	F1	mIoU	AP	AR	F1	mIoU	AP	AR	F1
All																		
	CODA	1412	8.9	13.2	26.2	17.5	10.8	13.7	33.5	19.5	9.2	16.2	21.2	18.4	11.8	16.8	28.3	21.1
	CODA-ONCE	1034	8.9	14.0	27.1	18.4	8.8	11.1	29.4	16.1	9.0	16.9	21.5	18.9	9.9	14.2	24.5	18.0
	CODA-KITTI	307	10.9	13.3	29.0	18.3	14.9	19.0	40.9	25.9	11.8	17.4	25.0	20.6	16.0	22.3	36.3	27.6
	CODA-nuScenes	71	0.4	0.7	1.0	0.8	0.4	0.6	1.4	0.9	0.3	1.0	0.6	0.7	0.5	0.9	1.1	1.0
Boxes																		
	CODA	1412	23.1	50.8	26.2	34.6	31.2	81.4	33.5	47.5	21.2	73.8	21.2	33.0	28.3	100.0	28.3	44.2
	CODA-ONCE	1034	24.4	57.6	27.1	36.9	27.2	78.2	29.4	42.7	21.5	77.9	21.5	33.7	24.5	100.0	24.5	39.4
	CODA-KITTI	307	24.0	37.8	29.0	32.8	38.2	85.3	40.9	55.3	25.0	74.9	25.0	37.5	36.3	100.0	36.3	53.3
	CODA-nuScenes	71	1.0	7.6	1.0	1.8	1.4	69.1	1.4	2.7	0.6	8.5	0.6	1.1	1.1	100.0	1.1	2.3
Overlap																		
	CODA	1304	32.4	55.0	37.9	44.9	40.5	81.4	44.6	57.6	23.0	79.9	23.0	35.7	28.3	100.0	28.3	44.2
	CODA-ONCE	985	33.9	60.5	38.8	47.2	36.6	78.2	40.7	53.6	22.6	81.8	22.6	35.4	24.5	100.0	24.5	39.4
	CODA-KITTI	250	35.0	46.4	44.2	45.3	47.1	85.3	51.3	64.0	30.8	92.0	30.8	46.1	36.3	100.0	36.3	53.3
	CODA-nuScenes	69	2.4	7.9	2.5	3.8	1.9	69.1	1.9	3.7	0.6	8.7	0.6	1.1	1.1	100.0	1.1	2.3

Table A.4: The evaluation results (in %) on CODA are presented for entire images. *Individual Metrics* refer to the computation of the given metric for all points in each image, followed by calculating the mean of these metrics across all images. *Aggregated Metrics* are obtained by consolidating all points from all images into a single confusion matrix and then calculating the metrics based on this unified confusion matrix. Additionally, the metrics for the unclustered ground truth are shown.

Part	Dataset	Count	Individual Metrics				Aggregated Metrics				Individual Metrics$_{noclustering}$				Aggregated Metrics$_{noclustering}$			
			mIoU	AP	AR	F1	mIoU	AP	AR	F1	mIoU	AP	AR	F1	mIoU	AP	AR	F1
Boxes																		
	CODA	5142	19.6	34.4	22.7	27.3	31.2	81.2	33.6	47.5	18.0	50.5	18.0	26.6	28.4	100.0	28.4	44.2
	CODA-ONCE	4321	21.1	37.5	24.4	29.6	27.3	78.1	29.6	42.9	19.1	53.2	19.1	28.1	24.6	100.0	24.6	39.5
	CODA-KITTI	397	22.5	34.3	27.7	30.7	38.3	84.9	41.1	55.4	24.6	71.5	24.6	36.7	36.4	100.0	36.4	53.4
	CODA-nuScenes	424	0.8	2.4	0.8	1.1	1.8	76.3	1.8	3.5	0.7	2.8	0.7	1.1	1.3	100.0	1.3	2.6
Overlap																		
	CODA	4279	29.2	41.3	34.9	37.8	40.5	81.2	44.6	57.6	21.6	60.7	21.6	31.9	28.4	100.0	28.4	44.2
	CODA-ONCE	3693	30.9	43.9	36.6	39.9	36.8	78.1	41.0	53.8	22.4	62.3	22.4	32.9	24.6	100.0	24.6	39.5
	CODA-KITTI	305	34.6	44.7	45.0	44.8	47.0	84.9	51.3	64.0	32.1	93.1	32.1	47.7	36.4	100.0	36.4	53.4
	CODA-nuScenes	281	1.5	3.6	1.5	2.1	2.4	76.3	2.4	4.7	1.0	4.3	1.0	1.6	1.3	100.0	1.3	2.6

Table A.5: The evaluation results (in %) on CODA are presented for single anomalies. *Individual Metrics* refer to the computation of the given metric for all points in each image, followed by calculating the mean of these metrics across all images. *Aggregated Metrics* are obtained by consolidating all points from all anomalies into a single confusion matrix and then calculating the metrics based on this unified confusion matrix. Additionally, the metrics for the unclustered ground truth are shown.

B List of Figures

2.1 Frustum for extracting a part of the point cloud based on the 2D bounding box. . 4
2.2 Visualization of the mean shift algorithm, adapted from [39]. 4
2.3 Visualization of the DBSCAN algorithm, adapted from [22]. 5

3.1 Categorization of single-source corner cases based on used sensor, reprinted from [18] 8
3.2 Overview of anomaly detection approaches based on camera, lidar, radar, multi-modal, and abstract *object level* data, reprinted from [2]. 9

4.1 WD-Pascal: Two examples of synthetically inserted anomalies into real-world scenes, reprinted from [4]. 12
4.2 Cumulated masks of all contained anomalies within the respective datasets, reprinted from [4]. 13
4.3 Detection results (%) on CODA, reprinted from [25]. 14
4.4 Distribution of the top-4 categories in: **A** ONCE, **B** Kitti, and **C** NuScenes, reprinted from [25]. 15
4.5 The Corner Case Proposal Generation (COPG) pipeline. The input to the pipeline includes the point cloud and the camera image of a given scene. The point cloud is used to compute (a), while the camera image (b) is utilized to produce (c) and (d). The results from (c) and (d) are then used to remove invalid proposals. The final output (g) consists of a set of bounding boxes indicating the proposed corner cases in the camera image, reprinted from [25]. 15

5.1 Lidar coordinate for each original dataset [14, 19, 31]. 18
5.2 KITTI sensor coordinates, reprinted from [14]. 18
5.3 The manual evaluation is based on several visual representations. Firstly, the DBSCAN clustering results are depicted in an image, followed by presenting all identified clusters in the 3D scatter plot in the center. In the right 3D scatter plot, the selected cluster is highlighted in red. Similarly, the same process is repeated for the mean shift clustering, with the corresponding results displayed in the bottom row. 21

5.4 Sartoris proposed an approach [38] that consists of two main components: the SV part and the SSV part. In the SV part, each point is assigned a semantic motion class, which represents the motion state of the point. This assignment is achieved by combining an SV semantic segmentation model and an SV motion object segmentation model. These models provide information about the semantic category of the point and its motion characteristics, respectively. In the SSV part, a displacement vector is predicted for each non-ground point using an SSV scene flow model. Additionally, the ego-motion of the vehicle is predicted using an SSV odometry model. By combining the predicted displacement vectors and ego motion, a motion label is assigned to each point. A comparison is then made between the motion labels obtained from the SV part and the SSV part for each point. Points that have inconsistent motion labels are identified as potential anomalies. These inconsistent points are clustered together, forming clusters that serve as indications of anomalies in the scene, reprinted from [38]. 23
5.5 Image 1137 from CODA, an example of all possible scenarios in one image. . . . 25
5.6 The visualized boxes represent the various aspects evaluated at the image level. The first image corresponds to the evaluation of all points from both the detection method and the ground truth combined (scenarios 1-14). The middle image corresponds to the part *boxes* (scenarios 1-10), and the bottom image corresponds to the part *overlap* (scenarios 1-8). 26
5.7 The confusion matrix for all scenarios in 5.2. The true positive (TP) labels are 13 and 14. The false positive (FP) labels are -13, -14, 3, and 4. The false negative (FN) labels are 11, 12, and 10. The true negative (TN) labels are -11, -12, 1, 2, and -10. 27

6.1 Number of occurrences of supercategories in the respective datasets. 30
6.2 Results of the manual inspection of the two different clustering methods mean shift and DBSCAN. The different labels refer to those defined in 9. On the left is the distribution over all CODA anomalies, and on the right the distribution is split into the original datasets. 30
6.3 In the left image, an example is presented where a foreground object cluster is selected due to its proximity to the center of the 2D bounding box. In the right image, an illustration is provided of a fence-like structure with a low surface area. 31
6.4 Distibution of images and anomalies in CODA across the three original datasets. . 31
6.5 Difference in lidar point density between the different datasets: ONCE - left; KITTI - middle; nuScenes - right 32
6.6 The distribution of points labeled as anomalies in the three datasets. 32
6.7 In the left image, an example of a distant object is depicted, with no single lidar point hitting it (near the middle above the white car). The right image displays two cars, with one significantly occluding the other. The mean shift algorithm correctly clustered both cars together as a single cluster. 33

6.8 The top-left diagram represents the distribution for the CODA dataset, while the other three diagrams pertain to the individual components of the CODA dataset. Notably, all y-scales are logarithmized, and the scenarios correspond to those from Sartoris' original work 5.1. 34
6.9 The diagram illustrates the original distribution of points across various scenarios for the KITTI odometry dataset, reprinted from [38]. The y-scale is logarithmized, and the colors are adopted from Sartoris' original scenario definition. 35

A.1 Results of the manual inspection of the two different clustering methods mean shift and DBSCAN devided according to supercategory. The different labels refer to those defined in 9. 42
A.2 Confusion matrices as piecharts for CODA and ONCE. 43
A.3 Confusion matrices as piecharts for KITTI and nuScenes. 44

C List of Tables

4.1 Overview over all analyzed datasets, clustered by the benchmark in which they were presented, adapted from [4] 12

5.1 All possible scenarios that can occur when comparing the labels between the SV and the SSV part, reprinted from [38]. 24

5.2 Overview of all possible scenarios encountered during the comparison of labels between the ground truth and the anomaly detection method. The *New Label* represents the combined labels for each point, while the *Confusion Matrix* denotes the corresponding position of each scenario within the matrix. The *Color* column indicates the color assigned to visualize each scenario during the evaluation process. 25

6.1 The evaluation results (in %) on CODA are presented for entire images. *Individual Metrics* refer to the computation of the given metric for all points in each image, followed by calculating the mean of these metrics across all images. *Aggregated Metrics* are obtained by consolidating all points from all images into a single confusion matrix and then calculating the metrics based on this unified confusion matrix. 36

6.2 The evaluation results (in %) on CODA are presented for single anomalies. *Individual Metrics* refer to the computation of the given metric for all points in each image, followed by calculating the mean of these metrics across all images. *Aggregated Metrics* are obtained by consolidating all points from all anomalies into a single confusion matrix and then calculating the metrics based on this unified confusion matrix. 37

6.3 The evaluation results (in %) on CODA are presented for selected single anomaly categories. *Individual Metrics* refer to the computation of the given metric for all points in each image, followed by calculating the mean of these metrics across all images. *Aggregated Metrics* are obtained by consolidating all points from all anomalies into a single confusion matrix and then calculating the metrics based on this unified confusion matrix. 38

6.4 The evaluation results (in %) on CODA are presented for entire images, with no previous clustering of the 2D bounding boxes. *Individual Metrics* refer to the computation of the given metric for all points in each image, followed by calculating the mean of these metrics across all images. *Aggregated Metrics* are obtained by consolidating all points from all images into a single confusion matrix and then calculating the metrics based on this unified confusion matrix. 38

A.1 Evaluation results (in %) on CODA for single anomaly supercategories. *Individual Metrics* correspond to calculating the given metric for all points in every anomaly and calculating the mean of all these metrics. *Aggregated Metrics* are achieved by cumulating all points of all annotations together into one confusion matrix and calculating the metrics on this one confusion matrix. Additionally, the metrics for the unclustered ground truth are shown. 42
A.2 Evaluation results (in %) on CODA for single anomaly categories. *Individual Metrics* correspond to calculating the given metric for all points in every anomaly and calculating the mean of all these metrics. *Aggregated Metrics* are achieved by cumulating all points of all annotations together into one confusion matrix and calculating the metrics on this one confusion matrix. Additionally, the metrics for the unclustered ground truth are shown. 45
A.3 Evaluation results (in %) on CODA for single anomaly categories. *Individual Metrics* correspond to calculating the given metric for all points in every anomaly and calculating the mean of all these metrics. *Aggregated Metrics* are achieved by cumulating all points of all annotations together into one confusion matrix and calculating the metrics on this one confusion matrix. Additionally, the metrics for the unclustered ground truth are shown. 46
A.4 The evaluation results (in %) on CODA are presented for entire images. *Individual Metrics* refer to the computation of the given metric for all points in each image, followed by calculating the mean of these metrics across all images. *Aggregated Metrics* are obtained by consolidating all points from all images into a single confusion matrix and then calculating the metrics based on this unified confusion matrix. Additionally, the metrics for the unclustered ground truth are shown. . . . 46
A.5 The evaluation results (in %) on CODA are presented for single anomalies. *Individual Metrics* refer to the computation of the given metric for all points in each image, followed by calculating the mean of these metrics across all images. *Aggregated Metrics* are obtained by consolidating all points from all anomalies into a single confusion matrix and then calculating the metrics based on this unified confusion matrix. Additionally, the metrics for the unclustered ground truth are shown. 47

D Bibliography

[1] S. A. Baur, D. J. Emmerichs, F. Moosmann, P. Pinggera, B. Ommer, and A. Geiger. SLIM: Self-Supervised LiDAR Scene Flow and Motion Segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[2] D. Bogdoll, M. Nitsche, and J. M. Zöllner. Anomaly Detection in Autonomous Driving: A Survey. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022.

[3] D. Bogdoll, F. Schreyer, and J. M. Zöllner. Ad-datasets: A meta-collection of data sets for autonomous driving. In *International Conference on Vehicle Technology and Intelligent Transport Systems*, 2022.

[4] D. Bogdoll, S. Uhlemeyer, K. Kowol, and J. M. Zöllner. Perception Datasets for Anomaly Detection in Autonomous Driving: A Survey. *arXiv preprint:2302.02790*, 2023.

[5] J.-A. Bolte, A. Bär, D. Lipinski, and T. Fingscheidt. Towards Corner Case Detection for Autonomous Driving. In *IEEE Intelligent Vehicles Symposium*, 2019.

[6] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, and T. Fingscheidt. Systematization of Corner Cases for Visual Perception in Automated Driving. In *IEEE Intelligent Vehicles Symposium*, 2020.

[7] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, and T. Fingscheidt. Corner Cases for Visual Perception in Automated Driving: Some Guidance on Detection Approaches. *arXiv preprint:2102.05897*, 2021.

[8] J. Cen, P. Yun, J. Cai, M. Y. Wang, and M. Liu. Open-set 3D Object Detection. In *International Conference on 3D Vision (3DV)*, 2021.

[9] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41, 2009.

[10] X. Chen, S. Li, B. Mersch, L. Wiesmann, J. Gall, J. Behley, and C. Stachniss. Moving Object Segmentation in 3D LiDAR Data: A Learning-Based Approach Exploiting Sequential Data. *IEEE Robotics and Automation Letters*, 6, 2021.

[11] T. Cortinhal, G. Tzelepis, and E. E. Aksoy. SalsaNext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving. *Advances in Visual Computing*, 12510, 2020.

[12] G. Di Biase, H. Blum, R. Siegwart, and C. Cadena. Pixel-wise Anomaly Detection in Complex Driving Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[13] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer. Deep Multi-modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22, 2021.

[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research*, 2013.

[15] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[16] J. Han, X. Liang, H. Xu, K. Chen, L. Hong, J. Mao, C. Ye, W. Zhang, Z. Li, X. Liang, and C. Xu. SODA10M: A Large-Scale 2D Self/Semi-Supervised Object Detection Dataset for Autonomous Driving. *arXiv preprint:2106.11118*, 2021.

[17] D. Hawkins. *Identification of Outliers*. Springer Dordrecht, 1980.

[18] F. Heidecker, J. Breitenstein, K. Rösch, J. Löhdefink, M. Bieshaar, C. Stiller, T. Fingscheidt, and B. Sick. An Application-Driven Conceptualization of Corner Cases for Perception in Highly Automated Driving. In *IEEE Intelligent Vehicles Symposium*, 2021.

[19] Huawei Corporation. ONCE-Home. `https://once-for-auto-driving.github.io/`, 2022. Accessed: 2023-07-29.

[20] H. Iqbal, A. Al–Kaff, P. Marin, L. Marcenaro, D. M. Gomez, and C. Regazzoni. Detection of Abnormal Motion by Estimating Scene Flows of Point Clouds for Autonomous Driving. In *IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021.

[21] K. J. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian. Towards Open World Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[22] I. Khater, I. Nabi, and G. Hamarneh. A Review of Super-Resolution Single-Molecule Localization Microscopy Cluster Analysis and Quantification Methods. *Patterns*, 1, 2020.

[23] Y. Kittenplon, Y. C. Eldar, and D. Raviv. FlowStep3D: Model Unrolling for Self-Supervised Scene Flow Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[24] K. Lertniphonphan, S. Komorita, K. Tasaka, and H. Yanagihara. 2D to 3D Label Propagation For Object Detection In Point Cloud. In *IEEE International Conference on Multimedia & Expo Workshops*, 2018.

[25] K. Li, K. Chen, H. Wang, L. Hong, C. Ye, J. Han, Y. Chen, W. Zhang, C. Xu, D.-Y. Yeung, X. Liang, Z. Li, and H. Xu. CODA: A Real-World Road Corner Case Dataset for Object Detection in Autonomous Driving. In *Computer Vision - ECCV*, 2022.

[26] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal Loss for Dense Object Detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2018.

[27] C. Liu, X. Qian, X. Qi, E. Y. Lam, S.-C. Tan, and N. Wong. MAP-Gen: An Automated 3D-Box Annotation Flow with Multimodal Attention Point Generator. In *International Conference on Pattern Recognition (ICPR)*, 2022.

[28] E. Martí, M. A. de Miguel, F. Garcia, and J. Pérez. A Review of Sensor Technologies for Perception in Automated Driving. *IEEE Intelligent Transportation Systems Magazine*, 11, 2019.

[29] M. Masuda, R. Hachiuma, R. Fujii, H. Saito, and Y. Sekikawa. Toward Unsupervised 3d Point Cloud Anomaly Detection Using Variational Autoencoder. In *IEEE International Conference on Image Processing*, 2021.

[30] J. Nitsch, M. Itkina, R. Senanayake, J. Nieto, M. Schmidt, R. Siegwart, M. J. Kochenderfer, and C. Cadena. Out-of-Distribution Detection for Automotive Perception. In *IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021.

[31] nuScenes. NuScenes. `https://www.nuscenes.org/`, 2023. Accessed: 2023-07-29.

[32] A. Paigwar, O. Erkent, D. Sierra-Gonzalez, and C. Laugier. GndNet: Fast Ground Plane Estimation and Point Cloud Segmentation for Autonomous Vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[33] A. Paigwar, D. Sierra-Gonzalez, Ö. Erkent, and C. Laugier. Frustum-PointPillars: A Multi-Stage Approach for 3D Object Detection using RGB Camera and LiDAR. In *IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2021.

[34] Z. Peng, L. Chang, L. Wang, and Z. Yan. Application of multiple clustering algorithms, 2020.

[35] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum PointNets for 3D Object Detection from RGB-D Data. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[36] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*, 2021.

[37] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations*, 2020.

[38] F. Sartoris. Anomaly Detection in Lidar Data by Combining Supervised and Self-Supervised Methods. Bachelor's thesis, Karlsruhe Institute of Technology (KIT), 2022.

[39] K. Sippel, T. Kübler, W. Fuhl, G. Schievelbein, R. Rosenberg, and W. Rosenstiel. Eyetrace2014: Eyetracking Data Analysis Tool. In *International Conference on Health Informatics*, 2015.

[40] C. Tao, S. Fu, C. Wang, X. Luo, H. Li, Z. Gao, Z. Zhang, and S. Zheng. F-PVNet: Frustum-Level 3-D Object Detection on Point–Voxel Feature Representation for Autonomous Driving. *IEEE Internet of Things Journal*, 10, 2023.

[41] I. Tishchenko, S. Lombardi, M. R. Oswald, and M. Pollefeys. Self-Supervised Learning of Non-Rigid Residual Flow and Ego-Motion. In *International Conference on 3D Vision (3DV)*, 2020.

[42] K. Wada. Labelme: Image Polygonal Annotation with Python. `https://github.com/wkentaro/labelme`, 2023. Accessed: 2023-07-29.

[43] Z. Wang and K. Jia. Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal 3D Object Detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[44] K. Wong, S. Wang, M. Ren, M. Liang, and R. Urtasun. Identifying Unknown Instances for Autonomous Driving. In *Conference on Robot Learning*, 2020.

[45] M. E. Yabroudi, K. Awedat, R. C. Chabaan, O. Abudayyeh, and I. Abdel-Qader. Adaptive DBSCAN LiDAR Point Cloud Clustering For Autonomous Driving Applications. In *iceit*, 2022.

[46] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh. Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review. *Sensors*, 21, 2021.

[47] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[48] C. Zhang, Z. Huang, M. H. Ang, and D. Rus. LiDAR Degradation Quantification for Autonomous Driving in Rain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.