Anomaly Detection for Autonomous Driving

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften
(Dr.-Ing.)

von der KIT-Fakultät für Wirtschaftswissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Daniel Bogdoll, M.Sc.

Tag der mündlichen Prüfung:	11. Dezember 2025
Referent:	Prof. Dr. J. Marius Zöllner
Korreferent:	Prof. Dr. Hanno Gottschalk

Karlsruhe 2025

Acknowledgments

This dissertation is primarily the result of my time as a research scientist at the Research Center for Information Technology (FZI), specifically as a member of the department for Technical Cognitive Systems (TKS) within the Intelligent Systems and Production Engineering (ISPE) division. Additionally, I spent six months as a visiting research scholar with the Mcity group at the University of Michigan. As a doctoral student at the Karlsruhe University of Technology (KIT), I was a member of the group for Applied Technical Cognitive Systems (ATKS) at the Institute of Applied Informatics and Formal Description Methods (AIFB).

Thank you, Marius, for both the given freedom and the continuous, intensive, and genuine mentoring. Our regular meetings have truly helped me throughout the entire time to sharpen my focus and stay on track. I always felt supported by you, especially in the last and most difficult phase of this endeavor. Thank you, Hanno, for becoming my second supervisor. I am glad that you accepted my request so openly, and I am thankful for your valuable feedback along the way.

Thank you, Max, for the regular and passionate exchange. We were pushing each other constantly, and I really enjoyed this spirit! This dissertation would certainly look different without our frequent discussions and your practical feedback. Thank you, Svetlana, for being my main inspiration on how to approach research topics. Your scientific views have always impressed me, and your feedback along the way regularly helped improve my work. Thank you, Tim, for being the postdoc I never had. Quite knowledgeable, you were constantly there to support me both professionally and creatively. Robert, I want to thank you for being such a great and kind colleague in the office - I could not have wished for anyone else. Thank you, Helen, for all the nice chats that lightened up the daily routine. Thank you, Lars, for inviting me into the world of scientific writing at the FZI. Thank you, Max, Ahmed, Ferdinand, Helen, Johannes, Marcus, Melih, Nikolai, Robert, Svetlana, Tim, and Tobias, for proofreading the thesis and all your feedback!

Thank you, Marc, and the entire TKS leadership team for giving me the freedom and time to pursue my research. I truly appreciate the support I have received over the last years. The research leading to this dissertation was done mainly within four publicly funded projects: KIGLIS, KI WISSEN, KI Data Tooling, and jbDATA. Thank you to everyone involved for the pleasant cooperation.

Thank you, Greg and Henry, for inviting me to the University of Michigan - I enjoyed my time in Ann Arbor and at Mcity tremendously! I also want to thank the KIT Research Travel Grant and the KIT Graduate School UpGrade Mobility for providing financial support for this stay.

Finally, this dissertation would not have been possible without numerous student contributions. Thank you to all the students I had the pleasure to supervise through bachelor and master theses, seminars, labs, or as student assistants:

Finn Sartoris, Moritz Nekolla, Stefani Guneshka, Felix Schreyer, Johannes Jestram, Jonas Rauch, Moritz Wittig, Christin Scheib, Enrico Eisen, Maximilian Nitsche, Marcus Schilling, Meng Zhang, Simon Klaus, Nishanth Gowda, Jonas Hendl, Jing Qin, Lukas Namgyu Rößler, Iramm Hamdard, Felix Geisler, Muhammed Bayram, Felix Wang, Lukas Bosch, Vincent Geppert, Yitian Yang, Harsh Modasia, Louis Karsch, Yang Zheng, Noël Ollick, Jan Imhof, and Anushervon Tabarov.

Abstract

With small fleets of autonomous vehicles of SAE level 4, i.e., such without a safety driver, publicly available, the adoption of autonomous vehicles will only continue to increase. Embedded within shared mobility solutions, this technical advancement can lead to a more sustainable, safe, and comfortable future. Scaling autonomous vehicles more broadly, however, requires handling a wide variety of challenging scenarios, especially those with often rare anomalies. With rising fleet sizes, such scenarios appear with increasing frequency. As many Machine Learning systems follow a closed-world assumption based on a set of known classes, such unknowns remain challenging.

This dissertation addresses anomaly detection for autonomous driving from a holistic perspective, contributing to the generation of scenarios with anomalies, the detection of anomalies, and the handling of anomalies. The first part addresses external anomalies, i.e., such that occur in the environment. Generating scenarios involves providing normal data to train models and creating scenarios with anomalies to evaluate anomaly detection methods. Based on a theoretical systematization of anomalies from the literature, scenarios from all anomaly layers can be created. As generating such external anomalies is often dangerous or infeasible, data is provided through a simulation engine. Based on these scenarios, an anomaly detection method is presented, which is trained on unlabeled sensor data alone. It leverages a world model as a representation of normality, utilizing both camera and LIDAR data. Once detected, anomalies can be integrated into the training process of Neural Networks, removing their status as anomalies. The presented approach handles previously detected anomalies where controlled traffic rule exceptions are required. To achieve this, a situation-aware reward for Reinforcement Learning is introduced.

Next to challenges induced by external anomalies, the driving task can be equally impacted by internal anomalies, such as model failures. This dissertation contributes to the field of internal anomaly detection by detecting model failures without the need for labeled evaluation sets. This is achieved by analyzing the disagreements between two models trained on the same task, but with different learning paradigms. Based on real-world data, the method successfully reveals categorical model failures, most often in seemingly normal situations.

Summarizing, this dissertation presents a holistic set of contributions to the field of anomaly detection for autonomous driving, addressing the generation, detection, and handling of anomalies. This is emphasized by examining both internal and external anomalies.

Publications and Supervised Theses

The following first-authored and peer-reviewed publications and supervised student theses contribute to this dissertation. Parts of this dissertation have already been published in selected publications. Where this is the case, the publications are explicitly mentioned at the beginning of each chapter. In some instances, the texts of the original publications have been edited, shortened, and supplemented to illustrate better how the works form a cohesive whole. This includes but is not limited to the following: Active language has been reformulated into passive language. The tenses of sentences have been adjusted. Linguistic and grammatical improvements and corrections have been applied. Text passages have been revised for better comprehensibility. Cross-references have been added. Terms used have been unified. Introductory and summarizing sections have been adapted.

In the process of writing prior publications and this dissertation, Artificial Intelligence (AI)-based tools such as DeepL, Grammarly, LanguageTool, ChatGPT, Gemini, or Copilot have aided in translations and linguistic refinement.

Publications

D. Bogdoll, R. P. Ananta, A. Giridharan, I. Moore *et al.* Mcity Data Engine: Iterative Model Improvement Through Open-Vocabulary Data Selection. Accepted at *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2025

D. Bogdoll, F. Sartoris, V. Geppert, S. Pavlitska *et al.* Label-Free Model Failure Detection for Lidar-based Point Cloud Segmentation. In *IEEE Intelligent Vehicles Symposium (IV)*, 09/2025

D. Bogdoll, Y. Yang, T. Joseph, M. Yazgan *et al.* MUVO: A Multimodal Generative World Model for Autonomous Driving with Geometric Representations. In *IEEE Intelligent Vehicles Symposium (IV)*, 08/2025

D. Bogdoll, I. Hamdard, L.N. Rößler, F. Geisler *et al.* AnoVox: A Benchmark for Multimodal Anomaly Detection in Autonomous Driving. In *European Conference on Computer Vision (ECCV) Workshop*, 05/2025

D. Bogdoll, J. Imhof, T. Joseph, J.M. Zöllner. Hybrid Video Anomaly Detection for Anomalous Scenarios in Autonomous Driving. In *British Machine Vision Conference (BMVC) Workshop*, 12/2024

D. Bogdoll, N. Ollick, T. Joseph, J.M. Zöllner. UMAD: Unsupervised Mask-Level Anomaly Detection for Autonomous Driving. In *British Machine Vision Conference (BMVC) Workshop*, 12/2024

D. Bogdoll, J. Qin, M. Nekolla, A. Abouelazm *et al.* Informed Reinforcement Learning for Situation-Aware Traffic Rule Exceptions. In *IEEE International Conference on Robotics and Automation (ICRA)*, 08/2024

D. Bogdoll, L. Karsch, J. Amritzer, J.M. Zöllner. On The Impact of Replacing Private Cars with Autonomous Shuttles: An Agent-Based Approach. In *IEEE Forum for Innovative Sustainable Transportation Systems (FISTS)*, 04/2024

D. Bogdoll, S. Pavlitska, S. Klaus, J.M. Zöllner. Conditioning Latent-Space Clusters for Real-World Anomaly Classification. In *IEEE Symposium Series on Computational Intelligence (SSCI)*, 01/2024

D. Bogdoll, L. Bosch, T. Joseph, H. Gremmelmaier *et al.* Exploring the Potential of World Models for Anomaly Detection in Autonomous Driving. In *IEEE Symposium Series on Computational Intelligence (SSCI)*, 01/2024

D. Bogdoll, S. Uhlemeyer, K. Kowol, J.M. Zöllner. Perception Datasets for Anomaly Detection in Autonomous Driving: A Survey. In *IEEE Intelligent Vehicles Symposium (IV)*, 07/2023

D. Bogdoll, J. Hendl, F. Schreyer, N. Gowda *et al.* Impact, Attention, Influence: Early Assessment of Autonomous Driving Datasets. In *IEEE International Conference on Control and Robotics Engineering (ICCRE)*, 06/2023

D. Bogdoll, S. Guneshka, J.M. Zöllner. One Ontology to Rule Them All: Corner Case Scenarios for Autonomous Driving. In *European Conference on Computer Vision (ECCV) Workshop*, 02/2023

D. Bogdoll, M. Zhang, M. Nitsche, J.M. Zöllner. Experiments on Anomaly Detection in Autonomous Driving by Forward-Backward Style Transfers. In *IEEE International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 12/2022

D. Bogdoll, J. Rauch, J.M. Zöllner. DLCSS: Dynamic Longest Common Subsequences. In *IEEE International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 12/2022

D. Bogdoll, E. Eisen, M. Nitsche, C. Scheib *et al.* Multimodal Detection of Unknown Objects on Roads for Autonomous Driving. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 11/2022

D. Bogdoll, M. Nitsche, J.M. Zöllner. Anomaly Detection in Autonomous Driving: A Survey. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*, 08/2022

D. Bogdoll, M. Nekolla, T. Joseph, J.M. Zöllner. Quantification of Actual Road User Behavior on the Basis of Given Traffic Rules. In *IEEE Intelligent Vehicles Symposium (IV)*, 07/2022

D. Bogdoll, F. Schreyer, J.M. Zöllner. AD-Datasets: A Meta-Collection of Data Sets for Autonomous Driving. In *International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)*, 05/2022

D. Bogdoll, S. Orf, L. Töttel, J.M. Zöllner. Taxonomy and Survey on Remote Human Input Systems for Driving Automation Systems. In *Future of Information and Communication Conference (FICC)*, 03/2022

D. Bogdoll, J. Jestram, J. Rauch, C. Scheib *et al.* Compressing Sensor Data for Remote Assistance of Autonomous Vehicles using Deep Generative Models. In *Conference on Neural Information Processing Systems (NeurIPS) Workshop*, 11/2021

D. Bogdoll, J. Breitenstein, F. Heidecker, M. Bieshaar *et al.* Description of Corner Cases in Automated Driving: Goals and Challenges. In *IEEE/CVF International Conference on Computer Vision (ICCV) Workshop*, 11/2021

D. Bogdoll, P. Matalla, C. Füllner, C. Raack *et al.* KIGLIS: Smart Networks for Smart Cities. In *IEEE International Smart Cities Conference (ISC2)*, 10/2021

Supervised Theses

N. Ollick. Camera-based Anomaly Detection with Generative World Models. Bachelor Thesis, Karlsruhe Institute of Technology, 03/2024

Y. Yang. 3D Voxel Reconstruction and World Model for Autonomous Driving. Master Thesis, Karlsruhe Institute of Technology, 12/2023

Y. Zheng. Anomaly Detection With World Models For Autonomous Driving. Master Thesis, University of Stuttgart, 12/2023

L.N. Rößler. Benchmarking Anomaly Detection on Camera and Lidar Data with 3D Voxel Representation. Bachelor Thesis, Karlsruhe Institute of Technology, 11/2023

L. Karsch. Sustainability of Autonomous Vehicles: An Agent-based Simulation of the Private Passenger Sector. Master Thesis, Karlsruhe Institute of Technology, 09/2023

V. Geppert. Anomaly Detection with Model Contradictions for Autonomous Driving. Bachelor Thesis, Karlsruhe Institute of Technology, 08/2023

J. Qin. Reinforcement Learning for Controlled Traffic Rule Exceptions. Master Thesis, Karlsruhe Institute of Technology, 06/2023

S. Klaus. Anomaly Detection in the Latent Space of VAEs. Bachelor Thesis, Karlsruhe Institute of Technology, 10/2022

M. Schilling. Anomaly Detection in 3D Space for Autonomous Driving. Master Thesis, Karlsruhe Institute of Technology, 06/2022

F. Sartoris. Anomaly Detection in Lidar Data by Combining Supervised and Self-Supervised Methods. Bachelor Thesis, Karlsruhe Institute of Technology, 06/2022

S. Guneshka. Ontology-based Corner Case Scenario Simulation for Autonomous Driving. Bachelor Thesis, Karlsruhe Institute of Technology, 03/2022

Contents

Acronyms

A2D2 Audi Autonomous Driving Dataset. 19

ABS Absolute Error. 73, 77

AD Autonomous Driving. 1, 6, 9, 17, 24, 27, 40, 55, 58, 81

AI Artificial Intelligence. v, 2, 44, 45

AIFB Institute of Applied Informatics and Formal Description Methods. i

AP Average Precision. 76–78, 110, 111

AR Average Recall. 110, 111

ATKS Applied Technical Cognitive Systems. i

AUPRC Area under the Precision-Recall Curve. 52

AUROC Area under the Receiver Operating Curve. 24, 52, 76–78

BDL Bayesian Deep Learning. 12, 13

BEV Bird's-Eye-View. 23, 48, 58–61, 64–67, 70, 79, 83, 85, 88, 91, 104, 119, 129, 131

CAD Computer-Aided Design. 31

CARLA Car Learning to Act. 1, 4, 6, 26, 29, 33, 35–37, 42, 44–46, 49, 55, 61, 64, 79, 87, 93, 115, 123, 134

CEO Chief Executive Officer. 2

CIFAR Canadian Institute For Advanced Research. 71

CLIP Contrastive Language-Image Pre-Training. 59, 72

CMCDOT Conditional Monte Carlo Dense Occupancy Tracker. 26

CNN Convolutional Neural Network. 19, 21, 23, 74, 88

CODA Corner Case Dataset. 42, 55, 56, 105, 106, 109–112, 132, 134

COOOL Challenge Of Out-Of-Label. 56

CS Cityscapes. 19, 25, 26, 42, 44, 49, 51

CUHK Chinese University of Hong Kong. 26

CWA Closed World Assumption. 11

DARPA Defense Advanced Research Projects Agency. 1

DBSCAN Density-Based Spatial Clustering of Applications with Noise. 24, 103

DDPG Deep Deterministic Policy Gradient. 82

DeepSAD Deep Semi-Supervised Anomaly Detection. 23

DeLORA Deep LIDAR Odometry for Robotic Applications. 105

DIN German Institute for Standardisation. 13, 14

DINO Detection Transformer with Improved Denoising Anchor Boxes. 72

DM Domain Mismatch. 22

DQN Deep Q-Network. 88

DS Domain Shift. 16, 65

E2E End-to-End. 6, 90, 112, 113, 116, 120

EDS Euclidean Distance Sum. 24

EMD Earth-Mover's Distance. 22

FPR_{95} False Positive Rate at 95% True Positive Rate. 52, 54, 76–78, 110

FPS Frames Per Second. 64

FS Fishyscapes. 18, 19, 21, 42, 51, 56, 75

FZI Research Center for Information Technology. i

GAIA Generative AI for Autonomy. 59, 61

GAN Generative Adversarial Network. 21, 22, 59

GPU Graphics Processing Unit. 59, 119

GRU Gated Recurrent Unit. 63

HD High Definition. 59

HMM Hidden Markov Model. 26

ID In-Distribution. 14, 20, 42, 56

IL Imitation Learning. 94

IoU Intersection over Union. 54, 65, 68, 69

ISPE Intelligent Systems and Production Engineering. i

ISSU Semantic Segmentation in the Presence of Unknowns. 56

KIT Karlsruhe University of Technology. i

KITTI Karlsruhe Institute of Technology and Toyota Technological Institute. 1, 19, 23, 42, 51, 102, 105–111, 131, 132

LaF Lost and Found. 2, 19–21, 25, 51

LIDAR Light Detection and Ranging. iii, 4, 6, 7, 17, 18, 22–28, 30, 41, 42, 45, 46, 48, 50–52, 55–64, 66–70, 72, 79, 83, 93, 98, 100, 102–104, 106, 107, 109, 110, 112, 113, 115–118, 120, 127–129, 131–134

LLM Large Language Model. 59, 94, 95, 118, 119

LTL Linear Temporal Logic. 84, 86, 89, 134

MILE Model-based Imitation Learning. 61, 66

mIoU mean Intersection over Union. 110, 111

ML Machine Learning. iii, 5, 9, 12, 13, 15, 58, 82, 97, 98, 112, 119

ML-MemAE-SC Multi-Level Memory modules in an Autoencoder with Skip Connections. 53, 128

MLP Multilayer Perceptron. 62, 63

MLUC Metric Learning with Unsupervised Clustering. 24

MNAD Memory-guided Normality for Anomaly Detection. 71, 75–79, 116

MNIST Modified National Institute of Standards and Technology. 71

MSE Mean Squared Error. 22, 27, 73, 76–78

MUAD Multiple Uncertainties for Autonomous Driving. 42

MVTec AD MVTec Anomaly Detection. 71

NF Normalizing Flows. 21

NN Neural Network. iii, 2, 18, 20, 118

ODD Operational Design Domain. 1, 2, 15, 30, 55

ONCE One Million Scenes. 42, 105, 109–111, 132

OOD Out-of-Distribution. 6, 13, 14, 20, 28, 42, 43, 56, 94, 99

OoDIS Out-of-Distribution Instance Segmentation. 56

OSIS Open-Set Instance Segmentation. 23, 24

OWA Open World Assumption. 11, 12

PID Proportional-Integral-Derivative. 88

POMDP Partially Observable Markov Decision Process. 85

PPO Proximal Policy Optimization. 82

PPV Positive Predictive Values. 52

PSNR Peak Signal-to-Noise Ratio. 22, 27, 65, 69, 71

RA Road Anomaly. 19

RADAR Radio Detection and Ranging. 4, 99

RbA Rejected By All. 51, 52, 72, 128, 134

REAL Redundancy Classifier. 51, 52, 128, 134

ResNet Residual Network. 65–67, 129

RGB Red Green Blue. 4, 22, 25, 42, 45, 46, 48, 52, 54–61, 64, 72, 83, 85, 88, 102, 106, 107, 117, 118, 128, 134

RGB-D Red Green Blue-Depth. 25

RL Reinforcement Learning. iii, 4, 7, 81–88, 90–94, 116, 119, 127, 130

RNN Recurrent Neural Network. 59

RO Road Obstacles. 19

RQ Research Question. 5, 115, 116

RSS Responsibility-Sensitive Safety. 2

S2M Score To Segmentation Mask. 72

SAA Segment Any Anomaly. 71, 72

SAE Society of Automotive Engineers. iii, 1, 10, 11, 13, 14

SAM Segment Anything Model. 71, 72, 75, 77, 78, 134

SCAL Scene-Class Affinity Loss. 64

SH StreetHazards. 19

SMIYC Segment Me If You Can. 19, 21, 51, 56, 75

SotA State-of-the-Art. 6, 17, 21, 22, 42, 50–52, 55, 57, 70, 75, 78, 79, 88, 93, 109, 116, 117, 128, 134

SSIM Structural Similarity Index Measure. 27, 74, 77

ST ShanghaiTech. 26, 71

STL Signal Temporal Logic. 84, 94

STU Spotting the Unexpected. 56

SVM Support Vector Machine. 22

TKS Technical Cognitive Systems. i

U2Seg Unsupervised Universal Segmentation. 75–78, 130, 134

UCSD University of California, San Diego. 26, 71, 75

UGainS Uncertainty Guided Anomaly Instance Segmentation. 72

VAD Video Anomaly Detection. 53, 54, 128, 134

VAE Variational Autoencoder. 24, 25, 27, 53, 128

VGG Visual Geometry Group. 21, 22, 74

ViT Vision Transformer. 66, 129

VLM Vision Language Model. 56, 94, 95, 117–119

VOS Virtual Outlier Synthesis. 19

VRU Vulnerable Road User. 45

WD WildDash. 19

It's harder than putting a man on the moon.

Doug Field, 2022 [314]

1. Introduction

Autonomous vehicles hold the promise of safer transportation. Alongside improvements in comfort and efficiency, they can increase accessibility and inclusion. If deployed correctly, they might also lead to a more sustainable future [BOG 10, 19][STU 3]. Autonomous Driving (AD) has come a long way since the early days of the Defense Advanced Research Projects Agency (DARPA) Grand Challenges between 2004 and 2007 [267]. These events were significant milestones, as decades of prior research culminated in impressive demonstrations in the real world [107, 407, 423]. In 2012, the rise of deep learning was accompanied by the rise of public data in autonomous driving, as both the AlexNet architecture [239] and the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) dataset [146] were introduced to the world. In 2016, Cityscapes [101] was released as the first diverse dataset with driving scenes from a variety of cities. These events were followed by the introduction of the Transformer architecture [426] and the release of Car Learning to Act (CARLA), the first open-source simulation engine dedicated to autonomous driving [117], in 2017. The release of BDD100K [491], the first large-scale driving dataset, followed shortly after in 2018. Arguably, the rise of autonomous driving would not have been possible without such advances in deep learning and the availability of more and more data. Fueled by enormous investments [298, 448], these advancements have led to small deployed fleets of autonomous vehicles without a safety driver as of 2025 [446, 405]. The most advanced fleets operate at Society of Automotive Engineers (SAE) level 4 autonomy [332] in the USA and China, each with a geographically small Operational Design Domain (ODD).

1.1. Motivation

Despite the impressive progress made, scaling up autonomous vehicles remains a challenging problem. The root cause for this is the long tail distribution of rare events, i.e., the existence of many events with a low probability of occurrence, which are often challenging to detect and handle. Such events can come in many forms, such as a zoo breakout [39], a trailer with a tree on it [183], a stop sign on a billboard [450], an overturned truck on a highway [134], a red light runner [231, 12], a delivery robot [35], or simply a vehicle being towed [447] – the list goes on [51, 52, 57]. It is important to keep in mind that for such an event to be challenging for an autonomous vehicle, it does not necessarily need to be odd or challenging from a human perspective.

While events from the long tail are rare from the perspective of a single human driver, they become more frequent and thus increasingly problematic for a fleet of autonomous vehicles all running the same software. This challenge of the long tail of rare events is a long-standing and well-established research problem in both academia and industry:

It's all about the long tail - 99.9999...%
— A. Karpathy, Sr. Director of AI at Tesla, 2019 [215]

The remaining problem we have is [...] the long tail of crazy things
— J. Schneider, Professor at Carnegie Mellon University, 2019 [376]

There is a lot of rare situations, and all of them need to be handled well
— D. Anguelov, Principal Scientist at Waymo, 2019 [11] (sic)

[...] solve the long tail of all the things that might happen in the world
— R. Urtasun, CEO at Waabi, 2021 [424]

Distribution of rare scenarios has a very long tail
— D. Dolgov, Co-CEO at Waymo, 2024 [114]

[...] you can't ever identify all the edge cases
— P. Koopman, Professor at Carnegie Mellon University, 2024 [232]

I have witnessed firsthand the difficulties of addressing the "long tail"
— E. Dagan, President at Wayve, 2024 [104]

Detecting and handling rare events from this long tail is necessary to scale autonomous vehicles beyond a tightly constrained ODD. An initial approach to detect anomalies with Neural Networks (NNs) in the context of autonomous driving without relying on classical methods using stereo vision was presented in 2015 by Creusot and Munawar [103]. As they did not disclose their evaluation data, no comparison of different approaches was possible. This already changed in 2016, when Pinggera et al. generated the first public anomaly detection benchmark Lost and Found (LaF) [347] with labeled obstacles on the road. Such generated anomalies are necessary for the development and evaluation of anomaly detection methods. Subsequently, detected anomalies can be leveraged to improve the handling of such. As the field of anomaly detection continued to gain attention [BOG 13, 22], as discussed in more detail in Chapter 3, Shalev-Shwartz et al. touched upon the handling of anomalies in their 2017 Responsibility-Sensitive Safety (RSS)

framework [386], where they present a set of rules as a proposal for a "mathematical model for safety assurance". Here, they also take into account atypical obstacles on the road.

For a long time, anomalies in the context of autonomous driving were loosely and inconsistently defined. This started to change in 2020, when Breitenstein et al. introduced an expert-defined taxonomy of different anomalies [51], focusing on affected sensors or occurrences in the environment surrounding the vehicle. Subsequently, Heidecker et al. introduced a data-defined taxonomy of anomalies in 2021 [178], focusing on internal errors introduced through data processing methods, such as false negatives during object detection. Such data-based perspectives also put a larger emphasis on the definition of anomalies in relation to the training data, representing normality, available to models. Section 2.5 further elaborates on the definitions of anomalies as used in this dissertation.

While there has been tremendous progress in all three areas — anomaly generation, anomaly detection, and anomaly handling — the long tail of rare events remains challenging to this day. This dissertation highlights research problems, identifies research questions, and presents multiple contributions to the field.

1.2. Scope and Contributions

This dissertation contributes to the generation of scenarios with anomalies, the detection of anomalies, and the handling of anomalies. In this section, research problems, identified research questions, and the contributions of this thesis are presented. As this is a broad field, the scope of this thesis is defined as follows:

- For the generation of scenarios, all anomalies in the environment are created in simulation, as many anomalies cannot be recorded in the real world in a safe and feasible manner.
- For the detection of external anomalies[1], this dissertation focuses on localizable[2] anomalies in raw sensor data. Non-localizable anomalies on the domain level [51] in the form of domain shifts [371, 400] are not addressed. Non-localizable anomalies on the sensor layer [178], such as camera failures [381], are not addressed. Different data sources, such as time-series [345] or trajectory [453] data, are not addressed.
- For the handling of external anomalies, this dissertation focuses on a learnable approach by integrating previously detected anomalies into the training dataset, eliminating their status as anomalies during inference. Different approaches, such as remote assistance [BOG 26] during deployment, are touched upon but not explicitly addressed.

[1] Definitions of both external and internal anomalies are introduced in Section 2.5.
[2] A localizable anomaly can be identified within a specific region of a frame, rather than classifying a whole frame as anomalous.

- For the detection of internal anomalies, this dissertation focuses on a complementary learning approach where disagreements between two models, trained for the same task but with different learning paradigms, are used to detect internal anomalies.
- This dissertation considers Red Green Blue (RGB) cameras and Light Detection and Ranging (LIDAR) as typical sensor modalities. Radio Detection and Ranging (RADAR) is not considered, as the utilized CARLA simulation environment only provides a low-fidelity RADAR sensor model based on raycasting [289], and the majority of the used datasets [146, 261, 253, 58, 294] do not contain RADAR data.
- This dissertation focuses on both external and internal anomalies. However, specific sensor-related anomalies from the sensor layer are not considered.

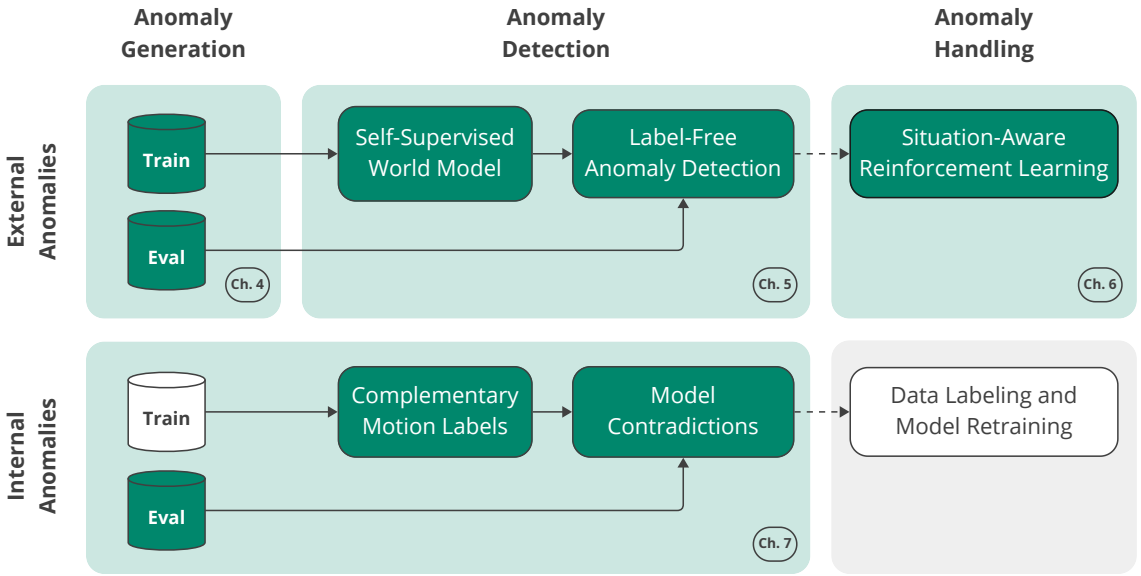


Figure 1.1.: **Dissertation Overview:** Anomalies can be separated into external ones, i.e., occurrences in the environment, and internal ones, i.e., failures introduced by the system itself. Chapter 4 introduces a challenging multimodal dataset including anomalies. Chapter 5 uses that data to demonstrate anomaly detection without the need for labeled data or outlier exposure. Subsequently, Chapter 6 presents situation-aware Reinforcement Learning (RL), handling previously detected anomalies through controlled traffic rule exceptions. Finally, Chapter 7 addresses internal anomalies and introduces model failure detection for the task of point cloud segmentation. Contributions are shown in green.

The remainder of this section first introduces the structure of this dissertation, as shown in Figure 1.1, and presents identified research questions and contributions based on it subsequently. After this introduction, Chapters 2 and 3 provide a technical background and an overview of anomaly detection approaches and related datasets from the literature. Chapters 4 – 7 present contributions in the

Chapter / Anomaly Level	Domain	Object	Scene	Scenario	Input	Model	Deployment
State of the Art (Ch. 3)	✓	✓	✓	✓	—	—	—
Anomaly Generation (Ch. 4)	✓	✓	✓	✓	—	—	—
Anomaly Detection (Ch. 5)	—	✓	—	—	—	—	—
Anomaly Handling (Ch. 6)	—	✓	—	—	—	—	—
Anomaly Detection (Ch. 7)	—	✓	—	—	✓	✓	✓

Table 1.1.: **Anomaly levels addressed in this dissertation:** Overview of anomaly levels [51, 178, 177] considered per chapter. Chapters 3 - 6 focus on external anomalies, while Chapter 7 addresses internal ones. The different anomaly levels are introduced in more detail in Section 2.5.

fields of anomaly generation, anomaly detection, and anomaly handling. During this section, the content of these chapters will be introduced in more detail, motivated by individual research questions. All chapters are aligned with a theoretical systematization of anomalies developed primarily by Breitenstein and Heidecker [51, 178, 177], as shown in Table 1.1.

To better understand research gaps with respect to anomaly detection methods and utilized datasets, Breitenstein et al. have provided a structured overview focusing on camera-based approaches and related datasets [52]. While this work remains highly relevant, it does not discuss other sensor modalities or combinations. This motivates the first Research Question (RQ):

> **RQ1**: What are the patterns of anomaly detection methods and related datasets for typical autonomous vehicle sensor modalities?

Chapter 3 presents an extensive survey and characterization of anomaly detection methods in autonomous driving, also examining the datasets and benchmarks utilized. A major identified pattern is the focus on methods that work with camera data only. Shortcomings include, amongst other things, varying definitions of anomalies and the need for outlier exposure, i.e., the inclusion of exemplary anomalies, during training for well-performing methods.

Building on these findings, this dissertation focuses on datasets next, as these are the basis for all Machine Learning (ML) approaches. While there exist over 200 datasets in autonomous driving [BOG 21][269], only a fraction is concerned with anomaly detection. However, none of these include anomalies from all considered external anomaly levels [51]. This leads to the second research question:

> **RQ2**: How can theoretical anomaly definitions from the literature be converted into datasets containing anomalies?

Chapter 4 first presents a methodology to generate expert-defined scenarios with anomalies from all levels. The scenario descriptions are based on an ontology, meaning that all scenarios are structured in a comparable way, allowing for later coverage analysis. This approach allows for the generation of a large-scale, structured scenario catalog. While such a catalog is useful to test an autonomous

driving function, generating expert-defined scenarios requires a great deal of manual effort. This results in a set of very specific scenarios. In an open world, however, a wide variety of situations can occur.

In the remainder of Chapter 4, a more scalable methodology for the generation of object-level and scenario-level anomalies is introduced. For a well-defined benchmark, it provides both training and evaluation data to ensure that anomalies are absent during training. For multimodality, a sensor setup is employed with ground truth provided based on camera and LIDAR data. To allow for a comparison between anomaly detection methods using different sensors, the ground truth is additionally provided in a voxelized form. All data is generated in the CARLA [117] simulation environment. By using CARLA, the largest open-source simulation ecosystem in AD is supported [259].

Next, this dissertation focuses on object-level anomaly detection methods, as prevalent in the literature. In the current State-of-the-Art (SotA), the predominant paradigm is to utilize strong semantic segmentation networks, combined with outlier exposure to provide examples of true positives [316, 108, 473]. Such known outliers are often synthetically generated by augmenting scenes with extracted patches or objects from Out-of-Distribution (OOD) data [411, 152, 316, 473] or by creating anomalies with generative models [150, 108]. These anomaly detection approaches require large amounts of labeled data and introduce biases towards exemplary anomalies. Methods defining normality based on raw training data are rare. In addition, most methods leverage only a single sensor modality. These limitations lead to the following research question:

> **RQ3**: How can unlabeled sensor data from multiple modalities be leveraged for the detection of object-level anomalies?

To learn a representation of normality without the need for labels, Chapter 5 first introduces a multimodal world model. For self-supervised training, a dataset representing normality, as introduced in Chapter 4, is utilized. A world model allows for the reconstruction of input data as well as for the prediction of future frames. Based on this learned normality through the world model, a reconstruction-based approach for anomaly detection is introduced. The performance of this approach is further improved by a self-supervised mask refinement. Based on the benchmark introduced in Chapter 4, the presented anomaly detection method outperforms the most relevant label-free SotA method and sets a new baseline.

Once anomalies are detected, they can be used to improve the handling of such for the task of driving. While some works consider anomalies that occur at inference time during the planning of trajectories [386, 323], it has not been extensively studied how to integrate previously detected anomalies into the training process and learn how to handle them accordingly. This is crucial to enable learned End-to-End (E2E) autonomous driving systems [449, 432, 202], which show potential to improve situation-aware driving behavior dramatically. The following research question is identified:

> **RQ4**: How can identified object-level anomalies benefit the training process of learned trajectory planning?

Chapter 6 addresses this research question by introducing situation-aware RL. Here, known anomalies are integrated into the training process. One way to handle object-level anomalies blocking a lane is to perform a controlled rule exception, utilizing an oncoming lane for progress. Classically, such behavior is punished through a static reward function. The presented approach introduces a dynamic and situation-aware reward, showing strong performance improvements in the handling of anomalies.

As framed by Heidecker et al. [178], anomalies do not only exist in the external environment surrounding the ego vehicle[3] but can also be induced by internal models and methods used. Even seemingly normal situations can be challenging for models, e.g., if the training data distribution does not match the one during deployment. Detecting both external and internal anomalies is crucial, as both can affect the downstream task of driving equally. For a holistic perspective, the following research question is identified and addressed:

> **RQ5**: How can model failures be detected in an open world without access to ground truth labels?

Chapter 7 investigates the detection of internal anomalies in a deployment-like setting. As limited validation and test sets cannot represent the open world, new failures are to be expected during deployment. Here, a self-supervised model is trained on the same task as an existing supervised legacy model. Disagreements between the models are leveraged to detect model failures for further inspection through an oracle. The approach is demonstrated for the segmentation of LIDAR point clouds in real-world environments, focusing on the differentiation between static and dynamic objects. The evaluation highlights the detection of a multitude of model failures in scenarios where the model repeatedly fails. In addition, the sensitivity of the approach towards external object-level anomalies is examined. The evaluation shows a particular emphasis on atypical objects, which are hard to classify by the legacy model.

Finally, Chapter 8 summarizes the contributions of this dissertation and provides an outlook for future work. It outlines the general development of the field of anomaly detection for autonomous driving and provides future research directions with respect to the contributions presented in this dissertation.

Summarizing, this dissertation contributes to the generation of scenarios with anomalies, the detection of anomalies, and the handling of previously detected anomalies. This holistic perspective is emphasized by addressing both internal and external anomalies. None of the approaches presented in this dissertation require labeled data during training. Next, the following Chapter 2 provides the technical background necessary for a better comprehension of the remainder of this dissertation.

[3] An ego vehicle is a vehicle in consideration, which perceives the environment through sensors

Personally, I am always ready to learn, although I do not always like being taught [...]

Winston Churchill, 1952 [421]

2. Background

This chapter provides context for central technical aspects touched upon in this dissertation. This background information is intended to facilitate a deeper comprehension of the following chapters. Specifically, this chapter provides background for different learning paradigms in ML in Section 2.1 and an overview of AD in Section 2.2. In addition, it provides introductions to closed and open world views and uncertainties in Sections 2.3 and 2.4. Finally, it introduces relevant definitions of anomalies in Section 2.5.

2.1. Learning Paradigms

Machine Learning can be categorized into different training paradigms. The most important ones in the context of this dissertation will be introduced in this section. Let $X = \{x_1, x_2, ..., x_n\}$ represent a training dataset, where x_i denotes the i-th sample.

Supervised Learning: In supervised classification learning, the goal is to learn from labels associated with the samples in the training dataset [308]. Each sample $x_i \in X$ has an associated label $y_i \in Y$, where Y is the set of possible labels. The objective is to learn a mapping function $g : X \to Y$ to predict the label for novel samples. The learning process aims to minimize the error between predicted labels $\hat{y}_i = g(x_i)$ and the true labels y_i. In regression settings, the target labels $y_i \in \mathbb{R}$ are continuous.

Active Learning: In Active Learning, supervised learning is designed in iterations. In each iteration, samples from an unlabeled subset $X_U \subseteq X$ are queried, i.e., selected, for labeling. An oracle, such as a human annotator or a slow but more accurate model, provides labels for the selected samples. The labeled training set is subsequently used for training. A main focus in active learning is on query strategies to select which samples to label in order to improve model performance [384], as this is more efficient than labeling all available training data.

Curriculum Learning: Curriculum Learning is a training strategy that increases the difficulty during training to improve generalization capabilities [36]. The difficulty is assigned by a predefined, designed, or learned difficulty measure $d(x_i)$ associated with each sample $x_i \in X$. As the focus is on the ordering of samples, Curriculum Learning can be applied to many training paradigms [36, 313].

Unsupervised Learning: In unsupervised learning, the goal is to discover patterns inherent in the training data X [308, 466, 187]. The objective is to learn a function $h : X \to Z$, where Z is a learned representation of the data, without relying on explicit labels. Typical applications are clustering or dimensionality reduction.

In anomaly detection specifically, unsupervised often refers to a setting where only normal data is used during training [175]. This is different from using any methods trained in a supervised setting, such as semantic segmentation, for the detection of anomalies. To avoid a conflict with the introduced definition of unsupervised learning, the term *label-free* is used to represent methods that do not use labeled data, which includes labeled anomalies.

Self-Supervised Learning: In self-supervised learning, the training objective follows the same setting as supervised learning [210]. However, similar to unsupervised learning, there are no associated labels to the training dataset X. Labels are first derived from the data itself by a function $k : X \to X' \times Y$ where for each $x \in X$, $x' \in X'$ either equals x or is a partial or transformed version, and $y \in Y$ is the target output. While there is no consensus [26, 25], self-supervised learning is often seen as a type of unsupervised learning due to the absence of explicit labels in the training data [272, 210, 503, 6, 325].

Reinforcement Learning: Reinforcement learning focuses on training an agent that can perform actions $a_i \in A$ in an environment by interacting with it [401]. Unlike previous paradigms, this is different from using a static set of training data X. The agent learns through trial and error by receiving rewards or penalties for its actions. The goal is to learn a policy $\pi : S \to A$ that maps states S to actions A in order to maximize the cumulative reward over time.

2.2. Autonomous Driving

The SAE standard J3016 – *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles* [332] – defines a set of six levels to categorize driving automation, as shown in Figure 2.1. As represented by the blue boxes, levels 0 - 3 still require a human driver. While many of the contributions of this dissertation might be used to improve features on those levels, they are not emphasized. Only levels 4 and 5 can operate without a human driver on board. However, it should be noted that support from a human operator is still possible through remote assistance [332] [BOG 15]. While there is no consensus on whether to use the term *automated* or *autonomous* for driverless vehicles [297, 416, 332], SAE levels 4 and 5 are referred to as autonomous driving in this dissertation.

As of 2025, there are popular industrial offers for advanced level 2, 3, and 4 systems. The Tesla Full Self-Driving (Supervised) [406] feature is a level 2 system, as the driver still has to pay attention. Due to its advanced state compared to other level 2 features, it is sometimes referred to as a level 2+ feature [233]. The Mercedes-Benz Drive Pilot [303] is a level 3 feature focusing on motorways. Finally, the Waymo

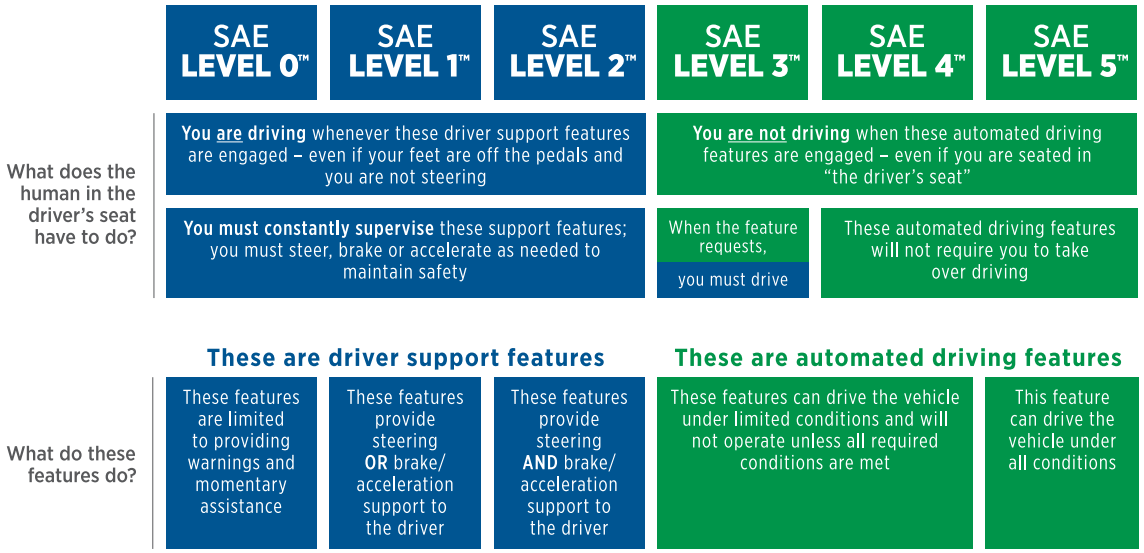


Figure 2.1.: **SAE J3016 standard:** The standard defines six levels of driving automation. Blue boxes represent functions that require a driver. Green boxes represent automated driving features. Adapted from [370].

One service is a level 4 offer [446] similar to a taxi service, with no driver present. Due to the broad definition of SAE level 5, it is currently not considered by the industry.

2.3. Closed and Open World

The world can be viewed from a closed or open perspective, sometimes also called closed-set or open-set scenarios. A closed view assumes that a given set of categories is complete. Such categories are often introduced through the semantic classes of labeled datasets [101], such as "road", "person", or "sky". This setting is typically used in classical object detection or semantic segmentation tasks, where such a set of known classes is given [262] and used for training and inference. For formal logical systems operating under a Closed World Assumption (CWA), the absence of knowledge regarding a queried statement implies its falsity [358].

On the other hand, an Open World Assumption (OWA) is much closer to the real world, where the existence of unknowns is acknowledged [418]. Figure 2.2 shows how a long-tailed distribution is only well-defined for known events – typical head and rare tail events – in the closed-world setting, as those have been observed and define the distribution. A core challenge is that the individually rare events of the tail make up a significant portion of all occurrences. However, yet unknown events in the open world are often assumed to be rare, but their frequency is truly unknown. Under a large domain shift, they can also appear frequently. Open-world [277, 420, 419], open-vocabulary [462], or zero-shot [471] tasks are among the approaches trying to identify the unknown. For formal logical

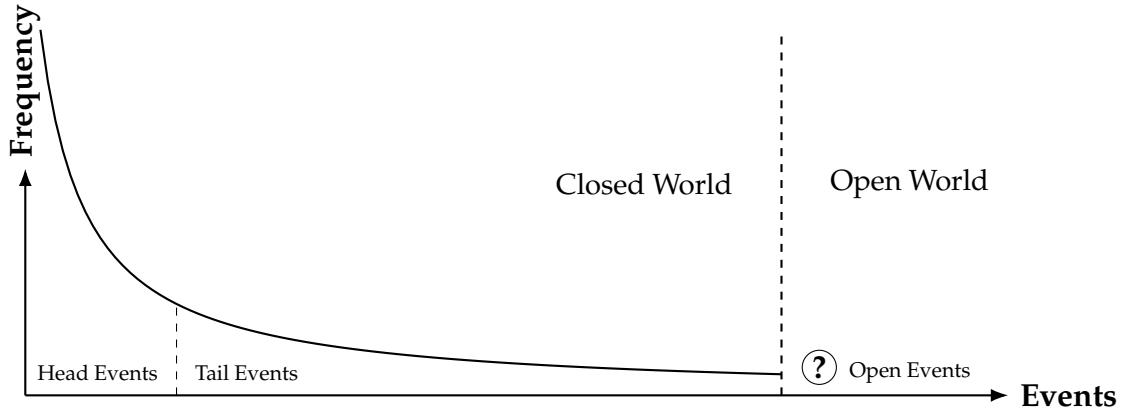


Figure 2.2.: **Long tail of rare events:** Long-tailed distribution in a closed world setting, with further unknown events in the open world. Adapted from [277].

systems operating under an OWA, the absence of knowledge regarding a queried statement does not imply its falsity, allowing for non-definite answers [358].

2.4. Uncertainty

The real world is full of uncertainty, and the same applies to complex systems. However, there are different sources of uncertainty, namely *aleatoric* and *epistemic*, which have been widely studied [161, 180]. More recently, these types of uncertainty have gained more attention in the context of ML [219, 199], as models tend to assign high probability values to erroneous predictions, even though those do not always reflect modeled model confidence [139].

Aleatoric uncertainty refers to "the notion of randomness" [199] which is inherent to the system and is irreducible, i.e., impossible to decrease. An example in the context of autonomous driving is the prediction of the movements of traffic participants such as pedestrians. It is impossible to be certain about where a pedestrian will be several seconds into the future. Similarly, partial observability contributes to aleatoric uncertainty [33].

Epistemic uncertainty refers to a "lack of knowledge" [199] and is reducable, e.g., by acquiring knowledge. An example in the context of autonomous driving is an underrepresented class in a training dataset, such that a model struggles to generalize during inference. Including a larger variety of examples in the training dataset can lead to reduced epistemic uncertainty.

One way to model uncertainty is Bayesian Deep Learning (BDL). Rather than learning fixed weights, BDL aims to find a distribution over the model parameters. As the computation of the predictive distribution remains challenging [456], approximations such as Monte Carlo Dropout [140] or Deep Ensembles [242] are

commonly employed. BDL has shown positive effects for both accuracy and calibration given OOD samples [382]. While specialized ML methods to predict either epistemic or aleatoric uncertainty exist, differentiating between the two types remains challenging [310]. As many methods require more than a single forward pass, they are slow and difficult to employ in a real-world setting. Progress on deterministic methods is being made, but distributional methods currently show better performance [350, 310].

2.5. Definition of Anomaly

The term *anomaly* is widely used but often utilized interchangeably with other related terms or as an umbrella term, as shown by multiple surveys [178, 372, 483, 77, 51, 69, 331]. For example, Chandola et al. describe an anomaly vaguely as a "pattern that does not conform to expected normal behavior" [77] in their widely established survey on anomaly detection. Foorthuis defines anomalies as “occurences [...] that are in some way unusual” [135]. This is complicated by the fact that perception-based anomaly detection is also present in other domains such as medicine [29], industrial processes [37], or surveillance [463], often with a domain-specific definition of what constitutes an anomaly [77].

Related terms, such as *corner case*, *outlier*, or *Out-of-Distribution*, which are often used interchangeably, do not follow clear definitions either [427, 483]. In the following, examples of how these terms are used are provided, followed by the introduction of the taxonomy used in this dissertation. Chou et al. [96] define *corner cases* as situations in which ensuring safety is difficult but possible. Bolte et al. define a corner case as a relevant and unpredictable object in a relevant location [45]. Zhou and Beyerer differentiate between internal corner cases as “interpretation problems of neural networks” and external influences such as sensor failures or unexpected behaviors [511]. Breitenstein, Heidecker et al. propose a multi-layer corner case taxonomy ranging from sensor issues to atypical behaviors of other traffic participants [51, 52, 178]. Ouyang et al. define samples as corner cases if their perturbation leads to model failures [334]. Pfeil et al. [345] propose a corner case taxonomy based on three different causes, comprising external environment anomalies, internal functional constraints, and system-internal conditions. Heidecker et al. focus on ML corner case samples that generate a model-specific “high predictive uncertainty” [176]. The DIN-SAE specification 91381 [516] – *Terms and Definitions Related to Testing of Automated Vehicle Technologies* – defines a corner case as a “scenario in which two or more parameter values are each within the capabilities of the system, but together constitute a rare condition that challenges its capabilities”. Similarly, Koopman et al. define corner cases as rare “combinations of normal operational parameters”, i.e., situations that can be anticipated [234].
An *edge case* is defined by Koopman et al. as a rare and novel situation that was not considered during the design process and that requires addressing it [234]. The

DIN-SAE specification 91381 [516] defines an edge case as a "scenario in which the extreme values or even the very presence of one or more parameters results in a condition that challenges the capabilities of the system". Karunakaran et al. define an edge case as a scenario that is difficult to predict, unknown, and unsafe [217]. Eliot defines both corner cases and edge cases as rare or unusual [123].
Boult et al. [49, 48, 47] touch upon the term *novelty* and formalize it for the spaces world, observation, and agent. Both the world and the observed space are external to an agent, but it has only access to the observed space. The agent space is internal and influences its actions. In their view, a novelty "depends on dissimilarity between a [...] novel world and the experience of some non-novel world", where such dissimilarities are task-dependent [49]. Greer and Trivedi characterize novelties as "unexpected scenarios that autonomous vehicles struggle to navigate" [154].
Chen et al. [87] define novelties both as an *Out-of-Distribution* case, where the data is dissimilar to the training dataset, and an adversarial case, where the data is similar but perturbed, resulting in a prediction change of the model. Yang et al. [483] define OOD data as coming from a "distribution that is different from the training distribution", focusing on OOD samples with known labels not present in the In-Distribution (ID) data. Differently, Mao et al. define OOD data as samples not aligning with user expectations, provided in natural language [295].
Shafaei et al. [385] describe OOD samples as *outliers*. Grubbs defines an outlier as an observation "that appears to deviate markedly from other members of the sample in which it occurs" [155].
Focusing on a safety perspective, Liu and Feng coined the term curse of rarity based on "*rare* safety-critical events" which are hard to define and identify [268]. Differently, Heidecker et al. see rare samples as known but hard to obtain [176].

While these works use different definitions of terms, two perspectives emerge: An internal one, focusing on the system[1] itself [178, 176, 511, 49, 345], and an external one, focusing on occurrences in the environment [51, 52, 49, 45, 345]. For autonomous driving, both perspectives are equally relevant as they directly influence the downstream task of driving. The multi-layer taxonomy developed primarily by Chandola, Breitenstein, and Heidecker [77, 51, 52, 178, 176] is the most systematic anomaly categorization describing both internal and external anomalies for autonomous driving. The taxonomy, as shown in Table 2.1, is used as a theoretical foundation in this dissertation and is presented in the following. This dissertation employs the term *anomaly* over alternatives to emphasize the broad spectrum of anomalous occurrences. One exception is the use of the term *outlier exposure* to describe the inclusion of known and exemplary anomalies into the training process, as it is a well-established term in the field.

2.5.1. Internal Anomalies

The anomaly systematization in autonomous driving by Breitenstein et al. [51] focuses primarily on external anomalies but also includes a hardware level on the

[1] The system of an autonomous vehicle consists of its software and hardware components.

Type	**Internal**				**External**				
Layer	**Method**			**Sensor**		**Content**			**Temporal**
Level	Input	Model	Deployment	Hardware	Physical	Domain	Object	Scene	Scenario

Table 2.1.: **Anomaly systematization:** The systematization shows all anomaly levels from the literature and how they are categorized into anomaly layers and anomaly types. Adapted from [51, 178, 177].

sensor layer, which addresses hardware degradations, such as pixel defects. As the hardware is not part of the surrounding environment, this level is considered internal. In addition, Heidecker et al. conceptualized a taxonomy for internal, system-induced anomalies [178, 177].

Definition 1 (Internal Anomaly)**.** An internal anomaly is an occurrence originating within the system that leads to erroneous outputs, regardless of whether it is triggered by the current environmental context.

As shown in Table 2.1, the method layer addresses internal ML model-related anomalies on three levels. As described by Heidecker et al. [176], the input level addresses issues based on the utilized training data, such as faulty labels or underrepresented classes. This can lead to erroneous model predictions during inference. Such data-related anomalies are also strongly emphasized by Zhou and Beyerer [511]. The model level is concerned with issues introduced through calibration problems and high epistemic uncertainties for predictions, indicating that the deployed model is not well-suited for handling such cases. Finally, the deployment level addresses organizational issues, focusing on concept shifts between the training data and the ODD where a vehicle is deployed.

2.5.2. External Anomalies

The systematization of external anomalies in autonomous driving was originally developed by Breitenstein et al. [51], focusing on camera data. Here, experts divided anomalies into different levels, in increasing order of detection complexity. This taxonomy was refined by Heidecker et al. and generalized for all typical sensor modalities, as depicted in Table 2.1. This dissertation addresses external anomalies from an ML-based perception perspective.

Definition 2 (External Anomaly)**.** An external anomaly is an occurrence in the environment surrounding the ego vehicle whose representation has low likelihood under the distribution induced by the training dataset $\mathcal{D}$.

The sensor layer addresses both internal anomalies on the hardware level and external anomalies on the physical level. Anomalies on the physical level are hardware-related but have their origin in the environment, such as overexposure,

and are thus external. The content layer consists of three levels. The domain level includes domain shifts, whereas the object level represents unknown artifacts such as lost cargo on the road. The scene[2] level focuses on contextual anomalies, such as people on a billboard. Finally, the temporal layer consists of scenarios[3] that cannot be detected in a single frame, such as someone performing a sudden braking maneuver.

In their original work, Breitenstein et al. [51] provide descriptive subcategories for each anomaly level. In the sensor layer, *global outliers* describe scenarios where "all or many pixels fall outside of the expected range", while *local outliers* describe a similar scenario for "one or few pixels". In the domain level, a *Domain Shift (DS)* describes a "shift in appearance, but not in semantics". In the context of a trained model, the concept of normality is fixed. A DS then corresponds to a covariate shift, while other shifts that alter task semantics, such as concept shift, are not considered. The object level includes *single-point anomalies* in the form of unknown objects. In the scene level, a *contextual anomaly* describes a "known object, but in an unusual location", and a *collective anomaly* describes "known objects, but in an unseen quantity". Finally, three categories exist for the scenario level. A *risky scenario* describes a temporal pattern that was seen during training but still has the potential for collision. Differently, a *novel scenario* is defined as an unknown pattern that does not have the potential for collision. Finally, an *anomalous scenario* is both unknown and has "high potential for collision". These subcategories were partially adopted for the refined taxonomy by Heidecker et al. [178] as shown in Table 2.1. Most of these expert-defined categories are compatible with the data-based definition of external anomalies as provided in Definition 2. For a *risky scenario* to comply with the definition, its description is adapted. As the potential for collision still differs from an *anomalous scenario*, the need for it to be a known pattern can be neglected.

It is important to note that internal and external anomalies are not mutually exclusive. To illustrate this point, consider a model struggling to classify a pedestrian as static or dynamic, but pedestrians are generally included in the training dataset. This does not classify as an external, but only as an internal anomaly. However, if a model struggles to detect a pedestrian in a costume, which was absent from the training dataset, this is classified as both an internal and external anomaly.

[2] A scene describes a "snapshot of the environment [...]" [422].
[3] A scenario describes a "temporal development between several scenes [...]" [422].

If I have seen further, it is by
standing on the shoulders of giants.

Isaac Newton, 1675 [321] *(proverb with origins in the 12^{th} century* [304])

3. State of the Art

Parts of this chapter have previously appeared in the following publication:

- D. Bogdoll et al. *Anomaly Detection in Autonomous Driving: A Survey*. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshop, 2022 [BOG 13]

3.1. Introduction

To better understand research gaps with respect to anomaly detection methods and utilized datasets in the context of autonomous driving, Breitenstein et al. provided an overview focusing on camera-based approaches and related datasets [52]. While extensive, their work does not address other typical sensor modalities and combinations. Addressing **RQ1**, this chapter derives patterns of anomaly detection methods and related datasets. It presents methods in four categories. First, methods using a typical sensor modality, either camera or LIDAR, are presented. Subsequently, methods using multimodal data sources are presented. Finally, methods based on abstracted sensor data are described. Based on this comprehensive examination of the SotA, patterns and weaknesses of anomaly detection methods are uncovered, which motivate large parts of this dissertation and are addressed in later chapters.

In the upcoming sections, this chapter provides an overview of anomaly detection methods in the domain of AD for different sensor modalities. Tables 3.1 - 3.4 provide overviews of the examined anomaly detection methods based on the utilized modality. Detection approaches are classified following Breitenstein et al. in five concepts: "reconstruction, prediction, generative, confidence scores, and feature extraction" [52]. Confidence score techniques can be subdivided into Bayesian approaches, learned scores, and scores obtained by post-processing. Reconstructive approaches try to reconstruct normality and consider any kind of deviation from it as anomalous. Generative approaches are closely related to the former reconstructive approaches, but also take into account the discriminator's decision or the distance to the training data. Feature extraction can be based on handcrafted or learned features to determine a class label or compare modalities on various feature levels. Prediction-based techniques predict the next frame(s) expected under normality. More details on the five detection concepts can be found in the work of Breitenstein et al. [52]. Figure 3.1 shows the distribution of anomaly detection methods covered in this chapter with respect to the five detection concepts,

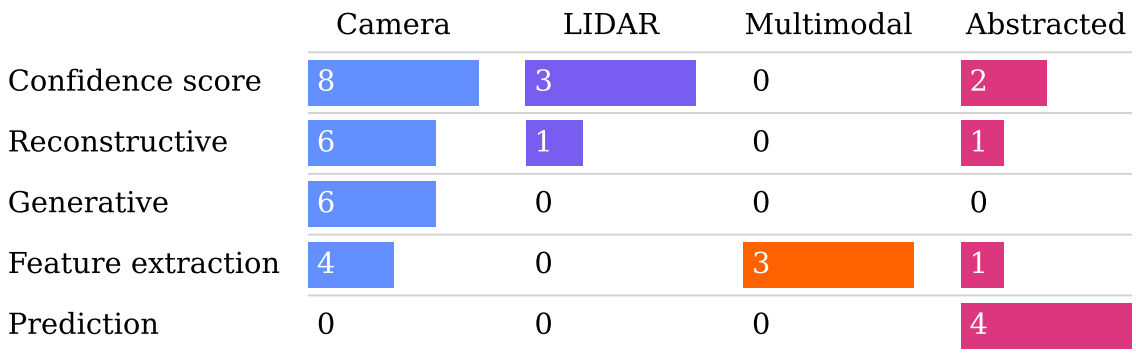


Figure 3.1.: **Overview of anomaly detection methods:** Distribution of anomaly detection approaches. There are 24 methods using camera data, 4 using LIDAR data, 3 using multimodal data, and 8 using abstracted data. Adapted from [BOG 13].

categorized by the underlying sensor modality. It shows that the vast majority of methods are based on camera data. Such camera-based approaches utilize a wide variety of detection approaches; only predictive approaches have not been addressed with raw camera data yet. A novel method using camera data for predictive anomaly detection is briefly introduced in Section 4.3. While LIDAR-based approaches show a focus on confidence score-based and reconstructive approaches, multimodal methods focus strongly on feature extraction. Finally, methods using abstracted data, which focus on scenario-level anomalies, use a wide variety of detection approaches.

3.2. Camera Data

Autonomous vehicles are often equipped with different camera systems, like stereo, mono, and fisheye cameras, to ensure a rich perception of the environment. Thus, anomaly detection in camera data holds great potential for more robust visual perception. For this section, two more criteria are introduced following the Fishyscapes (FS) benchmark [133]: outlier exposure and retraining. The former indicates whether an approach requires anomalous data during training. Retraining, however, specifies whether methods cannot use pre-trained models but require a special loss or retraining, which might decrease the performance [133]. All camera-based methods can be found in Table 3.1.

Confidence score: Approaches on the basis of confidence scores constitute a baseline for the detection of anomalies based on the estimation of uncertainty in NNs. As one of the earlier works, Kendall et al.'s Bayesian SegNet [218] derives the uncertainty of the semantic segmentation network SegNet by Monte Carlo dropout sampling, where higher variance of the classes indicates higher uncertainty. The uncertainty can be interpreted as a pixel-wise anomaly score to detect obstacles on roads [429, 330]. A similar approach to detect unknown obstacles on the road

Author(s)	Ref.	Approach	Outlier Exposure	Retrain	Anomaly Level	Data
Du et al.	[120]	Confidence	✗	✓	Object	PASCAL-VOC [127], BDD100K [492]
Jung et al.	[213]	Confidence	✗	✗	Scene	FS, LaF[43], RA [264]
Heidecker et al.	[179]	Confidence	✗	✗	Object	A2D2 [147]
Chan et al.	[76]	Confidence	✓	✓	Object	LaF [347], CS [101], FS [43]
Bevandić et al.	[41]	Confidence	✓	✓	Object	Vistas [320], CS [101], ImageNet [109], WD [40]
Malinin and Gales	[292]	Confidence	✓	✓	Object	FS, LaF [43]
Huang et al.	[198]	Confidence	✗	✗	Object	CamVid [53]
Kendall et al.	[218]	Confidence	✗	✗	Object	CamVid [53]
Vojir et al.	[429]	Reconstruction	✗	✗	Scene	LaF [347], RA [264], RO [263], FS, LaF [43]
Ohgushi et al.	[330]	Reconstruction	✗	✗	Scene	LaF [347], Highway dataset
Lis et al.	[263]	Reconstruction	✗	✓	Scene	FS, LaF [43], RO [263]
Di Biase et al.	[112]	Reconstruction	✓	✗	Object	FS, LaF [43]
Blum et al.	[44]	Reconstruction	✓	✗	Object	FS, LaF [43], FS [43]
Creusot and Munawar	[103]	Reconstruction	✗	✓	Scene	Recordings, YouTube
Nitsch et al.	[324]	Generative	✗	✓	Object	KITTI [145], nuScenes [58], ImageNet [109]
Grcić et al.	[149]	Generative	✗	✓	Object	WD-Pascal [40], LaF [347], SMIYC [75], SH [181]
Xia et al.	[469]	Generative	✗	✗	Object	CS [101], SH [181]
Löhdefink et al.	[278]	Generative	✗	✓	Domain	CS [101], BDD100K [492], KITTI [145]
Lis et al.	[264]	Generative	✗	✗	Object	LaF [347], RA [264]
Haldimann et al.	[167]	Generative	✗	✗	Scene	CS [101], Vistas [320] [320]
Xue et al.	[476]	Feature Extraction	✓	✓	Scene	LaF[347]
Bolte et al.	[46]	Feature Extraction	✗	✓	Domain	KITTI[145], CS [101], BDD100K [492]
Zhang et al.	[504]	Feature Extraction	✗	✗	Domain	Udacity [417]
Bai et al.	[22]	Feature Extraction	✓	✗	Scene	Urban dataset

Table 3.1.: **Camera-based anomaly detection:** The overview shows the used approach and anomaly level. In addition, it highlights whether outlier exposure or retraining is necessary. Finally, the used data is listed. Adapted from [BOG 13].

is proposed by Jung et al. [213]. They obtain class-conditioned "standardized max logits" of a segmentation network. This procedure is motivated by the finding that max logits have their own ranges for different predicted classes. The mean and standard deviations are thereby determined from the training samples. Thus, the standardization can be categorized as a learned confidence score approach. In addition to the standardization, they suppress class boundaries and apply dilated smoothing to consider local semantics in broad receptive fields. Heidecker et al. [179] model the epistemic uncertainty of Mask R-CNN [173] and quantify the class and positional uncertainty of instances. They outline a criterion to detect anomalies based on position and class uncertainty. Anomalies due to positional uncertainty are defined by the standard deviation of scaled bounding boxes exceeding a predefined threshold. In addition, instances are considered anomalous due to class uncertainty whenever the standard deviation of any class is above the predefined threshold. But Bayesian segmentation networks are slow in inference due to their multiple forward passes through the network with Monte Carlo dropout for each frame. Therefore, Huang et al. [198] simulate the sampling procedure via region-based temporal aggregation in frame sequences. To ensure the correct uncertainty estimation of moving objects, the previous segmentation is warped via optical flow. Bevandić et al. [41] present a multi-task network to simultaneously segment the input frame into semantics as well as output an anomaly probability map. The latter overrides the semantic segmentation whenever a probability exceeds a threshold to calibrate the confidence score when the model faces outliers. Du et al. [120] present the general learning framework Virtual

Outlier Synthesis (VOS), which contrastively shapes the decision boundary of NNs by synthesizing virtual outliers. At first, they estimate a class-conditioned multivariate Gaussian distribution in the penultimate latent space. Afterwards, outliers are sampled from a sufficiently small ϵ-likelihood region of this learned distribution. These virtual outliers near the class boundary encourage the model to form a compact decision boundary between ID and OOD data. Furthermore, they propose a novel training objective with free energy as an uncertainty measurement, where ID data has negative, and the virtual outliers have positive energy. During inference, OOD objects are detected with a logistic regressor based on the uncertainty score.

Reconstructive: Reconstructive and generative approaches are predominantly used for anomaly detection on the object level, since the models learn to reproduce the normality of the training data without any outlier exposure with anomalous objects. For instance, a work by Vojir et al. [429] proposes the reconstruction module JSR-Net to detect road anomalies based on a pixel-wise score. They enhance trained semantic segmentation networks by incorporating their information from known classes into the anomaly score. The network architecture consists of a reconstruction and a semantic coupling module. The former is connected to the backbone of the semantic segmentation network and reconstructs the road in a discriminative way, meaning it reduces the reconstruction loss of the road while increasing the loss for the remaining environment. In the subsequent module, the resulting pixel-error map is coupled with the output logits of the semantic segmentation to end up with a pixel-wise anomaly score. The extension module is trained on augmented road images, where patches of noise or a part of the input image are randomly positioned on the road and labeled as anomalous. The evaluation on various datasets shows the superiority of JSR-Net in comparison to others [264, 263, 40, 103] while preserving the closed-set segmentation performance.

A similar approach is evaluated by Ohgushi et al. [330] against the LaF benchmark on a highway dataset with real and synthetic road obstacles. In contrast to Vojir et al., they combine the entropy loss of the semantic segmentation network with the perceptual loss between the real and reconstructed image to form an anomaly map. They outline a set of post-processing steps where the final obstacle score map depends on the semantic information, the aforementioned anomaly map, and a superpixel division to refine local regions.

Di Biase et al. [112] leverage image re-synthesis [264] by combining the reconstruction error with two uncertainty maps of the segmentation network. The network outputs the softmax entropy and the distance between the two largest softmax values in addition to the segmentation output. Similar to [330], the perceptual difference is used as the reconstruction loss between the input and synthesized image. All predicted maps and the input image are fused in a spatial-aware dissimilarity module with three parts: encoder, fusion module, and decoder. In the fusion module, the encoded and re-synthesized inputs and the semantic image are concatenated and fused. The resulting feature map is evaluated against the jointly

encoded uncertainty and perceptual difference via point-wise correlation. The final pixel-wise anomaly segmentation is provided by decoding the fused features and spatial-aware normalization with the semantic information.

Generative: According to the FS, LaF, and Segment Me If You Can (SMIYC) obstacle track benchmarks, the dense anomaly detection with NFlowJS of Grcić et al. [149] outperforms all contemporary techniques and represents the SotA of camera-based anomaly detection[1]. NFlowJS simultaneously trains a Normalizing Flows (NF) to generate synthetic negative patches over regular images and a dense-prediction network on the resulting mixed-content images. The generated negative patches are thereby defined as the anomaly mask. During training, the discriminative model is encouraged to yield a uniform predictive distribution for the generated patch. This induces the generative distribution of the NF to move away from the inliers. At the same time, it is trained to maximize the likelihood of inliers. These opposing objectives support the generation of images at the boundary of the training data distribution while sensitizing the discriminative model for anomalies. In contrast to former generative models, the NFlowJS relies only on anomaly synthesis during training. Blum et al. [44] also evaluate an NF-based approach with logistic regression on their FS benchmark. However, the results are incomparable with NFlowJS.

Nitsch et al. [324] adopt and enhance a generative approach of Lee et al. [247] for the detection of object anomalies. Lee et al. propose an auxiliary Generative Adversarial Network (GAN) which encourages an object classifier to provide low confidence for samples outside the training distribution. Nitsch et al. extend the approach by a post hoc network statistic, which estimates a class-conditioned Gaussian distribution over the network's weights of the bottleneck layer. A cosine similarity metric determines the distribution distance and classifies a given sample based on an empirical threshold. Since they only perform classification, the localization of objects has to be done in advance.

Similarly, Lis et al. [264] adopt GANs to re-synthesize the input image and detect anomalies on the object level by the difference in appearance. Unlike prior works [429, 330], however, their image generation is based on the final semantic segmentation map rather than on intermediate feature representations. As the semantic segmentation preserves the scene layout but loses the precise scene's appearance, regular reconstruction errors, like the perceptual loss, would output a high overall difference without informative results. To overcome this, they propose a discrepancy network using the input image, a resynthesized image, and the semantic segmentation. Encodings from the input and the re-synthesized image are generated with two Visual Geometry Group (VGG)16 [394] networks with shared weights. A Convolutional Neural Network (CNN) processes the one-hot encodings of the semantic labels. At each encoding stage, the features of all three networks are fused and used as input for a decoding CNN on multiple stages, which up-scales the feature maps to the original image size to overlay the input

[1] As of the date of the original publication [BOG 13].

image with a pixel-wise anomaly mask. The semantic-to-image synthesis is also adopted and evaluated by [469, 167] in the form of a conditional GAN with a subsequent dissimilarity scoring.

Addressing domain-level anomalies, as introduced in Section 2.5, Löhdefink[2] et al. [278] present an approach for the detection of domain shifts. An autoencoder learns the domain of a given dataset in a self-supervised manner. The approach characterizes the training data domain via the distribution of the autoencoder's Peak Signal-to-Noise Ratio (PSNR). During inference, the Domain Mismatch (DM) is estimated by comparing the learned and incoming PSNR distribution of the data via the Earth-Mover's Distance (EMD). The evaluation shows a strong rank order correlation between the autoencoder's DM metric and the decrease of semantic segmentation performance when faced with target domains different than the source domain.

Feature Extraction: Another domain shift detection is proposed by Bolte et al. [46], where the Mean Squared Error (MSE) of feature maps is compared. The MSE is evaluated over entire datasets or batches. Similarly, Zhang et al. [504] propose the DeepRoad framework to validate single input images based on the distance to the training embedding of VGG network features [394]. Bai et al. [22] detect anomalies in urban road scenes and classify entire input scenes as anomalous. They identify a set of representatives for normal urban scenes via the k-means clustering of scale-invariant feature transform features. Finally, images are classified by a one-class Support Vector Machine (SVM).

Overall, many of the previously outlined techniques work without external data but require a retraining of the proposed extension module or entire detection architecture. However, the well-performing NFlowJS [149] technique utilizes outlier exposure, which has become a standard technique among SotA methods. The anomaly detection method presented in Chapter 5 does not require anomaly exposure during training, avoiding a bias towards known anomalies.

3.3. LIDAR Data

Most often, autonomous vehicles do not solely rely on camera data. Although RGB camera data has a high resolution and rich semantic information, it lacks an accurate depth measurement. Therefore, LIDAR sensors, which provide a three-dimensional depth map of the environment, are often found in sensor setups. While there is much research about local denoising of LIDAR point clouds [357, 27], this section focuses on anomalies on the object and domain level, where an entire cluster of points or a large and constant shift in appearance is considered as anomalous. Especially weather conditions like rain, snow, and fog heavily influence the data. All covered LIDAR-based methods can be found in Table 3.2.

[2]After first meeting Jonas at CVPR in 2022, he tragically passed away just a few days later. Reading his name here still fills me with sadness. I wish his family all the strength they need.

Author(s)	Ref.	Approach	Anomaly Level	Data
Zhang et al.	[499]	Confidence	Domain	Urban dataset
Cen et al.	[70]	Confidence	Object	UDI [70], KITTI [145]
Wong et al.	[459]	Confidence	Object	TOR4D [459], Rare4D
Masuda et al.	[296]	Reconstruction	Object	ShapeNet [78]

Table 3.2.: **LIDAR-based anomaly detection:** The overview shows the used approach and anomaly level. In addition, the used data is listed. Adapted from [BOG 13].

Confidence score: Research by Zhang et al. [499] shows that rain, in the context of a domain shift, affects the LIDAR measurement quality, as resulting point clouds are sparser, noisier, and the average intensity is lower. Therefore, they aim to quantify the LIDAR degradation with the Deep Semi-Supervised Anomaly Detection (DeepSAD) approach [367]. They first project 3D LIDAR data into a 2D intensity image. DeepSAD then transforms the images into a latent space, where all normal images, i.e., the scans without rain, fall into a hypersphere and all abnormal, i.e., rain-affected, images are mapped away from the hypersphere's center. Finally, the distance of a transformed test image to the learned center of the hypersphere is interpreted as the anomaly score. As the model architecture defines anomalies as those that fall out of the hypersphere, the proposed methodology is classified as a learned confidence detection approach. The trained DeepSAD reaches a Spearman's correlation of up to 0.82 between the rainfall intensity and degradation score on dynamic, simulated test data. This indicates a considerably accurate quantification of anomaly detection due to weather conditions.

In the past, several architectures have been proposed to detect objects in point clouds, like VoxelNet [514], PointRCNN [390], and PointNet++ [352]. However, these are based on a closed-set setting, thus being only capable of detecting classes that were included in the training set. In contrast, open-set detection methods are able to explicitly classify objects outside the closed set as unknown upon the regular detection of the predefined classes. The open-set setting, therefore, loosens the constraint to classify all detections as one of the predefined classes. Consequently, one expects the false positive rate to improve and the model to acknowledge the novelty of objects upon never-before-seen instances.

The idea of an open-set detector for 3D point clouds was first implemented by Wong et al. [459]. They propose an Open-Set Instance Segmentation (OSIS) network, which learns a category-agnostic embedding to cluster points into instances regardless of their semantics. The inference is based on a Bird's-Eye-View (BEV) LIDAR frame and consists of two stages: the closed-set and open-set perception. In the first stage, a backbone of 2D convolutions extracts multi-scale features, which are then fed into a detection and an embedding head. The latter is the core of OSIS and learns the category-agnostic embedding space. Moreover, the embedding head yields the prototypes of possible closed-set classes. Points are then associated with prototypes of known categories by the learned embedding space.

In the second stage, the remaining unassociated points are considered unknown. Those are clustered into instances of unknown objects via Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [126]. The outlined approach falls into the category of learned confidence scores, as the prototypes are learned during training, and unknown objects are identified by their uncertainty of class association. OSIS is evaluated on two large-scale, non-public datasets. Here, the technique outperforms other adapted deep learning-based instance segmentation algorithms for the detection of single-point anomalies on the object level.

The OSIS network is later used as a baseline for comparison of the Metric Learning with Unsupervised Clustering (MLUC) network developed by Cen et al. [70]. They focus on two primary tasks: identifying regions of unknown objects with high probability and enclosing these regions' points with proper bounding boxes. In the context of the first problem, the paper shows that the Euclidean Distance Sum (EDS), based on metric learning, is more suitable than a naive softmax probability metric to differentiate between regions of known and unknown objects. They replace the classifier of closed-set detections with the Euclidean distance representation to all prototypes of the embedding space. The Euclidean distance-based probability is incorporated into the loss function, such that the embedding vector of known classes is close to the corresponding prototypes of the respective class. However, unknown objects are mapped close to the center of the embedding, having a smaller EDS. The EDS measures the uncertainty of closed-set detections. Therefore, boxes with an EDS lower than a threshold λ_{EDS} are considered as regions of unknown objects. Similarly to OSIS, these bounding boxes of low confidence are then refined by unsupervised depth clustering. The MLUC considerably outperforms OSIS.

Reconstructive: Masuda et al. [296] show an approach to detect whether an object point cloud is anomalous or not. In contrast to the preceding methods, this technique is based on point clouds of single encapsulated objects. Since automotive LIDARs provide full environment scans, single objects or regions of interest would need to be extracted by detection or clustering approaches first. The proposed Variational Autoencoder (VAE) is based on the FoldingNet decoder [484] and learns to reconstruct the set of known objects which are considered normal. The point cloud is then classified as anomalous based on the reconstruction and the Chamfer distance as an anomaly score. The approach is evaluated on the ShapeNet [78] dataset, which also includes a variety of objects outside the AD domain. The results are promising, as the model achieves an average Area under the Receiver Operating Curve (AUROC) of 76.3%, where known classes were defined as anomalies.

Overall, anomaly detection on the object level in LIDAR data is gaining momentum, after research has already led to various closed-set detection architectures. The anomaly detection method presented in Chapter 5 leverages both camera and LIDAR data.

3.4. Multimodal Data

Autonomous vehicles are typically equipped with multiple modalities. In the following, an overview of techniques that identify anomalies based on irregularities between the individual sensors or by fusing information is provided. All covered multimodal methods can be found in Table 3.3.

Author(s)	Ref.	Approach	Anomaly Level	Data
Sun et al.	[398]	Feature Extraction	Scene	CS [101]
Ji et al.	[207]	Feature Extraction	Scene	Field environment
Gupta et al.	[158]	Feature Extraction	Scene	LaF [347]

Table 3.3.: **Multimodal anomaly detection:** The overview shows the used approach and anomaly level. In addition, the used data is listed. Adapted from [BOG 13].

Feature Extraction: Sun et al. [398] present a real-time fusion network for semantic segmentation based on Red Green Blue-Depth (RGB-D) data. The primary goal of the multimodal architecture is to improve image segmentation by incorporating depth information. Furthermore, they argue that the multi-source segmentation framework is also capable of detecting unexpected road obstacles, providing a unified pixel-wise scene understanding. However, the evaluation on the Cityscapes (CS) dataset [101] does not provide detection performance measures for the unexpected obstacles, as the approach concentrates on the semantic segmentation of closed-set classes. Another RGB-D based detection of road obstacles is implemented by Gupta et al. [158] in the form of MergeNet. As the architecture's name suggests, the model merges two networks, the Stripe-net and Context-net, via a third meta Refiner-net. The Stripe-net extracts low-level features of the RGB and depth data in parallel, based on images split in stripes. This forces the network to learn discriminative features within narrow bands of information and a small subset of parameters. Moreover, this allows for a more reliable detection of small road obstacles. In contrast, the Context-net is trained on the entire RGB image and is determined to learn high-level features. The Refiner-net acts as a meta network to combine the complementary features and end up with a form of curriculum learning. As a result, MergeNet is trained to discriminate between road, off-road, and small obstacles, where the latter is considered abnormal.

Ji et al. [207] propose a supervised VAE to merge multiple sensor modalities of different dimensionality. They show experiments with high-dimensional LIDAR data and low-dimensional data from wheel encoders. They abandon the decoder after training and use the learned encoder as a feature extractor. The modalities' latent representation is then, along with other encoded modalities, fed into a fully connected layer to identify an anomalous operation mode of the vehicle.

In summary, as shown in Figure 3.1, all of the multimodal anomaly detection techniques are based on the comparison of the individual modalities' extracted

features. As multimodal detection broadens the search space for potential anomalies while reducing the risk of false positives, the anomaly detection method presented in Chapter 5 leverages both camera and LIDAR data. Contrary to the feature-based approaches presented in this section, the method combines both reconstruction and prediction in order to provide pixel-based anomaly masks rather than classifications.

3.5. Abstracted Data

The previous sections present an overview of anomaly detection techniques suitable for specific sensor modalities. The following approaches focus on a more abstract level of pattern analysis, i.e., the detection of anomalous behavior in scenarios that are not necessarily bound to a sensor modality. Thus, the approaches are designed to detect anomalies on the scenario level [51] and deal with risky and abnormal driving behavior of non-ego vehicles. All covered abstraction-based methods can be found in Table 3.4.

Author(s)	Ref.	Approach	Anomaly Level	Data
Yang et al.	[480]	Prediction	Scenario	CARLA [117]
Bolte et al.	[45]	Prediction	Scenario	CS [101]
Liu et al.	[271]	Prediction	Scenario	CUHK [280], UCSD [290], ST [283]
Yuan et al.	[504]	Prediction Confidence	Scenario	Driving videos
Zhang et al.	[504]	Feature Extraction	Scenario	Udacity [417]
Stocco et al.	[397]	Reconstructive Confidence	Scenario	Udacity [417]

Table 3.4.: **Abstraction-based anomaly detection:** The overview shows the used approach and anomaly level. In addition, the used data is listed. Adapted from [BOG 13].

Prediction: Yang et al. [480] assess the behavior of driving vehicles based on Hidden Markov Models (HMMs) to detect anomalous scenarios. The observation states of the Markov model are provided by the Conditional Monte Carlo Dense Occupancy Tracker (CMCDOT) framework [368] and comprise real-time velocity as well as vehicle position through probabilistic occupancy grids. The framework derives these observations based on point cloud and odometry data. As a result, the pipeline can infer risky and abnormal driving behaviors in simulated multi-lane highway scenarios with two non-ego vehicles.

Bolte et al. [45] propose an anomaly detection on the scenario level, where patterns are observed over a sequence of sensor data, i.e., camera images. They quantify the anomalous behavior of moving objects, such as pedestrians or cars, by computing the error between the real and a predicted frame. The predicted frame is generated by an adversarial autoencoder and is based on the past sequence of input frames.

Hence, the anomaly score can also be interpreted as the non-predictability of the model. The model is evaluated with MSE, PSNR, and Structural Similarity Index Measure (SSIM) [443] metrics, and anomalous scenarios are determined by a threshold. They localize anomalous behaving objects by dividing the input image into grid cells of user-specific size and weight closer objects higher, as those pose a higher risk of collision.

A similar but more comprehensive approach is outlined in the paper of Liu et al. [271]. They adopt U-Net [365] as an image-to-image translation model to predict the next frame based on the past sequence of frames. In contrast to the former approach [45], their framework considers also temporal information of scenarios. They extend their objective function by an optical flow constraint to retain the motion information of moving objects. The optical flow is calculated via Flownet [116]. They leverage adversarial training to discriminate between real and fake images to further boost the performance of future frame prediction. Anomalous scenarios are again identified by the PSNR of the real and predicted frame exceeding a predefined threshold.

Reconstructive: Stocco et al. propose SelfOracle [397] for the detection of safety-critical misbehavior, like collisions and out-of-bound episodes. The architecture uses a VAE to reconstruct a set of preceding input images of a current scene and calculates the corresponding reconstruction errors. During the training on normal data, the model fits a probability distribution to the observed reconstruction errors via maximum likelihood estimation. The estimated distribution can then be used to determine a threshold value to distinguish between anomalous and normal behavior. In addition, SelfOracle implements a time-aware anomaly scoring by applying a simple autoregressive filter on the sequence of reconstruction errors, as the current error might be susceptible to single-frame outliers. While they evaluate SelfOracle only in a simulation environment, the approach seems promising and even outperforms the author's implementation of the DeepRoad framework.

Anomaly detection using abstracted data heavily depends on human driving behavior, as most methods leverage predictions. Therefore, with the rise of autonomous vehicles on the road, AD will experience a large concept drift in behavior prediction. Such an evolving normality is an expected phenomenon of data-based approaches. Similarly, the anomaly detection method presented in Chapter 5 is based on a data-defined representation of normality and leverages both reconstruction and prediction for the detection of anomalies. Abstract object data is used to refine detected anomalies with object-level masks.

3.6. Conclusion

This chapter provides an extensive overview of anomaly detection techniques for autonomous driving. Answering **RQ1**, it examines methods based on camera and LIDAR as typical sensors for autonomous vehicles. This way, trends and patterns

in the field of anomaly detection for autonomous driving are identified. Most of the recent advancements are concerned with image-based anomaly detection, while other modalities are still struggling to gain momentum. One reason for this is the absence of benchmarks, which are so far only established for camera-based methods. In addition, many well-performing approaches require outlier exposure, which poses the risk of missing unknown anomalies in the open world.

Addressing these identified patterns, Chapter 4 introduces a challenging and multimodal anomaly detection benchmark supporting both camera and LIDAR sensors. It includes both object-level and scenario-level anomalies. Subsequently, Chapter 5 introduces a self-supervised and multimodal anomaly detection method, not requiring labeled data or outlier exposure.

3.6.1. Recent Advances

The field has continued to evolve since the development and publication of the work underlying this chapter [BOG 13]. Recently, multiple surveys have confirmed the relevance and timeliness of the identified patterns of anomaly detection methods. Similarly to the structure of this chapter, as adapted from Breitenstein et al. [52], Rahmani et al. [355] classify anomaly detection methods into *reconstructive and generative*, *confidence*-based, *feature extraction*-based, and *other*, often predictive, methods. Shoeb et al. [391] categorize approaches into *Mask2Former* [92]-based, *uncertainty*-based, *generative*, and *other* approaches. This underscores the value of the structured analysis in this chapter and shows an intensified trend toward methods based on semantic segmentation models.

Addressing the complexity of existing benchmarks, Shoeb et al. [391] find that current benchmarks are saturated, and they agree to findings in [BOG 22], where anomaly detection benchmarks are examined in more detail, that "OOD detection suffers from under-complex street scenes" [391]. Existing anomaly detection benchmarks are discussed in more detail in Section 4.3.1. In addition, a focus on 2D image data remains a limiting factor [355, 396]. Furthermore, anomaly detection methods continue to rely on labeled datasets and outlier exposure during training [355, 396, 391]. Shoeb et al. are especially concerned about using outlier exposure, stating a "risk of overfitting to seen examples" [391].

These recent works underline the timeliness of the identified research problems presented in this chapter and show the continued relevance of the foundational research problems addressed in this dissertation. Looking forward, Shoeb et al. conclude that anomaly detection benchmarks require "[...] complexities incorporating temporal dynamics, multimodal sensor inputs" [391] and Rahmani et al. find a "growing interest in developing [...] methods to work more effectively with multi-modal sensor data" [355]. In line with these conclusions, Chapter 4 presents a challenging, multimodal benchmark with both object-level and scenario-level anomalies. Subsequently, Chapter 5 utilizes this benchmark for the evaluation of a multimodal anomaly detection method.

> "Data! data! data!" he cried impatiently. "I can't make bricks without clay."
>
> *Sherlock Holmes by A. C. Doyle, 1892* [118]

4. Anomaly Generation

Multiple supervised student theses have contributed to this chapter [STU 2, 7]. Parts of this chapter have previously appeared in the following publications:

- D. Bogdoll et al. *One Ontology to Rule Them All: Corner Case Scenarios for Autonomous Driving*. In European Conference on Computer Vision (ECCV) Workshop, 2023 [BOG 5]
- D. Bogdoll et al. *AnoVox: A Benchmark for Multimodal Anomaly Detection in Autonomous Driving*. In European Conference on Computer Vision (ECCV) Workshop, 2025 [BOG 6]
- D. Bogdoll et al. *Hybrid Video Anomaly Detection for Autonomous Driving*. In British Machine Vision Conference (BMVC) Workshop, 2024 [BOG 8]

4.1. Introduction

As shown in Chapter 3, anomaly detection benchmarks currently focus on camera data and include overly simplified scenes. In addition, a previous survey [BOG 22] found that anomalies are often ill-defined based on no clear definition of normality. Generally, no framework exists to generate scenarios with external anomalies based on the theoretical levels from the anomaly taxonomy introduced in Section 2.5.2 and shown in Table 2.1 [BOG 3]. Addressing **RQ2**, this chapter presents two methodologies to convert theoretical anomaly definitions from the literature into datasets containing anomalies. The first method offers comprehensive coverage for all considered anomaly levels, but focuses on the generation of individual scenarios. The second method focuses on specific types of anomalies, but allows for the scalable generation of numerous scenarios. All data is generated in the CARLA [117] simulation environment.

In general, there are two types of scenario generation approaches: knowledge-driven and data-driven [172]. While both require expert knowledge at some stage, knowledge-driven approaches "require substantial expertise" [59] on topics such as traffic dynamics or long-tail cases to create relevant scenarios, whereas data-driven approaches exploit "information contained in source data" [59]. Data-driven approaches are common when typical behaviors of traffic participants are extracted from real-world driving recordings, e.g., to achieve more realistic simulation environments. As the focus of this dissertation is on atypical scenarios, both

approaches presented in this chapter leverage knowledge about the theoretical anomaly levels introduced in Section 2.5 and are thus knowledge-driven.

Section 4.2 presents a methodology to generate expert-defined scenarios with anomalies from all levels. The scenario descriptions are based on an ontology[1], meaning that all scenarios are structured in a comparable way, allowing for later coverage analysis. This approach allows for the generation of a large-scale, structured scenario catalog. While such a catalog is useful to test an autonomous driving function, expert-defined scenarios require a great deal of manual effort, as each individual scenario is explicitly designed by a human. This results in a set of very specific scenarios. In an open world, however, a wide variety of situations can occur. Thus, the subsequent Section 4.3 focuses on the variability of scenarios with anomalies as a challenging benchmark for anomaly detection methods.

Section 4.3 presents a scalable methodology for the generation of object-level and scenario-level anomalies. Here, experts only define the ODD, such as location, time of day, and weather, and set some parameter limits. The scenarios are then generated programmatically. As a well-defined benchmark, it provides both training and evaluation data. This enables a fair comparison of anomaly detection methods, as they are trained based on the same definition of normality. For multimodality, a sensor setup of a typical recording vehicle is employed with ground truth labels provided for camera and LIDAR data. To allow for a comparison between anomaly detection methods using different sensors, ground truth is additionally provided in a voxelized form.

4.2. Individual Scenario Generation

This section presents a scenario generation method that is capable of generating data from all anomaly levels described by Breitenstein et al. [51], which was not possible previously. Based on the popular OpenSCENARIO [17] framework[2], human scenario designers can generate a wide variety of scenarios with anomalies. Using the resulting scenario-describing ontologies, synthetic data of scenarios with anomalies is generated automatically in simulation. The presented methodology is demonstrated by a scenario catalog comprising nine scenarios, including combinations of such containing different anomaly levels.

While there are other methods of generating scenarios [BOG 3], most are based on purely scripted descriptions [444]. This makes it challenging to understand the scenario coverage of a large-scale scenario catalogue. Differently, ontologies as used here support strong reasoning capabilities to infer scenario types purely based on generated ontologies [2]. While such an inference is not carried out in

[1] An ontology is a "formal explicit description of concepts in a domain" [327] that "includes machine-interpretable definitions of basic concepts [...] and relations among them" [327].

[2] OpenSCENARIO uses elements like Storyboard, Story, Act, ManeuverGroup, Maneuver, Event, and Action to describe scenarios. Details can be found in the OpenSCENARIO User Guide [17].

this dissertation due to the small, exemplary scenario catalog, the design choice of ontologies supports the systematic structuring and analysis of large-scale scenario catalogs.

4.2.1. Related Work

Ontologies are being widely used for the description of scenarios and have proven to be able to describe scenarios in great detail [184]. In the following, multiple scenario generation approaches are introduced and compared with respect to their ability to generate scenarios containing anomalies, as shown in Table 4.1. For this purpose, the requirements for an ontology to be able to describe all levels of anomalies, as introduced in Section 2.5, are derived first.

To describe all levels of anomalies, an ontology generally needs to be able to describe both static scenes and temporal scenarios. Furthermore, it needs to be able to describe arbitrary environments and arbitrary objects. Following an open-world assumption, arbitrary is defined with respect to environments and objects as the possibility to include such without changing any classes or properties of the ontology. This means, e.g., referencing external sources, such as OpenDRIVE files for environments or Computer-Aided Design (CAD) files for objects. An ontology needs to be designed in a way that the described scenarios can also be simulated. Finally, information about the anomaly levels needs to be included for details and knowledge extraction. To be useful, an ontology should also be available beyond its description in a scientific work. While some authors, such as [137, 201], released their ontologies previously, the provided links do not contain them anymore, which is why outdated sources are excluded.

Author(s)	Temporal Scenario Description	Arbitrary Environments	Arbitrary Objects	Scenario Simulation	Anomaly Categorization	Ontology Availability
Fuchs et al. [137]	-	-	✓	-	-	-
Hummel [201]	-	✓	-	-	-	-
Hülsen et al. [200]	✓	✓	-	-	-	-
Armand et al. [15]	-	-	-	-	-	-
Zhao et al. [509]	-	✓	-	-	-	✓
Bagschik et al. [21]	✓	-	-	-	-	-
Chen and Kloul [88]	✓	-	-	-	-	-
Huang et al. [197]	✓	-	-	-	-	-
Menzel et al. [302]	✓	✓	-	✓	-	-
Li et al. [257]	✓	✓	-	✓	-	-
Tahir and Alexander [402]	✓	-	-	✓	-	-
Hermann et al. [184]	-	✓	✓	✓	-	-
ASAM [16]	-	-	-	-	-	-
Presented Method	✓	✓	✓	✓	✓	✓

Table 4.1.: **Overview of ontology-based scenario descriptions:** Analysis of ontologies with respect to their suitability to describe and generate scenarios that include all levels of anomalies. Adapted from [BOG 5].

Based on these necessary attributes of ontologies to describe and generate anomalies on all considered levels, several related works are analyzed and compared

in Table 4.1. In the work of Bagschik et al. [21], an ontology is presented which describes simple highway scenarios based on a set of pre-defined keywords. In a later work, Menzel et al. [302] extend the concept to generate OpenSCENARIO [19] and OpenDRIVE [18] scenarios, while many of the relevant details were not modeled in the ontology itself, but in post-processing steps. For the description of the surrounding environment of a vehicle, Fuchs et al. [137] especially focus on lanes and occupying traffic participants, while neglecting their actions. Li et al. [257] also create scenarios which are executed in a simulation environment, covering primarily situations, where sudden braking maneuvers are necessary. Thus, their ontology is very domain-specific. They build upon their previous works [404, 460, 227]. Tahir and Alexander [402] propose an ontology that focuses on intersections due to their high collision rates. They show that their scenarios can be executed in simulation, focusing on changing weather conditions. While they claim to have developed an ontology, the released code [495] only contains scripted scenarios, which might be derived from an ontology structurally. Hermann et al. [184] propose an ontology for dataset creation, with a demonstrated focus on pedestrian detection, including pedestrian occlusions. Their ontology is structurally inspired by the Pegasus model [377] and consists of 22 sub-ontologies. It is capable of describing a wide variety of scenarios and translating them into simulation. However, since the ontology itself is neither described in detail nor publicly available, it does not become clear whether each frame requires a separate ontology or whether the ontology itself is able to describe temporal scenarios. In the OpenXOntology project by ASAM [16], an ontology is being developed with the purpose of unifying their different products, such as OpenSCENARIO or OpenDRIVE. Based on the large body of previous work in the field of scenario descriptions, this ontology is promising for further development. However, at the moment[3], it serves the purpose of a taxonomy. Finally, Gelder et al. [106] propose an extensive framework for the development of a "full ontology of scenarios". However, they have not developed the ontology itself, which is why their work cannot be compared to existing ontologies.

Next to ontologies which are explicitly designed to describe scenarios, more exist which also focus on decision-making aspects. In this category, Hummel [201] developed an ontology capable of describing intersections to a degree, where the ontology can also be used to infer knowledge about the scenes. While this is a general attribute of ontologies, she provides a set of rules for the analysis. Hülsen et al. [200] also describe intersections based on an ontology, focusing on the road layout, while interactions between entities cannot be modeled in detail. Armand et al. [15] address this issue and focus on such interactions. They also propose rules to infer knowledge from their ontology. These rules are partly attributed to the decision-making of an ego vehicle, e.g., whether it should stop or continue. Due to their strong focus on actions and interactions, they struggle to describe complex scenarios in a more general way. Zhao et al. [509] present a set of three ontologies, namely "Map", "Car", and "Control". Based on these, they are capable of describing complex scenes for vehicles only. While the scenes

[3] As of the date of the original publication [BOG 5].

do contain temporal information, such as paths for vehicles, these are only broad descriptions and not detailed enough to model complex scenarios. Huang et al. [197] present a similar work that is able to describe a wide variety of scenarios based on classes for road networks for highway and urban scenarios, the ego vehicle and its behavior, static and dynamic objects, as well as scenario types. However, it is designed to derive driving decisions from the descriptions instead of simulating these scenarios. Chen and Kloul [88], on the other hand, propose an ontology that is primarily designed to describe highway scenarios, with a special focus on weather circumstances.

Table 4.1 provides an overview of the related works and highlights the research gap addressed by the approach presented here. A trend can be observed where recent approaches focus more on the aspect of scenario simulation. However, no ontology has been able to describe and simulate long-tail anomaly events on all anomaly levels. The approach presented in this section fills this gap, being able to generate ontology scenarios for all considered anomaly levels and execute them in simulation.

4.2.2. Method

In order to generate anomaly scenarios, a developed master ontology[4] is the foundation for the creation of specific scenarios and provides the structure for all elements of a scenario. Based on this, all considered external anomaly levels, as introduced in Section 2.5, can be addressed. An overview of this process can be found in Figure 4.1. For the creation of scenarios, an ontology generator module is the interface to human scenario designers, who do not need any expertise in the field of ontologies in order to design scenarios. For each designed scenario, a scenario ontology is created. This is a major advantage over purely coded scenarios, as the complete scenario description is available in a human- and machine-readable form, which directly enables knowledge extraction, analysis, and further processing, such as exports into other formats or combinations of scenarios, for all created scenarios on any level of detail. Finally, the OpenSCENARIO conversion module converts this ontology into an OpenSCENARIO file, which can be directly simulated in the CARLA simulator.

Master Ontology

First, the master ontology is described, which is the skeleton of every concrete scenario. With its help, different scenarios can be described by instantiating the different classes, using individuals, and setting property assertions between them. The master ontology is closely aligned to the OpenSCENARIO documentation [17] since the ontology is used for the automatic generation of scenarios. Within the ontology, it is also possible to describe concrete anomalies based on the anomaly levels introduced in Section 2.5.

[4]The ontology and processing code are available on GitHub: https://github.com/fzi-forschungszentrum-informatik/corner_case_ontology

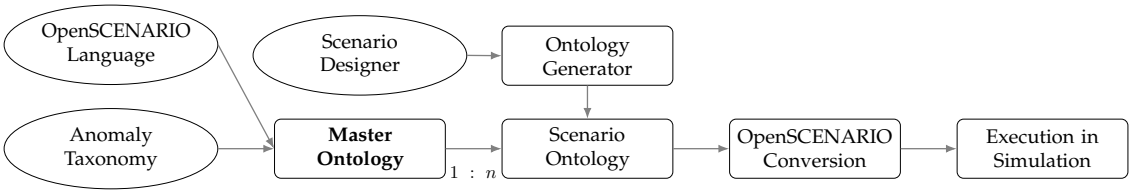


Figure 4.1.: **Generation of scenarios with anomalies:** Based on an anomaly taxonomy and the OpenSCENARIO language, a master ontology contains all necessary attributes to describe complex scenarios. In a $1 : n$ relation, ontologies describing individual scenarios can be derived. In an automated fashion, these scenarios are then converted into the OpenSCENARIO format, enabling the direct execution in simulation environments. Adapted from [BOG 5].

The master ontology, as shown conceptually in Figure 4.2 and in full detail in Section A.1 of the appendix, consists of 100 classes, 53 object properties, 44 data properties, 67 individuals, and 683 axioms. The 100 classes are either classes for the description of the anomaly category or derived from the OpenSCENARIO documentation [17], which means that the definitions of the different OpenSCENARIO elements can also be found there. They are used as parents for the individuals created within the ontology. The 53 object properties and the 44 data properties are used to connect the different parts of a scenario, in order to embed individuals into concrete scenarios. For a better understanding and more structured explanation, the master ontology can be divided into seven main groups: Scenario and Environment, Entities, Main Scenario Elements, Actions, Conditions, Weather and Time, and Anomaly Level. These will be described in more detail in the following, with each section highlighted in Figure A.1.

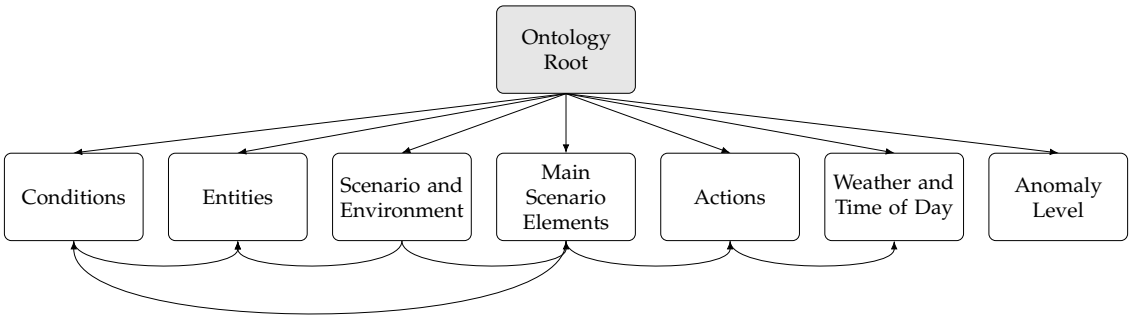


Figure 4.2.: **Master Ontology:** Main component groups of the master ontology and their relations. The complete ontology with these groups highlighted is displayed in full detail in Figure A.1.

Scenario and Environment: In order to be able to describe a scenario, the master ontology provides the scenario class, which is directly linked to the root of the ontology. Together with the scenario class, different object and data properties are provided. Those are used as connections between the different scenario elements,

such as the entities, towns, or the storyboard. Towns are CARLA-specific environments used in the ontology. CARLA allows users to create custom and, thus, arbitrary environments.

Entities: This group holds the different entities vehicle, pedestrian, bicycle, and misc. For arbitrary entities, the misc class can be utilized. If specific movement patterns are required, the classes vehicle, pedestrian, and bicycle are also already available. The individuals can be then connected to 3D assets from the CARLA blueprint library [65], which can be extended with external objects. This way, a scenario designer is able to add arbitrary assets into a scenario.

Main Scenario Elements: The main scenario elements are used to build the core of any scenario. The highest level is the storyboard, which includes an init and a story. A story has at least one act, which needs at least a *StartTrigger* and can optionally include a *StopTrigger*. Acts also are a container for different *ManeuverGroups*, that logically include maneuvers. The maneuvers then have to have a minimum of one event, which is also activated by a *StartTrigger*. Finally, each event needs to include at least one action. These are the main components of the OpenSCENARIO scenario description language and necessary parts of each scenario. For each of them, also a corresponding connecting property exists, i.e. *has_event*, *has_action*, *has_init_action*.

Actions: To be able to describe the maneuvers of the different entities, different actions are represented within the ontology, e.g., *TeleportAction*, which sets the position of an entity, or *RelativeLaneChangeAction*, which describes a lane change of an entity.

Conditions: As part of the *StartTrigger* and *StopTrigger* elements, conditions are used to activate them. Conditions are divided into two subclasses: *ByEntityCondition* and *ByValueCondition*. In general, the difference between those two is that the *ByEntityCondition* is always related to an entity, i.e., how close a vehicle is to another vehicle, while the *ByValueCondition* is always related to a value, i.e., the passed simulation time. Depending on the type of condition, different values must be met in order for the *StartTrigger* or *StopTrigger* to be activated. As an example, the *SimulationTimeCondition* can be used as a trigger with respect to the simulation time, using arithmetic rules.

Weather and Time of Day: To set the weather, the underlying CARLA town can be modified individually. This includes the weather conditions, which are subdivided into fog, precipitation, and the position of the sun. Also, the time of day can be set.

Anomaly Level: In the long tail of rare scenarios, each can be related to a specific anomaly level. The master ontology incorporates the top-level anomaly layers, as introduced in Section 2.5, but focuses on camera-based anomaly levels [51] in its current form. This makes extending the master ontology to include additional sensors straightforward, as they fall into the same top-level layers. Occurrences on the sensor layer, such as dead pixels or overexposure, can be simulated with subsequent scripts during the simulation phase and are not modeled by the

ontology alone. Details on those anomalies can be placed in the individual scenario ontologies by creating specific individuals of the respective anomaly classes of the master ontology.

Next to those groups, an additional 67 individuals exist, which are divided into *constants* and *default* individuals. There are two types of constants: OpenSCENARIO constants, such as arithmetic or priority rules, and CARLA constants, such as assets. The default individuals are used to help a scenario designer to create scenarios faster and easier. These include common patterns, such as default weather conditions or a trigger, which activates when the simulation starts running. In addition, a default ego vehicle is also included in the master ontology, which has a set of cameras and a bounding box attached to it. As the last part of the ontology, the 683 axioms represent the connections and rules between the entities and the properties within the ontology, along with the individuals.

Scenario Ontology Generation: Manual creation of ontologies is a time-consuming and error-prone process that requires expertise in the general field of ontologies and related software. To ensure that the OpenSCENARIO conversion module functions properly, the ontology generator module takes as input a scripted version of a scenario and creates a scenario ontology as a result. The concept behind the ontology generator is to use the master ontology as a base for a scenario description and automatically create the necessary individuals and property assertions between them, as shown in Figure 4.1. The master ontology is read by the ontology generator, and it uses, depending on the scenario, all classes, properties, and default individuals needed. The result is a new scenario ontology, which has the same structure as the master ontology with respect to classes and properties, but includes newly created individuals for the designed scenario.

Since the master ontology is built based on the OpenSCENARIO documentation [19], which is a very powerful and flexible framework, it allows for many possible combinations. This gives a scenario designer a large flexibility with respect to the design of new scenarios. This way, no prior experience with the OpenSCENARIO format is necessary. With the help of the ontology generator, every part that was defined within the master ontology can be utilized. Algorithm 1 shows, how a partly abstracted implementation, as done by a scenario designer, looks like. In Section A.2 of the appendix, an exemplary scenario ontology, which was generated by the ontology generator, is demonstrated. In this demonstration, the scenario ontology from Algorithm 1 is related to the visualization in Figure A.2.

Scenario Simulation: After a scenario is described with the help of individuals within a scenario ontology, it is read by the OpenSCENARIO conversion module, as shown in Figure 4.1. From these concrete scenarios, the conversion module generates OpenSCENARIO files. These can be directly simulated without any further adjustments. Since the OpenSCENARIO files include simulator-specific details, the ontology focuses on the CARLA simulation environment [66]. When the ontology generator module is used to create the scenario ontologies, their structural integrity is ensured, which is a necessary requirement for the conversion

Algorithm 1: Creation of a scenario ontology with the ontology generator including a domain-level anomaly, where the ego vehicle enters a foggy area (incl. abstract elements)

import OntologyGenerator **as** OG
import MasterOntology **as** MO
$ego\_vehicle \leftarrow MO.ego\_vehicle$ `//Default ego vehicle individual`
$weather\_def \leftarrow MO.def\_weather$ `//Default weather individual`

Initialize teleport_action(ego_vehicle), speed_action(ego_vehicle)
$init\_scenario \leftarrow OG.newInit(speed\_action, teleport\_action, weather\_def)$
`//Starting conditions for storyboard`

Initialize traveled_distance_condition
$Trigger \leftarrow OG.newStartTrigger(traveled\_distance\_condition)$ `//Trigger condition: Ego vehicle traveled defined distance`

Initialize weather(sun, fog, precipitation)
Initialize time_of_day, road_condition
$env \leftarrow OG.newEnv(time\_of\_day, weather, road\_condition)$
$env\_action \leftarrow OG.newEnvAction(env)$ `//Foggy environment after trigger`

Initialize Event, Maneuver, ManeuverGroup, Act, Story, Storyboard
`//Necessary OpenSCENARIO elements`

Export ScenarioOntology

module. This means that each scenario ontology is correctly provided to the conversion module. Theoretically, scenario ontologies can also be created manually to be processed by the conversion module. However, human errors are likely during such manual processes, preventing the correct processing by the conversion module.

While each scenario ontology is able to cover multiple anomalies, the created ontologies are fully modular. This means, given the same environment, the method is capable of combining multiple, already existing scenario ontologies into a new single scenario ontology. In such cases, where the number of scenario individuals is $n > 1$, a pre-processing stage is triggered, which extends the ontology to combine all n provided scenarios into a single new scenario S_{fusion}. For this purpose, this stage creates a new scenario, storyboard, and init. Subsequently, for every included scenario, the algorithm goes through its stories, entities, and init actions and merges them in S_{fusion}. For the final creation of the OpenSCENARIO file, the conversion module utilizes the property assertions between individuals to create the according Python objects, which are then used by the pyoscx library [351] to create the OpenSCENARIO file. These files can then be read by the ScenarioRunner [67] and executed in CARLA. In the following Section 4.2.3, a set of nine simulated scenarios is demonstrated.

4.2.3. Evaluation

For the evaluation, a diverse scenario catalog containing scenarios from all considered anomaly levels has been created. These cover different levels of complexity, starting with simpler content layer cases and ending with highly complex temporal layer cases. This serves as a qualitative evaluation to demonstrate the feasibility of the methodology as shown in Figure 4.1. Here, descriptions made by a scenario designer are converted into scenario ontologies and executed in simulation.

#	Anomaly Level	Individuals	Scenario Description
(a)	Domain Level	94	Domain Shift: Sudden weather change
(b)	Object Level	93	Single-Point Anomaly: Unknown object on the road
(c)	Scene Level	164	Collective Anomaly: Multiple known objects on the road
(d)	Scene Level	111	Contextual Anomaly: Known non-road object on the road
(e)	Scenario Level	94	Novel Scenario: Unexpected event in another lane
(f)	Scenario Level	104	Risky Scenario: A risky maneuver
(g)	Scenario Level	95	Anomalous Scenario: Unexpected traffic participant behaviour
(h)	Combined: (c) and (e)	156	Combined: Collective and Novel Scenario
(i)	Combined: (e) and (g)	122	Combined: Novel and Anomalous Scenario

Table 4.2.: **Overview of scenarios:** The scenario ontologies are derived from the master ontology and executed in simulation. These exemplary scenarios cover all considered external anomaly levels. Adapted from [BOG 5].

For the selection of the exemplary anomaly scenarios, three types of sources were considered. First, examples provided by the literature were used, such as the ones provided by Breitenstein et al. [52]. Second, various video sources, such as third-person videos, and dash-cam videos of traffic situations [288] were used for inspiration. Third, multiple brainstorming sessions took place, where personal experiences were collected. Afterward, the selection was narrowed down to a set of seven representative scenarios. Two more were created by combining two of those seven scenarios. An overview of these scenarios can be found in Table 4.2. These scenarios demonstrate that all considered anomaly levels, as introduced in Section 2.5, can be generated with the presented approach.

Visualizations of all scenarios can be found in Figure 4.3. The (a) domain-level scenario shows a sudden weather change, where the ego vehicle suddenly drives into dense fog. For the (b) object-level scenario, a falling vending machine on the road is simulated. The (c) collective scene-level anomaly includes a lot of running pedestrians in front of the ego vehicle, which could happen during a sports event. The next scenario, (d) contextual scene level, also has falling objects on the road, but in this case, the objects are traffic signs. This can be considered, for example, in a very windy environment. For the (e) novel scenario-level scenario, the scenario includes a cyclist performing unexpected maneuvers in the opposite lane. The (f) risky scenario-level scenario shows a close cut-in maneuver in front of the ego vehicle. The last anomaly category is the (g) anomalous scenario-level, where a pedestrian suddenly runs in front of the ego vehicle. To demonstrate the scalability of the approach, scenarios are also combined. In the combined scenario (h), a

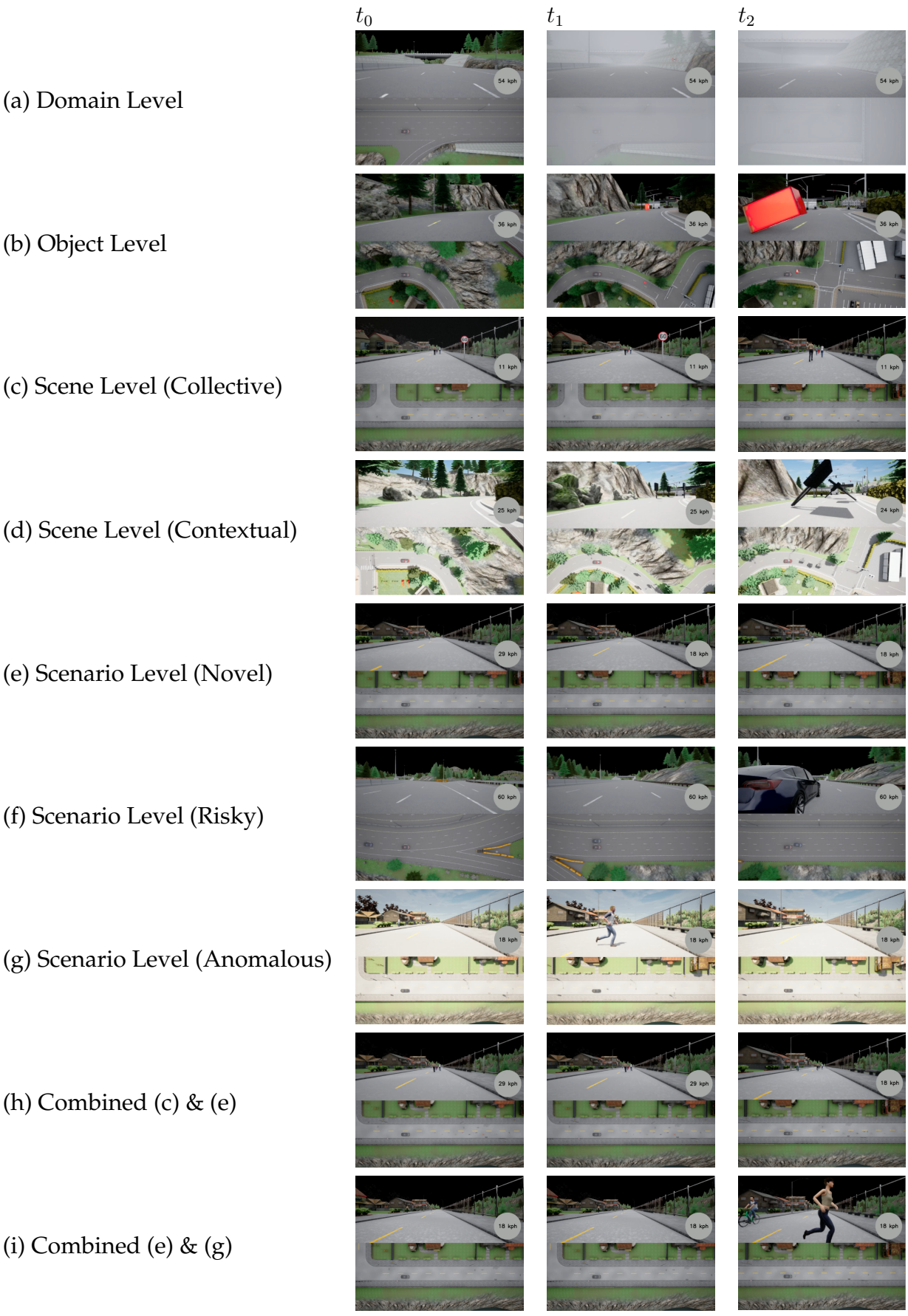


Figure 4.3.: **Scenarios in simulation:** Visualization of the simulated scenarios with anomalies, as listed in Table 4.2. Adapted from [BOG 5].

collective and a novel anomaly are included, where a lot of running pedestrians are in front of the ego vehicle, while a cyclist performs unexpected maneuvers in the opposite lane, next to the pedestrians. In addition, the novel scenario is combined with the anomalous scenario, resulting in (i), where a pedestrian walks in front of the ego vehicle and the cyclist. At the core of each demonstrated scenario lies a scenario ontology, as shown in Figure 4.1. More details on these scenario ontologies can be found in Section A.2 of the appendix, where the construction of the scenario ontology for the (a) domain-level scenario, where a vehicle enters a foggy area, is presented.

4.2.4. Summary

In the context of autonomous driving, the presented methodology enables the generation of scenarios with anomalies from all considered external anomaly levels, as introduced in Section 2.5.2, in simulation. Based on a master ontology and human-designed scenarios, concrete scenario ontologies are automatically derived and used for execution in simulation. The approach was demonstrated with a set of nine concrete scenarios. While these scenarios demonstrate a wide variety of anomalies, their design is still labor-intensive. This is valuable for the generation of a specific scenario catalog, but does not allow for larger-scale testing. In addition, the focus of this section is on the generation of scenarios and not on the benchmarking of anomaly detection algorithms, which requires additional effort. In the following, Section 4.3 demonstrates a scalable approach that focuses on large-scale scenario generation and anomaly detection benchmarking, rather than the design of individual scenarios.

4.3. Scalable Scenario Generation

As demonstrated in Section 4.2 and as is evident from the literature, expert perspectives are commonly used to judge individual long tail data points [51, 52, 178, 345, 366][BOG 5, 3]. To generate a larger number of scenarios with anomalies, this section introduces a scalable and adaptable method to generate data with anomalies to benchmark anomaly detection methods in AD. The approach focuses on anomalies at the object and scenario levels, as introduced in Section 2.5.

Temporal scenarios with a high scene complexity and many frames without anomalies require low false-positive rates for good anomaly detection performance. In related benchmarks involving reduced scene complexity and anomalies in every frame, false positives are less likely to occur. The approach presented here generates sensor data and ground truth in all modalities, as well as a spatial voxel representation, as shown in Figure 4.4. Classically, anomaly detection methods are evaluated in their respective sensor space. This prohibits the comparison between methods using different sensors. Here, the evaluation takes place in a voxelized

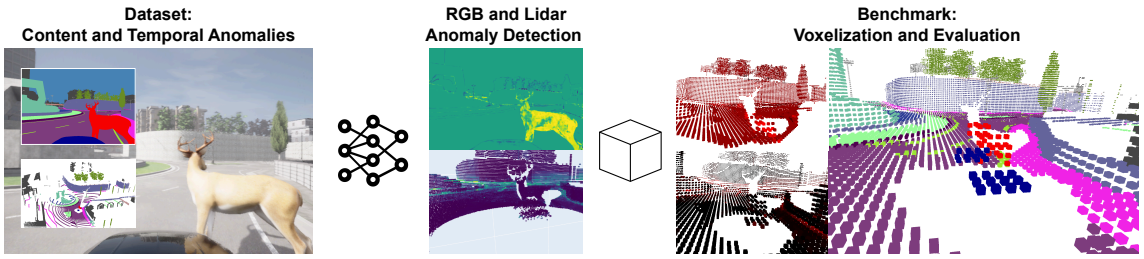


Figure 4.4.: **Multimodal Anomaly Detection Benchmark:** The stages of the anomaly detection benchmark are exemplarily shown with a deer as an object-level anomaly. Left: Scene and ground truth provided by the dataset for both camera and LIDAR data. Middle: Results for anomaly detection methods in camera and LIDAR data. Right: Anomaly detection results are converted into a common voxelized space and evaluated based on the voxelized ground truth. Reprinted from [BOG 6].

3D space, enabling the comparison of methods based on camera and LIDAR data. Next to providing a benchmark with labeled anomalies, the method is also able to generate large-scale training data that follows a clear definition of normality and does not include anomalies. For anomaly detection methods that are only trained on raw, unlabeled data, this clear definition of normality is of high importance. This is especially relevant given the surge in foundation models [68, 333], world models [503, 192], and large-scale pre-training approaches [185, 481], which often do not need labeled data for training and can be utilized for anomaly detection. Providing the means to generate large-scale training data that is aligned with the data included in the benchmark addresses many limitations of current benchmarks. Both the generated training data and the evaluation benchmark are utilized for anomaly detection in Chapter 5.

4.3.1. Related Work

While there exists a considerable number of datasets in autonomous driving [269, 251][BOG 21], only a few are designed for anomaly detection [BOG 22, 13]. These existing datasets, however, have significant limitations. First, the majority are camera-only, despite autonomous driving relying on multi-modal sensor setups. Second, temporal information is often overlooked, limiting anomaly detection to single frames. Additionally, most benchmarks show overly simplified traffic scenes, for example, an otherwise-empty road with only an object-level anomaly on it. This allows for the assumption that anything on the road might be an anomaly [287]. Most anomalies in benchmarks are human-defined, such as dogs on the street [44]. This can make the detection of such anomalies challenging, as these classes are typically also included in training datasets but not annotated [101, 119] [STU 9]. As a result, anomaly detection methods can miss anomalies they

Dataset	Data	#Frames	Ano. Source	Ano. Layer	#Ano. Classes	Ground truth	Temp. Data	Ego Act.	Reg. Tasks	Normality
Fishyscapes [43, 44]										
FS Lost and Found	RGB	375	Recording	Content	1	Sem. Mask (2D)	—	—	—	—
FS Static	RGB	1,030	Data Augmentation	Content	1	Sem. Mask (2D)	—	—	—	Cityscapes$^{\circledast}$
Crash to Not Crash [221]										
YouTubeCrash	RGB	2,400	Web Sourcing	Temporal	1	Bbox (2D)	✓	—	✓	YouTubeCrash
GTACrash	RGB	154,400	Simulation	Temporal	1	Bbox (2D)	✓	—	✓	**GTA V**
CAOS [181]										
StreetHazards	RGB	1,500	Simulation	Content	1	Sem. Mask (2D)	✓	—	✓	CARLA†
BDD-Anomaly	RGB	810	Class Exclusion	Content	3	Sem. Mask (2D)	—	—	✓	BDD100K$^{\circledast}$
SegmentMeIfYouCan [75]										
RoadAnomaly21	RGB	110	Web Sourcing	Content	1	Sem. Mask (2D)	—	—	—	Cityscapes†
RoadObstacle21	RGB	412	Recording	Content	1	Sem. Mask (2D)	✓	—	—	Cityscapes†
Rare Road Objects [55]										
Synthetic Fire Hydrants	RGB	30,000	Simulation	Content	1	Bbox (2D)	—	—	—	CARLA†
Synthetic Crosswalks	RGB	20,000	Simulation	Content	1	Bbox (2D)	—	—	—	CARLA†
CODA [253]										
CODA-KITTI	RGB, LIDAR	309	Void Classes	Content	6	Bbox (2D)	—	—	—	KITTI$^{\circledast}$
CODA-nuScenes	RGB, LIDAR	134	Void Classes	Content	17	Bbox (2D)	—	—	—	nuScenes$^{\circledast}$
CODA-ONCE	RGB, LIDAR	1,057	OOD Detection	Content	32	Bbox (2D)	—	—	—	ONCE$^{\circledast}$
CODA2022-ONCE	RGB, LIDAR	1,057	OOD Detection	Content	29	Bbox (2D)	—	—	✓	ONCE$^{\circledast}$
CODA2022-SODA10M	RGB	8,711	OOD Detection	Content	29	Bbox (2D)	—	—	✓	SODA10M$^{\circledast}$
W-OOD Tracking [287]										
Street Obstacle Sequences	RGB, Depth	1,129	Recording	Content	13	Inst. Mask (2D)	✓	—	—	Cityscapes†
CARLA-WildLife	RGB, Depth	1,210	Simulation	Content	18	Inst. mask (2D)	✓	—	✓	**CARLA**
Misc										
Lost and Found [347]	Stereo RGB	2,104	Recording	Content	42	Sem. Mask (2D)	✓	—	—	—
WD-Pascal [40]	RGB	70	Data Augmentation	Content	1	Sem. Mask (2D)	—	—	—	WildDash$^{\circledast}$
Vistas-NP [151]	RGB	11,167	Class Exclusion	Content	4	Sem. Mask (2D)	—	—	✓	Mapillary Vistas$^{\circledast}$
MUAD [136]	RGB, Depth	4,641	Sim., Class Exclusion	Content	9	Sem. Mask (2D)	—	—	✓	MUAD
DeepAccident [440]	RGB, LIDAR	57,000	Simulation	Temporal	9	Bbox (3D)	✓	—	✓	**CARLA**
Presented Benchmark	**RGB, LIDAR Depth**	**245,600**	Simulation	**Content Temporal**	**178**	**Inst. mask (2D,3D) Voxel (3D)**	✓	✓	✓	**CARLA**

Table 4.3.: **Perception-based anomaly detection benchmarks:** In the *Normality* column, † denotes a domain shift between normal data and the proposed dataset, and $^{\circledast}$ denotes that the data with anomalies is based on a subset of the normal data. Adapted from [BOG 22].

do not perceive as atypical when benchmarks do not differentiate well enough between ID and OOD data.

For the analysis of the SotA, works with published open-access perception datasets from an ego perspective that provide pixel- or point-wise ground truth have been included. Frameworks or methodologies that do not provide explicit, downloadable data [474, 383, 170, 454, 359], such with missing or incomplete data [148, 83, 235], and any that provide only frame-wise annotations [371, 497, 413, 8] are neglected. As shown in Table 4.3, it can be observed that most benchmarks are small and designed for camera-based content anomaly detection, providing ground truth in the form of semantic masks. Mostly, included anomalies on the content layer fall into the object or scene level, as introduced in Section 2.5. While some datasets provide only a single anomaly class, it has become more common to provide more granular labels for included anomalies. Among the datasets including content anomalies, just the Corner Case Dataset (CODA) [253] family includes LIDAR data but only provides ground truth in the form of 2D bounding boxes in the camera space. The DeepAccident [440] benchmark is the only one to provide temporal 3D labels for LIDAR point clouds.

There are different categories of how anomalies are integrated [BOG 22]. *Recording* and *Simulation* are similar in the way that selected anomalies are directly introduced into the data. This way, the anomalies are truly part of the envi-

ronment [347, 55, 287]. The definition of what counts as an anomaly can vary between benchmarks, though. *Data Augmentation* synthetically manipulates a given scene [BOG 27, 16], typically following a copy-and-paste pattern, where images of anomalies are pasted onto an already existing scene from another dataset. This way, a distribution shift between the anomaly and the underlying data is introduced [44, 40]. *Web Sourcing* describes the process of manually curating images from the web that are deemed anomalous [75, 221]. *Class Exclusion* is based on existing datasets and removes selected classes from the training data, thus treating them as anomalous, while the classes themselves remain rather normal from a human point of view [150, 181]. While this definition of anomalies aligns with the one presented in Section 2.5, it prevents simultaneously detecting anomalies and regular objects from the classes now removed from the training data. *Void Classes* utilizes void or misc classes from existing datasets and labels them as anomalous. This can be done with additional labeling guidelines to only relabel selected ones [253]. Finally, *OOD Detection* uses an anomaly detection method to derive anomaly proposals from a dataset which can then be labeled, typically after a human quality inspection [253]. Table 4.3 provides further information about dataset characteristics. While some datasets include temporal data in the form of sequences and provide labels for regular tasks, such as object detection or semantic segmentation, none include state information about the ego vehicle. However, during the deployment of an autonomous vehicle, state information is generally available and can be leveraged for the detection of anomalies. Most benchmarks do not provide a definition of normality [44, 75, 181]. Sometimes, even unlabeled anomalies occur in evaluation data [440]. This makes it particularly hard for un- and self-supervised methods to precisely detect anomalies if the semantic training distribution is not fully known or not defined at all. Especially desirable is a well-defined normality that allows for the generation of compliant training data. This is only feasible in simulation, where full control over both the training and evaluation data is available.

4.3.2. Method

This section presents the generation of both data that represents normality and data including content and temporal anomalies. The presented methodology can be used with arbitrary vehicle setups and a wide selection of parameters to create data. This does not only allow for the detection of anomalies in known environments but also for the detection of anomalies under domain shifts. First, a formalized definition of normality is provided. It is demonstrated how the generated training data follows this definition of normality. Next, the possible types of scenarios are presented. Finally, an overview of a generated exemplary dataset is provided.

Definition of Normality

In the literature, the definition of normality was often not extensively discussed when anomaly detection benchmarks were presented. A typical solution is to define normality as all semantic classes from the Cityscapes dataset [44, 75]. However, such a conceptual definition is not necessarily related to the content of the data used for training. To link it to the training data requires fully labeling the dataset in order to be aware of all semantic classes representing normality.

The presented method provides full control of both normality and anomalies in synthetic environments. This ensures that anomalies included in the benchmark are true anomalies and are not included in an unlabeled training dataset. For a fair benchmark of anomaly detection methods, it is important that they share the same definition of normality as defined by the training data, rather than by expert-defined concepts. Otherwise, different anomaly detection methods might use different training data, and thus different representations of normality. This can lead to a misalignment of what constitutes an anomaly, which can harm the performance on benchmarks, where anomalies are sometimes arbitrarily defined and are not always known to benchmark participants. Defining normality through training data requires the possibility of generating large amounts of data following a formal definition of normality, which is challenging in the real world, as anomalies would certainly occur in fleet-sized, unlabeled datasets.

In addition, in the field of autonomous driving as a subfield of embodied Artificial Intelligence (AI), there is more to the training data than just frames: There is a recording entity that performs actions, and there is temporal context, all of which contribute to a definition of normality. A formal definition of normality based on three categories is provided in the following. Subsequently, a concrete definition of normality for the CARLA simulation engine is provided in Table 4.4 that allows for the generation of compliant datasets. The definition has an ego, domain, and physical entities component:

Ego: Domain shifts in data can be induced not only by novel environmental conditions but also by different capturing methods. In autonomous driving, this especially refers to sensor types and configurations. In addition, temporal changes in the environment are heavily influenced by an agent's own actions. Thus, the behavior of the ego vehicle also counts towards normality.

Domain: With the domain, the static environment around the vehicle is described. This includes the geographical areas the vehicle has traversed, but also seen weather types and time of day specifications.

Physical Entities: These are the dynamic actors in the scene, most typically other vehicles, cyclists, and pedestrians. However, also categories such as animals or potentially moving objects can be included here.

Such a formal definition of normality allows for a clear understanding of what constitutes an anomaly. As training data can be generated based on the formal

Category	Description	Presented Methodology
Ego		
Ego vehicle	Recording vehicle	Lincoln MKZ 2020
Sensor configuration	Sensor types, placements	Configurations Mono, Stereo, Multi, Surround
Ego behavior	Driving characteristics	Behavior Agent (actions)
Domain		
Area	Geographical area	Towns 01,02,03,04,05,06,07,10HD
Environment	Weather, time of day	ClearNoon, CloudyNoon, WetNoon, WetCloudyNoon, HardRainNoon, SoftRainNoon, ClearSunset, CloudySunset, WetSunset, WetCloudySunset, MidRainSunset, HardRainSunset, SoftRainSunset
Physical entities		
Traffic participants	Vehicles, VRUs	Vehicles, Walkers $\in$ Blueprint library
Vehicle behavior	Driving characteristics	Traffic Manager Autopilot
Pedestrian behavior	Movement characteristics	AI Walker

Table 4.4.: **Definition of normality:** The first two columns list categories and their descriptions of attributes formally defining normality. The third column shows how these attributes can be implemented to define normality with the CARLA simulation engine to generate training data that aligns with the formal definition. Adapted from [BOG 6].

definition of normality, as shown in Table 4.4, it is guaranteed that all anomalies that are introduced in the remainder of this chapter are exclusive to the evaluation dataset and thus true anomalies.

Scenario Generation

The presented method is designed for configurable, large-scale datasets and supports generating training data representing normality that contains only normal samples as well as evaluation data that includes anomalies[5]. As shown in Figure 4.5, first, the vehicle sensor configuration needs to be set. An arbitrary number of camera, LIDAR, and depth sensors can be positioned freely on the ego vehicle of choice. This allows for the replication of existing sensor setups, the alignment with other datasets, or the testing of new configurations. Four such vehicle sensor configurations come pre-defined. As shown in Figure 4.6, the *mono* configuration consists of a LIDAR and a camera, which are centered on top of the vehicle, centered forward. The *stereo* setup consists of two cameras at the front edge and both a camera and LIDAR on top. The *multi* setup adds rear-facing cameras and additional LIDAR sensors at the front and rear, positioned at a lower level compared to the roof-mounted LIDARs. Finally, the *surround* setup provides a full 360° camera view next to a top-mounted LIDAR. Every RGB camera is automatically configured with an accompanying depth camera.

[5]The data generation code is available on GitHub: https://github.com/fzi-forschungszentrum-informatik/anovox

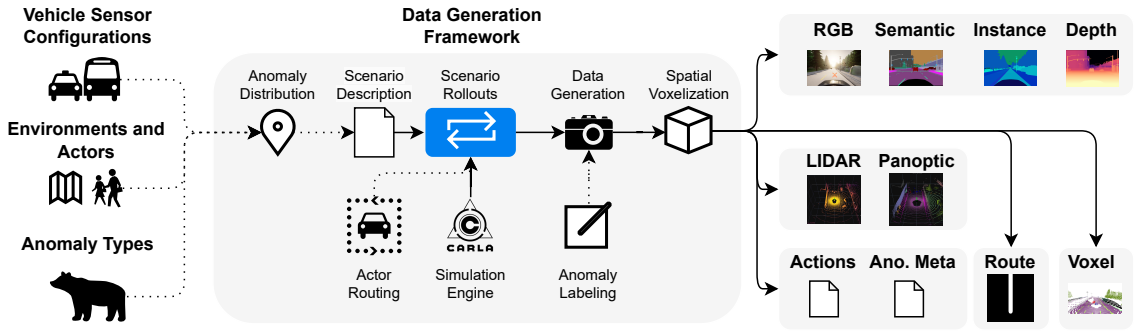


Figure 4.5.: **Data Generation:** Highly configurable scenario creation for both data representing normality or including content or temporal anomalies. Generated datasets include rich labels and ground truth in 2D, 3D, and a spatial voxel space. Adapted from [BOG 6].

Second, the environment and actors need to be set. Eight different regions and 14 weather and time of day presets are supported. Pedestrians, cyclists, as well as multiple types of vehicles, can be spawned. Third, the type of anomaly needs to be defined. The approach supports the creation of normality without anomalies, the placement of object-level anomalies, and the activation of scenario-level anomalies. By removing domains from the training dataset, it also supports the detection of anomalies under domain shifts. Given these configurations, scenario flows are pre-computed and stored as scenario descriptions. Thus, metadata describing all scenarios is available and allows for effortless dataset analysis.

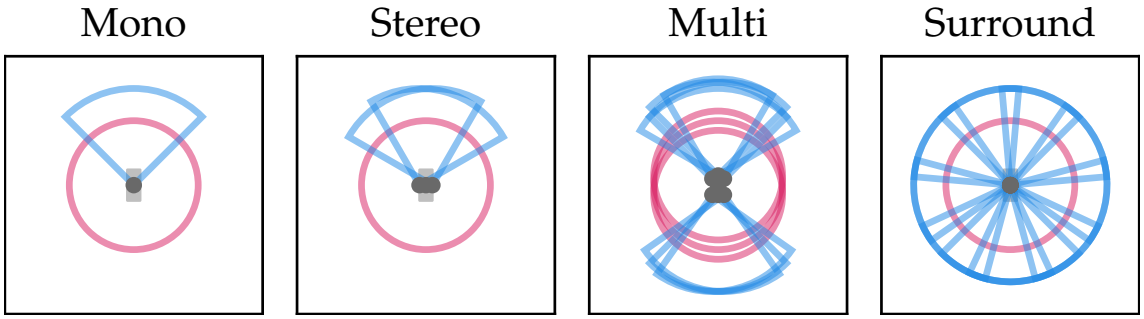


Figure 4.6.: **Preconfigured sensor configurations:** Blue wedges visualize RGB cameras, red circles visualize LIDAR sensors. Reprinted from [BOG 6].

Next, the method executes the driving scenarios in simulation. A custom-built CARLA simulation engine [117] includes all content anomalies that were manually collected and processed. Each scenario has a length of 20 seconds and is recorded at 10 Hz, resulting in 200 frames. When a scenario starts, the ego vehicle is spawned in the world and follows a given route to its target. At some point along the way, a content or temporal anomaly appears. To guarantee that the anomaly is reached in time, a green wave is activated along the route of the ego vehicle. Since physics computations remain active in the simulation, the ego vehicle will make contact with the anomalies, which leads to realistic collisions.

Actor Routing: Anomalies on the road do the same thing in simulation as they would in real life - they cause traffic jams. This makes it often unfeasible to reach the anomaly for the ego vehicle. Thus, a filter and rerouting algorithm for all other vehicles in a scenario is deployed. First, all actors close to the anomaly spawn point and on the direct path toward the anomaly are filtered out. Then, all planned paths are continuously monitored, and vehicles are rerouted whenever they would enter a lane with an anomaly on it. This rerouting technically changes their driving behavior. As shown in Table 4.4, in the training data, only vehicles are present, which show an autopilot driving behavior. Rerouting makes them switch into a behavior agent. This is addressed by providing novel labels, as further explained later on. This way, false positives, which might occur due to a violation of the alignment with normality, can be filtered out for evaluation.

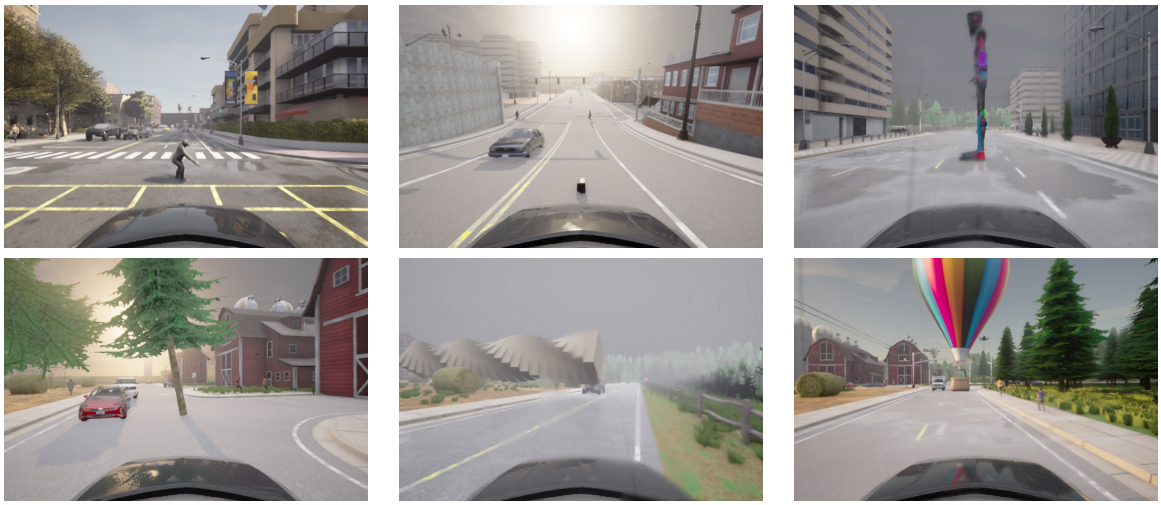


Figure 4.7.: **Content anomalies:** Examples from the six categories: An ape as an animal anomaly, an old tv as a home anomaly, a statue as a special anomaly, a tree as a nature anomaly, a pillar as a falling anomaly, and a hot air balloon as an airplane anomaly. Reprinted from [BOG 6].

Content Anomalies: A total of 178 different content anomalies are provided in five different size classes: tiny, small, medium, big, and huge. Semantically, they hierarchically belong to six different super-classes. Every anomaly has an individual label next to its super-class and size label. The class *animal* includes 33 animals of different sizes. The category *home* includes 53 typical household items such as furniture, tables, backpacks, or cardboard boxes. The category *special* includes 67 objects of rather atypical types and such that fall into a misc category, some of which could appear in the real world in the form of costumes or cuddly toys. The class *nature* includes 12 outdoor objects, such as rocks or wood. In the category *falling*, 9 large objects are provided, such as novel trees, that were spawned in an unstable position, which made them fall over. Finally, the *airplane* class consists of four types of large flying objects.

As especially large anomalies sometimes require manual positioning, the classes *home*, *special*, and *animal* are used for automated scenario generation. Here, all

anomalies are placed in critical positions along the way distributed around lane centers. For the classes *nature*, *falling*, and *airplane*, a manually curated dataset is generated with anomalies in geographic areas with large free-space areas.

Temporal Anomalies: As both Section 4.2 and prior work have shown that the generation of knowledge-driven temporal anomalies requires high manual engineering efforts [BOG 5, 12][8], only a single type of temporal anomalies is considered. Here, sudden braking scenarios of a lead vehicle are implemented, which are both safety-critical and typical scenarios in everyday traffic [221]. While a planned route for the ego vehicle is set in the scenarios containing object-level anomalies, here, the same route is used for a lead vehicle. Then, the ego vehicle follows the lead vehicle. Along the route, the lead vehicle will perform a sudden brake with negative acceleration values much higher than those seen during training. While braking, the lead vehicle is labeled as an anomaly directly in the sensor and voxel data in the same way as labeled content anomalies.

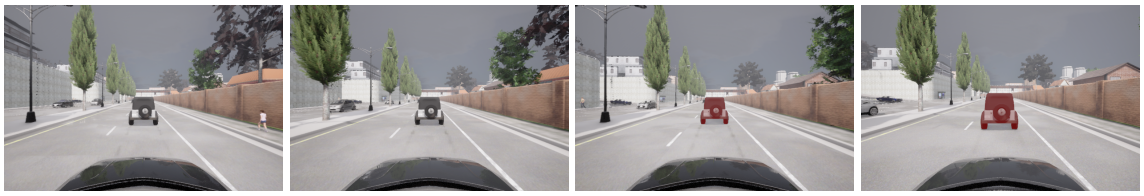


Figure 4.8.: **Temporal anomalies:** This scenario shows the implemented type of temporal anomalies. The first two images show the regular vehicle following mode. The last two images show the active braking maneuver with overlayed ground truth in red. Reprinted from [BOG 6].

Data Generation: As shown in Figure 4.5, sensor data for all positioned RGB cameras, depth cameras, and LIDAR sensors is provided. For regular perception tasks, panoptic masks for both camera and LIDAR are provided. For each frame, also information about the state of the ego vehicle is collected, such as its actions throttle, street, and brake. In addition, a standard-format BEV representation of the planned route is provided to support approaches that might be used for the detection of anomalies and require additional information about the planned route [507, 28, 191][BOG 23]. This data is also leveraged by the anomaly detection method presented in Chapter 5.

For the anomalous instances, metadata, such as their positions and their size, is provided. Ground truth is embedded into the semantic masks for both camera and LIDAR. As the benchmark is designed for the comparison of methods that use different modalities for the detection of anomalies, all anomalies are also represented in a 3D voxel grid with a customizable grid size. This voxel grid is used in Section 4.3.3 for the evaluation of anomaly detection methods irrespective of the sensor modality used. Based on depth maps and LIDAR point clouds, all visible points from all sensors are fused in 3D and quantized into the voxel grid. More precisely, depth maps are used to map pixels from camera data into 3D. The voxel grid is of a fixed size, only considering 3D points within this defined range.

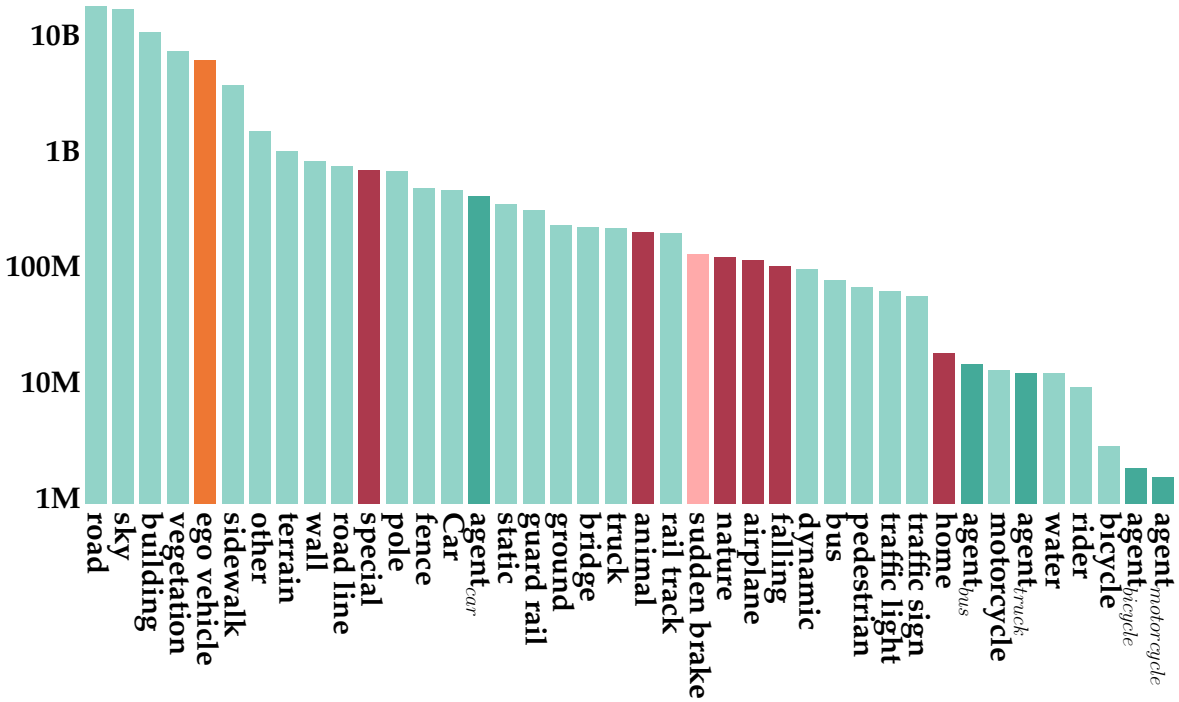


Figure 4.9.: **Number of pixels per class:** Light green represents standard classes, dark green vehicles in agent behavior mode, orange the ego vehicle, and red anomalies. Reprinted from [BOG 6].

Semantic classes are assigned to individual voxels based on the closest point to the voxel center.

The utilized CARLA simulation engine has inherent flaws, such as imperfect behavior agents [222]. Addressing these, minor data cleaning is performed to remove scenarios that contain collisions with pedestrians in the evaluation data and scenarios that show high deceleration values for other vehicles in the training data. This ensures that unlabeled anomalies are avoided in the evaluation data and that temporal anomalies are distinct from all behaviors seen during training.

Labels: The generated data provides 40 classes in the semantic masks, as shown in Figure 4.9. Labels are provided for standard tasks as used by Cityscapes and CARLA [101, 117]. Additionally, the ego vehicle is labeled if visible. Some vehicles in the scene might switch from autopilot to a behavior agent. As this driving behavior is not present in training data, additional labels are provided for all vehicle classes while controlled by a behavior agent. Finally, labels for the included content and temporal anomalies are provided. Labels for the super-classes can be found in the semantic masks, while fine-granular, individual anomaly labels are provided in metadata. Voxels are assigned an anomaly label when the closest point to their center is anomalous.

Dataset and Statistics

The presented methodology was used to generate a large-scale dataset[6]. A total of 1,117 scenarios are provided based on the *mono* sensor configuration, 76 for the *stereo* configuration, and 35 for the *multi* configuration. Data is provided for eight different areas, and for each, normality training data as well as evaluation data with content and temporal anomalies is provided, resulting in 24 different types of scenarios. For each of those 24, varying settings such as the weather, time of day, or spawned anomaly are employed. For content anomalies, 14.8% of all frames contain visible anomalies in camera data and 74.8% in LIDAR data. For temporal anomalies, 15.5% of all frames include anomalies equally visible in both sensor modalities. As shown in Figure 4.10, anomalies are well distributed over the visible space but with a focus on critical areas in front of the ego vehicle.

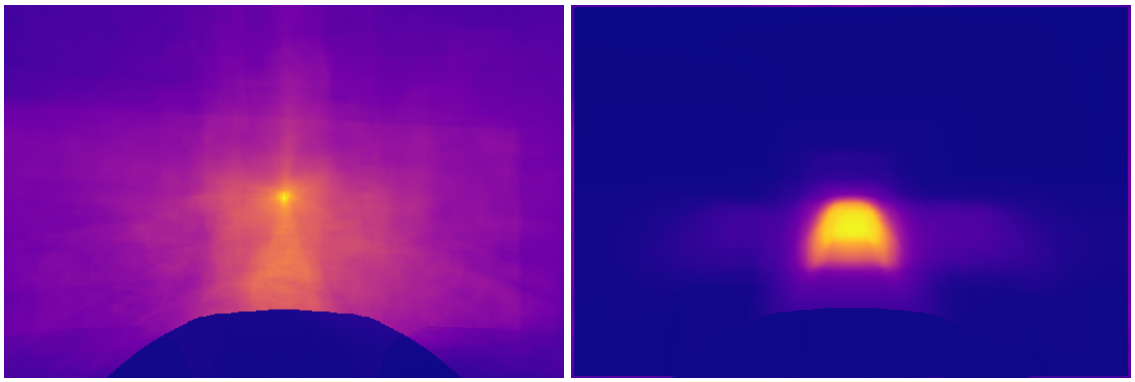


Figure 4.10.: **Spatial distribution of anomalies:** The distributions show content anomalies (left) and temporal anomalies (right). Reprinted from [BOG 6].

4.3.3. Evaluation

In addition to the generation of data, a full evaluation suite for camera and LIDAR-based anomaly detection methods for content anomalies is provided. Based on pixel- or pointwise anomaly scores, results are voxelized and compared against the voxelized anomaly ground truth. The evaluation takes place on a voxel grid of size $100 \times 100 \times 64$ m, where each voxel has a length of 0.5 m. Ablation studies on smaller voxel sizes have been performed, but no significant impact on the results could be found.

Anomaly Detection on the Content Layer

To get first insights into how current anomaly detection methods for *content* anomalies perform on the benchmark, two SotA methods have been evaluated,

[6]The dataset is hosted on Zenodo: https://zenodo.org/communities/anovox/

one based on camera data and one based on LIDAR data. For the selection, existing surveys and benchmark results were analyzed [BOG 13, 22][44, 75], resulting in 14 candidate methods [329, 252, 142, 153, 111, 348, 356, 4, 76, 312, 38, 316, 71, 112]. To emphasize the definition of normality based on training data alone, methods were neglected that require outlier exposure with anomalies during training. Finally, the best-performing methods with open-source implementations, based on existing benchmarks, were selected for evaluation. For camera data, the Rejected By All (RbA) method from Nayal et al. [316], and for LIDAR data, the Redundancy Classifier (REAL) framework by Cen et al. [71] were selected. While both methods allow for the usage of outlier exposure during training, no fine-tuning on anomalies was performed here. For RbA this means that the "Outlier Data Exposure" [316] was not used, and for REAL this means that only the "predictive distribution calibration" without "unknown object synthesis" [71] was used.

The *Rejected By All (RbA)* model proposed by Nayal et al. [316] uses camera data and is based on the Mask2Former model [92]. The authors proposed that object queries are specialized on single classes, such that anomalies can be detected when the input is rejected by all queries. RbA was trained on Cityscapes and evaluated on Segment Me If You Can anomaly and obstacle tracks, as well as on the Fishyscapes LaF track. The *Redundancy Classifier (REAL)* approach from Cen et al. [71] uses LIDAR data and is based on the Cylinder3D [515] framework. To assign high anomaly scores to novel classes, the authors utilize a calibration loss, where the second-highest prediction per point is assigned to the *unknown* class. This class is used for uncertainty prediction during inference while also performing closed-set semantic segmentation. In their experiments, without this calibration loss, anomalies are falsely classified as known classes with high probability. REAL was trained and evaluated on both SemanticKITTI [34] and nuScenes [58].

Training: For a fair comparison, both methods were trained on the same training dataset. For this, a small *normality* instantiation was created in the size of the CS dataset that consists of 2,975 frames with temporal scenarios. Contrary to the setting of REAL, where unknown objects are included in the training set but ignored for the loss computation, this training dataset does not include any anomalies. The standard training procedures and parameters as provided by the original authors [316, 71] were applied.

Evaluation: For the evaluation, a small instantiation with *content* anomalies was created. To evaluate both methods, their anomaly scores need to be mapped to the voxel space first. This is necessary, as point- and pixel-wise anomaly scores of LIDAR and camera data cannot be compared directly. While this is straightforward for LIDAR data, the anomaly scores from RbA are lifted into 3D using ground truth depth data. Due to this evaluation in voxel space, it must be noted that the class imbalance between normal data and anomalies is much larger than in the sensor space due to quantization effects. Thus, results cannot be compared to reported values from the SotA. However, evaluations on the sensor data were also performed directly, confirming higher scores.

Model	AUPRC ↑	AUROC↑	F1↑	PPV↑	FPR_{95}↓
REAL	0.14	43.30	0.0	0.0	100
$REAL_{+norm}$	0.04	43.53	0.0	0.0	100
$REAL_{big}$	0.17	44.7	0.0	0.0	100
$REAL_{medium}$	0.11	42.8	0.0	0.0	100
$REAL_{small}$	**0.21**	**55.1**	0.0	0.0	100
RbA	0.7	57.3	2.6	1.4	100
RbA_{+norm}	0.2	57.6	2.6	0.4	100
RbA_{big}	**2.3**	54.9	**3.7**	**2.7**	100
RbA_{medium}	0.6	**60.5**	0.0	0.0	100
RbA_{small}	0.01	53.2	0.0	0.0	100

Table 4.5.: **Evaluation of SotA anomaly detection methods:** Evaluation of the anomaly detection methods REAL (LIDAR-based) and RbA (RGB-based), with **best** and second-best results highlighted. Each model is evaluated in five settings. First, only the frames that consider anomalies are considered. In the $+norm$ setting, all frames, also those displaying normality, are considered. Finally, size-based subsets of the included anomalies are considered, evaluating the model performance for big, $medium$, and $small$ anomalies. Adapted from [BOG 6].

As shown in Table 4.5, both methods perform poorly across every metric and thus have significant issues detecting anomalies in the presented challenging setting. While stable training performance can be observed, and improved performance was shown when trained on larger datasets, the poor results are rather surprising. RbA is able to detect some anomalies successfully, as shown in Figure 4.4, where it masks the deer as anomalous, but struggles with most. REAL on the other hand, while generating well-performing closed-set predictions, is unable to generate any meaningful uncertainties in the absence of unknown objects in the training data, as exemplarily shown in Figure 4.11.

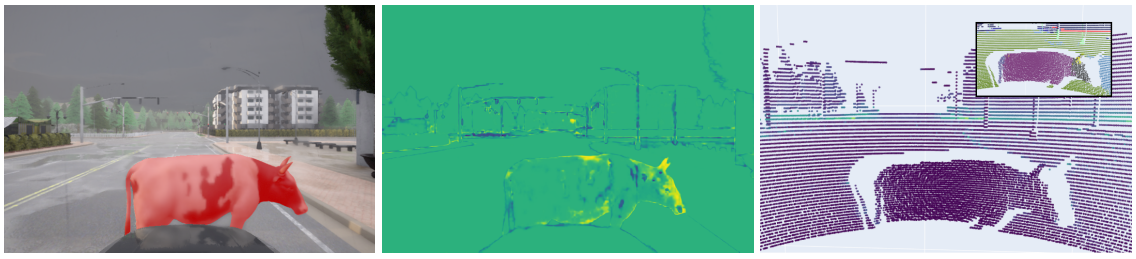


Figure 4.11.: **Exemplary SotA anomaly detection:** Scene with a cow as an object-level anomaly (left). The RbA (middle) anomaly detection method is only able to detect some border parts of the cow as anomalous, highlighted in yellow. Similarly, REAL (right) assigns the same low uncertainty to the cow as it does to the ground, as shown in violet. In the accompanying closed-set detection (top right), the cow is mostly classified as a car. Reprinted from [BOG 6].

This shows that the task of anomaly detection becomes much harder on a challenging benchmark, where lots of frames without visible anomalies and particularly small anomalies raise the bar additionally. This setting, combined with the induced class imbalance due to the quantization loss during voxelization, makes it more challenging to perform well on this benchmark compared to existing benchmarks.

Anomaly Detection on the Temporal Layer

In addition to the evaluation of detection methods for anomalies on the content layer, this benchmark also enables the evaluation of scenario-level anomalies. In this context, the Video Anomaly Detection (VAD) method HF^2-VAD$_{AD}$ [BOG 8] is evaluated with the introduced sudden brake maneuvers. This subsection briefly introduces the HF^2-VAD$_{AD}$ methodology and results. More details can be found in [BOG 8].

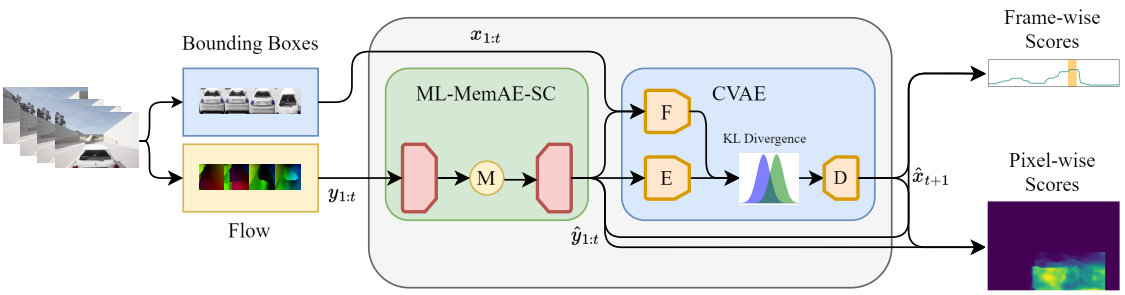


Figure 4.12.: **Anomaly detection method HF^2-VAD$_{AD}$:** Optical flows $y_{1:t}$ and bounding box patches $x_{1:t}$ for relevant objects are generated for each frame. Multi-Level Memory modules in an Autoencoder with Skip Connections (ML-MemAE-SC) reconstruct the optical flows $\hat{y}_{1:t}$ with memory modules M. A Conditional VAE predicts a future patch $\hat{x}_{t+1}$. Finally, image-wise and pixel-wise anomaly scores are generated. Reprinted from [BOG 8].

HF^2-VAD$_{AD}$ is an adaptation and extension of the hybrid framework for VAD method HF^2-VAD [275], which was developed and evaluated in the setting of surveillance videos from static cameras. The method was designed to classify entire frames as anomalous when atypical activity was detected. As shown in Figure 4.12, HF^2-VAD$_{AD}$ is adapted to autonomous driving by generating dense, pixel-wise anomaly scores for the whole frame. Bounding boxes are generated with an off-the-shelf object detection model [305]. Flow reconstructions and flow-guided frame predictions are learned from the perspective of an ego-vehicle rather than from the perspective of a static surveillance camera.

For the evaluation of HF^2-VAD$_{AD}$, anomalous scenarios with the introduced sudden braking maneuvers by lead vehicles are utilized, as shown in Figure 4.8.

Such scenarios are not present in the training data. As shown in Table 4.6, various experiments with varying conditions are performed. Here, the False Positive Rate at 95% True Positive Rate (FPR_{95}) metric provides insights into the false-positive rates if nearly all anomalies are correctly detected. Since detected bounding boxes do not fully match the existing ground truth, the FPR_{95} is evaluated based on the overlapping area. The Intersection over Union (IoU) metric shows the quality of generated bounding boxes. Comparing city and highway scenarios, a strong sensitivity of the model with respect to these environments is observed, but no clear trend emerges. For highway settings with bad weather, the model performs poorly. As the pixel-wise anomaly scores are only calculated within bounding boxes and all other pixels are set to 0, there is generally a low false positive rate. It can be observed that bounding boxes are best detected in highway settings with good weather. The IoU performance suffers especially from poor detections of distant objects. Since mostly irrelevant vehicles are missing in the detections, this can even lead to improved FPR_{95} values.

Domain	**Weather**	**FPR_{95} ↓**	**IoU ↑**
All	All	2.58	48.10
City	Sunshine	<u>2.48</u>	48.09
City	Rain	2.68	<u>51.30</u>
Highway	Sunshine	3.15	**60.71**
Highway	Rain	**1.34**	41.09

Table 4.6.: **Evaluation of HF^2-VAD$_{AD}$:** Evaluation under different scenarios, with **best** and <u>second-best</u> results highlighted. Adapted from [BOG 8].

Based on the presented benchmark results with HF^2-VAD$_{AD}$, it is demonstrated that HF^2-VAD, a framework originally developed for detecting anomalies in surveillance systems, can be effectively transferred to autonomous driving. This shows the utility of the presented benchmark for the evaluation of scenario-level anomalies.

4.4. Conclusion

This chapter presents two approaches to generate scenarios with anomalies, addressing all considered external anomaly levels as shown in Table 1.1. Answering **RQ2**, both methods convert existing theoretical anomaly levels from the literature, as introduced in Section 2.5, into datasets containing anomalies. First, a methodology to generate expert-defined scenarios is presented in Section 4.2. Here, human scenario designers are enabled to generate scenarios from all anomaly levels, including combinations of multiple levels. Next, a more scalable approach is presented in Section 4.3. Focusing on anomalies on the content and temporal layer, an automated data generation framework provides data from both RGB

camera LIDAR sensors. In addition, a benchmark suite based on a 3D voxel representation enables the comparison between anomaly detection methods that use different sensor modalities. This is demonstrated by evaluating and comparing a camera-based and a LIDAR-based SotA anomaly detection method.

The following Chapter 5 builds upon the data generated in this chapter. It presents a multimodal anomaly detection approach for anomalies on the content layer, which is trained based on a concept of normality, such as shown in Table 4.4. Subsequently, the presented benchmark suite from Section 4.3 is utilized for the evaluation of the approach.

4.4.1. Recent Advances

The field has continued to evolve since the development and publication of the works underlying this chapter [BOG 5, 6, 8]. Recently, multiple works have built upon the results presented here, confirming the relevance of improving benchmarks for anomaly detection methods in the context of autonomous driving. Extending upon to the presented generation concept for individual scenarios in Section 4.2, BridgeGen by Hao et al. [171] combines a knowledge-driven scenario description with data-driven optimization methods to compute concrete scenario parameters in order to generate critical scenarios with a focus on trajectories while also ensuring coverage of a broader ODD. Their 5-layer ODD includes road layout, traffic infrastructure, temporary manipulations, objects, and the environment. Due to introduced inter-layer constraints, traffic participants have to follow the provided road layout, e.g., pedestrians can only cross on crosswalks. Similar to the work presented in Section 4.2, their ontology allows for the inclusion of *misc* traffic participants and is designed for direct simulation in the CARLA environment. However, they demonstrate their approach only with a single intersection and two vehicles. CornerSim [105] by Daoud et al. generates synthetic data in CARLA [117] in a more scalable fashion, similar to the results presented in Section 4.3. They claim support for multiple anomaly levels as introduced by Breitenstein [51] but only provide a static dataset with 2,000 images and object-level anomalies [204].

Providing real-world anomaly data, multiple works draw inspiration from the simulated data presented in this chapter. The OpenAD Benchmark for 3D Object Detection [470] allows for both 2D and 3D anomaly detection. The authors acknowledge the benchmark presented in Section 4.3 as the only simulated benchmark that supports both 2D and 3D anomaly detection. Similar to LIDAR-CODA [BOG 20][253], a dataset used for the evaluation in Chapter 7, OpenAD includes labeled anomalies in both RGB and LIDAR data. The authors leverage an anomaly detection approach to annotate anomalous objects in five existing datasets [457, 146, 58, 294, 399]. The dataset includes over 200 classes of both uncommon objects and common ones in atypical variations, e.g., cars with open doors. They provide labels that loosely indicate whether an object class is included in the training dataset. Also addressing 3D anomaly detection, Nekrasov

et al. released the real-world Spotting the Unexpected (STU) dataset with labeled anomalies in LIDAR data [317] and synchronized raw camera data. Similar to the work presented here, the authors put an emphasis on clearly separating ID training data from OOD evaluation data with anomalies. This is achieved through an extra class *unlabeled* which includes classes that are often included in training data but without labels, such as parking meters. The authors provide both instance and semantic labels for temporal scenarios. Their evaluation confirms the results presented here, with all evaluated anomaly detection approaches struggling due to the much harder task compared to saturated benchmarks such as FS or SMIYC.

Further works also aim to explicitly address the saturation of anomaly benchmarks on RGB data, but do not introduce additional sensor modalities. Nekrasov et al. released the Out-of-Distribution Instance Segmentation (OoDIS) benchmark [319] as an extension of the Fishyscapes and Segment Me If You Can benchmarks [44, 75], providing labeled anomaly instances. This changes the task from semantic anomaly segmentation to a more challenging instance anomaly segmentation. Laskar et al. presented the Semantic Segmentation in the Presence of Unknowns (ISSU) benchmark [244], which is a real-world benchmark following several ideas presented in Section 4.3, such as providing labels of both known and unknown classes and generating temporal evaluation data. The Challenge Of Out-Of-Label (COOOL) benchmark [9] consists of labeled dashcam videos with objects and roadway hazards. It does not provide training data, but only labeled sequences for evaluation. The focus is on the prediction of hazards by known or unknown object classes.

Similar to the benchmarks in autonomous driving, current benchmarks in industrial settings are saturated as well [175]. In line with the design choices of the benchmark presented in this chapter, new large-scale benchmarks [175, 237] with high-resolution images and a higher variance of normal data have increased the benchmark difficulty significantly.

For the evaluation of Vision Language Models (VLMs) for anomaly detection, Chen et al. presented CODA-LM [83], an extension of the CODA dataset [253]. They annotated 9,768 driving scenarios with question-answer pairs. This way, VLMs can be tested on the tasks general perception, regional perception, and driving suggestions, where they are tasked to describe the influence of other road entities on driving behavior and suggest next steps for the ego vehicle. These recent advances highlight the relevance and impact of the data generation methods presented in this chapter.

The revolution will not be supervised.

Alyosha Efros, 2019 [121]

5. External Anomaly Detection

Multiple supervised student theses have contributed to this chapter [STU 11, 10, 5]. Parts of this chapter have previously appeared in the following publications:

- D. Bogdoll et al. *MUVO: A Multimodal Generative World Model for Autonomous Driving with Geometric Representations*. In IEEE Intelligent Vehicles Symposium (IV), 2025 [BOG 23]
- D. Bogdoll et al. *UMAD: Unsupervised Mask-Level Anomaly Detection for Autonomous Driving*. In British Machine Vision Conference (BMVC) Workshop, 2024 [BOG 14]

5.1. Introduction

As shown in Chapter 3, existing anomaly detection methods often neglect multimodal sensor setups, which are typical in autonomous driving, and focus on RGB data alone. In addition, they often require underlying semantic segmentation models [92] and outlier exposure during training. This makes it challenging to utilize large amounts of unlabeled data. Addressing **RQ3**, this chapter presents a label-free anomaly detection approach, leveraging unlabeled data from multiple sensor modalities for the detection of object-level anomalies. Section 5.2 presents a world model representing a defined notion of normality, as shown in Chapter 4. The model is trained in a self-supervised fashion, requiring no labeled data at all. This model is subsequently utilized in Section 5.3 to detect object-level anomalies. In both cases, both RGB and LIDAR data are leveraged. To further improve the detections, a self-supervised segmentation model is used to refine instance-wise masks. Finally, the presented anomaly detection approach is evaluated on the benchmark previously introduced in Section 4.3. The method outperforms the most relevant label-free SotA anomaly detection method and sets a new baseline.

5.2. Self-Supervised Normality Learning

World models are generative models that embed observations into latent states, predict future states conditioned on actions, and decode these latent predictions into the observation space [245][BOG 2]. They can be trained in a self-supervised way and are well-suited to represent normality, as shown in Table 4.4, due to their

temporal nature and the inclusion of planned actions. In Machine Learning, recent world models like Cosmos [5] or Genie [54, 340] have demonstrated the capability to take sequences of high-resolution input images, conditioned on instructions or actions, and generate future images, predicting possible future scenarios.

In Autonomous Driving, the majority of world models focus on camera-based inputs [513, 285, 349, 143, 482, 442, 192, 124], only a few work with LIDAR data [503, 85, 517]. These works neglect typical sensor setups of autonomous vehicles, as they only consider a single sensor modality. Only two recent works leverage both camera and LIDAR data [506, 468]. However, they rely on BEV features as part of their sensor fusion strategy, which is an acknowledged bottleneck due to missing height information [506]. Finally, world models that predict future 3D occupancies have been proposed, which are highly actionable but rely on visual inputs only [485] or operate in the occupancy space alone [436]. While the progress made in world models is immense, it is still unclear how much they can benefit from multimodal sensor setups, such as those typically used in autonomous driving. The general benefit of leveraging both RGB and LIDAR data is well-established [505, 276, 94], but previous works did not evaluate the impact on future predictions by a world model.

This chapter presents the first multimodal world model leveraging both camera and LIDAR data without requiring limiting BEV representations. For the design of the final architecture, an extensive set of experiments is performed to determine the influence of design choices with respect to sensor fusion strategies, latent space dimensionality, and additional 3D occupancy prediction. A simple world model architecture is chosen to perform the set of experiments, also comparing against multiple BEV-based sensor fusion baselines. The chapter introduces the model architecture in Section 5.2.2, describing the encoder, fusion, transition, and decoder components of the world model in detail. Next, it provides an extensive overview of experiments in Section 5.2.3, highlighting the impact of different fusion strategies, latent space dimensionality, and additionally predicting 3D occupancy. The best-performing model, determined by the extensive evaluation, is subsequently used in Section 5.3 for the detection of object-level anomalies.

5.2.1. Related Work

The method presented in this chapter is at the intersection of world modeling, sensor fusion, and 3D occupancy prediction. Leveraging advances from all three fields, an extensive set of experiments evaluates the impact of different sensor fusion strategies and the inclusion of 3D occupancy prediction on the quality of future predictions by a world model. This overview of related works facilitates a better understanding of the subsequently presented method.

World Models

World models are generative models that embed observations into latent states, predict future states conditioned on actions, and decode these latent predictions into the observation space [245][BOG 2]. Many such world models rely on labels, privileged information[1], or expert-designed state spaces, limiting their ability to scale. A typical use case is the prediction of BEV semantic labels based on supervised training [191, 434, 144]. DriveDreamer [441] conditions real-world RGB images on High Definition (HD) maps and labeled 3D bounding boxes. Based on a diffusion model [364], future frames and actions are jointly predicted. The style of predictions is guided by Contrastive Language-Image Pre-Training (CLIP) [354] embeddings, using annotated scenes during training. Contrary to these approaches, the method presented in this section does not require labeled training data.

There also exist self-supervised world models. DreamerV3 is capable of predicting future observations in Minecraft [165, 166]. DriveGAN [223] was trained on real-world data and acts as an action-conditioned neural simulator. DayDreamer [464, 164] learns robotic tasks from real-world visual inputs. A world model from Tesla [124], trained on proprietary multi-camera RGB data, demonstrated the prediction of future observations and semantic or spatial data based on supervised fine-tuning. Classically, world models use Recurrent Neural Networks (RNNs) for the prediction of future states [163, 162, 164, 165], which shows poor scaling properties. Inspired by the progress of Large Language Models (LLMs), more recent methods approach the task through Transformer-based sequence modeling [265, 465, 99, 192, 143], which shows better scaling properties but is computationally demanding. Among those, Generative AI for Autonomy (GAIA)-1 [192] uses vector quantization [425] to tokenize the data and perform autoregressive prediction. It was trained on proprietary real-world camera data and can be conditioned with both actions and textual inputs. Based on a video diffusion decoder, it achieves temporally consistent, high-resolution predictions. Similarly, Vista [143] further increases the image resolution used by previous models. Compared to such recent self-supervised world models, the methodology presented in this section is computationally efficient and does not require hundreds of Graphics Processing Units (GPUs) for training.

In spatial domains, several world models exist based on LIDAR data [503, 85, 517] or 3D occupancy grids [485, 436]. The presented method differs from those approaches by leveraging multimodal data. Most similar to the presented experiment setup, BEVWorld [506] and HoloDrive [468] leverage both camera and LIDAR data. The authors of BEVWorld propose a multi-model encoder that generates a unified BEV representation. Upsampled voxel features are used to predict camera and LIDAR data. Differently, HoloDrive has separate models for image and LIDAR generation and introduces 2D-to-3D and 3D-to-2D structures to improve a joint generation leveraging BEV representations. In both cases, BEV features

[1] Privileged information is additional data that is only available during the training phase.

lack height information and are thus a bottleneck. The approach presented in this chapter does not require BEV features.

Sensor Fusion

In autonomous driving, sensor fusion approaches typically use camera and LIDAR sensors. Recently, a shift towards Transformer-based [426, 196] architectures can be observed. Many works [260, 208, 293, 276, 498] perform BEV camera-LIDAR fusion [346]. Others improve upon this by utilizing 3D voxel features [256, 508]. Sparse representations [472, 61, 255], modality interactions, and intermediate fusions are becoming more common [266, 94, 487, 478]. While many works deal with robustness against LIDAR failures, dealing with inferior image conditions is underrepresented [23]. Other works focus on interpretability [387], real-time performance [337], event streams [433], modality agnosticity [90], or auxiliary supervision [353]. This chapter presents an extensive set of sensor fusion experiments to better understand the influence of sensor fusion strategies on the quality of future predictions of a world model.

3D Occupancy Forecasting

Forecasting is a task similar to future predictions produced by a world model, but purely based on past data without conditioning future frames, i.e., without planned actions of an ego vehicle. In the context of predicting 3D voxels, the OpenOcc benchmark [393] was the first benchmark to include voxel-wise flow information, similar to OpenScene [100]. An occupancy network by Tesla predicts motion flow vectors for voxels [125]. Khurana et al. combine LIDAR data with motion sensors to predict future 3D occupancy [220]. Liu et al. introduced the task of occupancy completion and forecasting [273], whereas others utilize input images to forecast 3D occupancy [84, 501, 505, 486]. The approach presented in this chapter evaluates the impact of also predicting 3D occupancy as an additional head of the world model.

5.2.2. Method

A world model is able to represent normality purely based on training data. However, in the context of autonomous driving, no world model exists that fuses the common sensor modalities RGB camera and LIDAR without relying on limiting BEV representations. Here, a variety of sensor fusion strategies are evaluated in order to determine the influence of different strategies on the prediction quality of a world model. In addition, the influence of further predicting 3D occupancy is examined. For this evaluation, the experiment setup[2] follows the fundamental ar-

[2]The code is available on GitHub: https://github.com/fzi-forschungszentrum-informatik/muvo

chitecture of Model-based Imitation Learning (MILE) [191], which is much reduced in complexity compared to other approaches, such as GAIA-1 or Vista [193, 143].

As shown in Figure 5.1, changes to the architecture of MILE are introduced to allow for sensor fusion of a typical sensor setup of autonomous vehicles, comprising stereo cameras and LIDAR [145, 445], and predict raw sensor data rather than low-resolution BEV masks based on camera data. An additional head to predict 3D occupancy is introduced to analyze the effects of a spatial loss in a sensor-independent space. First, RGB camera data and LIDAR point clouds are processed, encoded, and fused. Second, the latent representations of the sensor data are fed to a transition model to derive a probabilistic model of the current state, followed by sampling, while concurrently predicting the probabilistic model of future states and sampling from it. Lastly, both current and future states are decoded from the probabilistic models, predicting raw RGB images, point clouds, and 3D occupancy grids. Qualitative predictions of the world model can be found in Figure 5.2. In the following, the different components of the world model and the experiment setup are introduced in more detail.

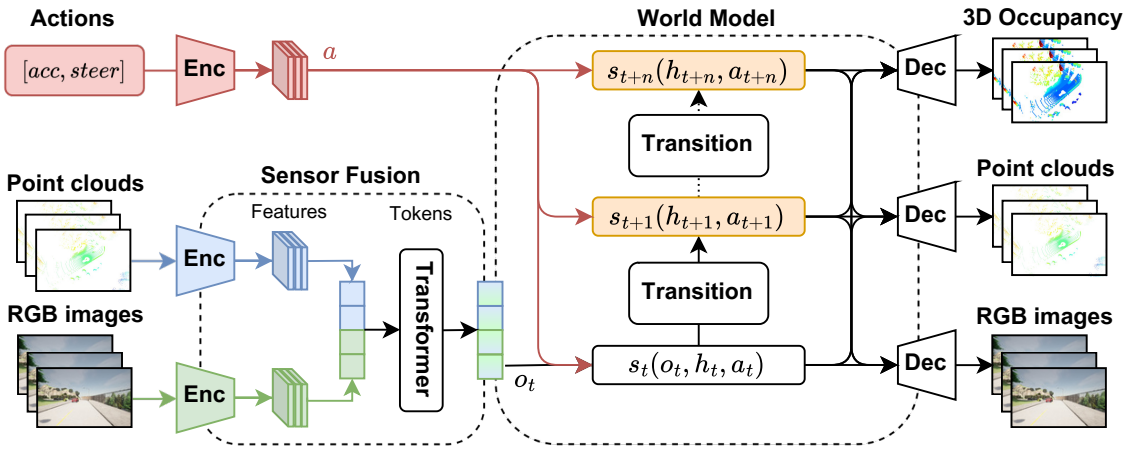


Figure 5.1.: **Self-supervised world model representing normality:** Raw camera images and LIDAR point clouds are processed and fused. The resulting latent representations are fed into a transition model. Conditioned on actions, future states are predicted. Finally, future states are decoded into 3D occupancy grids, raw point clouds, and raw images. Reprinted from [BOG 23].

Observation Encoder

As shown in Chapter 4, RGB images from a front camera and point clouds from a top-mounted LIDAR are used as input data, based on the CARLA simulation engine. The 3D point cloud, comprising up to $60,000$ points, is projected into a 2D cylindrical range view projection for a pixel-based representation [249, 128, 230]. For images $\mathcal{I} \in \mathbb{R}^{3 \times H_i \times W_i}$, the approach follows Hu et al. [191] and uses an input size of 600×960 pixels. For images $\mathcal{I}$ and point clouds $\mathcal{R} \in \mathbb{R}^{4 \times H_r \times W_r}$ in range

view representation, a pre-trained backbone is utilized for feature extraction. Feature maps are derived from different model layers similar to Hu et al. [191] and fused, culminating in image features $\mathcal{F}_c \in \mathbb{R}^{C \times H_c \times W_c}$ and point cloud features $\mathcal{F}_l \in \mathbb{R}^{C \times H_l \times W_l}$.

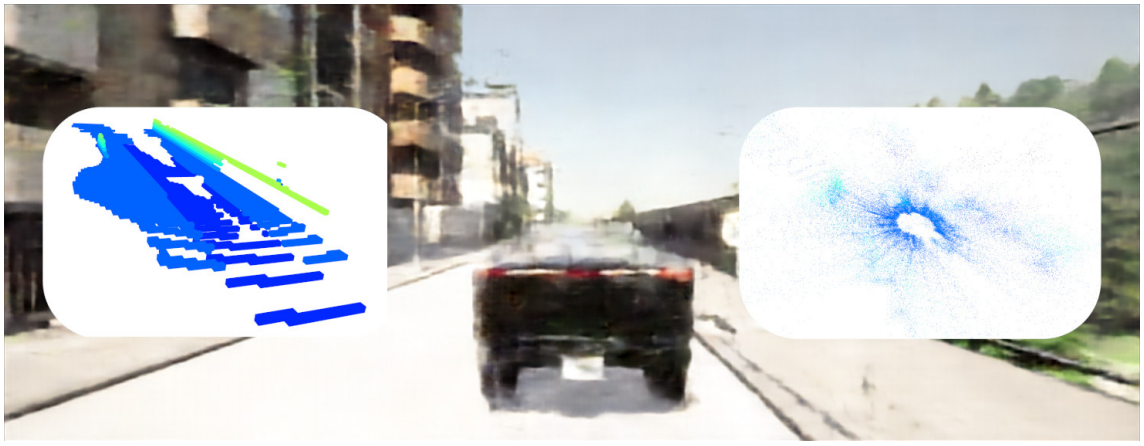


Figure 5.2.: **Exemplary world model predictions:** Qualitative output of a sensor fusion experiment with occupancy prediction activated. The predictions shown for camera and LIDAR sensors and 3D occupancy are based on past camera and LIDAR inputs. Reprinted from [BOG 23].

Multimodal Fusion

As shown by previous works [115, 94, 387], the self-attention mechanism of a Transformer [426] is employed to fuse features of different sensors. It takes a sequence of tokens as input, where each token is a D_t-dimensional feature vector, so the input sequence is $\mathbf{t}_{in} \in \mathbb{R}^{D_t \times N_t}$, with N_t representing the number of tokens in the sequence. The flattened H and W dimensions of the features $\mathcal{F}$ obtained from the encoder described in the subsection of the observation encoder result in tokens $\mathbf{f} \in \mathbb{R}^{D \times HW}$. Subsequently, 2D sinusoidal positional embeddings [426, 94] $\mathbf{e} \in \mathbb{R}^{D \times HW}$ are incorporated into each token to introduce spatial inductive biases. The learnable sensor embeddings $\mathbf{s} \in \mathbb{R}^{D \times N_s}$ are added, introducing a sensor category, where N_s is the number of sensors. The resulting tokens $\mathbf{t} \in \mathbb{R}^{D \times HW}$ are obtained, with each token $\mathbf{t}_i(x, y) = \mathbf{f}_i(x, y) + \mathbf{e}_i(x, y) + \mathbf{s}_i$, where i indicates the i-th sensor, and (x, y) denotes the coordinate index of that token within the sensor feature. These tokens from all sensors are concatenated and fed into a Transformer encoder comprising k layers, each consisting of multi-head self-attention, Multilayer Perceptrons (MLPs), and layer normalizations, resulting in new tokens $\mathbf{t}_{new} \in \mathbb{R}^{D \times (\sum_i H_i W_i)}$.

Transition Model

The input consists of fused observation features $\mathbf{o}_{0:t}$ and encoded actions $\mathbf{a}_{0:t} \in \mathbb{R}^{T \times D_a}$, based on a simple MLP, assuming access to a policy or motion planner. The output includes stochastic hidden states $\mathbf{s}_{0:t} \in \mathbb{R}^{T \times D_s}$ and deterministic historical states $\mathbf{h}_{0:t} \in \mathbb{R}^{T \times D_h}$, predictions for future states $\mathbf{s}_{t:t+n}$, and $\mathbf{h}_{t:t+n}$, T represents the number of frames, also referred to as the sequence length, and D_a, D_s, D_h are the dimensions of each vector respectively. The deterministic historical variable $\mathbf{h}_{t+1} = f_\theta(\mathbf{h}_t, \mathbf{s}_t)$ is modeled by a Gated Recurrent Unit (GRU) [95] f_θ, enabling the model to remember past states. The posterior hidden state probability distribution is given by $q(\mathbf{s}_t \mid \mathbf{o}_{\leq t}, \mathbf{a}_{<t}) \sim \mathcal{N}_\phi(\mathbf{o}_t, \mathbf{h}_t, \mathbf{a}_t)$, while the prior hidden state probability distribution, without the input of observed feature $\mathbf{o}_t$, is given by $p(\mathbf{s}_t \mid \mathbf{h}_t, \mathbf{a}_{t-1}) \sim \mathcal{N}_\theta(\mathbf{h}_t, \mathbf{a}_{t-1})$. Here, $\mathcal{N}_\phi$ and $\mathcal{N}_\theta$ are probability models modeled by a MLP. Given observations, $\mathbf{s}_t$ is sampled from the posterior distribution q. In the absence of observations, i.e., during prediction, $\hat{\mathbf{s}}_\mathbf{t}$ is sampled from the prior distribution p.

The output tokens $\mathbf{t}_{new}$ from the sensor fusion stage come in the form of a two-dimensional latent state and are utilized as the encoded observations $\mathbf{o}_t$ for the transition model. Compared to 1D states, the stochastic hidden states $\mathbf{s}_t$, and deterministic historical states $\mathbf{h}_t$ are set to shape $C_h \times (\sum_j T_j)$, where T_j is the number of tokens of each output modality. For f_θ, all fully connected layers are replaced with convolutional layers to utilize two-dimensional states. The probability models $\mathcal{N}_\phi$ and $\mathcal{N}_\theta$ are modeled by a Transformer decoder. Learnable embeddings, which have the same shape as the stochastic hidden states $\mathbf{s}_t$, are used as queries, where $\mathbf{h}_t, \mathbf{a}_t, (\mathbf{o}_t)$ are concatenated as key-value pairs. Then, the state information in these is queried through the attention mechanism to obtain stochastic hidden state tokens.

Multimodal Decoder

The world model decodes latent representations into camera and LIDAR data and introduces an optional head for 3D occupancy. The input is a latent dynamic state $(\mathbf{s}_t, \mathbf{h}_t)$ with shape $C \times \sum_j T_j$ which is provided by the transition model, where T_j is the number of tokens of each output modality. Those tokens are divided based on modalities, and each modality's tokens $C \times T_j$ are reshaped to fit the output shape. The occupancy decoder is of shape $C \times X \times Y \times Z$. For camera and LIDAR data, first, the input is reshaped to $C \times H_0 \times W_0$, where H_0 and W_0 are determined by the final output resolution $H \times W$. Subsequently, upscaling with convolutional networks is performed similarly to prior world models [162, 160] to produce a feature map of size $C_n \times H \times W$. For camera and LIDAR, two-dimensional convolutions are utilized, while three-dimensional convolutions are employed for voxels.

5.2.3. Evaluation

For the evaluation, the utilized training setup is presented first, followed by the evaluation of sensor fusion strategies. First, the influence of differently sized latent spaces is examined and shown in Figure 5.3. Next, the impact of different fusion strategies is evaluated, as shown in Figure 5.4. Finally, the effects of the optional 3D occupancy prediction are examined. Figure 5.5 shows the relation between camera-LIDAR-based pre-training and 3D occupancy prediction, and Figure 5.6 shows the reverse impact of predicting occupancy on the quality of sensor predictions.

Training Setup

Training Losses: For each modality, downsampling is performed multiple times with ratios of 1, 2, and 4. With this multi-scale approach, losses are computed at different resolutions. For images, the output RGB data aligns with the size of the input, and the common L1 loss $\mathcal{L}^{\text{img}}$ is utilized for the minimization of the absolute discrepancies between target and prediction. For point clouds, range view images of dimensions $4 \times H_r \times W_r$ are generated, which can be converted into $N \times 3$ point cloud data. The target is the range view image transformed from the ground truth, where an L2 loss $\mathcal{L}^{\text{p, xyz}}$ is applied to minimize the Euler distance and an L1 loss $\mathcal{L}^{\text{p, r}}$ is based on range r. For 3D occupancy, voxel grids of size $192 \times 192 \times 64$ with 0.5 m voxels contain the binary occupancy. The target is obtained by voxelizing fused depth maps from depth cameras and point clouds from LIDAR. The loss for voxel grids uses a Scene-Class Affinity Loss (SCAL) [62] $\mathcal{L}^{\text{V, scal}}$. The total loss is given by Equation 5.1:

$$\mathcal{L} = \sum \lambda_i \Big(\lambda_{\text{img}} \mathcal{L}_i^{\text{img}} + \lambda_{\text{pcd}} (\mathcal{L}_i^{\text{pxyz}} + \mathcal{L}_i^{\text{pr}} + \mathcal{L}_i^{\text{pcd}}) + \lambda_{\text{V}} \mathcal{L}_i^{\text{V, scal}} \Big) \tag{5.1}$$

Datasets: The training dataset $\mathcal{D}_{train}$ was collected in the CARLA simulation environment [117]. The data collection encompasses four towns (Town01, Town03, Town04, Town06) and four weather conditions (Clear Noon, Wet Noon, Hard Rain Noon, Clear Sunset), gathered at a frequency of 10 Frames Per Second (FPS). For each town, 25 runs were executed, each lasting 300 seconds, with randomly selected weather conditions, amounting to 300,000 frames of data. The following sensor data were collected: RGB image $\mathcal{I} \in \mathbb{R}^{3\times600\times960}$, depth map $\mathcal{I}_D \in \mathbb{R}^{1\times600\times960}$, e.g., derived from stereo cameras, point cloud $\mathcal{P} \in \mathbb{R}^{\leq 60{,}000\times3}$ obtained from a LIDAR with 64 vertical channels, route map $route \in \mathbb{R}^{1\times64\times64}$ as the planned route in BEV space, speed $\mathbf{v} \in \mathbb{R}$, and actions $\mathbf{a} \in \mathbb{R}^2$ in the form of acceleration and steering angle.

The same setup is adapted for two distinct validation sets. For each town, five 300-second-long driving sessions were executed with the following settings:

$\mathcal{D}_{val}^{RL}$: This set uses the same cities and weather conditions as the training set. However, the driving routes are randomized. The goal is to evaluate the effectiveness of the model in Representation Learning in familiar environments.

$\mathcal{D}_{val}^{DS}$: The same cities as in the training set are maintained, but different weather conditions are introduced. The driving routes are also randomized to evaluate the model's performance under Domain Shifts.

Training Parameters: Data was sampled at intervals of 0.2 seconds, creating sequences of length twelve to serve as training inputs. All twelve frames were treated as known data. In the experiments containing voxel reconstructions, the length of sequences was reduced to six to speed up the training. It was trained with a batch size of 16 and the AdamW optimizer [279] with a learning rate of 10^{-4} and a weight decay of 0.01. For validation, six resp. four frames were used as given observations, while six resp. two served as ground truth. For all experiments, a pre-trained Residual Network (ResNet)18 [174] was used as the baseline backbone.

Sensor Fusion Strategies

Several prior multimodal world models rely on naive fusion approaches [464, 392, 144]. Here, such approaches are compared to a Transformer-based architecture. To evaluate the effect of different sensor fusion strategies, several metrics are used based on the sensor modality: For assessing the quality of image predictions, the PSNR is used to assess average differences. The Chamfer Distance is used to evaluate the accuracy of point cloud predictions. For the predictions of 3D occupancy grids, the metrics IoU, Precision, and Recall are used. Here, IoU^{+} represents occupied voxels and IoU^{-} empty ones.

Decoders and fusion methods examined in this chapter are presented first. Subsequently, an overview and a comparison of all analyzed combinations are provided, as shown in Figure 5.4. The following naming scheme A-B-C is applied: **A** represents the method of processing point clouds: *PP* stands for the use of PointPillars as the encoder; *RV* indicates the conversion of point clouds into range view. **B** denotes the approach of image processing: *BEV* implies mapping to BEV followed by feature extraction with a backbone; *WOB* denotes that no BEV mapping is performed. **C** describes the method of sensor fusion: *AVG* stands for the averaging of 1D features; *FC* means that concatenation followed by a fully connected layer is performed; *TR* denotes that the Transformer-based multi-head self-attention mechanism was used, as described in Section 5.2.2.

Encoders: For image features $\mathcal{F}_c$, the standard encoder introduced in the encoder subsection is compared to approaches that map features to BEV space [346, 506, 191, 260]. Here, features are first elevated into a 3D space. Then, these 3D feature voxels are aggregated into the BEV space, leading to image features $\mathcal{F}_b \in \mathbb{R}^{C \times H_b \times W_b}$. Multiple representations are evaluated for point clouds. A range view-based representation is compared with PointPillars [243] as an encoder, where point

clouds are segmented into discrete pillars along the X and Y axes, followed by data processing and feature extraction, resulting in a 2D BEV pseudo-image.

Latent Space: In prior works, such as MILE, the latent space is commonly modeled through one-dimensional vectors [191, 162], which may limit the model performance by introducing a representational bottleneck. Experiments with both a 1D and a 2D latent space are performed. In addition, an additional perceptual loss [211] and a Vision Transformer (ViT) backbone [301] are examined.

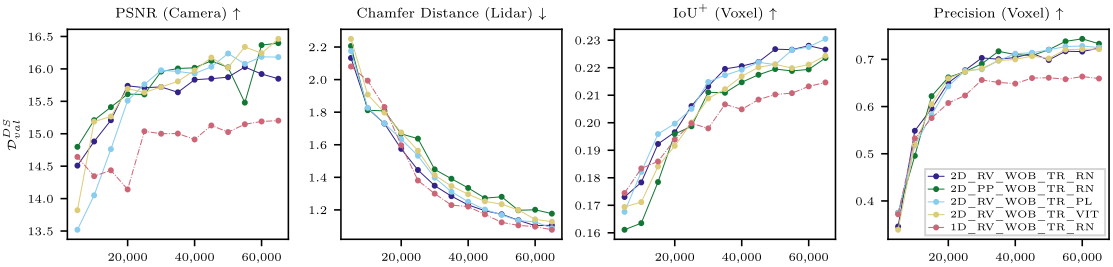


Figure 5.3.: **Two-dimensional latent space:** Comparison of a 1D baseline (red) with a set of 2D latent spaces, where the influence of a Vision Transformer backbone and an additional perceptual loss term (PL) are also examined. For the backbone, ResNet18 (RN) and MobileViT-V2 (VIT) are evaluated. Reprinted from [BOG 23].

Figure 5.3 shows four evaluation graphs depicting the prediction performance for camera, LIDAR, and 3D occupancy on $\mathcal{D}_{val}^{DS}$. The 2D latent state significantly benefits predictions for camera images and spatial voxel occupancies, while LIDAR predictions do not see any benefit. This might be since camera data is much more complex with respect to semantic information than LIDAR data. Compared to the baseline 2D model (dark blue), there is no strong effect of utilizing a perceptual loss, as it produces visually poorer reconstructions and does not show any significant advantages. Using the ViT as an encoder does provide advantages for the prediction of camera images, but shows no effect on other metrics. This shows that the 2D latent space itself provides the largest boost in performance, while other changes have little effect.

Fusion Methods: A Transformer-based sensor fusion approach, as described in the subsection on fusion strategies, is compared to naive combinations of encoded 1D features from each sensor modality. Experiments are performed for both averaging features and concatenating them, followed by a fully connected layer. To generate such latent states, the output tokens are reshaped into their original shape after the encoding, namely $\mathcal{F}_c^{new} \in \mathbb{R}^{C \times H_c \times W_c}$ and $\mathcal{F}_L^{new} \in \mathbb{R}^{C \times H_L \times W_L}$. Each feature is then downsampled by convolutional layers, followed by pooling layers to get one-dimensional features $\mathbf{f}_d \in \mathbb{R}^D$, which are subsequently concatenated and then passed through fully connected layers to reduce its dimensionality, producing the vector $\mathbf{o}_t \in \mathbb{R}^D$.

The prediction performance of eight encoder-fusion combinations is evaluated, as visible in Figure 5.4. In the following, the effects on image predictions are discussed first, followed by the effects on point cloud predictions.

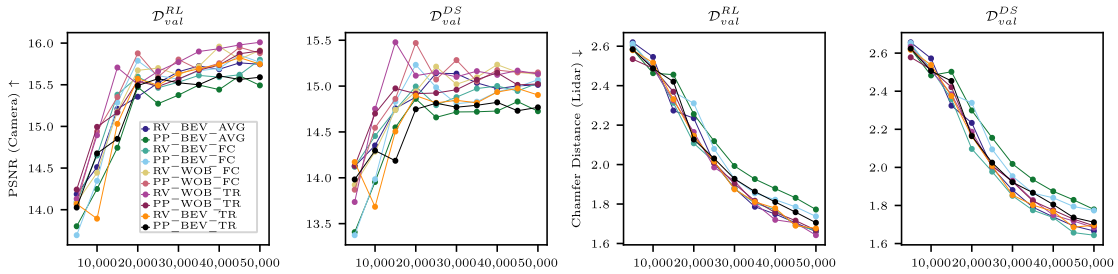


Figure 5.4.: **Sensor Fusion:** With $\mathcal{D}_{val}^{RL}$ representation learning is evaluated, while $\mathcal{D}_{val}^{DS}$ examines robustness. Feature averaging (AVG) [81], feature concatenation (FC) [464], and a Transformer-based architecture (TR) [94] are examined. For LIDAR encodings, PointPillars (PP) [243] and a range view (RR) [249] representation followed by a ResNet [174] are evaluated. For camera data, direct encoding without BEV (WOB) and a BEV mapping are evaluated [346]. Reprinted from [BOG 23].

Image Prediction: The impact of the different experiments on the quality of camera predictions is shown in the first two graphs of Figure 5.4. It shows a drop in performance for all networks in the $\mathcal{D}_{val}^{DS}$ dataset compared to $\mathcal{D}_{val}^{RL}$, but the relative performance of different networks remains consistent across both datasets. Generally, the Transformer-based architecture *RV-WOB-TR* performs on par or better compared to the other combinations, and range view-based LIDAR encodings show clear advantages over PointPillars. Methods with an additional BEV mapping of image features perform worse, and combinations with PointPillars suffer especially. It can be seen that the effectiveness of introducing a Transformer-based architecture depends on the encoder used. It outperforms other approaches when combined with a ResNet-18 for feature extraction. In contrast, when combined with PP and BEV, its performance is lower than concatenating but higher than averaging.

Point Cloud Prediction. The impact of the different experiments on the quality of camera predictions is shown in the last two graphs of Figure 5.4. Examining the Chamfer Distance plots, where lower values mean better performance, reveals no significant disparity in performance between the two validation datasets. For $\mathcal{D}_{val}^{RL}$, the Transformer-based architecture *RV-WOB-TR* performs on par or better compared to the other combinations. However, on $\mathcal{D}_{val}^{DS}$, its performance drops. As before, range view-based methods demonstrate superiority over PointPillars. Utilizing BEV features shows no clear disadvantage for this task. Transformer-based architectures generally outperform other fusion techniques.

The experiments determine the Transformer-based architectures with a 2D latent space and range-view representations for point clouds as an optimal fusion strategy, while performance benefits are more pronounced for camera predictions.

3D Occupancy Prediction

In addition to analyzing fusion strategies, the effects of additionally predicting more actionable 3D occupancies are examined. The experiments shown in Figure 5.5 analyze whether occupancy predictions can benefit from a pre-trained model that was trained by only predicting camera and LIDAR data. Subsequently, it is analyzed whether occupancy prediction improves the prediction of camera and LIDAR data, as shown in Figure 5.6.

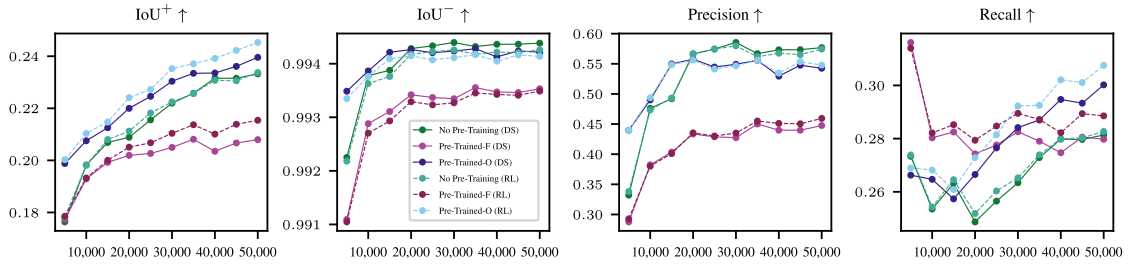


Figure 5.5.: **Pre-Training:** Influence of camera-LIDAR pre-training for 50,000 steps on 3D occupancy prediction. Evaluation on both $\mathcal{D}_{val}^{RL}$ and $\mathcal{D}_{val}^{DS}$. The green lines show a benchmark without pre-training. Violet lines show frozen weights of the pre-trained model, and weights remained open for the blue lines. Reprinted from [BOG 23].

3D Occupancy Prediction: Experiments in three scenarios are performed, as shown in Figure 5.5. As the effect of encoded knowledge of predicting camera and LIDAR data on 3D occupancy is examined, first, a model is trained as a pre-trained starting point that predicts camera and LIDAR data alone for 50,000 steps. For the first scenario, a pre-trained model is employed, but all of its weights are frozen (F) so that only the weights of the voxel decoder are trained. This approach allows for assessing the impact of fine-tuning only the voxel-specific aspects of the model while keeping the rest of the network, in particular all encoders, constant to evaluate if any information about a discrete geometry of the world is already encoded based on camera and LIDAR data. For the second scenario, the pre-trained weights were used as a starting point, but the entire network was open (O) for weight updates during training. Here, it is analyzed how the pre-trained weights influence the learning process when the whole network adapts and evolves during training. For the third scenario, no pre-training is utilized, and the network is trained from scratch.

In Figure 5.5, it can be observed that the model trained from scratch, without pre-training, exhibits a similar performance on both validation datasets across all four metrics, while the other two models using pre-trained weights generally performed better on $\mathcal{D}_{val}^{RL}$ than on $\mathcal{D}_{val}^{DS}$ across three metrics, excluding IoU^{-}. Interestingly, for IoU^{-}, an opposite behavior can be observed, where the models perform better on $\mathcal{D}_{val}^{DS}$. This is attributed to voxel occupancy grid predictions

focusing more on occupied grids. Since voxel grids are mostly empty, models on $\mathcal{D}_{val}^{DS}$ tend to predict more noise, leading to lower IoU$^-$ scores.

Comparing the setting with open weights to not performing pre-training, the open model shows advantages early on, supporting the idea that pre-trained weights contribute valuable spatial knowledge. However, in the later stages of training, the model trained from scratch overtakes the open model in precision, while the model with open weights remains superior for IoU$^-$ and recall. This indicates that the non-pre-trained model adopts a more conservative strategy for 3D occupancy prediction. When the scenario with frozen weights is examined, although the model underperforms compared to the other two, its performance improves over time by only training the voxel decoder. This improvement underscores that the pre-trained weights already contain some, however limited, spatial information, indicating that the model partially integrates image and point cloud features to form spatial voxel features even when trained only on these two modalities.

As learning 3D occupancy is computationally intensive, it can be concluded that pre-training strategies on only camera and LIDAR data are generally recommendable, as they both speed up training and show overall superior performance.

Sensor Data Predictions: Experiments are performed to determine whether knowledge encoded through occupancy can be leveraged by LIDAR and camera predictions, as shown in Figure 5.6. Based on the Chamfer Distance for point clouds and the PSNR metric for images, only slightly increased performance gains for both modalities can be observed when occupancy prediction is included, with a more pronounced benefit for camera predictions under the $\mathcal{D}_{val}^{RL}$ setting.

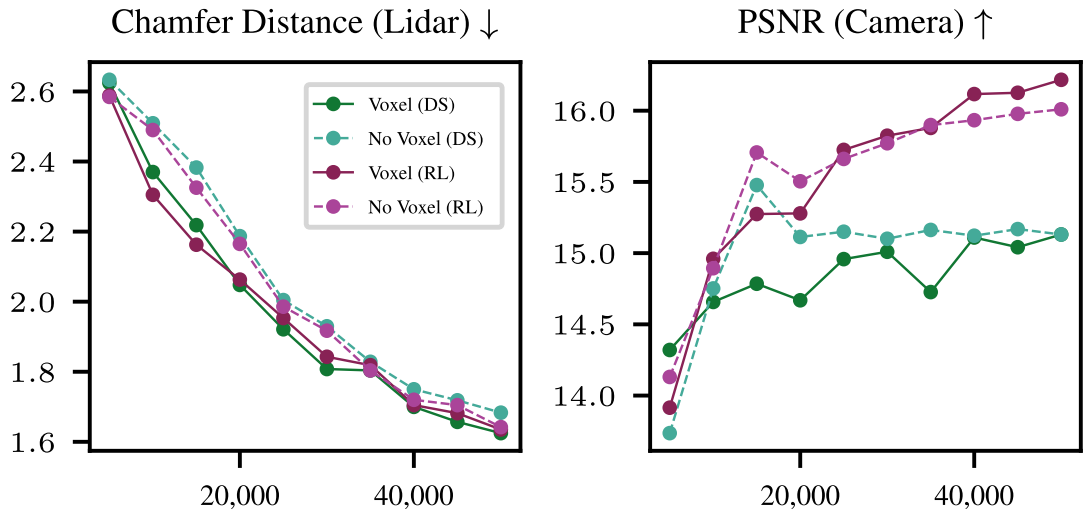


Figure 5.6.: **Occupancy:** Impact of predicting 3D occupancy on the quality of camera and LIDAR predictions, evaluated on both $\mathcal{D}_{val}^{RL}$ and $\mathcal{D}_{val}^{DS}$. Reprinted from [BOG 23].

5.2.4. Summary

The presented world model, as shown in Figure 5.1, is the first to leverage multimodal sensor data in the form of camera and LIDAR data without relying on limiting BEV representations. An extensive set of experiments is performed, examining different sensor fusion strategies and the effect of additionally predicting 3D occupancy. The experiments demonstrate that range view-based LIDAR representations with a standard Transformer-based fusion and an increased 2D latent space are beneficial in the case of camera-LIDAR fusion, confirming that the introduction of a BEV feature representation, as typical in the literature [506, 468], acts as a bottleneck, as it misses height information. This world model now represents a data-defined definition of normality and can be leveraged for anomaly detection. The following Section 5.3 utilizes the presented world model to detect object-level anomalies that are absent from the training data.

5.3. Label-Free Anomaly Detection

As shown in Chapter 3, anomaly detection is often based on highly specialized methods, focusing on the content layer [316, 108, 318]. However, as shown in Section 5.2, the perpendicular line of work of world models focuses on a more general understanding of the world. Such generative world models have shown promising results in autonomous driving [190, 192, 503, 143] [BOG 23]. They embed sensory data into latent states, reconstruct observations based on those, and predict action-conditioned future states. For anomaly detection, however, they have not been utilized yet [BOG 2]. In this section, the world model introduced in Section 5.2 is used for multimodal anomaly detection, employing both the reconstructive and predictive capabilities of the world model. By leveraging advances from the field of image segmentation, the detections are further refined in a self-supervised fashion to better identify anomalous instances. The presented approach outperforms the most relevant SotA method in label-free anomaly detection on the benchmark introduced in Section 4.3, setting a new baseline in label-free anomaly detection for autonomous driving.

5.3.1. Related Work

Recent trends in anomaly detection have shown that utilizing semantic segmentation models and including exemplary anomalies for outlier exposure data during training achieves close-to-perfect benchmark results [44, 75, 108], as described in Chapter 3. In addition, most works focus on camera data, with only few that leverage multimodal data but require models trained in a supervised fashion [BOG 4][253]. However, normality should be learned from raw sensory data and thus in a label-free setting, as introduced in Section 2.1. Including anomalies during training poses the risk of missing anomalies in a never-ending open-world

setting, and utilizing supervised semantic segmentation [316, 4, 112], bounding boxes [275, 129], or language [409] limits the definition of normality to labeled training data, which does not scale well. This section revisits the field of label-free anomaly detection and also explores mask-level approaches for enhancing detections.

Label-Free Anomaly Detection: As introduced in Section 2.1, the term label-free refers to methods that do not use labeled data during training. While modeling uncertainty of models on computer vision tasks in an unsupervised way has already been addressed [219, 140, 141, 159], these models were not evaluated on anomaly detection benchmarks. Instead, they have been evaluated on their ability to model uncertainty in general computer vision tasks. Since anomaly detection is not only relevant in autonomous driving, there are also unsupervised anomaly detection methods in other domains. For example, Zhou et al. [512] have developed an anomaly detection model on retinal images, e.g., for detecting retinal diseases or lesions, and many works [438, 250, 203] have evaluated their anomaly detection models on the MVTec Anomaly Detection (MVTec AD) dataset [37] for industrial inspection. Similarly, self-supervised detection methods exist in label-free settings [380, 209, 496]. Others use the Modified National Institute of Standards and Technology (MNIST) [246, 10, 182] or Canadian Institute For Advanced Research (CIFAR) [238, 182, 431] datasets, which contain images of only small sizes for their evaluation. Tu et al. address self-supervised anomaly detection in autonomous driving by synthesizing anomalies [415], effectively introducing outlier exposure.

In anomaly detection in the surveillance setting, there is also a trend towards supervision requiring labeled training data [275, 129]. However, there are two recent label-free methods. Abati et al. [1] have developed a novelty detection model that uses a deep autoencoder in combination with an autoregressive parametric density estimator, using real-world data with the University of California, San Diego (UCSD) Ped2 [74] and the ShanghaiTech [284] datasets. Similar to Abati et al. [1], Park et al. [339] trained Memory-guided Normality for Anomaly Detection (MNAD) on datasets with images from the real world [74, 284, 280], which partly contain semantic classes that can also be found in autonomous driving, e.g., pedestrians, bicycles, and cars. They compare the reconstruction of an autoencoder to the initial input image by using the L2 distance and the PSNR in order to calculate anomaly scores.

Mask-Level Anomaly Detection: A general trend to improve anomaly detection methods, which typically predict anomaly scores for independent pixels, is to use learned masks to generate instance-level detections. For detecting masks of anomalous instances in an image, the zero-shot Segment Anything Model (SAM) [224] was quickly used for the localization of anomalies in images. In the following, an overview of recent methods using segmentations during post-processing is given, as shown in Table 5.1.

Segment Any Anomaly (SAA)+ [63] utilizes pre-trained foundation models for mask-level anomaly detection without further training. The authors first em-

Method	Supervision	Temporal	Multimodal	Outlier Exposure	Extra Networks
SAA+ [63]	✓	—	✓	—	✓ [270, 224]
UGainS [318]	✓	—	—	✓	✓ [316, 224]
S2M [510]	✓	—	—	✓	✓ [360, 274, 224]
ClipSAM [254]	✓	—	✓	—	✓ [354, 224]
Presented method	—	✓	✓	—	✓ [BOG 23][325]

Table 5.1.: **Overview of mask-level anomaly detection methods:** The table shows methods that use segmentation masks for post-processing. Supervision refers to the necessity of labeled data. Temporality denotes the incorporation of temporal context. Multimodal models utilize further modalities besides RGB data, such as text or LIDAR, for anomaly detection. Outlier exposure shows whether exemplary anomalies are needed during training. Finally, all extra needed networks are shown. Adapted from [BOG 14].

ploy Grounding Detection Transformer with Improved Denoising Anchor Boxes (DINO) [270], which provides bounding boxes for regions defined by a prompt. To refine those box regions into masks, they utilize SAM [224]. Similarly, Score To Segmentation Mask (S2M) [510] generates bounding boxes that include anomalies, followed by SAM. Comparable to many other anomaly detection models, they use outlier exposure during training. Uncertainty Guided Anomaly Instance Segmentation (UGainS) [318] uses the existing anomaly detection model RbA [316] in combination with SAM for localizing anomalous instances in the observation. Finally, ClipSAM [254] utilizes CLIP text and image encoders [354] to generate an initial anomaly mask and refines it with SAM.

Recent trends have moved away from label-free anomaly detection, and benchmarks are saturated with near-perfect results, as described in Chapter 3. While label-free anomaly detection methods from other domains are available, no label-free anomaly detection model for autonomous driving has been proposed so far. Here, the core difference lies in the scene complexity. Anomaly detection in industrial or medical settings focuses on static scenes with mostly single objects, while traffic scenes are highly complex. In addition, the recent trend of mask-level anomaly detection methods works in a supervised manner. Thus, there is a clear need to revisit the field of label-free anomaly detection in order to use vast amounts of unlabeled data for training, as typically available in autonomous driving.

5.3.2. Method

As shown in Section 5.3.1 and Table 5.1, the presented work is the first label-free mask-level anomaly detection method. In the context of autonomous driving, this means it can be trained purely based on unlabeled sensor recordings without the need to record abnormal driving situations. An overview of the approach is

shown in Figure 5.7. First, multimodal sensor data from several different sensors is used as input for the world model presented in Section 5.2 to reconstruct and predict future frames. Furthermore, semantic masks are derived from camera data. For visual differences, a reconstruction of the current observation is compared to the accompanying sensor data frame based on multiple methodologies. For temporal differences, only multiple future predictions from the world model are compared. After a weighted fusion of the pixel-wise scores, the resulting anomaly map is refined based on the generated masks.

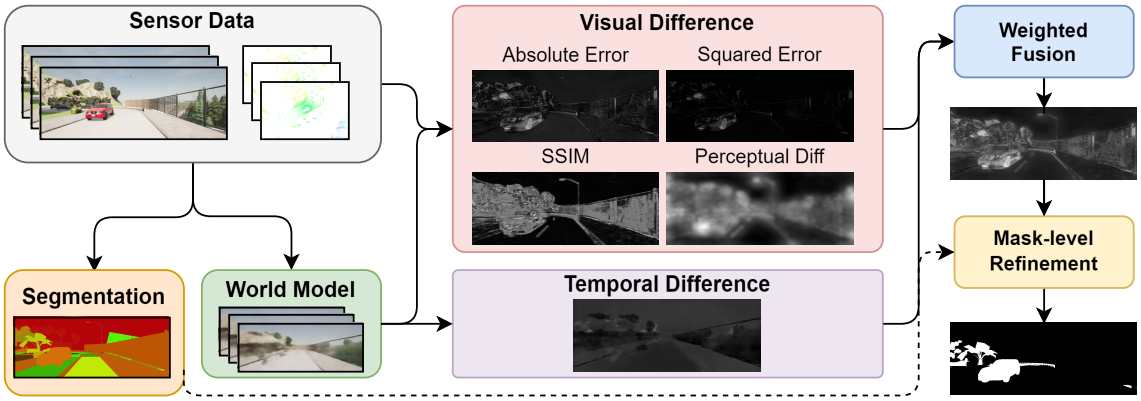


Figure 5.7.: **Label-free anomaly detection:** Multimodal sensor data is fed into a world model to reconstruct and predict frames, and semantic masks are derived from camera data. For *visual differences*, a reconstruction of the current observation is compared to the accompanying sensor data frame based on multiple methodologies. For *temporal differences*, only multiple future predictions from the world model are compared. After a weighted fusion of the pixel-wise scores, the resulting anomaly map is refined based on the generated masks. Reprinted from [BOG 14].

The approach first uses the world model to generate a reconstruction of the current frame. This reconstruction is then compared to the ground truth sensory data from the camera sensor of the autonomous vehicle. While the approach only uses camera data, the world model is grounded and conditioned by further sensor modalities, planned actions, and the provided route. To compute *visual differences*, several image comparison methods are employed in order to evaluate their influence on the anomaly detection performance. The Absolute Error (ABS) Δ_{ABS} and MSE Δ_{MSE} are calculated for each pixel individually and measure the differences in the r, g, b color channels of the reconstruction $\hat{x}$ and the sensory image x.

$$\Delta_{ABS} = \frac{|r_x - r_{\hat{x}}| + |g_x - g_{\hat{x}}| + |b_x - b_{\hat{x}}|}{3} \tag{5.2}$$

$$\Delta_{MSE} = \frac{(r_x - r_{\hat{x}})^2 + (g_x - g_{\hat{x}})^2 + (b_x - b_{\hat{x}})^2}{3} \tag{5.3}$$

Contrary to this, SSIM [443] compares images based on their structure by utilizing batches of multiple proximate pixels, rather than focusing on individual pixels alone, using sliding window patches. In Equation 5.4, μ denotes means and σ (co)variances, with constants κ_1 and κ_2 for numerical stability [429, 443].

$$\Delta_{SSIM} = \frac{(2\mu_x\mu_{\hat{x}} + \kappa_1)(2\sigma_{x\hat{x}} + \kappa_2)}{(\mu_x^2 + \mu_{\hat{x}}^2 + \kappa_1)(\sigma_x^2 + \sigma_{\hat{x}}^2 + \kappa_2)} \tag{5.4}$$

Finally, perceptual difference Δ_{PD} [112] is an image comparison method that leverages a pre-trained deep CNN to extract features and compare two images pixel-wise based on their content. Similar to Di Biase et al. [112], weights that are pre-trained on the ImageNet [109] dataset are utilized. In Equation 5.5, F^i denotes the i-th layer of a VGG network [394], and M and N refer to elements and layers, respectively.

$$\Delta_{PD} = \sum_{i=1}^{N} \frac{1}{M_i} \|F^i(x) - F^i(\hat{x})\|_1 \tag{5.5}$$

For *temporal differences* Δ_{TD}, multiple predictions of the world model are compared to each other. The temporal difference is calculated by comparing prior predictions for the current time step to each other. For this, the mean of the absolute errors between n past predictions $\hat{x}_{t-i}$ for time t and the current reconstruction $\hat{x}_t$ is computed, as shown in Equation 5.6.

$$\Delta_{TD} = \frac{1}{n} \left(\sum_{i=1}^{n} \Delta_{ABS} \left(\hat{x}_{t-i}, \hat{x}_t \right) \right) \tag{5.6}$$

All K difference maps are then normalized and can thus be fused by assigning weights $w_i \in [0, 1]$ with $\sum_{i=1}^{K} w_i = 1$ to compute the final mask, as shown in Equation 5.7.

$$\Delta_{Total} = \sum_{i=1}^{K} w_i \Delta_i \tag{5.7}$$

While the resulting anomaly map assigns anomaly scores to each pixel in the image, it does not classify instances in an observation as anomalous. For this, the scores are refined with instance masks to generate mask-level anomaly scores. By utilizing an image segmentation approach for mask generation, the presented method iterates through each observed mask in the observation and calculates average instance-wise anomaly scores.

5.3.3. Experiments

As shown in Chapter 3, common anomaly benchmarks, such as Fishyscapes [44] or Segment Me If You Can [75], are limited to camera data and do not contain data on actions, e.g., steering wheel angle, or additional sensory data. Among existing anomaly detection benchmarks [BOG 22], the framework introduced in Section 4.3 is the only one providing multimodal sensory data and action data of the ego-vehicle. Based on the presented data generation pipeline as shown in Figure 4.5, the evaluation in this section is performed with object-level anomalies. For this purpose, 16 abnormal driving scenarios with 200 frames each were generated to create a small-sized benchmark with anomalies that is comparable to current benchmarks. The scenarios take place in different towns under different weather conditions and contain static anomalies, e.g., an object or an animal standing on the street, as depicted in Figure 5.8. The dataloader for the world model samples each 10^{th} frame, i.e., every second.

Experimental Setup: The presented method requires both a self-supervised world model and a self-supervised segmentation model. The training dataset of the world model introduced in Section 5.2 does not contain anomalies and thus establishes the baseline for typical behavior in the context of anomaly detection.

For image segmentation, all prior works shown in Table 5.1 utilize the Segment Anything Model [224]. However, SAM was trained in a supervised manner, limiting the use of large-scale, unlabeled datasets as typically available in autonomous driving. Differently, Unsupervised Universal Segmentation (U2Seg) is an image segmentation model that is capable of generating panoptic segmentation masks by using self-supervised learning and clustering. This conceptually enables the learning of both the world model and the segmentation model on the same large-scale, unlabeled dataset. It would have been beneficial to train U2Seg on the target domain, but as it was trained on the entirety of ImageNet [109], the necessary resources for training were unavailable, and a provided checkpoint is used. Experiments with both SAM and U2Seg are performed.

Baseline: As described in Section 5.3.1, there are only two relevant SotA label-free anomaly detection models. While both models demonstrate similar performances, Abati et al. [1] only provide inference, but no training code for their approach. Thus, the approach presented in this chapter is evaluated against MNAD by Park et al. [339]. The authors provide code for both prediction and reconstruction tasks, but focus on frame-wise evaluations. To verify their evaluation, the experimental results of Park et al. [339] could be reproduced in a first step.

For the evaluation, MNAD was trained on a reduced version of the dataset that was used to train the world model introduced in Section 5.2. Each 100^{th} frame was sampled from it, resulting in 2,725 frames. This ensures that MNAD was trained on images from the same towns, with the same driving conditions, and thus with the same semantic structure as the world model. The sampling was necessary to prevent overfitting, as UCSD Ped2, which was originally used by

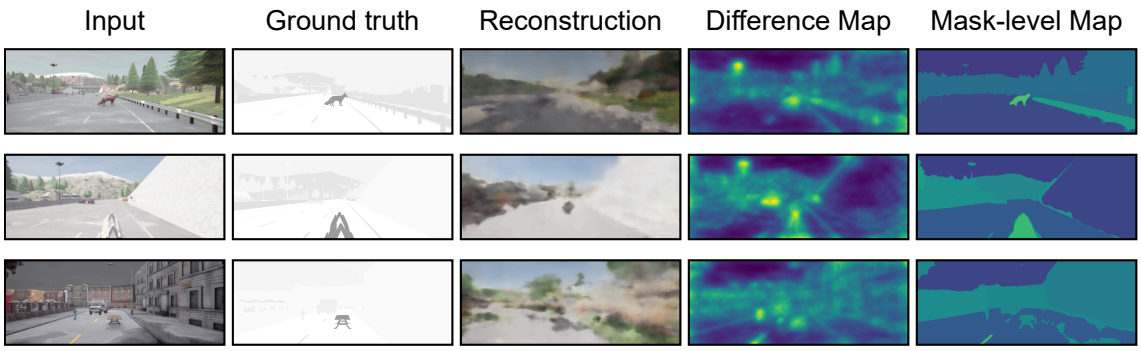


Figure 5.8.: **Exemplary Detections**: The first columns show the input image and the corresponding ground truth. World model reconstructions are utilized to generate difference maps, which are finally refined to mask-level maps. Masks are generated by the label-free segmentation model U2Seg. The first two rows show positive cases, while the third row shows a failure case. Adapted from [BOG 14].

Park et al. [339], only contains 2,550 images. The data sampling thus allows a dataset size which is comparable to the one used to train MNAD in the original experimental setup. [339]. Following Park et al. [339], the model was trained for 60 epochs. Contrary to the approach presented in this chapter, MNAD only localizes anomalies as an intermediate step and uses additional metrics for the final frame-wise score. While frame-wise scores can also be used in the context of autonomous driving [BOG 17][STU 4] to detect frames including anomalies, they do not allow for the localization of anomalies. Thus, based on these intermediate reconstructions, the L2 distance is used to compute pixel-wise anomaly scores.

5.3.4. Evaluation

For the evaluation of the presented method, a wide variety of combinations of the five introduced visual and temporal difference components are examined. The Average Precision (AP), FPR_{95}, and the AUROC metrics are used, as they are common metrics in anomaly detection benchmarks for autonomous driving [44, 75]. All combinations, as shown by the used weights w_i, and the results can be found in Table 5.2.

Experimental Results: Here, the findings on the performance of the presented approach compared to the MNAD baseline are presented. Since the visual differences and the temporal differences are normalized, they can be individually weighted and combined in order to form an anomaly map. This process is done in the weighted fusion component. In the following, the impact of the single difference metrics and their combinations is also evaluated.

Since MNAD does not use masks, first, the pixel-wise L2 distance of MNAD is compared to the similarly calculated MSE of the presented method on the

u_{ABS}	u_{MSE}	u_{SSIM}	u_{PD}	u_{TD}	AP ↑	FPR_{95} ↓	AUROC ↑	AP ↑	FPR_{95} ↓	AUROC ↑
					Ground truth			**SAM**		
1	0	0	0	0	17.68	35.56	65.23	13.72	50.58	65.16
0	1	0	0	0	19.05	38.92	63.61	13.77	52.22	64.93
0	0	1	0	0	19.77	21.26	79.03	11.43	46.79	68.26
0	0	0	1	0	**29.90**	**16.93**	**83.18**	**18.93**	**42.32**	**71.88**
0	0	0	0	1	11.41	52.70	49.15	7.11	69.26	47.72
0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	27.50	17.81	82.47	17.11	44.01	70.83
$\frac{1}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	0	26.21	18.16	82.07	16.02	44.55	70.88
$\frac{1}{3}$	$\frac{1}{3}$	0	$\frac{1}{3}$	0	25.52	20.67	79.73	17.11	43.83	71.74
$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	0	18.85	22.88	77.73	12.85	46.44	70.20
0	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	25.28	18.53	81.83	14.30	45.18	69.85
$\frac{1}{4}$	0	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	24.34	19.39	81.27	14.74	44.87	69.95
$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{1}{4}$	$\frac{1}{4}$	22.28	21.43	79.05	16.15	45.28	70.88
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{1}{4}$	17.25	24.15	76.92	12.60	48.12	68.42
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	0	23.47	19.04	81.34	15.55	44.35	71.12
$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	22.38	19.71	80.74	14.52	45.02	70.21
					U2Seg			**Max. Value**		
1	0	0	0	0	14.04	60.20	59.55	19.00	59.68	40.68
0	1	0	0	0	14.54	60.98	59.93	18.86	59.52	40.76
0	0	1	0	0	12.17	58.44	62.88	10.87	67.03	33.30
0	0	0	1	0	**18.88**	56.74	64.77	17.26	57.68	42.55
0	0	0	0	1	9.02	68.89	54.44	11.01	74.23	25.97
0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	17.70	56.76	65.09	20.97	52.01	**48.57**
$\frac{1}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	0	17.13	56.73	**65.50**	18.71	52.63	47.85
$\frac{1}{3}$	$\frac{1}{3}$	0	$\frac{1}{3}$	0	17.99	57.07	65.47	**21.91**	**51.83**	48.49
$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	0	13.64	58.31	63.97	18.51	60.24	40.52
0	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	17.08	**56.54**	65.13	16.69	56.77	43.76
$\frac{1}{4}$	0	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	16.35	56.92	65.18	15.44	57.88	42.90
$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{1}{4}$	$\frac{1}{4}$	17.15	57.09	64.97	18.44	56.67	43.68
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{1}{4}$	12.17	58.44	62.88	16.05	63.27	37.25
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	0	17.16	56.84	62.88	19.86	53.36	47.22
$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	16.38	57.06	65.04	20.01	56.84	43.82
					No Mask			**Single Mask**		
1	0	0	0	0	6.80	78.19	60.19	5.04	93.57	50.54
0	1	0	0	0	7.03	78.49	60.68	5.04	93.57	50.53
0	0	1	0	0	4.72	50.87	73.02	5.83	92.83	51.12
0	0	0	1	0	**10.86**	**32.91**	**79.51**	10.40	88.49	53.26
0	0	0	0	1	4.09	73.37	53.05	5.06	93.57	50.59
0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	9.51	37.37	53.05	**12.66**	**86.31**	**54.48**
$\frac{1}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	0	9.29	38.99	78.70	8.88	89.93	52.59
$\frac{1}{3}$	$\frac{1}{3}$	0	$\frac{1}{3}$	0	9.42	42.34	76.24	8.83	89.94	52.54
$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	0	6.93	52.40	72.32	5.05	93.56	50.64
0	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	8.29	39.37	77.51	8.88	89.93	52.57
$\frac{1}{4}$	0	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	8.14	40.26	77.17	10.37	88.48	53.35
$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{1}{4}$	$\frac{1}{4}$	8.50	44.02	75.05	7.30	91.39	51.79
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{1}{4}$	6.16	53.28	71.14	4.30	94.29	50.26
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	0	8.83	40.07	77.62	8.83	89.93	52.57
$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	8.11	41.12	76.69	8.07	90.66	52.18
	MNAD [339]				6.37	89.61	61.96	—	—	—

Table 5.2.: **Evaluation Results:** The six experiments shown use the following settings: Ground truth segmentation; segmentation from SAM and U2Seg; instance-wise maximum anomaly value selection; no mask segmentation; selection of a single mask instance with the highest anomaly score. **Best** and second-best results are highlighted. Adapted from [BOG 14].

raw pixel-wise output without masks. With the presented method in the setting of using the MSE as visual difference, the AP is 7.03, the FPR_{95} 78.49, and the AUROC 60.68. Compared to MNAD, AP is 10.36% higher and FPR_{95} 12.41% lower. Compared to these improvements, AUROC is 2.11% higher for MNAD. Even better results are achieved when using the *perceptual difference* for visual difference. Here, the presented method achieves by far the highest AP, lowest FPR_{95}, and highest AUROC in the pixel-wise setup without masks. AP is 70.49% higher, FPR_{95} 63.27% lower, and AUROC 28.32% higher compared to the experimental results for MNAD.

When improving the results with masks, the improvements become even more pronounced. Using masks in a label-free setting that are generated with the image segmentation model U2Seg and the perceptual difference as visual difference, the method presented in this chapter achieves an AP that is 196.39% higher than the AP in the evaluation of MNAD.

Ablation Studies: In order to better understand the presented method, a set of ablation studies is performed. First, next to utilizing U2Seg, the possible performance gains of using SAM [224] or ground truth masks are of interest. SAM is a zero-shot image segmentation model that is also used by SotA anomaly detection models. While SAM was trained with labeled data, it performs well in the context of zero-shot inference. The effects of not refining the anomaly map with masks at all are examined as well. Second, rather than averaging all anomaly scores per mask, it is of interest whether picking the maximum value, inspired by Liu et al. [275], impacts the performance. Table 5.2 shows these results in the section "Max. Value". As shown in the section "Single Mask", picking only the mask with the highest anomaly score, neglecting the rest, is also examined.

When using the zero-shot image segmentation approach SAM, which is also used as an image segmentation approach in prior anomaly detection models, it is possible to further improve the experimental results. With the perceptual difference as visual difference in the setup, AP in this setup is 18.93%, FPR_{95} is 52.77% lower, and AUROC 16.01% higher than in the respective results for MNAD. To evaluate the full potential of utilizing masks for anomaly detection, the presented approach is also evaluated with masks from a ground truth instance segmentation map. This setup achieves by far the best experimental results, again showing the huge potential of leveraging masks in anomaly detection. The best AP score with this experimental setup is 29.90, the best FPR_{95} is 16.93, and the best AUROC is 83.18. In the prior experimental setups, the average anomaly score of the masks is used for evaluation. Interestingly, it shows that the perceptual difference is not suitable for anomaly detection when assigning the maximum anomaly score to masks rather than their average score. Generally, substituting the average anomaly score per instance with the maximum score does not achieve better results. Worst results are achieved when only considering the instance with the highest anomaly score. In this setting, often not the anomalous object, but a different object in the observation has the highest anomaly score. This then results in completely ignoring the abnormal object.

5.4. Conclusion

This chapter presents an approach to first learn a representation of normality and then leverage it to detect external object-level anomalies, as introduced in Table 2.1. Addressing **RQ3**, both methods presented in this chapter – training the world model and using it for anomaly detection with mask-based refinements – leverage unlabeled sensor data from multiple modalities. The world model introduced in Section 5.2 is the first to leverage both camera and LIDAR data without a BEV bottleneck representation. Based on a concept of normality as shown in Table 4.4, carefully selected data from the CARLA simulation environment was used for training. Here, no anomalies are present, allowing the world model to learn a data-based representation of normality. The world model is used subsequently in Section 5.3 for anomaly detection. It is evaluated with the benchmark introduced in Section 4.3, focusing on object-level anomalies. It outperforms the most relevant label-free SotA anomaly detection method, and is further improved by mask-based refinements, as shown in Table 5.2.

The following Chapter 6 presents an anomaly handling approach for the task of driving. Here, previously detected object-level anomalies are integrated into the training dataset to improve situations in which the lane ahead is blocked by an obstacle, requiring controlled traffic rule exceptions.

5.4.1. Recent Advances

The field has continued to evolve since the development and publication of the works underlying this chapter [BOG 23, 14]. Recently, multiple works [132, 286, 414, 80] have examined the field of world models for autonomous driving. Among the approaches analyzed by those works [481, 479, 486, 506, 168, 91, 410, 144, 502, 281, 468, 84], only BEVWorld [506] and HoloDrive [468] are included as multimodal world models comparable to the approach presented in this chapter, not requiring labeled data during training. The reliance of both on a bottleneck BEV representation as the key difference has already been addressed in Section 5.2.1, showing that the presented approach is still the only multimodal world model not limited by a BEV bottleneck.

In label-free anomaly detection, i.e., the detection of anomalies without the need for labeled data, many works remain in the low-complexity setting of anomalies in industrial and medical settings [455, 488, 461, 315]. For autonomous driving, recent advances in the general field of anomaly detection have already been discussed in Section 3.6.1. Shining a light on a rare label-free approach, Cai et al. present an unsupervised anomaly detection approach for LIDAR data [60]. Their work is inspired by MNAD [339] and focuses on the detection of adversarial attacks at the scene level without atypical objects. While the approach cannot be compared directly to the presented method in Section 5.3, no architectural novelty compared to MNAD can be observed. MNAD was clearly outperformed by the anomaly

detection method presented in this chapter, as shown in Table 5.2. These advances underline the continued relevance of the methods presented in this chapter, as label-free anomaly detection remains underexplored in the field of autonomous driving.

But obstacles don't have to stop you.

Michael Jordan, 1994 [212]

6. Anomaly Handling

A supervised student thesis has contributed to this chapter [STU 6]. Parts of this chapter have previously appeared in the following publication:

- D. Bogdoll et al. *Informed Reinforcement Learning for Situation-Aware Traffic Rule Exceptions*. In IEEE International Conference on Robotics and Automation (ICRA), 2024 [BOG 18]

6.1. Introduction

External anomalies detected by anomaly detection methods, such as the method presented in Chapter 5, can be subsequently addressed in either an online setting, i.e., when encountered by an autonomous vehicle during operation, or an offline setting, i.e., when collected and used for the general improvement of the system. In online settings, anomalies can either be addressed while driving, e.g., by maintaining a higher safety distance to detections considered unknown to address potentially unknown behavior [391] or via human support. In the latter case, an autonomous vehicle comes to a stop and remote assistance [BOG 15, 11, 9, 26, 29, 30] is activated.

This chapter discusses the handling of object-level anomalies in an offline setting. Previously detected anomalies are integrated into the training data for future model training, effectively removing their status as anomalies [234]. This is especially useful for whole categories of detected anomalies, which can be handled appropriately. Addressing **RQ4**, this chapter improves learned trajectory planning by enabling an autonomous vehicle to perform controlled traffic rule exceptions through prior knowledge of hierarchical traffic rules. Based on a RL setting with curriculum learning, a situation-aware reward design is introduced to provide a dynamic reward signal for situations that allow for controlled rule exceptions.

6.2. Situation-Aware Reinforcement Learning

Navigating complex traffic scenarios with object-level anomalies requires a high level of flexibility. Especially in the field of Reinforcement Learning for Autonomous Driving, often very simple and conflicting reward functions are being used, which do not have the potential to solve such scenarios [229]. Especially

hierarchical traffic rules, which sometimes override others in specific situations, are typically neglected in reward functions but are necessary in everyday traffic [72, 13]. Additionally, even though RL has made strides in behavior planning and control instructions for autonomous driving, the potential of RL in direct trajectory generation is not extensively researched [307].

The method presented in this chapter leverages the capabilities of informed RL[1] [BOG 32] to enhance the decision-making and adaptability of autonomous vehicles, especially in scenarios requiring traffic rule exceptions. Hierarchical traffic rules can be used as a source of knowledge for an informed and dynamic reward in order to handle situations that include anomalies. This means that rule violations, which are situationally permitted by a hierarchical traffic rule, are not statically penalized in the reward function.

6.2.1. Related Work

This section reviews related work on the application of Reinforcement Learning in motion planning for autonomous vehicles, typical traffic scenarios for training, and traffic rule formalization. It facilitates a better understanding of the presented work and introduces methods that are adapted for later experiments.

Reinforcement Learning for Motion Planning

Motion planning in the context of autonomous driving can be split into behavioral planning, trajectory planning, and control instructions [13]. In the context of behavioral planning, Fayjie et al. [130] proposed an RL-based autonomous driving strategy for urban traffic scenarios, where the discrete action space consists of "left", "right", and "keep going" to symbolize lane changing behaviors. Ye et al. [489] proposed a strategy for automatic lane changing with RL based on Proximal Policy Optimization (PPO). This strategy enables a trained agent to make efficient lane-changing decisions even in dense traffic scenarios. Furthermore, numerous studies have been conducted exploring RL-based behavioral planning for autonomous vehicles, with findings demonstrating reliable performance [194, 50, 64, 189, 157].

In the context of trajectory planning, Feher et al. [131] learn waypoints that an agent should follow. For that, they use the Deep Deterministic Policy Gradient (DDPG) algorithm. A limitation of this methodology lies in its sole focus on lateral planning. Moghadam et al. [307] propose an RL agent that learns input parameters for a trajectory planner on the Frenet Space for highway scenarios. They use a continuous action space with processed time-series data as observation space instead of raw sensory observations. Coad et al. [97] present an RL agent with a continuous action space in a static occupancy grid. The agent's action

[1] Informed ML is concerned with the integration of "prior knowledge into learning systems" [430]

space is a sequence of changes in curvilinear coordinates, lateral displacement, and velocity with a fixed longitudinal step. Lu et al. [282] propose a hierarchical Reinforcement Learning framework for trajectory planning given a state space composed of BEV images and LIDAR data. The framework consists of a high-level action responsible for choosing the direction of motion and a medium-level action sampling the vehicle's next waypoint from a fixed-size semi-circle, which can also sample off-road waypoints.

In the context of control instructions, with a focus on end-to-end learning, many methods leverage raw sensor inputs as the input space and directly output control commands for autonomous vehicle control, which include steering angle and acceleration [14, 206, 311, 374, 493]. However, these approaches are challenging to interpret, as no planned trajectory is available.

The method presented in this chapter is most related to RL works in the context of trajectory planning. Neither behavior plans nor control instructions can be precisely analyzed with respect to their degree of compliance given a traffic rule. The method is most related to the work presented by Moghadam et al. [307], as it also learns trajectories in Frenet space. However, a core difference is the utilized observation space. While their work relies on processed time-series data, the method presented in this chapter utilizes RGB observations.

Traffic Scenarios

Most RL-based autonomous driving studies set up a specific autonomous driving environment for the vehicle. Given the relatively straightforward nature of highway traffic conditions, these environments present comparatively less complex scenarios for autonomous driving. Consequently, a substantial number of studies utilize highway scenarios as the benchmark for evaluating RL-based autonomous driving strategies [475, 14, 24, 189, 311, 458]. However, further approaches focus on urban area traffic, encompassing elementary urban traffic, intersections, as well as dense urban traffic situations [50, 130, 489, 500]. Nonetheless, there is a lack of literature considering scenarios that require controlled traffic rule exceptions [13][BOG 5]. Talamini et al. [403] utilize RL to train a driving strategy when controlled traffic rule exceptions become necessary. Their approach considers behavior planning with lateral motion only and does not provide a structured approach regarding the integration of hierarchical rules into the reward.

The method presented in this chapter is not confined to any category of traffic scenarios. Different from Talamini et al. [403], where regular traffic scenarios are examined, the method presented in this chapter addresses rule exceptions in atypical traffic scenarios involving previously detected anomalies.

Formalism of Traffic Rules

Many studies have explored formalizing traffic rules into a machine-readable format. A number of methods have been used, e.g., temporal logic [291], Linear Temporal Logic (LTL) [226], Signal Temporal Logic (STL) [7], Isabelle theorem proving [361], and fuzzy logic [309]. However, these studies primarily concentrate on translating individual rules without considering the prioritization among different rules. Censi et al. [72] introduced a theoretical rulebook to structure different rules, establish a hierarchy between rules, and analyze traffic rule exception scenarios, but did not provide a framework or implementation to utilize it.

The work presented in this chapter directly builds upon the rulebook developed by Censi et al. [72], but integrates it into a reward function rather than applying it during inference. Overall, there is a noticeable research gap in the development of RL algorithms for autonomous vehicles that not only address the trajectory generation in scenarios that require traffic rule exceptions, but also efficiently incorporate a structured set of traffic rules into the reward function.

6.2.2. Method

This section presents the methodology as visualized in Figure 6.1. First, the problem statement is introduced, which outlines present challenges. Next, the generation of vehicle trajectories using the Frenet Space [362] is detailed. Subsequently, the process of structuring traffic rules for computational interpretation is discussed, including using the rulebook and its integration into a reward function. The methodology can be utilized with arbitrary RL frameworks.

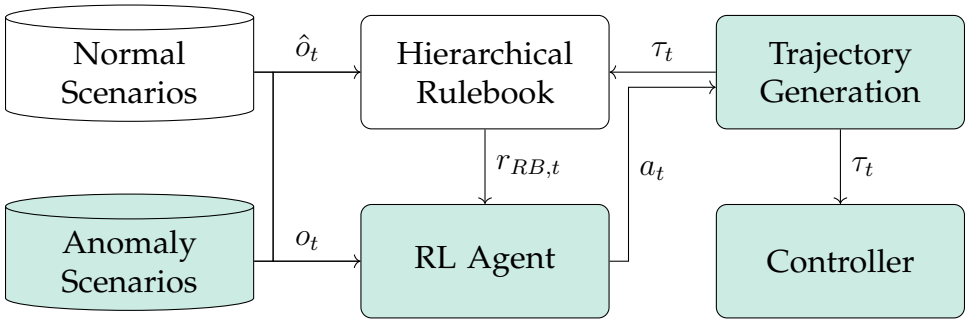


Figure 6.1.: **Architecture:** In a curriculum learning setting, normal scenarios are used first to learn basic driving behavior. Then, anomalies are provided to learn controlled rule exceptions. Given an observation o_t, the RL agent chooses an action a_t as the parametric input for generating a trajectory τ_t. The rulebook then evaluates the trajectory in the context of an abstracted environment $\hat{o}_t$ and provides the partial reward $r_{RB,t}$. Finally, a controller follows the trajectory. During evaluation, only the path in green is executed. Adapted from [BOG 18].

Situations requiring controlled traffic rule exceptions contain apparent conflicts of traffic rules, where one rule can override another. For an agent to solve the tasks, situation-awareness is necessary to apply the correct set of rules at any given time. As the system dynamics are unknown, the problem is modeled as a Partially Observable Markov Decision Process (POMDP), with the state space comprising high-dimensional sensory RGB data from a BEV camera and the action space consisting of input parameters for the generation of a trajectory in Frenet Space.

Trajectory Generation

To evaluate the compliance of actions with respect to given traffic rules, it is necessary to generate planned future trajectories. As shown in Figure 6.1, actions a_t are used as the input for a trajectory generation module. Following the approach presented by Werling et al. [451], trajectories are generated in Frenet Space. The goal state of a trajectory in Frenet space is fully defined by a target state $\{v, d, t\}$, which means that each set of parameters corresponds to a trajectory. This set is used as the input for the trajectory generation module. Here, v represents the desired velocity at the termination of the trajectory, d the lateral offset relative to the reference trajectory, and t the time needed to reach the desired target state. As shown in Figure 6.1, a controller is then utilized to follow the trajectory.

Situation-Aware Reward Design

Classic reward functions in RL are static and reward or punish a behavior irrespective of the context of a given situation. In the case of scenarios requiring traffic rule exceptions, some rules can override others. To adapt to this, the method presented in this chapter introduces a dynamic situation-aware reward. For this, a formal rulebook is used as part of the reward function to represent situation-specific hierarchies between different rules. Assuming no rule conflicts as the default state, the agent first needs to assess the current situation to activate dynamic rewards.

Situation Awareness: In order to assess traffic situations, an agent needs to have an understanding of traffic rules and capabilities to monitor them [BOG 12]. As shown in Figure 6.1, the abstracted environment $\hat{o}_t$ provides information for this purpose, such as map data or knowledge about relevant entities or objects in the environment. This information can be obtained either from processing sensory observations, a dedicated data source, or via ground truth in simulation. Information from the abstracted environment $\hat{o}_t$ is used, for example, to trigger hierarchical traffic rules and measure rule compliance. Similar to the definition of rules by Censi et al. [72], rule realizations are introduced. Let τ_t be a sequence of states, i.e., a trajectory. A rule realization $\psi : \tau_t \rightarrow \mathbb{R}$ assigns a real number $\psi(\tau_t) \in [0, 1]$ to τ_t. This is an expression of the degree of compliance of τ_t with respect to an underlying traffic rule, where a value of 1 indicates full compliance.

Based on this knowledge of existing traffic rules, an agent can then set rule coefficients ρ_j depending on the current situation, e.g., diminishing the relevance of a rule if it is overwritten by another one.

Hierarchical Rulebook: Inspired by the conceptual hierarchical rulebook by Censi et al. [72], an implementation of a rulebook is presented within the reward function of an RL agent. Following the definition of Censi et al. [72], a rulebook is defined as a tuple $(\Psi, \preceq)$, where Ψ represents a finite set of rule realizations ψ_i and $\preceq$ denotes a preorder on Ψ.

Linear Temporal Logic syntax is utilized for rule descriptions and their integration into the reward function. The rulebook is only activated when the situation awareness module detects a situation where a controlled rule exception becomes possible. The hierarchical structure of the rulebook is instrumental in determining which rules take precedence over others. Its hierarchical structure can be visualized as a graph, with each rule realization as a node and edges indicating priority relationships. Nodes of equal priority can be merged.

Figure 6.2.: **Hierarchical rulebook:** Graph representation of a rulebook $\mathcal{R}$ with rule realizations ψ_i and hierarchy coefficients ρ_j, where j indicates the hierarchy index. Adapted from [BOG 18].

For instance, in Figure 6.2, ψ_1 holds the highest hierarchy, ψ_2 and ψ_3 share the same, followed by ψ_4, and ψ_5 with the lowest. Such a representation assures the rulebook's scalability.

Linear Temporal Logic: LTL is a powerful logic language utilized in defining sequences of events or states. Its syntax contains various logical operators: negation ($\neg$), conjunction ($\wedge$), disjunction ($\vee$), and implication ($\rightarrow$), along with temporal operators: *Next* (**X**), *Globally* (**G**), *Finally* (**F**), and *Until* (**U**). To evaluate a trajectory τ_t, which can be defined as a sequence of vehicle states, the LTL is applied for each rule realization ψ_i. This reward calculation can be described by a function $f(\psi_i, \tau_t)$. More details on the utilization of LTL can be found in Section 6.2.3.

Reward Design: The rulebook's hierarchical structure is incorporated into the reward function based on hierarchy coefficients $\rho_j \in [0, 1]$ for each hierarchy, as shown in Figure 6.2. The coefficient scales the reward or penalty associated with a given level's rules so that higher-hierarchy rules have a higher weight in the reward. This way, the reward is dynamically adapted to the current situation. By default, the hierarchy coefficients are set to unity, such that $\rho_j = 1 \forall j$. The realization is shown in Equation 6.1.

$$r_{RB,t} = \sum_{\psi_i \in \Psi} \left(\prod_{\psi_j = \psi_i}^{\psi'} \rho_j \right) f(\psi_i, \tau_t) c_{\psi i} \tag{6.1}$$

Here, Ψ are all the rules involved in the rule exception, ψ' is the rule with the highest priority in the rulebook, and ρ_j is the hierarchy coefficient of ψ_j. $f(\psi_i, \tau_t)$ is the reward value. $c_{\psi i}$ is a scaling factor for each rule that allows fine-tuning.

6.2.3. Experiments

This section presents the experimental setup for the training and evaluation of the Reinforcement Learning agents. It first provides details on the traffic scenarios and rules examined. Next, the state and action spaces and the reward function are presented. Finally, it describes the training process and the parameters used.

Traffic Scenarios: For the experiments, a typical scenario in everyday urban traffic is chosen. Based on the German road traffic regulations, it is generally forbidden to cross a solid line. However, in certain situations, e.g., when the lane of the ego vehicle is blocked, there exists a rule exception [138], as shown in Figure 6.3.

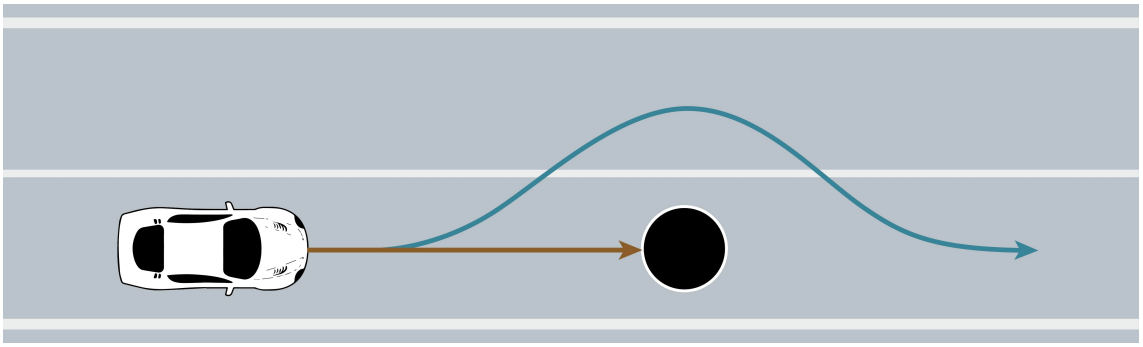


Figure 6.3.: **Scenario with controlled rule exception:** Traffic scenario that shows an atypical scenario with an object-level anomaly. In this scenario, the ego vehicle's lane is blocked. To deviate from continuing following its lane (brown), a controlled rule exception can be performed (blue). Reprinted from [BOG 18].

A benchmark[2] is provided with 1,000 such scenarios in the CARLA simulation environment [117] and also the codebase to generate more if needed. Each scenario is defined by a reference trajectory with a length of 80 meters and an object-level anomaly from the CARLA blueprint library [65] that blocks the lane at some point along the trajectory, as shown exemplarily in Figure 6.4. Focusing on object-level anomalies, the benchmark consists of low-complexity scenarios collected in the static CARLA Town 1 environment.

[2]The benchmark is available on GitHub: https://github.com/fzi-forschungszentrum-informatik/informed_rl

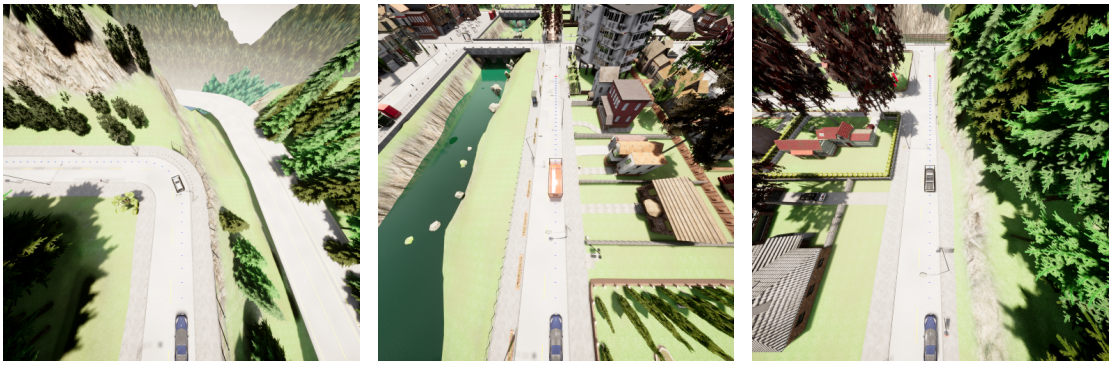
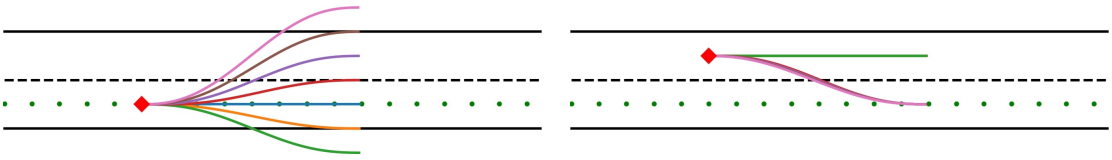


Figure 6.4.: **Benchmark:** Exemplary scenarios with different object-level anomalies blocking the lane. The dotted lines in front of the ego vehicle represent the reference trajectories towards the goal. Reprinted from [STU 6].

Agent Selection: For the experiments, two established Reinforcement Learning Models are selected. First, the current SotA model-based algorithm DreamerV3 is utilized, which demonstrates superior performance in a wide variety of domains [165]. Second, the model-free Rainbow algorithm [186] is utilized, an improved version of the well-known Deep Q-Network (DQN). In both cases, CNNs are used to encode the observations.

State Space: The state space comprises BEV RGB images with a resolution of 128×128 pixels, as shown in Figure 6.7. This state space inherently captures essential aspects like road geometry, the ego vehicle's position, obstacles, and the planned path.

Action Space: In order to generate trajectories in Frenet Space, the target state $\{v, d, t\}$ is utilized, as introduced earlier in Section 6.2.2. To focus on the agent's ability to avoid obstacles, the discrete action space is simplified. v and t are set to constants and d to specific values in dependence on the vehicle's position, as illustrated in Figure 6.5. As shown in Figure 6.1, a Proportional-Integral-Derivative (PID) controller is implemented to follow the generated trajectory subsequently.



(a) Action space for $d = 0$ (b) Action space for $d \neq 0$

Figure 6.5.: **Dynamic action space in Frenet space:** Visualization of trajectories τ_t based on selected actions a_t. The left side shows a scenario where the ego vehicle is in its intended lane, while it is in the opposite lane on the right side. Reprinted from [BOG 18].

Situation-Aware Reward: For the total reward r_t, two aspects are combined, as shown in Equation 6.2. The first component utilizes the current state of the ego vehicle, and the second one is based on the situation-aware rulebook.

$$r_t = r_{ego,t} + r_{RB,t} \tag{6.2}$$

The first component r_{ego} is shown in Equation 6.3. It consists of $r_{finish} = 10$ if the vehicle reaches the target distance but not the target lane, 60 if both are reached, and 0 otherwise. Additionally, r_{speed} is set to -1 if the speed is not within $10 - 50$ km/h, otherwise, it is 0. The trajectory length traveled in the past step is denoted by l. All values were determined by a small set of experiments.

$$r_{ego} = r_{finish} + r_{speed} l \tag{6.3}$$

For the second reward component, $r_{RB,t}$, a set of simplified traffic rules necessary for the designed scenarios is utilized. As shown in Table 6.1, the agent monitors three traffic rules, focusing on collision avoidance and adherence to road layout. By default, the agent should adhere to all rules.

Rule	Realization	LTL Formula	j	ρ_j
Avoid collisions	ψ_1	**G**(no_collision)	1	1
Stay in lane	ψ_2	**G**(in_lane)	2	0.1
Stay on road	ψ_3	**G**(no_out_road)	2	0.1

Table 6.1.: **Rule overview:** The table shows rule realizations ψ_i, LTL formulas, hierarchy levels j, and coefficients ρ_j. Reprinted from [BOG 18].

All rules can be monitored based on the temporal operator **G** from LTL, as introduced in Section 6.2.2. As shown in Equation 6.4, states breaking the rule receive a penalty of -1 per rule realization, otherwise 0. The expression $\tau_t \not\models \mathbf{G}\psi$ refers to whether the states in the generated trajectory satisfy a rule. The trajectory is considered to violate a rule if any state does not satisfy it.

$$f(\mathbf{G}\psi, \tau_t) = \begin{cases} -1, & \text{if } \tau_t \not\models \mathbf{G}\psi \\ 0, & \text{Otherwise} \end{cases} \tag{6.4}$$

Given the concrete set of rules and rule realizations from Table 6.1 and the rulebook reward as defined in Equation 6.1, $r_{RB,t}$ is expressed as follows:

$$r_{RB,t} = \rho_1 r_{collision} c_{col} + \rho_1 \rho_2 r_{in\_lane} l c_{lane} + \rho_1 \rho_2 r_{no\_out\_road} l \tag{6.5}$$

When the scenarios demand it, controlled rule exceptions become necessary to proceed. Based on ground truth through $\hat{o}_t$, the situation awareness module of the agent activates the rulebook when it approaches an obstacle. This becomes evident in Equation 6.5, where all coefficients ρ_j are then set to their values as defined in Table 6.1 instead of their default value 1. Thus, when necessary, the

agent can leave the road with only a minor negative influence on the reward in order to perform a controlled rule exception.

Curriculum Learning: The training strategy is divided into two steps. First, the agent shall learn regular driving behavior. After that, situations are introduced that require controlled traffic rule exceptions, as shown in Figure 6.1. Thus, for the first 3,000 steps, the agent is trained in a simple urban environment. Subsequently, the training is continued with scenarios that require the previously introduced traffic rule exceptions.

6.2.4. Evaluation

This section compares and analyzes the results of the presented experiments from Section 6.2.3. Two RL agents are compared, and a variety of ablation studies are performed in order to attribute the performance of the approach to the individual components. Both quantitative results for the whole training process, as well as qualitative demonstrations of how the most successful agent performs in scenarios that require controlled traffic rule exceptions, are shown.

Quantitative Evaluation

For the evaluation, two key metrics based on the vehicle's performance in avoiding obstacles are used, focusing on returning to the original lane and adhering to traffic rules: The metric *arrived distance* represents the distance the vehicle was able to travel along the s-axis in the Frenet coordinate at the end of each episode, reflecting the distance traveled along the lane. The metric *finished score* is the value ranging from 0 to 1 that quantifies the success in completing the scenario navigation task. A value of 1 denotes full success, 0.5 indicates returning to the correct longitudinal but not lateral position, and 0 is assigned otherwise. These metrics collectively assess the agent's ability to manage scenarios that require controlled traffic rule exceptions.

As existing RL models are extended with a trajectory generation component and the situation-aware reward design, four types of ablation studies are performed. Setting a baseline, the RL agents are implemented in an E2E setting with a discrete control-based action space, consisting of three possible acceleration values $\{-1, 0, 1\}$ and three possible angular velocity values $\{-1, 0, 1\}$. Evaluating the impact of trajectories, only the trajectory generation is implemented without the situation-aware reward. This means that all coefficients ρ_j are set to 1 constantly. Examining the isolated rulebook, only the situation-aware reward function is implemented, but trajectory generation is not utilized. In this case, Equation 6.4 only checks the state of the vehicle at each timestep. Finally, examining the combined effect of the approaches, both the trajectory generation and the situation-aware reward function are implemented.

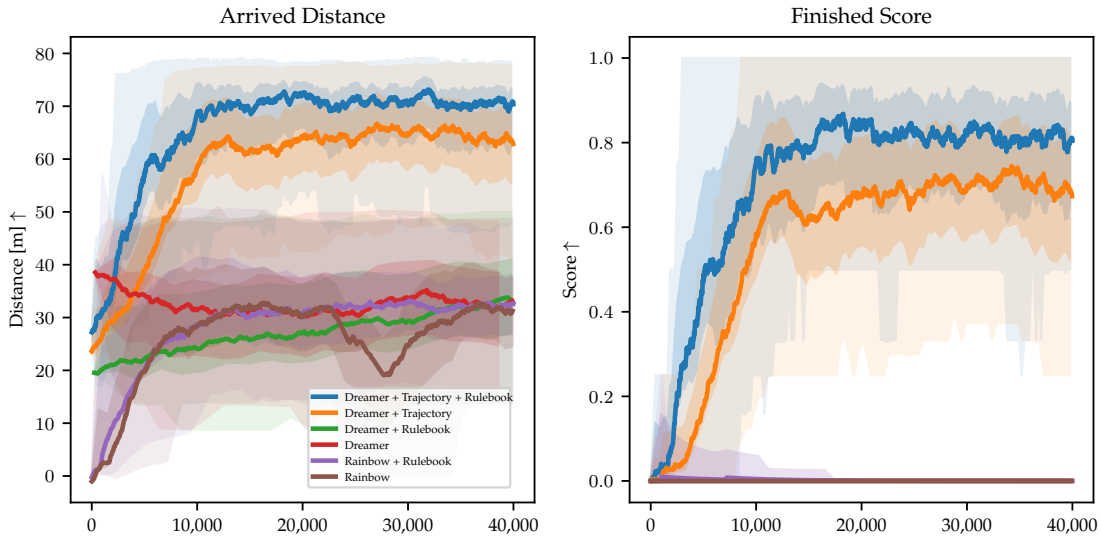
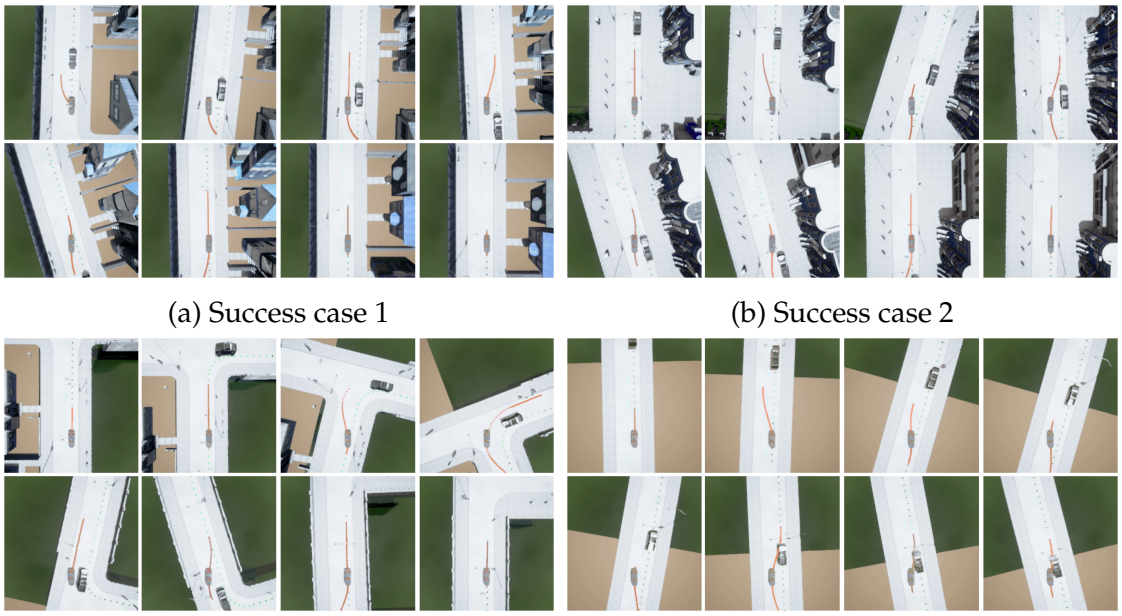


Figure 6.6.: **Evaluation:** The figures show the *arrived distance* and *finished score* metrics during training, visualizing the running average, standard deviation, and 5th and 95th percentiles over 40,000 steps. Agents were compared that worked with direct controls as their output or a *trajectory* and utilized either a conservative reward or the *rulebook*. Reprinted from [BOG 18].

A total of six different models were trained in accordance with the ablation study design, as shown in Figure 6.6. Independent of the underlying RL model, the scenarios in combination with the presented reward are too challenging for both baseline models. Including only the rulebook has no clear influence. At this point, the focus is put on the generally more capable DreamerV3 agent. In combination with the trajectory planning module, DreamerV3 consistently outperforms other methods on both metrics *arrived distance* and *finished score*. This approach exhibited a steeper learning curve, suggesting rapid adaptation to guide the vehicle efficiently. When the situation-aware reward function is additionally activated, the performance of the DreamerV3 model improves further. This shows that the reward function is beneficial for the agent's learning process, as the total performance stays consistently above the approach without the situation-awareness, while not achieving a 100% success rate in either of the metrics. In order to better understand failure cases, a qualitative evaluation is presented in the following.

Qualitative Evaluation

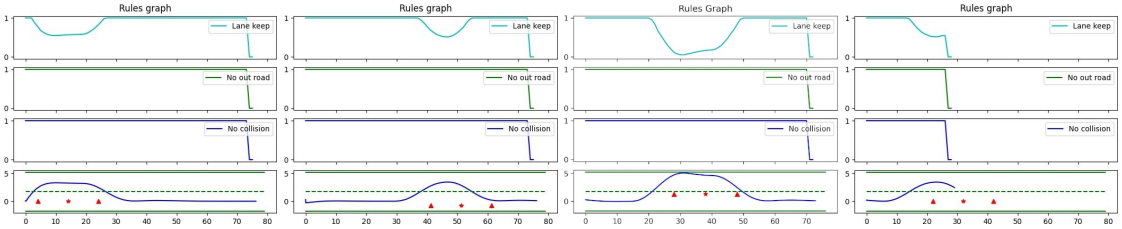
For a better understanding of individual scenarios, both the agent's driving performance and its adherence to the defined traffic rules are visualized in Figure 6.7 and Figure 6.8. Figure 6.7 shows observations o_t from the BEV camera, including the planned trajectory during an episode. This visualization illustrates the vehicle's



(a) Success case 1 (b) Success case 2

(c) Success case 3 (d) Failure case 1

Figure 6.7.: **Qualitative results:** The RL agent detects situations in which controlled rule exceptions are necessary. Trajectories that avoid obstacles are learned, changing to the oncoming lane temporarily, and returning to the default state as soon as possible. Reprinted from [STU 6].



(a) Success case 1 (b) Success case 2 (c) Success case 3 (d) Failure case 1

Figure 6.8.: **Compliance with traffic rules:** Rule adherence for the scenarios shown in Figure 6.7. The bottom row depicts the scenario, where ⋆ shows the position of the obstacle and ▲ visualizes the area in which the rulebook is activated. The top three rows show rule adherence, where 1 means full compliance and 0 violation. Adapted from [STU 6].

ability to change lanes to avoid obstacles and then return to the original lane immediately. The corresponding trajectory and traffic rule compliance graphs, plotted in Frenet coordinates, are provided in Figure 6.8. As can be seen from the quantitative results (a) - (c), the agent performs controlled rule exceptions successfully most of the time. Most failure cases occur due to too early returns to the original lane of the ego vehicle, as shown in the last scenario (d).

6.3. Conclusion

This section introduces situation-aware RL to perform controlled traffic rule exceptions in autonomous driving under the presence of previously detected object-level anomalies. Addressing **RQ4**, it demonstrates that learned trajectory planning benefits from the inclusion of previously detected anomalies into the training process. By designing a dynamic reward function that reacts to the presence of anomalies on the road, faster learning convergence and better performance are observed. This reward is based on a hierarchical rulebook [72] representing real-world traffic laws. Any given RL algorithm can be improved, as only the reward function is adapted. The act of integrating previously detected anomalies into the training process progressively diminishes their status as anomalies based on Definition 2.

This section concludes the arc of generating, detecting, and handling external anomalies in this dissertation, as shown in Figure 1.1 and Table 1.1. Summarizing, Chapter 3 provides an extensive overview of current anomaly detection methods for external anomalies and highlights several research gaps, including saturated benchmarks, a lack of multimodal anomaly detection works, and a strong reliance on semantic segmentation networks and outlier exposure during training. Subsequently, Chapter 4 presents data generation methods for all anomaly levels as introduced in Table 2.1. Focusing on object-level anomalies, Chapter 5 presents an anomaly detection technique that utilizes both camera and LIDAR data and does not require any labels during training. Based on an evaluation with the benchmark introduced in Chapter 4, it outperforms the most relevant SotA model. Finally, this Chapter 6 presents handling previously detected anomalies in an offline setting. Integrating anomalies into the training process of RL agents – eliminating their status as anomalies –, the handling of complex scenarios that require knowledge about hierarchical traffic rules is drastically improved. The data used in these chapters is based on the CARLA simulation environment, as the generation of anomalies as shown in Chapter 4 is not feasible in the real world.

The following Chapter 7 presents anomaly detection from a different perspective. After examining external anomalies in Chapters 3 through 5, it presents an anomaly detection approach for internal anomalies, providing a holistic perspective on the field of anomaly detection for autonomous driving. Different from the experiments performed in simulation, the detection of internal anomalies is evaluated on real-world datasets, closer to an open-world deployment setting.

6.3.1. Recent Advances

The field has continued to evolve since the development and publication of the work underlying this chapter [BOG 18]. Recently, multiple works have continued to address the presented topics. Salem et al. [373] focus on integrating assumptions into behavior specifications – like the rulebook utilized in this chapter – to treat insufficient specifications. These insufficiencies can occur, as specifications are

created at an early stage, where many assumptions are necessary. The authors address this through a scenario-based evaluation process. They provide a formalized method for the specification of target behaviors and model facts and relations in an ontology to infer maneuver options. The authors do not consider anomalies and only present an exemplary case study without integrating their work into an actual maneuver planner, which makes the utility and scalability of their approach difficult to judge. However, the authors suggest integrating their concept into learned approaches [341], similar to the approach presented in this chapter.

Grundt et al. [156] address the relevance of infused knowledge – as done here through the situation-aware reward function – and perform experiments in an RL setting. Similar to the method presented here, they consider scenarios where a model initially struggles. Formulating the requirement that a vehicle should always stop "in a distance of 2-15 meters to a static obstacle", the original model fails to do so under reduced road friction. Their solution is to include rainy scenarios with reduced road friction in the training data to eliminate domain-level anomalies for such scenarios. Different from the method presented in this chapter, their observation space changes based on the integrated knowledge. First, their observation space is limited to the distance between the vehicle and the object and the ego velocity, limiting the scalability of the approach. The authors then extend the observation space with the friction coefficient. Their experiments show that this additional input is sufficient for the agent to successfully perform the task. Similarly, Abouelazm et al. [3] further evaluate the impact of hierarchical rewards in RL, also in the presence of static obstacles. More similar to the method presented in this chapter, their approach utilizes a constant observation space based on sensory data and performs planning in Frenet coordinates. Their hierarchical reward consists of four components for "safety, progress, comfort, and traffic rule conformance". By integrating all four components into the reward function, their evaluation shows a reduction in collisions of 21 % in comparison to a more naive baseline reward. These results confirm the approach presented here, as both new works arrive at similar results compared to those presented in this chapter.

Patrikar et al. [341] present an approach combining a learned Imitation Learning (IL) planner with a hierarchical rule-based planner [428], where traffic rules are modeled through STL. They do not insert rules into the learning stage but use the rule-based planner during inference: A classifier estimates whether a given scenario matches the training data distribution and applies the rule-based planner as a fallback solution only in OOD scenarios. Their OOD scenarios consist of domain-level anomalies in the form of new geographic areas, where the IL planner sometimes struggles to follow the general road layout.

Following a different approach, Sinha et al. [395] use LLMs and VLMs to detect anomalies and react to them. They utilize a two-stage approach. First, an embedding-based method compares current observations to a cache of embedding vectors collected from a dataset representing normality, similar to the setting presented in Section 4.3.2. If the generated anomaly score crosses a threshold, the second stage is activated. Here, a VLM generates a textual description of the

scenes and queries an LLM, providing several trajectories as options to follow. They perform experiments with scene-level anomalies [122] and evaluate their approach on two scenarios: Stop signs on a billboard and traffic lights in the back of a pickup truck.

Summarizing, most works focus on scenarios with anomalies as described in Section 2.5, which underlines the timeliness of the topic. Leveraging VLMs and LLMs represents an especially exciting research direction, as the world knowledge embedded in these models can support the handling of atypical situations that require context that can only rarely be extracted from driving data alone. However, real-time performance on the edge and hallucinations will remain challenges in the near future. As object-level anomalies were not addressed by these works, the approach presented in this chapter remains an important cornerstone in the field of anomaly handling.

Do I contradict myself? Very well
then ... I contradict myself [...]

Walt Whitman, 1855 [452]

7. Internal Anomaly Detection

Multiple supervised student theses have contributed to this chapter [STU 8, 1]. Parts of this chapter have previously appeared in the following publication:

- D. Bogdoll et al. *Label-Free Model Failure Detection for Lidar-based Point Cloud Segmentation*. In IEEE Intelligent Vehicles Symposium (IV), 2025 [BOG 20]

7.1. Introduction

As shown in Section 2.5, anomaly detection can be viewed from both an internal and an external perspective. So far, this dissertation has extensively addressed external anomalies, i.e., occurrences in the environment surrounding the ego vehicle, in Chapters 3 - 6. This chapter changes the perspective and focuses on the detection of internal anomalies, i.e., those that have their origin within the system. Examining both internal and external anomalies leads to a more holistic understanding of root causes for model failures and is crucial, as the downstream task of driving can be equally impacted by both. Addressing the final **RQ5**, this chapter presents a method for the detection of model failures under the assumption of an open-world setting without access to ground truth labels.

As shown in Table 2.1, there are three levels of internal anomalies on the method layer. Anomalies on the input level originate from the training data itself, e.g., in the form of an imbalanced class distribution or label errors. Anomalies on the model level are introduced through design choices of selected ML model architectures which introduce "inductive model bias" [177]. Anomalies on the deployment level stem from misspecifications between the used training data and the environment where an autonomous vehicle is deployed, and can come in the form of domain shifts. As the approach presented in this chapter compares a supervised model trained on a labeled dataset with a self-supervised model that only uses the raw data, the method is able to detect input-level anomalies. As two different model architectures are used, anomalies on the model level can also be detected. Finally, the evaluation examines both data that is similar to the training data, but also datasets that represent different domains. This way, deployment-level anomalies are considered. Due to the complex nature of the levels, detected anomalies are not assigned to their respective type.

In the following, this chapter examines how internal anomalies on all levels can be detected based on real-world data closer to a deployment setting. Additionally,

it evaluates how external anomalies in the environment impact the detection approach. It does so by comparing a legacy supervised model with a self-supervised model, both trained for the same task. In the context of LIDAR point cloud segmentation, disagreements between the models are analyzed, and identified model failures are categorized.

7.2. Self-Supervised Model Failure Detection

Given a labeled dataset in autonomous driving, 70 - 85 % of the data is usually reserved for training, leaving only 15 - 30 % for both validation and testing [58, 399, 492]. These small evaluation datasets stand in stark contrast to the millions of kilometers driven on public roads during deployment [42]. As a result, many failure modes of ML models, be it in seemingly normal situations or due to external anomalies, are not captured in the evaluation sets. As large-scale unlabeled fleet data is generally available [294, 169, 258], there is an untapped potential to use this data for the detection of failure modes of ML models.

There are many active research areas dealing with the detection of failure modes. Active learning [240] is concerned with continuously enriching training data by querying samples from a set of unlabeled data points for a more efficient training process. Discrepancies between different sensor systems can also be used to query samples [216]. In error estimation, many approaches try to utilize unlabeled test sets to evaluate models [110]. Label refinement compares given labels, e.g., by an auto-labeling process, with new proposals [379]. All of these methods have in common that they utilize or compare two or more different results for the same task. However, there are no known approaches that take advantage of different training paradigms to detect internal anomalies in the form of model failures. In this chapter, the concept of complementary learning is introduced to leverage different data characteristics of the training dataset, as shown in Figure 7.3. Two models, one supervised and the other self-supervised, are trained on the task of point cloud segmentation to detect model failures based on disagreements. This approach resembles a typical deployment setting, where an existing supervised legacy model is assumed as the model under test. Beyond the labels necessary for the training of the supervised legacy model, the method presented in this chapter does not require any labels to detect internal anomalies and can thus be used with large-scale, unlabeled data recordings.

7.2.1. Related Work

The concept of comparing the outputs of two or more neural networks was already introduced in 1994 by Cohn et al., where they queried samples for active learning based on the disagreement between neural networks [98]. Since then, the variability in model predictions has been widely used to detect anomalies or

errors. Ensemble diversity is especially well studied, as it was shown to lead to better performance [113], robustness [338], uncertainty quantification [242], and detection of outliers or distribution shifts [300, 335, 437]. While no uniform metric for ensemble diversity exists, measures like disagreement of models, double fault measure, or output correlation are widely used [241]. Ensemble diversity can be implicitly enhanced via random initialization [242], noise injection or dropout, or explicitly via bagging, boosting, or stacking. Compared to ensembles, mixtures of experts [205] enforce higher model specialization and thus more component diversity, leading to better detection of OOD data [343, 342].

These approaches involve a combination of several neural networks with similar or identical architectures. Active learning is another research field interested in the detection of model failures. Here, uncertainty derived from ensembles is resource-intensive and thus only rarely used as part of a querying strategy [378, 363, 477]. Similar to ensembles, disagreements in a query-by-committee setting can be used to select samples [188]. In autonomous driving, also contradicting detections from sensors can be used as triggers, e.g., when RADAR and camera detections do not match [216]. Discrepancies between teacher and student models, typically known from knowledge distillation, can also be utilized [79]. As test sets are often small and not representative, directly estimating the accuracy of a model with only unlabeled data is of high interest [110, 344, 20, 82]. Here, simple classification tasks or approaches that estimate an overall error that cannot be applied to individual samples are typical. In some cases, generated pseudo-labels are utilized for further training steps [435, 494].

Disagreements can also be used for detecting erroneous labels. Ground truth labels in large vision datasets are often error-prone when auto-labeling processes based on large models are employed [73]. Detecting label errors with disagreements can be done by predicting a novel or refined label, and uncertainties can be generated by predicting multiple such labels [195, 379, 30]. This way, also noisy labels introduced by human errors can be detected [326].

Robustness during deployment is often achieved with sensor fusion, which, quite differently, purposefully aims to complement the weaknesses of one sensor with the strengths of another. Thus, disagreements are both typical and expected, with the aim of resolving them [236]. However, also data from a single sensor can be split into multiple streams to increase robustness. For example, object detection can be improved by combining appearance and geometry [389] or temporality and geometry [32, 248]. In performance monitoring [388, 56], but also in anomaly detection [108], typically, a primary model performing a regular task is accompanied by a learned or model-based module that provides some sort of uncertainty for the results of the regular task.

Many of the analyzed works utilizing disagreements deal with toy problems and only analyze classification tasks, which are not sufficient to truly understand the shortcomings of a model that is designed for the complex task of autonomous driving. Many works analyze model outputs of the same architecture, leveraging differences during training. However, this way, the same data characteristics

are being used during training. Existing disagreement-based approaches for the design of triggers for active learning [216] and for increased robustness during deployment [389] are most similar to the approach presented in this chapter. However, these industry demonstrations are not accompanied by scientific works and are thus hard to evaluate. Finally, no known work exists that utilizes different training paradigms to detect model failures through disagreements.

7.2.2. Method

To detect model failures without labeled validation or test sets, complementary learning for the same task is performed in order to detect model failures and classify challenging scenarios. The term *complementary learning* is introduced for the complementary use of different training paradigms, as introduced in Section 2.1, for a given purpose. For example, they can be used to detect model errors based on model predictions. The approach is demonstrated with the segmentation of LIDAR point clouds for autonomous driving into dynamic and static points, which are referred to as motion labels.

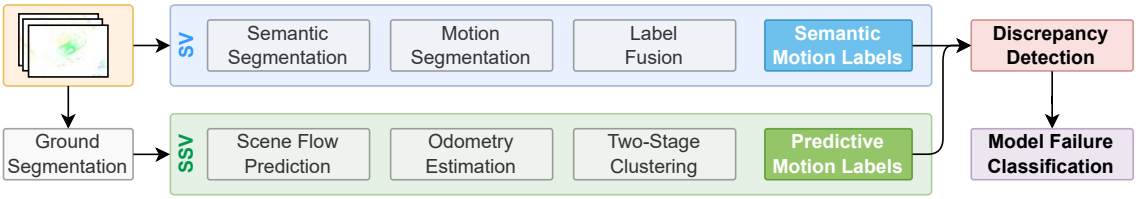


Figure 7.1.: **Overview:** Given point clouds, semantic motion labels are derived in a supervised fashion based on legacy models (blue). In addition, ground segmentation is performed and predictive motion labels are derived in a self-supervised fashion (green). Subsequently, point-wise discrepancy detection is performed, and potential model failures are classified. Reprinted from [BOG 20].

The ability to detect model failures is based on the concept that different training paradigms leverage different data characteristics from the same training dataset. As shown in Figure 7.1, first, motion labels are derived in a *supervised* and *self-supervised* fashion. Here, the first paradigm leverages human knowledge through labels, given only context from static scenes. On the other hand, the second paradigm leverages temporal information inherent in the data. Typically, these paradigms are combined either in a pre-training context [86] or with a combined loss during learning [93]. Based on a point-wise comparison, discrepancies are detected and clustered for better interpretation. Finally, an oracle examines and classifies the model failures to better understand challenging situations.

Supervised Semantic Motion Labels

Semantic motion labels are derived with a supervised semantic segmentation model [102] to determine whether a point belongs to a static or dynamic class. Some classes do not provide clear information about the motion state of the points, e.g., points assigned to the class *cyclist* at a traffic light may be static in the case of a red light and dynamic in the case of a green light. By also performing supervised motion segmentation [89], classes are further subdivided into semantic motion labels, as shown in Figure 7.2. The existence of such legacy models is expected in a typical deployment setting.

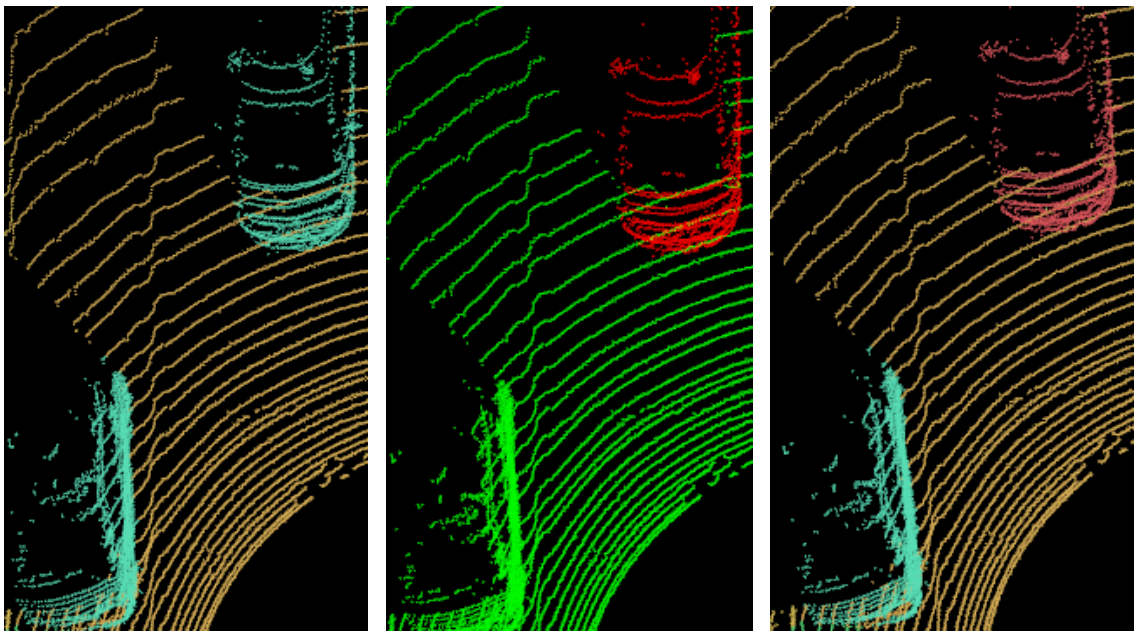


Figure 7.2.: **Supervised Semantic Motion Labels:** The left semantic segmentation [102] allows no distinction between the parked car at the bottom left and the moving car at the top right. The middle image shows a supervised motion segmentation [89], where the parked car was classified as static, and the moving car as dynamic. Finally, the right image shows the fused semantic motion labels to distinguish between static and dynamic instances of a class. Reprinted from [BOG 20].

Self-Supervised Predictive Motion Labels

To identify model failures of a supervised legacy data processing stream, a self-supervised processing stream is introduced. Based on these two streams, discrepancies can be detected. In order to predict motion labels for a given point cloud, first, the ground is filtered out [336] to focus on objects in the scene, a common pre-processing step of scene flow models [467, 306, 412, 225, 31]. Based on self-supervised flow prediction [225] of the remaining points, motion labels are derived, indicating whether a point is static or dynamic.

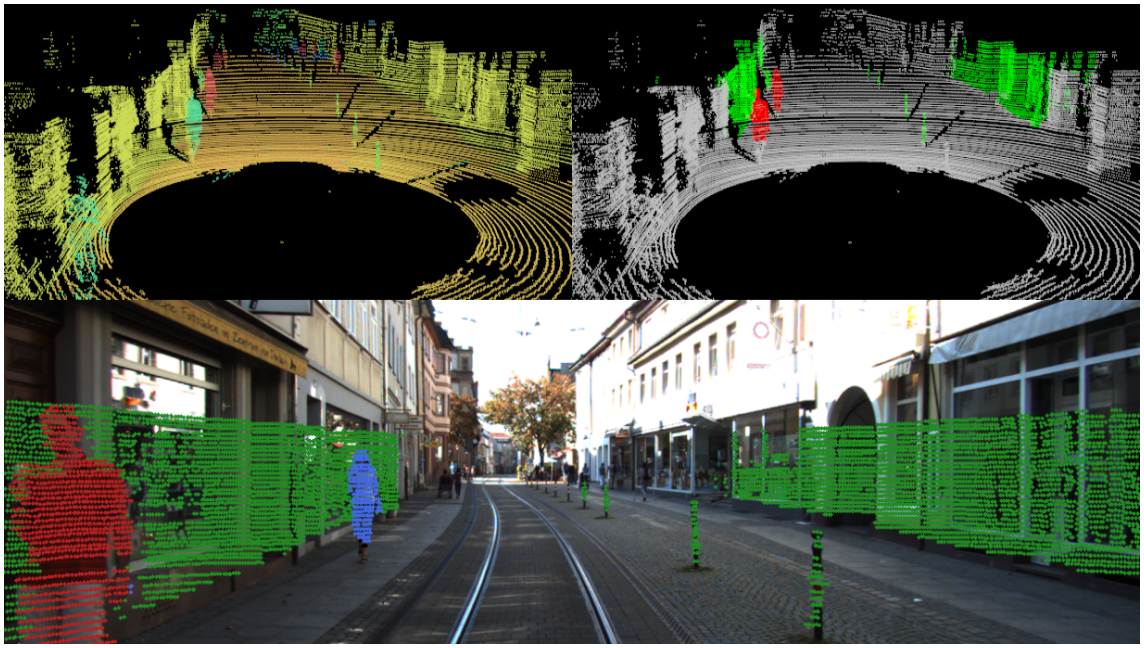


Figure 7.3.: **Model Failure Detection:** The top left point cloud shows a legacy *supervised* and the top right a *self-supervised* motion segmentation. The supervised model falsely classifies the pedestrian in the front left as static. The approach exposes this model failure, as highlighted in red in the bottom image. The color scheme is introduced in the subsection on discrepancy detection. Scene from the KITTI dataset [146]. Reprinted from [BOG 20].

As visible in the top right image in Figure 7.3, the model only performs predictions for points that are closer than 25 m and visible in the front RGB camera. The model takes consecutive point clouds as input and predicts the future motion for each LIDAR point in the form of a 3D displacement vector. The scene flow model does not distinguish between the point's own motion and the observer's ego-motion and represents the overall motion of a point between two consecutive frames. In order to derive relative displacements, it needs to be corrected for the ego-motion. This can be done by leveraging or learning odometry information [328]. For the approach to be more generalizable, here, odometry information is learned. After predicting the future point cloud $\hat{X}_{t+1} = X_t + f_t$, the learned rigid body transformation $T_{t+1 \to t}$ of an odometry model is applied, transforming the predicted point cloud back into the coordinate system of X_t. This results in the future point cloud $\tilde{X}_{t+1}$, which contains only the predicted relative motion without the ego motion. As a result, static objects line up closely with the original data of X_t, and only dynamic objects show a predicted displacement, as shown in Figure 7.5a. An analysis of the velocity values of the flow predictions shows that separating static from dynamic classes is infeasible in a point-wise fashion, as a strong overlap exists. However, a significant difference is found when considering instance-wise normalized standard deviations, as shown in Figure 7.4. As this analysis is performed with ground-truth labels, the necessity arises to form instance clusters

during inference, where labels are not present. This is achieved through a two-stage clustering process.

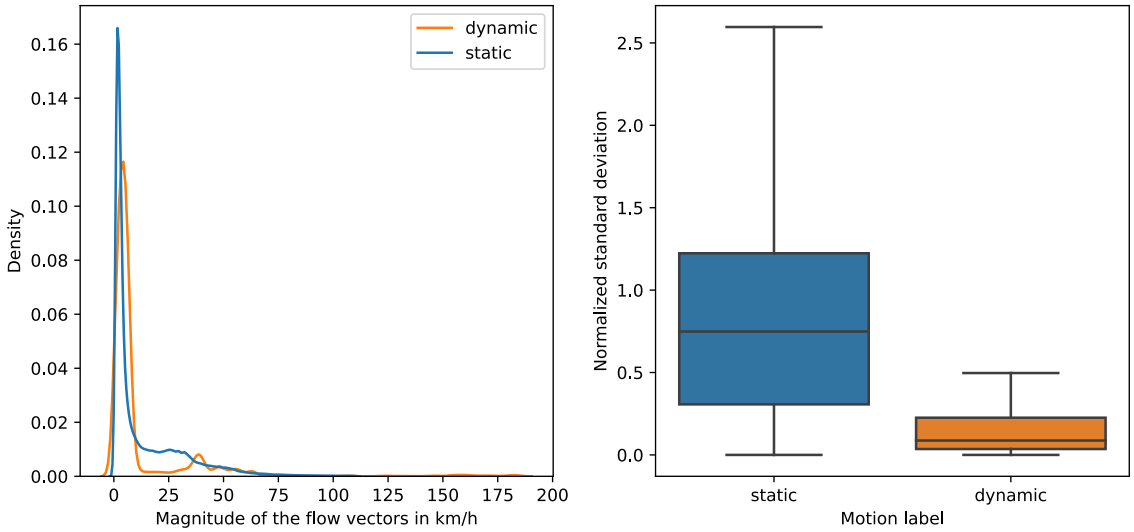


Figure 7.4.: **Self-Supervised Label Generation:** The left graph shows that the magnitude of point-wise flow vectors is insufficient to distinguish between dynamic and static points. Analyzing object instances rather than individual LIDAR points, the boxplot on the right shows that the normalized standard deviation per instance is significantly lower for dynamic instances. This allows for the distinction between dynamic and static instances. Reprinted from [STU 8].

Two-Stage Clustering during Inference: Figure 7.5a shows a scene where static objects in $\tilde{X}_{t+1}$, like parked cars, line up closely with the original data of X_t, and only dynamic objects, such as the two bicyclists, show a predicted displacement. The aim is to cluster dynamic objects in the environment based on this data, where the scene flow predictions have been compensated for ego motion. In the first stage, the DBSCAN [126] algorithm is used to spatially cluster the point cloud, as shown in Figure 7.5b. A cluster is classified as potentially dynamic if the normalized standard deviation of the cluster's velocity is below 0.12, a threshold identified through a grid search. This classification alone is insufficient, however, as static clusters are still sometimes classified as potentially dynamic due to noise and erroneous scene flow predictions. This becomes clear in Figure 7.5c, where all points shown are considered potentially dynamic after the first clustering stage.

To further reduce false positives, the potentially dynamic points are clustered in a second stage based on their flow vectors, with the same aim of distinguishing between static and dynamic clusters. Points with a similar flow are clustered irrespective of their spatial position. This changes the distribution of flow vectors per cluster, causing fewer static clusters to be incorrectly classified as dynamic. As shown in Figure 7.5c, the blue points on the left and right edges belonging to static objects now form a cluster. Black points are ignored by the DBSCAN algorithm as outliers and are considered static. Finally, the newly found clusters are classified

as dynamic if the median speed of the cluster is above $1.\overline{1}$ m/s. This threshold was chosen as it is a typical velocity profile of pedestrians, who are the slowest group of dynamic traffic participants, setting a lower limit. Slower movement is difficult to distinguish from noise, so such clusters are treated as static. This way, other moving entities, such as cars, can also be categorized as dynamic. As visible in Figure 7.5d, this second stage leads to only classifying the two moving bicyclists present in the scene as dynamic.

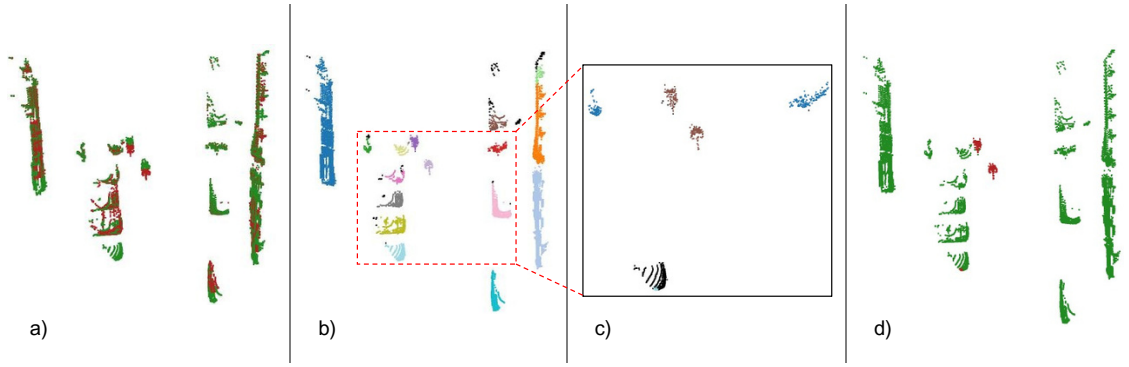


Figure 7.5.: **Self-Supervised Predictive Motion Labels:** The first image shows the original point cloud in green and the point cloud transformed by the scene flow model and compensated by the ego-motion in red. The second and third images show the result of the spatial and flow-based clustering, respectively. The fourth image shows the final predictive motion labels, with dynamic points in red and static points in green. Scenes are shown from a BEV perspective. Reprinted from [BOG 20].

Discrepancy Detection and Failure Classification

After obtaining motion labels from both the *supervised* and the *self-supervised* stream, contradictions between the labels are detected, see Figure 7.1. Only the LIDAR points per frame for which both streams predicted a label are considered. Given a semantic and a predictive motion label for each LIDAR point, there exist four categories: Points which both models deem static (green ●); points which both models deem dynamic (blue ●); points where the *supervised* stream predicts a static point and the *self-supervised* a dynamic one (red ●), and points where the *supervised* stream predicts a dynamic point and the *self-supervised* a static one (yellow ●). Examples of these categories can be found in Figures 7.7 and 7.8. Finally, instances with contradicting labels are clustered so that an oracle, such as a human expert, can classify model failures for single instances and complete scenes.

Implementation Details

For all models shown in Figure 7.1, publicly available models and model architectures are utilized to demonstrate the modularity of the approach. The supervised

semantic segmentation model SalsaNext [102] and the supervised motion segmentation model of Chen et al. [89] are trained on the KITTI-360 dataset [261, 375], as it is a large dataset that contains semantic labels, motion labels, and odometry data. Hyperparameters are taken from the original papers [102, 89]. For the remaining models, available pre-trained model weights are used. For ground segmentation, GndNet [336] is employed. For self-supervised scene flow estimation, FlowStep3D [225] is used. For the self-supervised odometry model, Deep LIDAR Odometry for Robotic Applications (DeLORA) [328] is utilized.

7.2.3. Evaluation

Model failures can occur in seemingly normal situations [178, 345, 511, 176], and models are also prone to failure in the presence of external anomalies [51, 52, 178, 427, 268] [BOG 13, 3]. For a comprehensive understanding of the presented approach, both settings are examined. In Section 7.2.3, the approach is first analyzed given regular data from the KITTI odometry dataset without labeled external anomalies. As it is unknown which scenarios might be challenging for a given model, i.e., no ground truth exists, a qualitative evaluation is performed by manually analyzing the method on over 20,000 frames. The goal is to evaluate whether the approach is able to detect model failures, such as false positives or negatives, given seemingly regular scenarios.

In Section 7.2.3, the focus is on the influence of external anomalies. Here, ground truth of external anomalies is available in the utilized CODA dataset [253], which provides anomaly labels for the KITTI [146], One Million Scenes (ONCE) [294], and nuScenes [58] datasets. A quantitative evaluation is performed to better understand the sensitivity of the method towards external anomalies. This is done by treating the output of the discrepancy detection module as a binary semantic segmentation for anomaly detection, where predictions either represent model agreements or disagreements. Disagreements are treated as indications of external anomalies, with the ground truth from CODA representing whether a point belongs to an external anomaly. The goal is to quantitatively evaluate whether model disagreements detected by the approach indicate the presence of external anomalies.

Figure 7.6 provides an overview of the distribution of the discrepancy detection results. The figure shows how often model agreements for static points (green), model agreements for dynamic points (blue), model disagreements with the supervised model classifying points as static (red), and model disagreements with the supervised model classifying points as dynamic (yellow) occurred. Here, each first solid bar represents the distribution on the KITTI dataset, while striped bars represent the different CODA subsets. For regular scenarios from the KITTI dataset, the majority of points are predicted as static by both models, and only around 5 % of the points show model disagreements. For scenarios from the CODA datasets with external anomalies, many more disagreements take place, as visible in the

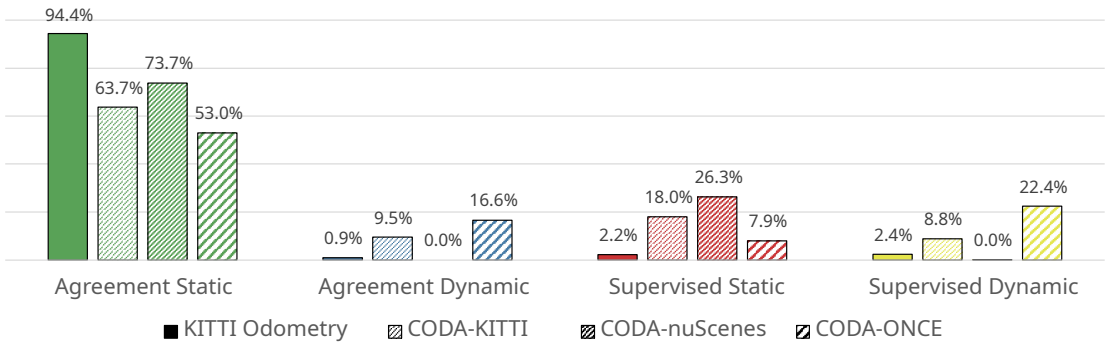


Figure 7.6.: **Discrepancy Detection:** The charts show the distribution of the four possible outcomes of discrepancy detection for four different datasets. Green and blue categories represent model agreements, and red and yellow categories represent model disagreements. Solid bars represent regular scenarios from the KITTI dataset, and striped bars represent data with external anomalies from the CODA subsets. Reprinted from [BOG 20].

last two categories. However, a large variety among the subsets of CODA can be observed. In the following, first, scenarios are examined qualitatively in which the models disagree. Subsequently, the relation between disagreements and external anomalies in the environment is quantitatively examined.

Regular Scenarios

Regular scenarios represent the majority of kilometers driven during deployment, and it is important to understand situations in which models fail. However, the evaluation under regular scenarios is challenging, as no ground truth is available. Thus, the approach was manually examined on over 20,000 frames. This includes manually inspecting projected LIDAR point clouds on the front RGB image and assessing whether the classifications of the approach are contradictory, as explained in Section 7.2.2 and shown, for example, in Figure 7.7. By human assessment, it is determined if an object that is deemed static or dynamic by the models is actually static or dynamic, which is possible as multiple frames forming a temporal scenario are available for each frame. This way, it can be determined which model is wrong in cases of disagreement, and also cases can be spotted where both models are wrong but agree.

Evaluation Data: For training, KITTI-360 and several sequences of the KITTI Odometry dataset were used. To minimize perceptual failures due to a domain shift, the qualitative evaluation is performed on the remaining 20,350 frames of the KITTI Odometry sequences 11-21. For both datasets, the provided motion-corrected LIDAR data is used, where distortions arising from the sensor's rotation during vehicle movement have already been compensated for. The datasets are

closely related, as both were captured in Karlsruhe, Germany, with a Velodyne HDL-64E LIDAR.

Evaluation: As shown in Figure 7.1, the final stage of the approach is the classification of model failure modes. The qualitative evaluation is performed by a human oracle. For visual inspection, LIDAR points mapped onto the corresponding RGB image are utilized for an improved scene understanding, as shown in Figure 7.7 with the color scheme introduced in Section 7.2.2. In most cases, both streams are correctly consistent. In the following, representative examples of detected model failures are qualitatively presented, and those that occurred frequently are highlighted, suggesting general model flaws.

Figure 7.7.: **Supervised Model Failures:** These exemplary images show model failures of the *supervised* stream, which can be detected due to contradicting outputs of the *self-supervised* model. Scenes from the KITTI dataset [146]. Reprinted from [BOG 20].

First, model failures of the supervised stream, based on the legacy models under test, are discussed. Model failures are detected through the disagreement between the two streams. Representative examples are shown in Figure 7.7. Scene 1 shows a turning car and two moving bicyclists, where one bicyclist is wrongly labeled as static by the supervised stream. Scene 2 contains two walking pedestrians that are wrongly classified as static by the supervised stream. Scene 3 shows a parked car misclassified as dynamic by the supervised stream. Scenes 4 and 5 show a car moving slowly and a car moving backward, respectively. These cases

demonstrate effectively that the approach enables the detection of regular but challenging scenarios that lead to model failures. Such model failures remain undetected in small evaluation datasets. Various weak points in each stream are found, characterized by repeated occurrence. Specifically, the supervised model under test has weaknesses in distinguishing between dynamic and static objects in specific situations, e.g., at red lights or when a car is parked directly in front of the ego vehicle. Examples of such situations are given in scenes 6 and 7 of Figure 7.7.

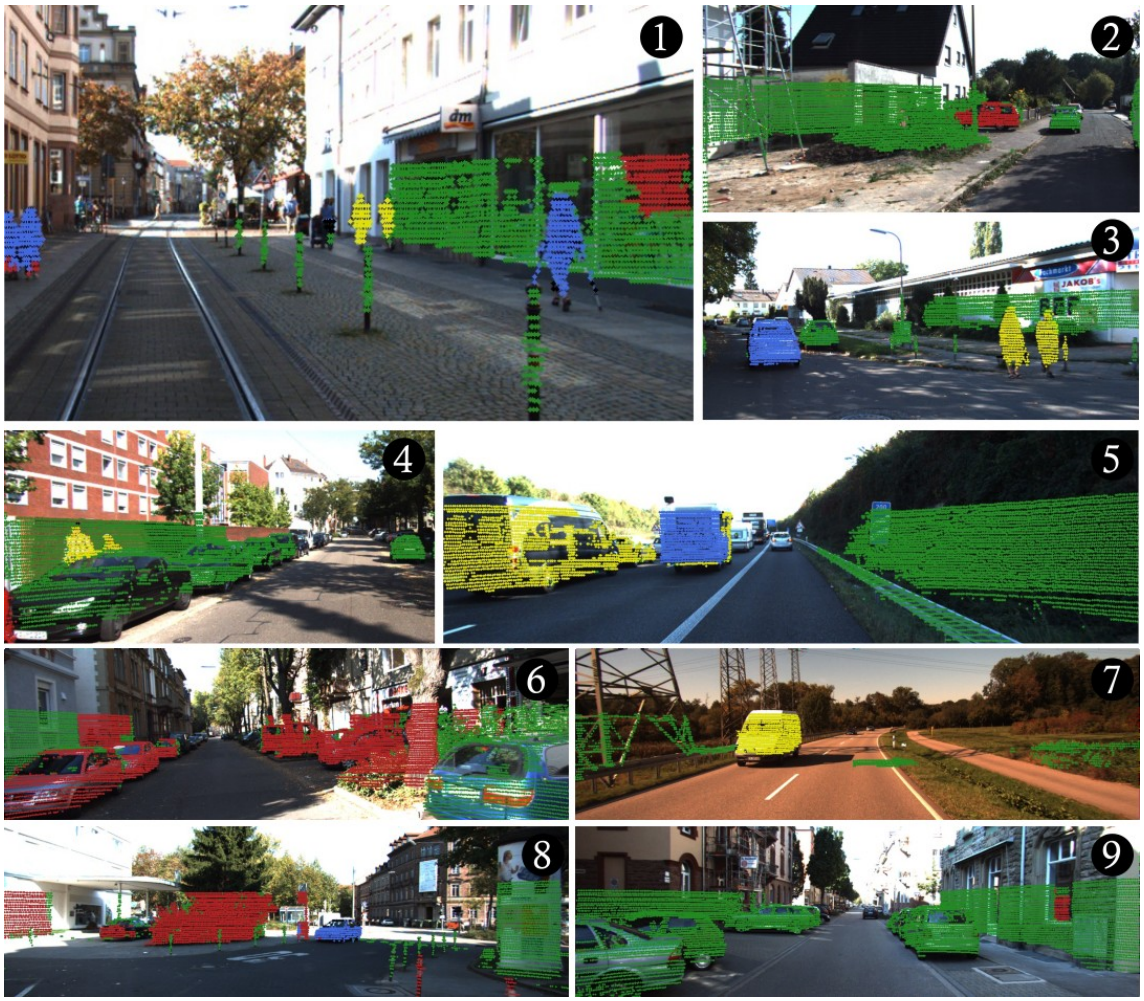


Figure 7.8.: **Self-Supervised Model Failures:** These exemplary images show failures of the *self-supervised* model, which can be detected due to contradicting outputs of the *supervised* stream. Scenes from the KITTI dataset [146]. Reprinted from [BOG 20].

While the focus of the approach presented in this chapter is on the detection of model failures induced by the supervised legacy stream, the introduced self-supervised stream can also introduce model failures. Next, scenarios where self-supervised model failures occur are analyzed, detected by correct predictions of the supervised stream. Figure 7.8 shows representative scenes. Scene 1 contains two distant pedestrians walking, wrongly classified as static by the self-supervised

stream. In scene 2, a parked car is misclassified as dynamic. Scenes 3, 4, and 5 show walking pedestrians or moving cars incorrectly classified as static. These cases demonstrate that the approach enables the detection of challenging temporal scenarios. The self-supervised stream classifies an above-average number of objects as dynamic when the ego-vehicle turns or goes over speed bumps. An example is shown in scene 6, where the vehicle turns, and in scene 8, where it drives over a speed bump. Another weak point is fast oncoming vehicles on highways, often classified as static, as seen in scene 7. Finally, a common weakness is small clusters on the side, which are incorrectly classified as dynamic, as in scene 9, where a window is classified as dynamic.

In rare cases, both models are incorrectly consistent, i.e., both streams agree, but the label is incorrect in both cases. Examples are shown in Figure 7.9. Here, the left scene shows two walking pedestrians that are incorrectly classified as static, and the right scene shows a parked car that is classified as dynamic.

Figure 7.9.: **Simultaneous Model Failures:** Examples where both streams produce model failures. Scenes from the KITTI dataset [146]. Both cases are misclassified and are, therefore, consistent. Reprinted from [STU 8].

Scenarios with external anomalies

Scenarios with external anomalies are known to lead to model failures, as first shown in Chapter 3. Based on SotA datasets, evaluating LIDAR-based anomaly detection models has been challenging. Evaluation datasets are either unavailable [459, 329] or utilize known classes but exclude them from training data [70]. Thus, first, an extension of the CODA dataset is introduced to convert existing labels in the camera data into LIDAR space. Subsequently, it is quantitatively evaluated whether model disagreements can indicate the presence of external anomalies in the environment. This is done to better understand the sensitivity of the approach in the presence of external anomalies.

Evaluation Data: For the evaluation, data from the CODA dataset [253] is utilized. The CODA dataset provides anomaly labels for objects based on three existing datasets: KITTI [146], ONCE [294], and nuScenes [58]. CODA defines an anomaly as an object that "blocks or is about to block a potential path of the self-driving vehicle" [253] and/or "does not belong to any of the common classes of autonomous driving benchmarks" [253]. While the first risk-aware definition is not always in line with the methodology of the approach presented here, where objects that block the path in front of the ego vehicle are not necessarily hard to segment, the

second one is well-suited. Novel classes are often more challenging for supervised methods compared to self-supervised approaches.

For the CODA-KITTI split, the authors of CODA manually reviewed all *misc* labels available in the ground truth and relabeled some as external anomalies according to their labeling policy. This split allows for a quantitative examination of the presented approach with only a small domain gap. For CODA-nuScenes, the authors similarly adopted available annotations in a manual process. Finally, for CODA-ONCE, they deployed an automated anomaly detection approach, making this subset the most relevant. CODA includes 1,500 scenes with a total of 5,937 external anomaly instances. Of those, 4,746 belong to the superclass *traffic_facility*, followed by 929 *vehicle* and 197 *obstruction* instances. Most *vehicle* instances, 396, can be found in CODA-KITTI.

The CODA dataset provides anomaly labels only in the form of 2D bounding boxes in image space. However, point-wise labels in 3D LIDAR space are necessary to utilize CODA for the evaluation of the approach presented in this chapter. Therefore, based on a frustum-based filter, subsequent clustering, and manual inspection, the original 2D labels from image space are transferred into refined, point-wise 3D labels that go beyond the coarse characteristic of the provided bounding boxes, as shown in Figure 7.10. Here, the different LIDAR systems utilized also become clearly visible. Due to the sparse point cloud of nuScenes, many small or distant labeled anomalies in the image space are only covered by a few or no LIDAR points.

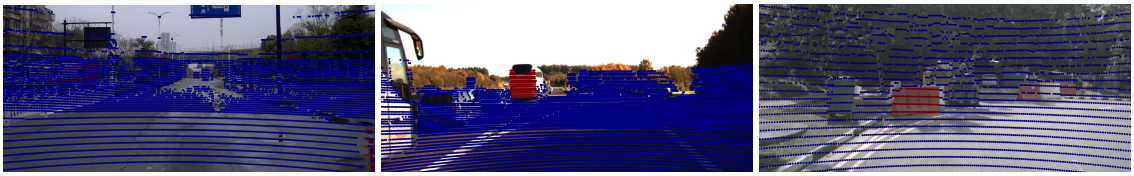


Figure 7.10.: **CODA with labeled LIDAR data:** Annotated LIDAR scenes from the three data splits ONCE [294], KITTI [146], and nuScenes [58], from left to right. Anomalies are shown in red. Reprinted from [STU 1].

Evaluation: To quantitatively evaluate the influence of external anomalies present in the environment on the model disagreement between the streams, the standard metrics mean Intersection over Union (mIoU), AP, Average Recall (AR), and F1 score in the context of binary semantic segmentation for anomaly detection are used, as shown in Tables 7.1 and 7.2. Due to the binary nature of the approach, the FPR_{95} metric cannot be computed. For a fair evaluation, all points of the LIDAR point cloud are considered, even if the approach does not label individual points, e.g., because they were filtered out during pre-processing. Such cases are counted as false negatives if an external anomaly is missed.

To better understand the suitability of CODA under introduced domain shifts, either due to new environments or due to new sensor setups, experiments are performed on the individual subsets, as shown in Table 7.1. The results clearly

Dataset	#Frames	mIoU ↑	AP ↑	AR ↑	F1 ↑
CODA	1,412	8.9	13.2	26.2	17.5
CODA-ONCE	1,034	<u>8.9</u>	**14.0**	<u>27.1</u>	**18.4**
CODA-KITTI	307	**10.9**	<u>13.3</u>	**29.0**	<u>18.3</u>
CODA-nuScenes	71	0.4	0.7	1.0	0.8

Table 7.1.: **Evaluation on CODA subsets:** Evaluation of the approach on CODA and its three subsets. Best results **bold** and second-best <u>underlined</u>. Reprinted from [BOG 20].

show that the approach struggles with the nuScenes subset, which is primarily due to the large domain shift with respect to the sensor setup. The approach is more sensitive towards anomalies for the subsets ONCE and KITTI. This is reflected in Figure 7.6, where the CODA subsets also show much higher detection rates of model failures compared to the analysis with regular scenarios. This aligns with the much higher number of external anomalies, even though the subsets reveal strongly varying behavior patterns.

Next, the sensitivity of the approach presented in this chapter towards the superclasses provided in CODA is investigated. CODA provides 43 fine-grained label categories split into the seven superclasses shown in Table 7.2. This evaluation examines whether model disagreements, when treated as anomaly detections, show different results for different types of external anomalies. As visible in Table 7.2, the method shows different levels of sensitivity given different types of external anomalies, being most sensitive to *cyclists* and objects of the *misc* class. The model performs worst on the class *animal*, which is difficult to interpret given the small number of only five instances. The *misc* class contains objects that are "unrecognizable or difficult to categorize" [253]. These results align well with the approach, where *cyclist* instances, which are hard to predict by the self-supervised stream, and *misc* instances, which are rare and thus hard to classify by the supervised stream, lead to model disagreements.

Superclass	#Instances	mIoU ↑	AP ↑	AR ↑	F1 ↑
Pedestrian	16	33.9	44.1	37.3	40.4
Cyclist	22	**41.6**	<u>58.3</u>	**49.5**	**53.5**
Vehicle	736	33.0	48.3	<u>41.0</u>	44.4
Animal	5	0.0	0.0	0.0	0.0
Traffic facility	3,360	28.6	39.9	33.9	36.7
Obstruction	125	20.5	34.0	22.9	27.4
Misc	15	<u>36.7</u>	**60.7**	37.9	<u>46.7</u>

Table 7.2.: **Evaluation on CODA superclasses:** Evaluation of the approach on external anomalies in the form of seven superclasses. Best results **bold** and second-best <u>underlined</u>. Reprinted from [BOG 20].

7.3. Conclusion

This chapter presents an approach for the detection of internal anomalies, i.e., model failures, for the segmentation of LIDAR point clouds, focusing on the differentiation between *static* and *dynamic* objects. Addressing **RQ5**, model failures of supervised legacy models are detected in an open world without access to ground truth labels. This is achieved by leveraging complementary learning paradigms to detect contradicting outputs on the same task, consisting of a *supervised* legacy stream for semantic motion labels and a *self-supervised* stream for predictive motion labels. This way, internal anomalies are detected on a scale far beyond the limited scope of a small evaluation dataset. For the evaluation, model failures are inspected in regular scenarios first. By manually analyzing over 20,000 frames qualitatively, model failures are detected in seemingly normal scenarios and categorized into frequently occurring cases. In such regular scenarios, the method categorizes around 95 % of the data as typical, which makes human analysis of the remaining 5 % feasible even for larger datasets.

In the second part of the evaluation, the sensitivity of the approach towards scenarios with external anomalies, as defined through the CODA dataset, is analyzed. In order to quantitatively examine the approach, a method to convert the coarse bounding-box labels provided by CODA in image space to finer point-wise labels in LIDAR space is introduced. The evaluation demonstrates that the approach presented in this chapter shows an increased sensitivity to hard-to-classify objects and hard-to-predict bicycles.

The approach effectively unveils internal anomalies in the form of model failures far beyond those that can be detected with small evaluation datasets. This leads to an increased understanding of the model performance in large-scale deployments, leveraging abundantly available unlabeled data. Model failures detected by the approach can be utilized to collect additional training data representing both static and temporally challenging scenarios.

As shown in Figure 1.1, after detecting internal anomalies in the form of model failures, anomaly handling can be performed subsequently with standard practices. While anomaly handling for internal anomalies is not addressed in this dissertation, issues can be handled through iterative sample selection with human inspection [BOG 31], data labeling, and model retraining [BOG 1].

7.3.1. Recent Advances

The field has continued to evolve since the development and publication of the work underlying this chapter [BOG 20]. Recently, multiple works have examined the evaluation of ML models without labeled test data for the detection of internal anomalies. Kaljavesi et al. [214] compare the outputs of a modular driving stack and an E2E driving stack and leverage disagreements between the different architectures for the detection of internal anomalies. While modular driving

stacks require labeled data for intermediate tasks, such as object detection, E2E approaches only require labels in the form of steering commands, which come for free. Additional auxiliary losses might require labeled data, but are optional. Similar to the work presented in this chapter, both systems follow different training paradigms. The authors define internal anomalies as a large difference between the two approaches at the planning stage, i.e., when the two modules propose different paths or target velocities. Similarly to the first evaluation presented here, the authors perform a qualitative evaluation due to the absence of ground truth. On a real-world test track, challenging scenarios were introduced through other road users, including overtaking maneuvers and pedestrians under occlusions. Similar to the results presented here, issues from both methods were detected, while the modular system was considered the method under test. This work demonstrates that the proposed architecture of *complementary learning* can be applied to tasks beyond environment perception and different types of architectures.

Wang et al. [439] examine root causes for errors in perception systems. Their approach requires ground truth and is not capable of detecting model discrepancies during inference, but is of relevance nonetheless. Their interventional root causal analysis aims to identify the perception module responsible for a failure. The authors examine both a camera-LIDAR fusion approach and a fusion approach more similar to the work presented here, where data from one LIDAR sensor is processed through two streams. One stream uses CenterPoint [490] and the other is based on Euclidean clustering. Similar to the quantitative evaluation results presented in this chapter, their experiments show that external anomalies, such as unknown traffic cones, contribute to model failures. In the context of the presented method in this chapter, their approach can be used to identify the model at fault once an oracle has determined a model failure. Summarizing, these approaches highlight the ongoing relevance of identifying internal anomalies in the form of model failures without access to labeled evaluation sets and show that internal and external anomalies are equally relevant.

I have made this longer than usual, only because I have not had the time to make it shorter.

Blaise Pascal, 1657 [228]

8. Conclusion and Outlook

This dissertation contributes to the field of anomaly detection in the context of autonomous driving. It presents a cohesive approach including data generation methods, anomaly detection, and a method to handle previously detected anomalies. For a holistic view, it examines both external anomalies, i.e., those occurring in the environment surrounding the ego vehicle, and internal anomalies, i.e., those with an origin within the system itself. All chapters align with a theoretical systematization of anomalies as shown in Table 1.1. All experiments on external anomalies are conducted within the CARLA simulation engine, and experiments for the detection of internal anomalies are performed on real-world datasets. None of the anomaly detection approaches require labeled training data.

8.1. Conclusion

This section revisits all RQs introduced in Section 1.2 and summarizes the contributions presented in this dissertation. After a general introduction in Chapter 1 and technical background in Chapter 2, Chapters 3 through 6 focus on external anomalies. Chapter 3 introduces an overview of anomaly detection methods, addressing RQ1:

> **RQ1**: What are the patterns of anomaly detection methods and related datasets for typical autonomous vehicle sensor modalities?

The chapter identifies patterns in the field of anomaly detection for autonomous driving for camera and LIDAR data and outlines several open research topics that are addressed in later chapters, such as saturated benchmarks, a focus on camera-based methods, and the need for outlier supervision and labeled data for semantic segmentation networks during training. Based on these insights, the generation of data with scenarios including anomalies is the focus of Chapter 4, addressing RQ2:

> **RQ2**: How can theoretical anomaly definitions from the literature be converted into datasets containing anomalies?

The chapter is based on the theoretical anomaly systematization, primarily developed by Breitenstein and Heidecker, shown in Table 2.1. In Section 4.2, it first demonstrates an approach to generate scenarios with external anomalies on

all considered anomaly levels, focusing on the generation of individual, expert-defined scenarios. Subsequently, the chapter presents a more scalable approach in Section 4.3, focusing on object-level and scenario-level anomalies. The section introduces both a well-defined normality for the training of anomaly detection methods and a challenging benchmark. The method provides data for the typical sensor modalities camera and LIDAR. The benchmark evaluation is performed in a 3D voxel space, enabling the comparison of anomaly detection methods using different sensor modalities. The utility of the benchmark is demonstrated for both types of anomalies, demonstrating how current SotA models struggle to detect external anomalies in more challenging settings compared to established benchmarks. Using this data for model training and evaluation, Chapter 5 introduces label-free anomaly detection, addressing RQ3:

> **RQ3**: How can unlabeled sensor data from multiple modalities be leveraged for the detection of object-level anomalies?

The anomaly detection method presented in Chapter 5 does not require semantic segmentation models or outlier exposure. Based on the data generation framework introduced in Chapter 4, it is trained on raw camera and LIDAR sensor data alone in a self-supervised fashion. To further improve the detection of anomalies, the approach leverages mask-based refinements of generated segmentation masks and outperforms the most relevant SotA model MNAD, as demonstrated in Table 5.2. Subsequently, Chapter 6 continues with the handling of such detected anomalies, addressing RQ4:

> **RQ4**: How can identified object-level anomalies benefit the training process of learned trajectory planning?

The chapter considers learned trajectory planning in the context of Reinforcement Learning. By integrating previously detected anomalies into the training process, they lose their status as anomalies and can be handled. The approach handles a wide variety of anomalies, categorizing them into a class of anomalies that require similar handling. As anomalies in the environment surrounding the ego vehicle can require complex maneuvering, the chapter focuses on the performance of controlled traffic-rule exceptions. The considered scenarios include a blocked lane in front of the ego vehicle that requires deviating to the opposite lane, which is only allowed under certain circumstances. The method achieves this through a situation-aware reward function, which gets triggered through the presence of identified obstacles on the road ahead. As shown in Figure 6.6, this leads to significant performance improvements, successfully demonstrating the effectiveness of leveraging previously detected anomalies during training. RL-based methods can be used in the trajectory planning of autonomous vehicles to suggest initial trajectories or to enable E2E systems. Finally, Chapter 7 changes the focus from external to internal anomalies, as both can equally influence the downstream task of driving. It introduces an anomaly detection method for internal anomalies, addressing RQ5:

> **RQ5**: How can model failures be detected in an open world without access to ground truth labels?

The chapter focuses on the detection of internal anomalies in the form of model failures. As the detection of model failures is challenging on small evaluation datasets, this chapter introduces a method for the detection of model failures in an open world during deployment. The analyzed setting deals with LIDAR point cloud segmentation, focusing on the separation of static and dynamic objects. The presented method detects model failures based on disagreements between a legacy model, trained in a supervised fashion, and a second model, trained for the same task in a self-supervised fashion. Similar to the anomaly detection approach presented in Chapter 5, this allows for the utilization of raw, unlabeled data. Based on an extensive qualitative and quantitative analysis, the chapter demonstrates that the approach successfully detects multiple failure modes, revealing internal anomalies. Such detected data points can then be integrated into training runs to eliminate these internal anomalies for future deployments.

Summarizing, this dissertation takes a holistic approach to the field of anomaly detection for autonomous driving by contributing to the generation, detection, and handling of anomalies. This is further emphasized by the analysis of internal and external anomalies, as both can equally impact the downstream task of driving. None of the anomaly detection methods presented in this dissertation require labeled data during training. The contributions of this dissertation on the detection and handling of anomalies might potentially contribute to better scaling properties of fleets of autonomous vehicles in the future. However, further research is still needed, as outlined in the following Section 8.2.

8.2. Outlook

This section outlines future trends in the field of anomaly detection for autonomous driving and provides concrete research directions that can be addressed based on the contributions presented in this dissertation. This section focuses first on external anomalies and addresses internal anomalies subsequently.

Anomaly Generation: In the context of external anomalies, many of the common anomaly detection benchmarks for autonomous driving are saturated [44, 75, 37], and the field is currently moving towards more challenging benchmarks [175, 319, 317, 470][BOG 6]. These benchmarks have shown dramatic performance decreases of current SotA anomaly detection methods, raising the need for novel approaches. In line with the presented benchmark in Chapter 4, recent benchmarks [317, 470, 244] address multiple important aspects. They focus more on a concise definition of normality, include temporal data, include both RGB and LIDAR sensor data, and provide labeled data for regular classes and anomalies to test both regular detection tasks and the detection of anomalies. Based on recent advances of VLMs,

semantically labeling detected anomalies in an open-world setting will most likely become a novel benchmarking task in the field.

A drawback of real-world anomaly detection benchmarks is that they would also need to provide a large-scale, fully labeled training dataset in order to clearly define anomalies. Labels are necessary to be fully aware of the semantic content of the training data in a real-world setting, even if the labels are not used for the training of anomaly detection methods. As this has not been achieved yet, simulation environments remain superior with respect to a clear definition of anomalies, enabling a fair comparison of anomaly detection methods. A large benefit of simulation environments is the possibility for large-scale data generation. However, the simulation-based methods introduced in Chapter 4 also have limitations. As they are knowledge-driven, the combinatorial scalability and the realism of included traffic participants are still limited. To address these constraints, automated variations of the generated scenarios can be introduced [257, 184, 302] to drastically increase the number of available scenarios. This way, a powerful combination of knowledge- and data-driven scenario generation can be achieved. In addition, simulated environments are criticized for their lack of realism with respect to generated sensor data. This aspect can be addressed with Sim2Real methods [408] or simulation environments based on generative NNs [BOG 28] to leverage the best of both worlds.

In a real-world context, the training data used to train NNs deployed in autonomous vehicles is what defines their normality. As this data is often not fully labeled, the definition of anomalies remains noisy. While this can make it challenging to detect what an expert might consider an anomaly in some cases, it does not represent a benchmark setting, where a clear definition is much more relevant for a fair comparison of anomaly detection methods. In these industrial deployment settings, advances in many fields are necessary to support the scale-up of fleets of autonomous vehicles. To better deal with the rare and unknown, advances in anomaly detection, uncertainty quantification, one- and few-shot learning, as well as open-world detection are needed.

Anomaly Detection: Advancements through novel and more challenging benchmarks will most likely launch a second wave of anomaly detection methods in autonomous driving. Novel methods are expected to leverage both LIDAR and RGB data, detect individual anomaly instances, and utilize temporal data to track anomalies. In addition, semantically classifying detected anomalies, similar to the field of open-world detection, is an expected future research topic that can be addressed with VLMs and LLMs. Using temporal data from both LIDAR and camera sensors, the method introduced in Chapter 5 contributes to this new research direction. In addition, it does not follow the typical assumptions of a labeled training dataset and known anomalies for outlier exposure during training. The method is trained in a self-supervised way without the need for labeled training data or known anomalies. This setting enables the usage of raw sensor data for a representation of normality. As outlier exposure is more and more

criticized recently [391, 317] and the use of it is already highlighted in some benchmarks [319, 75], it is likely that novel methods will focus on reducing the need for outlier exposure during training. However, it is unlikely that the field will move away from leveraging semantic segmentation models, as novel benchmarks, including the one provided in Chapter 4, provide all necessary labels to evaluate the simultaneous detection of known and unknown classes. This limits the field of anomaly detection to supervised methods. However, recent advances in ML move towards self-supervised learning to leverage the abundance of unlabeled data available.

In this spirit, self-supervised world models – as a representation of normality – are becoming more and more powerful [193, 369] and might prove useful in novel, more challenging anomaly detection benchmarks, as they allow for a scalable approach of representing normality. This is especially relevant for the usage of anomaly detection methods in real-world settings, where large-scale, unlabeled data recordings from fleets are readily available. Even though the semantic content of the training data might not be fully known, this still allows for the detection of yet unknown anomalies with respect to the used data. While Chapter 5 demonstrates the value of utilizing world models in the context of anomaly detection, the detection method is limited by the underlying world model. Situations where the world model introduced in Section 5.2 struggles to handle dynamic traffic participants directly affect the anomaly detection method presented in Section 5.3. While training a world model with an architecture with better scaling properties would have exceeded the available GPU resources, advances in the field of world models will lead to more effective models. These advances can be used in the future to improve the underlying world model.

Anomaly Handling: Generally speaking, the handling of detected anomalies is an underexplored field. While active learning for offline settings and remote assistance for online settings are well explored, the fields typically do not address anomalies explicitly. Active learning is mostly concerned with making the training process more efficient by labeling as few samples as possible, and remote assistance focuses on different support modes rather than the cause that started the remote assistance process. The method presented in Chapter 6 is one of only few works in an offline setting that explicitly integrates anomalies into the training process and leverages complex, hierarchical traffic rules in order to handle them. In order to evaluate the effect of integrating a situation-aware reward into RL, the presented method focuses primarily on simple scenarios with data from a BEV perspective. To address the scalability and real-world compatibility of the approach, more complex scenarios and raw sensor data can be examined. In more complex settings, it is also relevant to determine the relevance of detected anomalies with respect to the driving task first [BOG 25], which is also an underexplored field.

In an online setting, handling anomalies in the environment often requires a situational awareness that goes beyond the context necessary to solve typical traffic scenarios. Recent works suggest that VLMs and LLMs are well suited for the task, as such models are trained on data that goes far beyond the traffic domain.

This way, they might be able to suggest trajectories that take into account both the present traffic situation and broader knowledge from outside sources.

Internal Anomaly Detection: Finally, detecting internal anomalies in the form of model failures without labeled evaluation sets is also an underexplored field. While some works focus on the overall performance on an unlabeled test set, the detection of frames where a model fails, even on small regions of a frame, is rarely examined. The method presented in Chapter 7 leverages a self-supervised model trained on the same task as a legacy supervised model to detect model failures based on disagreements. The approach successfully detects multiple model failure modes in the setting of LIDAR point cloud segmentation. When both streams are wrong, model failures go undetected. This behavior is known and unavoidable [437, 435] and can be mitigated by deploying multiple approaches or triggers to detect challenging scenarios [216]. In addition, model failures introduced by the self-supervised stream can be reduced by training it on more data, as costly labels are not necessary. While it has been shown in the literature that the disagreement-based approach also works for the detection of internal anomalies at a planning stage by comparing a E2E and a modular driving stack, it is of interest to apply it to even more settings, such as full panoptic segmentation beyond the utilized static and dynamic classes [325], lane detection [322], or drivable-area segmentation [299], to further examine its generalizability.

In summary, benchmarks for anomaly detection methods in autonomous driving have become much more challenging, with future default benchmarks to be determined. It is expected that novel anomaly detection methods will adapt to those new benchmarks. Beyond that, the field lacks a holistic perspective, not taking into account how detected anomalies can be handled to improve the driving task itself. As it is of little relevance for the driving task whether erroneous model outputs stem from an external or an internal anomaly, the field should move toward more comprehensive model analysis to examine both external and internal anomalies in evaluation schemes. As a first step, this dissertation has made multiple contributions to the field, providing a more holistic view of anomaly detection for autonomous driving.

Appendix

A. Anomaly Generation

A supervised student thesis has contributed to this chapter [STU 2]. Parts of this chapter have previously appeared in the following publication:

- D. Bogdoll et al. *One Ontology to Rule Them All: Corner Case Scenarios for Autonomous Driving*. In European Conference on Computer Vision (ECCV) Workshop, 2023 [BOG 5]

A.1. Master Ontology

Figure A.1 shows the full master ontology, as described in Section 4.2.2 and outlined in Figure 4.2.

A.2. Scenario Ontology

At the core of each demonstrated scenario lies a scenario ontology, as shown in Figure 4.1. In the following, the construction of the scenario ontology for the *Domain Shift (a)* scenario, where a vehicle enters a foggy area, is presented. The scenario is visualized in Figure 4.3, and the scenario ontology is shown in Figure A.2. The ontology has 94 individuals, which means that 27 new and scenario-specific individuals are created, since the master ontology has 67 default individuals. Each individual which name starts with *indiv_* is a newly created part of the scenario ontology; every other individual is either a default or a constant that is already present in the master ontology. The graph starts from the top with the *Scenario* individual, which is connected to a CARLA town and a newly created *Storyboard*. Every *Storyboard* has an *Init* and a *Story*. In this particular *Init*, there are only the *Action*s, which are responsible for the position and the speed of the *EgoVehicle*, and connections to the default *EnvironmentAction*. The most interesting part of this *Scenario*, however, can be found deep within the *Story* - namely, the second *EnvironmentAction*, which creates dense fog inside the scenario. This *Action* gets triggered by the *indiv_DistanceStartTrigger*, which has a *TraveledDistanceCondition* as a *Condition*. Since this type of *Condition* is an *EntityCondition*, it requires a connection to an *Entity*, in this case the *ego_vehicle*. This *StartTrigger* gets activated when the *ego_vehicle* has traveled a certain distance. After this *Event* is executed, the *Scenario* comes to an end.

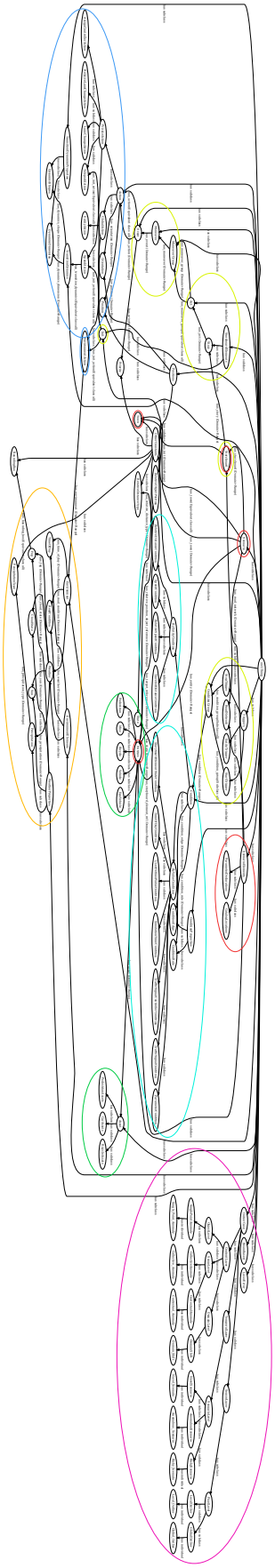


Figure A.1.: **Master Ontology:** Contains scenario and environment (red), entities (green), main scenario elements (yellow), actions (dark blue), conditions (light blue), weather and time of day (orange), and anomaly level (pink). Reprinted from [BOG 5].

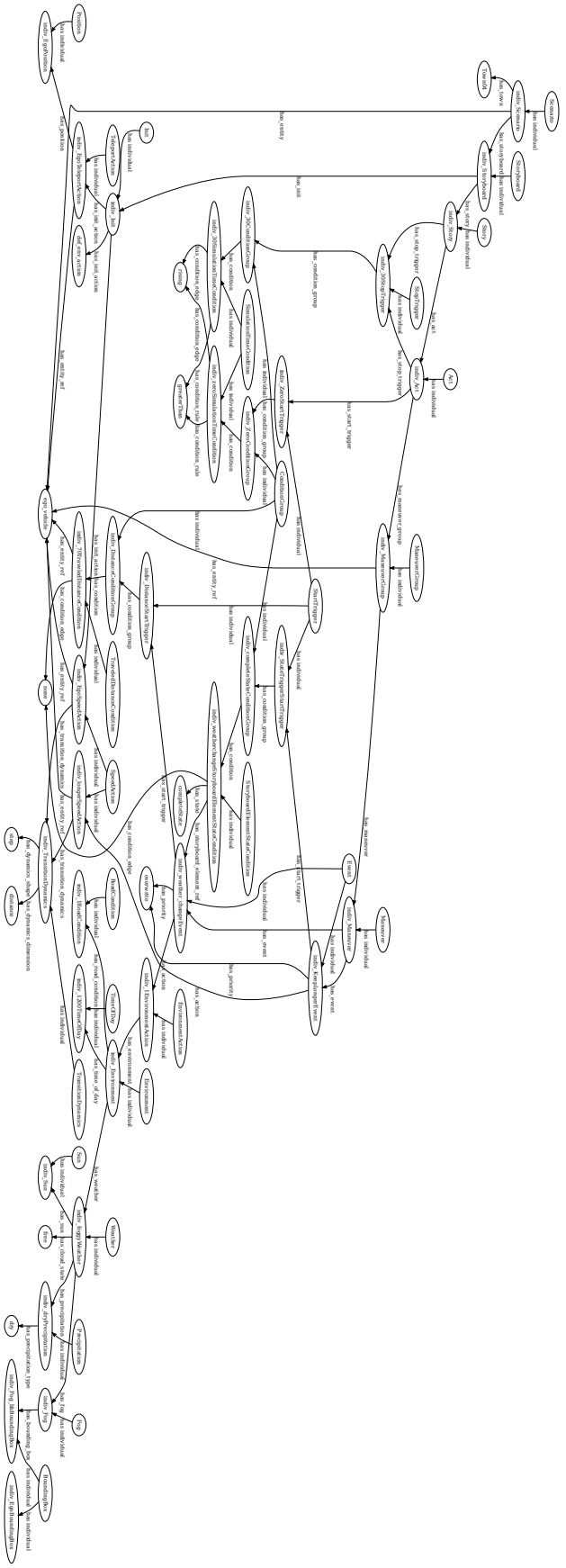


Figure A.2.: **Scenario Ontology:** Scenario describing a vehicle entering a foggy area. Reprinted from [STU 2].

List of Figures

List of Tables

Own Publications

[BOG1] Daniel Bogdoll, Rajanikant Patnaik Ananta, Abeyankar Giridharan, Isabel Moore, Gregory Stevens, and Henry X. Liu. Mcity Data Engine: Iterative Model Improvement Through Open-Vocabulary Data Selection. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2025. Accepted, to appear.

[BOG2] Daniel Bogdoll, Lukas Bosch, Tim Joseph, Helen Gremmelmaier, Yitian Yang, and J. Marius Zöllner. Exploring the Potential of World Models for Anomaly Detection in Autonomous Driving. In *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2023.

[BOG3] Daniel Bogdoll, Jasmin Breitenstein, Florian Heidecker, Maarten Bieshaar, Bernhard Sick, Tim Fingscheidt, and Marius Zöllner. Description of Corner Cases in Automated Driving: Goals and Challenges. In *IEEE/CVF International Conference on Computer Vision (ICCV) Workshop*, 2021.

[BOG4] Daniel Bogdoll, Enrico Eisen, Christin Scheib, Maximilian Nitsche, and J. Marius Zöllner. Multimodal Detection of Unknown Objects on Roads for Autonomous Driving. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2022.

[BOG5] Daniel Bogdoll, Stefani Guneshka, and J. Marius Zöllner. One Ontology to Rule Them All: Corner Case Scenarios for Autonomous Driving. In *European Conference on Computer Vision (ECCV) Workshop*, 2023.

[BOG6] Daniel Bogdoll, Iramm Hamdard, Lukas Namgyu Rößler, Felix Geisler, Muhammed Bayram, Felix Wang, Jan Imhof, Miguel de Campos, Anushervon Tabarov, Yitian Yang, Hanno Gottschalk, and J. Marius Zöllner. AnoVox: A Benchmark for Multimodal Anomaly Detection in Autonomous Driving. In *European Conference on Computer Vision (ECCV) Workshop*, 2025.

[BOG7] Daniel Bogdoll, Jonas Hendl, Felix Schreyer, Nishanth Gowda, Michael Färber, and J. Marius Zöllner. Impact, Attention, Influence: Early Assessment of Autonomous Driving Datasets. In *International Conference on Control and Robotics Engineering (ICCRE)*, 2023.

[BOG8] Daniel Bogdoll, Jan Imhof, Tim Joseph, and J. Marius Zöllner. Hybrid Video Anomaly Detection for Anomalous Scenarios in Autonomous Driving. In *British Machine Vision Conference (BMVC) Workshop*, 2024.

[BOG9] Daniel Bogdoll, Johannes Jestram, Jonas Rauch, Christin Scheib, Moritz Wittig, and J. Marius Zöllner. Compressing Sensor Data for Remote Assistance of Autonomous Vehicles using Deep Generative Models. *Conference on Neural Information Processing Systems (NeurIPS) Workshop*, 2021.

[BOG10] Daniel Bogdoll, Louis Karsch, Jennifer Amritzer, and J. Marius Zöllner. On The Impact of Replacing Private Cars with Autonomous Shuttles: An Agent-Based Approach. In *IEEE Forum for Innovative Sustainable Transportation Systems (FISTS)*, 2024.

[BOG11] Daniel Bogdoll, Patrick Matalla, Christoph Füllner, Christian Raack, Shi Li, Tobias Käfer, Stefan Orf, Marc René Zofka, Finn Sartoris, Christoph Schweikert, Thomas Pfeiffer, André Richter, Sebastian Randel, and Rene Bonk. KIGLIS: Smart Networks for Smart Cities. In *IEEE International Smart Cities Conference (ISC2)*, 2021.

[BOG12] Daniel Bogdoll, Moritz Nekolla, Tim Joseph, and J. Marius Zöllner. Quantification of Actual Road User Behavior on the Basis of Given Traffic Rules. In *IEEE Intelligent Vehicles Symposium (IV)*, 2022.

[BOG13] Daniel Bogdoll, Maximilian Nitsche, and J. Marius Zöllner. Anomaly Detection in Autonomous Driving: A Survey. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR) Workshop*, 2022.

[BOG14] Daniel Bogdoll, Noël Ollick, Tim Joseph, and J. Marius Zöllner. UMAD: Unsupervised Mask-Level Anomaly Detection for Autonomous Driving. In *British Machine Vision Conference (BMVC) Workshop*, 2024.

[BOG15] Daniel Bogdoll, Stefan Orf, Lars Töttel, and J. Marius Zöllner. Taxonomy and Survey on Remote Human Input Systems for Driving Automation Systems. In *Future of Information and Communication Conference (FICC)*, 2022.

[BOG16] Daniel Bogdoll, Shreyasha Paudel, and Tejaswi Koduri. Augmenting Real Sensor Recordings With Simulated Sensor Data, 2022. Patent US11455565B2.

[BOG17] Daniel Bogdoll, Svetlana Pavlitska, Simon Klaus, and J. Marius Zöllner. Conditioning Latent-Space Clusters for Real-World Anomaly Classification. In *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2023.

[BOG18] Daniel Bogdoll, Jing Qin, Moritz Nekolla, Ahmed Abouelazm, Tim Joseph, and J. Marius Zöllner. Informed Reinforcement Learning for Situation-Aware Traffic Rule Exceptions. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[BOG19] Daniel Bogdoll, Jonas Rauch, and J. Marius Zöllner. DLCSS: Dynamic Longest Common Subsequences. In *IEEE International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 2022.

[BOG20] Daniel Bogdoll, Finn Sartoris, Vincent Geppert, Svetlana Pavlitska, and J. Marius Zöllner. Label-Free Model Failure Detection for Lidar-based Point Cloud Segmentation. *IEEE Intelligent Vehicles Symposium (IV)*, 2025.

[BOG21] Daniel Bogdoll, Felix Schreyer, and J. Marius Zöllner. AD-datasets: A Meta-Collection of Data Sets for Autonomous Driving. In *International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)*, 2022.

[BOG22] Daniel Bogdoll, Svenja Uhlemeyer, Kamil Kowol, and J. Marius Zöllner. Perception Datasets for Anomaly Detection in Autonomous Driving: A Survey. In *IEEE Intelligent Vehicles Symposium (IV)*, 2023.

[BOG23] Daniel Bogdoll, Yitian Yang, Tim Joseph, Melih Yazgan, and J. Marius Zöllner. MUVO: A Multimodal Generative World Model for Autonomous Driving with Geometric Representations. *IEEE Intelligent Vehicles Symposium (IV)*, 2025.

[BOG24] Daniel Bogdoll, Meng Zhang, Maximilian Nitsche, and J. Marius Zöllner. Experiments on Anomaly Detection in Autonomous Driving by Forward-Backward Style Transfers. In *IEEE International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 2022.

[BOG25] Jasmin Breitenstein, Florian Heidecker, Maria Lyssenko, Daniel Bogdoll, Maarten Bieshaar, J. Marius Zöllner, Bernhard Sick, and Tim Fingscheidt. What Does Really Count? Estimating Relevance of Corner Cases for Semantic Segmentation in Automated Driving. In *IEEE/CVF International Conference on Computer Vision (ICCV) Workshop*, 2023.

[BOG26] Martin Gontscharow, Jens Doll, Albert Schotschneider, Daniel Bogdoll, Stefan Orf, Johannes Jestram, Marc René Zofka, and J. Marius Zöllner. Scalable Remote Operation for Autonomous Vehicles: Integration of Cooperative Perception and Open Source Communication. In *IEEE Intelligent Vehicles Symposium (IV)*, 2024.

[BOG27] Tejaswi Koduri, Daniel Bogdoll, Shreyasha Paudel, and Gautham Sholingar. AUREATE: An Augmented Reality Test Environment for Realistic Simulations. In *SAE World Congress Experience (WCX)*, 2018.

[BOG28] Ferdinand Muetsch, Helen Gremmelmaier, Nicolas Becker, Daniel Bogdoll, Marc René Zofka, and J. Marius Zöllner. From Model-Based to Data-Driven Simulation: Challenges and Trends in Autonomous Driving. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR) Workshop*, 2023.

[BOG29] Sven Ochs, Jens Doll, Daniel Grimm, Tobias Fleck, Marc Heinrich, Stefan Orf, Albert Schotschneider, Helen Gremmelmaier, Rupert Polley, Svetlana Pavlitska, Maximilian Zipfl, Helen Schneider, Ferdinand Mütsch, Daniel Bogdoll, Florian Kuhnt, Philip Schörner, Marc René Zofka, and J. Marius

Zöllner. One Stack to Rule them All: To Drive Automated Vehicles, and Reach for the 4th level. *arXiv:2404.02645*, 2024.

[BOG30] Hannes Reichert, Lukas Lang, Kevin Rösch, Daniel Bogdoll, Konrad Doll, Bernhard Sick, Hans-Christian Reuss, Christoph Stiller, and J. Marius Zöllner. Towards Sensor Data Abstraction of Autonomous Vehicle Perception Systems. In *IEEE International Smart Cities Conference (ISC2)*, 2021.

[BOG31] Lars Töttel, Maximilian Zipfl, Daniel Bogdoll, Marc René Zofka, and J. Marius Zöllner. Reliving the Dataset: Combining the Visualization of Road Users' Interactions with Scenario Reconstruction in Virtual Reality. In *International Conference on Intelligent Transportation Engineering (ICITE)*, 2022.

[BOG32] Julian Wörmann, Daniel Bogdoll, Christian Brunner, Etienne Bührle, Han Chen, Evaristus Fuh Chuo, Kostadin Cvejoski, Ludger van Elst, Philip Gottschall, Stefan Griesche, Christian Hellert, Christian Hesels, Sebastian Houben, Tim Joseph, Niklas Keil, Johann Kelsch, Mert Keser, Hendrik Königshof, Erwin Kraft, Leonie Kreuser, Kevin Krone, Tobias Latka, Denny Mattern, Stefan Matthes, Franz Motzkus, Mohsin Munir, Moritz Nekolla, Adrian Paschke, Stefan Pilar von Pilchau, Maximilian Alexander Pintz, Tianming Qiu, Faraz Qureishi, Syed Tahseen Raza Rizvi, Jörg Reichardt, Laura von Rueden, Alexander Sagel, Diogo Sasdelli, Tobias Scholl, Gerhard Schunk, Gesina Schwalbe, Hao Shen, Youssef Shoeb, Hendrik Stapelbroek, Vera Stehr, Gurucharan Srinivas, Anh Tuan Tran, Abhishek Vivekanandan, Ya Wang, Florian Wasserrab, Tino Werner, Christian Wirth, and Stefan Zwicklbauer. Knowledge Augmented Machine Learning with Applications in Autonomous Driving: A Survey. *arXiv:2205.04712*, 2022.

Supervised Student Theses

[STU1] Vincent Geppert. Anomaly Detection with Model Contradictions for Autonomous Driving. Bachelor Thesis, Karlsruhe Institute of Technology (KIT), 2023.

[STU2] Stefani Guneshka. Ontology-based Corner Case Scenario Simulation for Autonomous Driving. Bachelor Thesis, Karlsruhe Institute of Technology (KIT), 2022.

[STU3] Louis Karsch. Sustainability of Autonomous Vehicles: An Agent-based Simulation of the Private Passenger Sector. Master Thesis, Karlsruhe Institute of Technology (KIT), 2023.

[STU4] Simon Klaus. Anomaly Detection in the Latent Space of VAEs. Bachelor Thesis, Karlsruhe Institute of Technology (KIT), 2022.

[STU5] Noël Ollick. Camera-based Anomaly Detection with Generative World Models. Bachelor Thesis, Karlsruhe Institute of Technology (KIT), 2024.

[STU6] Jing Qin. Reinforcement Learning for Controlled Traffic Rule Exceptions. Master Thesis, Karlsruhe Institute of Technology (KIT), 2023.

[STU7] Lukas N. Rößler. Benchmarking Anomaly Detection on Camera and Lidar Data with 3D Voxel Representation. Bachelor Thesis, Karlsruhe Institute of Technology (KIT), 2023.

[STU8] Finn Sartoris. Anomaly Detection in Lidar Data by Combining Supervised and Self-Supervised Methods. Bachelor Thesis, Karlsruhe Institute of Technology (KIT), 2022.

[STU9] Marcus Schilling. Anomaly Detection in 3D Space for Autonomous Driving. Master Thesis, Karlsruhe Institute of Technology (KIT), 2022.

[STU10] Yitian Yang. 3D Voxel Reconstruction and World Model for Autonomous Driving. Master Thesis, Karlsruhe Institute of Technology (KIT), 2023.

[STU11] Yang Zheng. Anomaly Detection with World Models for Autonomous Driving. Master Thesis, University of Stuttgart, 2023.

Bibliography

[1] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent Space Autoregression for Novelty Detection. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2019.

[2] Konrad Abicht. OWL Reasoners still useable in 2023. *arXiv:2309.06888*, 2023.

[3] Ahmed Abouelazm, Jonas Michel, Helen Gremmelmaier, Tim Joseph, Philip Schörner, and J. Marius Zöllner. Balancing Progress and Safety: A Novel Risk-Aware Objective for RL in Autonomous Driving. In *IEEE Intelligent Vehicles Symposium (IV)*, 2025.

[4] Jan Ackermann, Christos Sakaridis, and Fisher Yu. Maskomaly: Zero-Shot Mask Anomaly Segmentation. In *British Machine Vision Conference (BMVC)*, 2023.

[5] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos World Foundation Model Platform for Physical AI. *arXiv:2501.03575*, 2025.

[6] Ben Agro, Quinlan Sykora, Sergio Casas, Thomas Gilles, and Raquel Urtasun. Uno: Unsupervised occupancy fields for perception and forecasting. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2024.

[7] Edgar A. Aguilar, Luigi Berducci, Axel Brunnbauer, R. Grosu, and D. Ničković. From STL Rulebooks to Rewards. *arXiv:2110.02792v1*, 2021.

[8] Mohammad Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Basura Fernando, Lars Petersson, and Lars Andersson. VIENA2: A Driving Anticipation Dataset. In *Asian Conference on Computer Vision (ACCV)*, 2018.

[9] Ali K. AlShami, Ananya Kalita, Ryan Rabinowitz, Khang Lam, Rishabh Bezbarua, Terrance Boult, and Jugal Kalita. COOOL: Challenge Of Out-Of-Label A Novel Benchmark for Autonomous Driving. *arXiv:2412.05462*, 2024.

[10] Jinwon An and Sungzoon Cho. Variational Autoencoder based Anomaly Detection using Reconstruction Probability. SNU Data Mining Center: Special Lecture on IE, 2015.

[11] Drago Anguelov. Taming the Long Tail of Autonomous Driving Challenges. https://www.youtube.com/watch?v=Q0nGo2-y0xY, 2019.

[12] AP News. A driverless car hits a person crossing against the light in China. https://apnews.com/article/china-autonomous-driving-accident-baidu-b0b4527ff355836f2df03868ff0bd0fc, 2024. Accessed: 2025-03-25.

[13] Szilárd Aradi. Survey of Deep Reinforcement Learning for Motion Planning of Autonomous Vehicles. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 2022.

[14] Szilard Aradi, Tamas Becsi, and Peter Gaspar. Policy Gradient Based Reinforcement Learning Approach for Autonomous Highway Driving. In *IEEE Conference on Control Technology and Applications (CCTA)*, 2018.

[15] Alexandre Armand, David Filliat, and Javier Ibañez-Guzman. Ontology-based context awareness for driving assistance systems. In *IEEE Intelligent Vehicles Symposium (IV)*, 2014.

[16] ASAM. ASAM OpenXOntology. https://www.asam.net/project-detail/asam-openxontology/, 2020. Accessed: 2022-02-28.

[17] ASAM. OpenSCENARIO Documentation. https://releases.asam.net/OpenSCENARIO/1.0.0/ASAM_OpenSCENARIO_BS-1-2_User-Guide_V1-0-0.html, 2020. Accessed: 2022-01-28.

[18] ASAM. ASAM OpenDRIVE. https://www.asam.net/standards/detail/opendrive/, 2022. Accessed: 2025-04-03.

[19] ASAM. ASAM OpenSCENARIO. https://www.asam.net/standards/detail/openscenario, 2022. Accessed: 2022-02-28.

[20] Christina Baek, Yiding Jiang, Aditi Raghunathan, and Zico Kolter. Agreement-on-the-Line: Predicting the Performance of Neural Networks under Distribution Shift. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[21] Gerrit Bagschik, Till Menzel, and Markus Maurer. Ontology based Scene Creation for the Development of Automated Vehicles. In *IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[22] Shuang Bai, Chao Han, and Shan An. Recognizing Anomalies in Urban Road Scenes Through Analysing Single Images Captured by Cameras on Vehicles. *Sensing and Imaging*, 2018.

[23] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hangbo Fu, and Chiew-Lan Tai. TransFusion: Robust LiDAR-Camera Fusion for 3D Object Detection with Transformers. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2022.

[24] Zhengwei Bai, Wei Shangguan, Baigen Cai, and Linguo Chai. Deep Reinforcement Learning Based High-level Driving Behavior Decision-making Model in Heterogeneous Traffic. In *Chinese Control Conference (CCC)*, 2019.

[25] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A Cookbook of Self-Supervised Learning. *arXiv:2304.12210*, 2023.

[26] Randall Balestriero and Yann LeCun. Contrastive and Non-Contrastive Self-Supervised Learning Recover Global and Local Spectral Embedding Methods. *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[27] Haris Balta, Jasmin Velagic, Walter Bosschaerts, Geert De Cubber, and Bruno Siciliano. Fast Statistical Outlier Removal Based Method for Large 3D Point Clouds of Outdoor Environments. In *IFAC PapersOnLine*, 2018.

[28] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst. In *Robotics: Science and Systems (RSS)*, 2019.

[29] Jinan Bao, Hanshi Sun, Hanqiu Deng, Yinsheng He, Zhaoxiang Zhang, and Xingyu Li. Bmad: Benchmarks for medical anomaly detection. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2024.

[30] Andreas Bär, Jonas Uhrig, Jeethesh Pai Umesh, Marius Cordts, and Tim Fingscheidt. A Novel Benchmark for Refinement of Noisy Localization Labels in Autolabeled Datasets for Object Detection. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR) Workshops*, 2023.

[31] Stefan Baur, David Emmerichs, Frank Moosmann, Peter Pinggera, Bjorn Ommer, and Andreas Geiger. SLIM: Self-Supervised LiDAR Scene Flow and Motion Segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[32] Stefan Baur, Frank Moosmann, and Andreas Geiger. LISO: Lidar-only Self-Supervised 3D Object Detection. In *European Conference on Computer Vision (ECCV)*, 2024.

[33] Philipp Becker and Gerhard Neumann. On Uncertainty in Deep State Space Models for Model-Based Reinforcement Learning. *Transactions on Machine Learning Research (TMLR)*, 2022.

[34] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[35] Rebecca Bellan. A Waymo robotaxi and a Serve delivery robot collided in Los Angeles. https://techcrunch.com/2024/12/31/a-waymo-robotaxi-and-a-serve-delivery-robot-collided-in-los-angeles/, 2024. Accessed: 2025-03-25.

[36] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning (ICML)*, 2009.

[37] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2019.

[38] Victor Besnier, Andrei Bursuc, David Picard, and Alexandre Briot. Triggering Failures: Out-Of-Distribution detection by learning from local adversarial attacks in Semantic Segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[39] Amy Beth Hanson and Betarice Dupuy. A vehicle backfiring startled a circus elephant into a Montana street. She still performed Tuesday. https://apnews.com/article/circus-elephant-escapes-butte-montana-572fdb1eef96a9c63815222788263d0d, 2024. Accessed: 2025-03-14.

[40] Petra Bevandić, Ivan Krešo, Marin Oršić, and Siniša Šegvić. Simultaneous Semantic Segmentation and Outlier Detection in Presence of Domain Shift. In *German Conference on Pattern Recognition (GCPR)*, 2019.

[41] Petra Bevandić, Ivan Krešo, Marin Oršić, and Siniša Šegvić. Dense Outlier Detection and Open-Set Recognition Based on Training with Noisy Negative Images. *Image and Vision Computing*, 2022.

[42] Richard Bishop. 60 Million Miles And Counting: Robotaxis Shift Into High Gear. https://www.forbes.com/sites/richardbishop1/2024/07/27/60-million-miles-and-counting-robotaxis-shift-into-high-gear/, 2024. Accessed: 2025-01-29.

[43] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. Fishyscapes: A Benchmark for Safe Semantic Segmentation in Autonomous Driving. In *IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2019.

[44] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. The Fishyscapes Benchmark: Measuring Blind Spots in Semantic Segmentation. *International Journal of Computer Vision (IJCV)*, 2021.

[45] Jan-Aike Bolte, Andreas Bar, Daniel Lipinski, and Tim Fingscheidt. Towards Corner Case Detection for Autonomous Driving. In *IEEE Intelligent Vehicles Symposium (IV)*, 2019.

[46] Jan-Aike Bolte, Markus Kamp, Antonia Breuer, Silviu Homoceanu, Peter Schlicht, Fabian Huger, Daniel Lipinski, and Tim Fingscheidt. Unsupervised Domain Adaptation to Improve Image Segmentation Quality Both in the Source and Target Domain. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR) Workshops*, 2019.

[47] T. Boult, D. S. Prijatelj, and W. Scheirer. *A Unifying Framework for Formal Theories of Novelty: Discussions, Guidelines, and Examples for Artificial Intelligence*, chapter A Unifying Framework for Novelty. Springer, 2024.

[48] T. E. Boult, P. A. Grabowicz, D. S. Prijatelj, R. Stern, L. Holder, J. Alspector, M. Jafarzadeh, T. Ahmad, A. R. Dhamija, C. Li, S. Cruz, A. Shrivastava, C. Vondrick, and W. J. Scheirer. A Unifying Framework for Formal Theories of Novelty:Framework, Examples and Discussion. *arXiv:2012.04226*, 2020.

[49] Terrance Boult, Przemyslaw Grabowicz, Derek Prijatelj, Roni Stern, Lawrence Holder, Joshua Alspector, Mohsen M. Jafarzadeh, Toqueer Ahmad, Akshay Dhamija, Chunchun Li, Steve Cruz, Abhinav Shrivastava, Carl Vondrick, and Scheirer Walter. Towards a Unifying Framework for Formal Theories of Novelty. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[50] Maxime Bouton, Alireza Nakhaei, David Isele, Kikuo Fujimura, and Mykel J. Kochenderfer. Reinforcement Learning with Iterative Reasoning for Merging in Dense Traffic. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2020.

[51] Jasmin Breitenstein, Jan-Aike Termöhlen, Daniel Lipinski, and Tim Fingscheidt. Systematization of Corner Cases for Visual Perception in Automated Driving. In *IEEE Intelligent Vehicles Symposium (IV)*, 2020.

[52] Jasmin Breitenstein, Jan-Aike Termöhlen, Daniel Lipinski, and Tim Fingscheidt. Corner Cases for Visual Perception in Automated Driving: Some Guidance on Detection Approaches. *arXiv:2102.05897*, 2021.

[53] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic Object Classes in Video: A High-Definition Ground Truth Database. *Pattern Recognition Letters*, 2009.

[54] Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative Interactive Environments. *arXiv:2402.15391*, 2024.

[55] Tom Bu, Xinhe Zhang, Christoph Mertz, and John M. Dolan. CARLA Simulated Data for Rare Road Object Detection. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2021.

[56] Cornelius Buerkle, Fabian Oboril, Johannes Burr, and Kay-Ulrich Scholl. Safe Perception - A Hierarchical Monitor Approach. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2022.

[57] Rahul Bulusu and Ashutosh Dhekne. Helping Autonomous Vehicles Maneuver Traffic Anomalies Using UWB. In *International Conference on Mobile Computing and Networking (MobiCom)*, 2024.

[58] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2020.

[59] Jinkang Cai, Weiwen Deng, Haoran Guang, Ying Wang, Jiangkun Li, and Juan Ding. A Survey on Data-Driven Scenario Generation for Automated Vehicle Testing. *Machines*, 2022.

[60] Mumuxin Cai, Xupeng Wang, Ferdous Sohel, and Hang Lei. Unsupervised Anomaly Detection for Improving Adversarial Robustness of 3D Object Detection Models. *Electronics*, 2025.

[61] Qi Cai, Yingwei Pan, Ting Yao, Chong-Wah Ngo, and Tao Mei. ObjectFusion: Multi-modal 3D Object Detection with Object-Centric Fusion. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[62] Anh-Quan Cao and Raoul De Charette. MonoScene: Monocular 3D Semantic Scene Completion. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2022.

[63] Yunkang Cao, Xiaohao Xu, Chen Sun, Yuqi Cheng, Zongwei Du, Liang Gao, and Weiming Shen. Segment Any Anomaly without Training via Hybrid Prompt Regularization. *IEEE Transactions on Cybernetics*, 2023.

[64] Zhong Cao, Diange Yang, Shaobing Xu, Huei Peng, Boqi Li, Shuo Feng, and Ding Zhao. Highway Exiting Planner for Automated Vehicles Using Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 2021.

[65] CARLA. CARLA Blueprint Library. https://carla.readthedocs.io/en/latest/bp_library/, 2022. Accessed: 2022-02-28.

[66] CARLA. CARLA Simulator. https://carla.org/, 2022. Accessed: 2022-02-28.

[67] CARLA. Scenario Runner Github. https://github.com/carla-simulator/scenario_runner, 2022. Accessed: 2022-02-28.

[68] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[69] Ander Carreño, Iñaki Inza, and Jose A. Lozano. Analyzing Rare Event, Anomaly, Novelty and Outlier Detection Terms under the Supervised Classification Framework. *Artificial Intelligence Review*, 2020.

[70] Jun Cen, Peng Yun, Junhao Cai, Michael Yu Wang, and Ming Liu. Open-Set 3D Object Detection. In *International Conference on 3D Vision (3DV)*, 2021.

[71] Jun Cen, Peng Yun, Shiwei Zhang, Junhao Cai, Di Luan, Michael Yu Wang, Ming Liu, and Mingqian Tang. Open-world Semantic Segmentation for LIDAR Point Clouds. In *European Conference on Computer Vision (ECCV)*, 2022.

[72] Andrea Censi, Konstantin Slutsky, Tichakorn Wongpiromsarn, Dmitry Yershov, Scott Pendleton, James Fu, and Emilio Frazzoli. Liability, Ethics, and Culture-Aware Behavior Specification using Rulebooks. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[73] Krystian Chachuła, Jakub Łyskawa, Bartłomiej Olber, Piotr Frątczak, Adam Popowicz, and Krystian Radlak. Combating Noisy Labels in Object Detection Datasets. *arXiv:2211.13993*, 2023.

[74] Antoni B. Chan and Nuno Vasconcelos. Modeling, Clustering, and Segmenting Video with Mixtures of Dynamic Textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2008.

[75] Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann, and Matthias Rottmann. SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[76] Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy Maximization and Meta Classification for Out-Of-Distribution Detection in Semantic Segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[77] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 2009.

[78] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. *arXiv:1512.03012*, 2015.

[79] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning Efficient Object Detection Models with Knowledge Distillation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[80] Haoqiang Chen, Yadong Liu, and Dewen Hu. Representation Learning for Vision-Based Autonomous Driving via Probabilistic World Modeling. *Machines*, 2025.

[81] Jianyu Chen, Shengbo Eben Li, and Masayoshi Tomizuka. Interpretable End-to-End Urban Autonomous Driving With Latent Deep Reinforcement Learning. In *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 2022.

[82] Jiefeng Chen, Frederick Liu, Besim Avci, Xi Wu, Yingyu Liang, and Somesh Jha. Detecting Errors and Estimating Accuracy on Unlabeled Data with Self-training Ensembles. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[83] Kai Chen, Yanze Li, Wenhua Zhang, Yanxin Liu, Pengxiang Li, Ruiyuan Gao, Lanqing Hong, Meng Tian, Xinhai Zhao, Zhenguo Li, Dit-Yan Yeung, Huchuan Lu, and Xu Jia. Automated Evaluation of Large Vision-Language Models on Self-driving Corner Cases. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2025.

[84] Min Chen, Zhao Dawei, Xiao Liang, Nie Yiming, and Dai Bin. UniWorld: Autonomous Driving Pre-training via World Models. *arXiv:2308.07234*, 2023.

[85] Runjian Chen, Hyoungseob Park, Bo Zhang, Wenqi Shao, Ping Luo, and Alex Wong. TREND: Unsupervised 3D Representation Learning via Temporal Forecasting for LiDAR Perception. *arXiv:2412.03054*, 2024.

[86] Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial Robustness: From Self-Supervised Pre-Training to Fine-Tuning. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2020.

[87] Valerie Chen, Man-Ki Yoon, and Zhong Shao. Task-Aware Novelty Detection for Visual-based Deep Learning in Autonomous Systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[88] Wei Chen and Leïla Kloul. An Ontology-based Approach to Generate the Advanced Driver Assistance Use Cases of Highway Traffic. In *International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KEOD)*, 2018.

[89] Xieyuanli Chen, Shijie Li, Benedikt Mersch, Louis Wiesmann, Jürgen Gall, Jens Behley, and Cyrill Stachniss. Moving Object Segmentation in 3D LiDAR Data: A Learning-based Approach Exploiting Sequential Data. *IEEE Robotics and Automation Letters (RA-L)*, 2021.

[90] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. FUTR3D: A Unified Sensor Fusion Framework for 3D Detection. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR) Workshops*, 2023.

[91] Yuntao Chen, Yuqi Wang, and Zhaoxiang Zhang. DrivingGPT: Unifying Driving World Modeling and Planning with Multi-modal Autoregressive Transformers. *arXiv:2412.18607*, 2024.

[92] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-Attention Mask Transformer for Universal Image Segmentation. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2022.

[93] Yong Cheng, Wei Wang, Lu Jiang, and Wolfgang Macherey. Self-Supervised and Supervised Joint Training for Resource-rich Machine Translation. In *International Conference on Machine Learning (ICML)*, 2021.

[94] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. TransFuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022.

[95] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[96] Glen Chou, Yunus Emre Sahin, Liren Yang, Kwesi J. Rutledge, Petter Nilsson, and Necmiye Ozay. Using Control Synthesis to Generate Corner Cases: A Case Study on Autonomous Driving. *IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems (TCAD)*, 2018.

[97] Josiah Coad, Zhiqian Qiao, and John M Dolan. Safe Trajectory Planning Using Reinforcement Learning for Self Driving. *arXiv:2011.04702*, 2020.

[98] David Cohn, Les Atlas, and Richard Ladner. Improving Generalization with Active Learning. *Machine Learning*, 1994.

[99] comma.ai. Commavq: A Dataset of Tokenized Driving Video and a GPT Model. https://github.com/commaai/commavq, 2023. Accessed: 2025-03-07.

[100] OpenScene Contributors. OpenScene: The Largest Up-to-Date 3D Occupancy Prediction Benchmark in Autonomous Driving. https://github.com/OpenDriveLab/OpenScene, 2023. Accessed: 2025-03-07.

[101] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2016.

[102] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. SalsaNext: Fast, Uncertainty-Aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving. In *International Symposium on Visual Computing (ISVC)*, 2020.

[103] Clement Creusot and Asim Munawar. Real-Time Small Obstacle Detection on Highways Using Compressive RBM Road Reconstruction. In *IEEE Intelligent Vehicles Symposium (IV)*, 2015.

[104] Erez Dagan. Solving the long-tail with e2e AI: "The revolution will not be supervised". https://wayve.ai/thinking/e2e-embodied-ai-solves-the-long-tail/, 2024. Accessed: 2025-03-14.

[105] Alaa Daoud, Corentin Bunel, and Maxime Guériau. CornerSim: A Virtualization Framework to Generate Realistic Corner-Case Scenarios for Autonomous Driving Perception Testing. *Procedia Computer Science*, 2024.

[106] Erwin de Gelder, Jan-Pieter Paardekooper, Arash Khabbaz Saberi, Hala Elrofai, Olaf Op den Camp, Steven Kraines, Jeroen Ploeg, and Bart De Schutter. Towards an Ontology for Scenario Definition for the Assessment of Automated Vehicles: An Object-Oriented Framework. *IEEE Transactions on Intelligent Vehicles (T-IV)*, 2022.

[107] Defense Advanced Research Projects Agency. DARPA Grand Chalenge 2004 - Final Report. https://www.esd.whs.mil/Portals/54/Documents/FOID/Reading%20Room/DARPA/15-F-0059_GC_2004_FINAL_RPT_7-30-2004.pdf, 2004. Accessed: 2025-03-13.

[108] Anja Delić, Matej Grcić, and Siniša Šegvić. Outlier Detection by Ensembling Uncertainty with Negative Objectness. In *British Machine Vision Conference (BMVC)*, 2024.

[109] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2009.

[110] Weijian Deng and Liang Zheng. Are Labels Always Necessary for Classifier Accuracy Evaluation? In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2021.

[111] Wenbang Deng, Kaihong Huang, Qinghua Yu, Huimin Lu, Zhiqiang Zheng, and Xieyuanli Chen. ElC-OIS: Ellipsoidal Clustering for Open-World Instance Segmentation on LiDAR Data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[112] Giancarlo Di Biase, Hermann Blum, Roland Siegwart, and Cesar Cadena. Pixel-wise Anomaly Detection in Complex Driving Scenes. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2021.

[113] Thomas G Dietterich. Ensemble Methods in Machine Learning. In *International Workshop on Multiple Classifier Systems (MCS)*, 2000.

[114] Dmitri Dolgov. The Waymo Way: Making Autonomous Driving a Reality. https://www.youtube.com/watch?v=s_wGhKBjH_U, 2024. Accessed: 2025-03-14.

[115] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[116] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning Optical Flow with Convolutional Networks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[117] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio López, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *Conference on Robot Learning (CoRL)*, 2017.

[118] Arthur Conan Doyle. *The Adventure of the Copper Beeches*. Strand Magazine, 1892.

[119] Xuefeng Du, Xin Wang, Gabriel Gozum, and Yixuan Li. Unknown-Aware Object Detection: Learning What You Don't Know from Videos in the Wild. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2022.

[120] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. VOS: Learning What You Don't Know by Virtual Outlier Synthesis. In *International Conference on Learning Representations (ICLR)*, 2022.

[121] Alexei Efros. KTH Machine Learning Seminar: The Revolution Will Not Be Supervised. https://www.csc.kth.se/cvap/cvg/ml-seminars/posts/post-6/, 2019. Accessed: 2025-04-22.

[122] Amine Elhafsi, Rohan Sinha, Christopher Agia, Edward Schmerling, Issa A. D. Nesnas, and Marco Pavone. Semantic Anomaly Detection with Large Language Models. *Autonomous Robots*, 2023.

[123] Lance Eliot. Whether Those Endless Edge Or Corner Cases Are The Long-Tail Doom For AI Self-Driving Cars. https://www.forbes.com/sites/lanceeliot/2021/07/13/whether-those-endless-edge-or-corner-cases-are-the-long-tail-doom-for-ai-self-driving-cars/, 2022. Accessed: 2025-03-28.

[124] Ashok Elluswamy. Foundation Models for Autonomy. Talk at CVPR Workshop on Autonomous Driving, 2023.

[125] Ashok Elluswamy. Occupancy Networks. Talk at CVPR Workshop on Autonomous Driving, 2023.

[126] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 1996.

[127] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision (IJCV)*, 2009.

[128] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. RangeDet: In Defense of Range View for LiDAR-based 3D Object Detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[129] Jianwu Fang, Jiahuan Qiao, Jie Bai, Hongkai Yu, and Jianru Xue. Traffic Accident Detection via Self-Supervised Consistency Learning in Driving Scenarios. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 2022.

[130] Abdur R. Fayjie, Sabir Hossain, Doukhi Oualid, and Deok-Jin Lee. Driverless Car: Autonomous Driving Using Deep Reinforcement Learning in Urban Environment. In *International Conference on Ubiquitous Robots (UR)*, 2018.

[131] Árpád Fehér, Szilárd Aradi, Ferenc Hegedüs, Tamás Bécsi, and Péter Gáspár. Hybrid DDPG Approach for Vehicle Motion Planning. In *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2019.

[132] Tuo Feng, Wenguan Wang, and Yi Yang. A Survey of World Models for Autonomous Driving. *arXiv:2501.11260*, 2025.

[133] Fishyscapes. Results - The Fishyscapes Benchmark. https://fishyscapes.com/results, 2022.

[134] Mike Flemming. Tesla Model 3 owner walks away uninjured after crashing into overturned truck on highway. https://driveteslacanada.ca/model-3/tesla-model-3-owner-walks-away-uninjured-after-crashing-into-overturned-truck-on-highway/, 2020. Accessed: 2025-03-25.

[135] Ralph Foorthuis. On the Nature and Types of Anomalies: A Review of Deviations in Data. *International Journal of Data Science and Analytics (JDSA)*, 2021.

[136] Gianni Franchi, Xuanlong Yu, Andrei Bursuc, Angel Tena, Rémi Kazmierczak, Séverine Dubuisson, Emanuel Aldea, and David Filliat. MUAD: Multiple Uncertainties for Autonomous Driving, a benchmark for multiple uncertainty types and tasks. In *British Machine Vision Conference (BMVC)*, 2022.

[137] Simone Fuchs, Stefan Rass, Bernhard Lamprecht, and Kyandoghere Kyamakya. A Model for Ontology-Based Scene Description for Context-Aware Driver Assistance Systems. In *International Conference on Ambient Media and Systems (AMBI-SYS)*, 2010.

[138] Führerscheine.de. Achtung durchgezogene Linie – Überholen Verboten! https://www.fuehrerscheine.de/bussgeldkatalog/durchgezogene-linie/, 2023. Accessed: 2023-09-15.

[139] Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[140] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *International Conference on Machine Learning (ICML)*, 2016.

[141] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete Dropout. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[142] Silvio Galesso, Max Argus, and Thomas Brox. Far Away in the Deep Space: Dense Nearest-Neighbor-Based Out-of-Distribution Detection. In *IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2023.

[143] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A Generalizable Driving World Model with High Fidelity and Versatile Controllability. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[144] Zeyu Gao, Yao Mu, Ruoyan Shen, Chen Chen, Yangang Ren, Jianyu Chen, Shengbo Eben Li, Ping Luo, and Yanfeng Lu. Enhance Sample Efficiency and Robustness of End-to-end Urban Autonomous Driving via Semantic Masked World Model. In *Conference on Neural Information Processing Systems (NeurIPS) Workshops*, 2022.

[145] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[146] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2012.

[147] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, Tiffany Fernandez, Martin Jänicke, Sudesh Mirashi, Chiragkumar Savani, Martin Sturm, Oleksandr Vorobiov, Martin Oelker, Sebastian Garreis, and Peter Schuberth. A2D2: Audi Autonomous Driving Dataset. *arXiv:2004.06320*, 2020.

[148] Lei Gong, Yu Zhang, Yingqing Xia, Yanyong Zhang, and Jianmin Ji. SDAC: A Multimodal Synthetic Dataset for Anomaly and Corner Case Detection in Autonomous Driving. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

[149] Matej Grcić, Petra Bevandić, Zoran Kalafatić, and Siniša Šegvić. Dense Anomaly Detection by Robust Learning on Synthetic Negative Data. *arXiv:2112.12833*, 2021.

[150] Matej Grcić, Petra Bevandić, Zoran Kalafatić, and Siniša Šegvić. Dense Out-of-Distribution Detection by Robust Learning on Synthetic Negative Data. *Sensors*, 2024.

[151] Matej Grcić., Petra Bevandić., and Siniša Segvić. Dense Open-set Recognition with Synthetic Outliers Generated by Real NVP. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2021.

[152] Matej Grcić, Petra Bevandić, and Siniša Šegvić. DenseHybrid: Hybrid Anomaly Detection for Dense Open-Set Recognition. In *European Conference on Computer Vision (ECCV)*, 2022.

[153] Matej Grcić, Josip Šarić, and Siniša Šegvić. On Advantages of Mask-level Recognition for Outlier-aware Segmentation. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR) Workshops*, 2023.

[154] Ross Greer and Mohan Trivedi. Towards Explainable, Safe Autonomous Driving with Language Embeddings for Novelty Identification and Active Learning: Framework and Experimental Analysis with Real-World Data Sets. *arXiv:2402.07320*, 2024.

[155] Frank E. Grubbs. Procedures for Detecting Outlying Observations in Sample. *Technometrics*, 1969.

[156] Dominik Grundt, Astrid Rakow, Philipp Borchers, and Eike Möhlmann. What Does AI Need to Know to Drive: Testing Relevance of Knowledge. *Science of Computer Programming (SCP)*, 2025.

[157] Jingqiu Guo, Senlin Cheng, and Yangzexi Liu. Merging and Diverging Impact on Mixed Traffic of Regular and Autonomous Vehicles. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 2021.

[158] Krishnam Gupta, Syed Ashar Javed, Vineet Gandhi, and K. Madhava Krishna. MergeNet: A Deep Net Architecture for Small Obstacle Discovery. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[159] Fredrik K. Gustafsson, Martin Danelljan, and Thomas B. Schon. Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR) Workshops*, 2020.

[160] David Ha and Jürgen Schmidhuber. Recurrent World Models Facilitate Policy Evolution. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[161] Ian Hacking. *The Emergence of Probability: A Philosophical Study of Early Ideas about Probability, Induction and Statistical Inference*. Cambridge University Press, 2 edition, 2016.

[162] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to Control: Learning Behaviors by Latent Imagination. In *International Conference on Learning Representations (ICLR)*, 2020.

[163] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning Latent Dynamics for Planning from Pixels. In *International Conference on Machine Learning (ICML)*, 2019.

[164] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with Discrete World Models. In *International Conference on Learning Representations (ICLR)*, 2021.

[165] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering Diverse Domains through World Models. *arXiv:2301.04104*, 2023.

[166] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 2025.

[167] David Haldimann, Hermann Blum, Roland Siegwart, and Cesar Cadena. This Is Not What I Imagined: Error Detection for Semantic Segmentation through Visual Dissimilarity. *arXiv:1909.00676*, 2019.

[168] Shadi Hamdan and Fatma Güney. CarFormer: Self-Driving with Learned Object-Centric Representations. In *European Conference on Computer Vision (ECCV)*, 2024.

[169] Jianhua Han, Xiwen Liang, Hang Xu, Kai Chen, Lanqing Hong, Jiageng Mao, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Xiaodan Liang, and Chunjing Xu. SODA10M: A Large-Scale 2D Self/Semi-Supervised Object Detection Dataset for Autonomous Driving. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[170] Niklas Hanselmann, Katrin Renz, Kashyap Chitta, Apratim Bhattacharyya, and Andreas Geiger. KING: Generating Safety-Critical Driving Scenarios for Robust Imitation via Kinematics Gradients. In *European Conference on Computer Vision (ECCV)*, 2022.

[171] Kunkun Hao, Wen Cui, Lu Liu, Yuxi Pan, and Zijiang Yang. Integrating Data-Driven and Knowledge-Driven Methodologies for Safety-Critical Scenario Generation in Autonomous Vehicle Validation. In *IEEE International Conference on Software Quality, Reliability, and Security Companion (QRS-C)*, 2024.

[172] Kunkun Hao, Lu Liu, Wen Cui, Jianxing Zhang, Songyang Yan, Yuxi Pan, and Zijiang Yang. Bridging Data-Driven and Knowledge-Driven Approaches for Safety-Critical Scenario Generation in Automated Vehicle Validation. *arXiv:2311.10937*, 2023.

[173] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[174] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2016.

[175] Lars Heckler-Kram, Jan-Hendrik Neudeck, Ulla Scheler, Rebecca König, and Carsten Steger. The MVTec AD 2 Dataset: Advanced Scenarios for Unsupervised Anomaly Detection. *arXiv:2503.21622*, 2025.

[176] Florian Heidecker, Maarten Bieshaar, and Bernhard Sick. Corner Case Definition in Machine Learning Processes for the Perception of Highly Automated Driving. *AI Perspectives & Advances (AIPP)*, 2024.

[177] Florian Heidecker, Maarten Bieshaar, and Bernhard Sick. Corner Cases in Machine Learning Processes. *AI Perspectives & Advances (AIPP)*, 2024.

[178] Florian Heidecker, Jasmin Breitenstein, Kevin Rösch, Jonas Löhdefink, Maarten Bieshaar, Christoph Stiller, Tim Fingscheidt, and Bernhard Sick. An Application-Driven Conceptualization of Corner Cases for Perception in Highly Automated Driving. In *IEEE Intelligent Vehicles Symposium (IV)*, 2021.

[179] Florian Heidecker, Abdul Hannan, Maarten Bieshaar, and Bernhard Sick. Towards Corner Case Detection by Modeling the Uncertainty of Instance Segmentation Networks. In *International Conference on Pattern Recognition)ICPR) Workshops*, 2021.

[180] Jon C. Helton and David E. Burmaster. Guest editorial: treatment of aleatory and epistemic uncertainty in performance assessments for complex systems. *Reliability Engineering & System Safety*, 1996.

[181] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling Out-of-Distribution Detection for Real-World Settings. In *International Conference on Machine Learning (ICML)*, 2022.

[182] Dan Hendrycks and Kevin Gimpel. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[183] Mario Herger. Waymo Confused Behind A Trailer With a Tree. https://thelastdriverlicenseholder.com/2024/05/14/waymo-confused-behind-a-trailer-with-a-tree/, 2024. Accessed: 2025-03-14.

[184] Martin Herrmann, Christian Witt, Laureen Lake, Stefani Guneshka, Christian Heinzemann, Frank Bonarens, Patrick Feifel, and Simon Funke. Using ontologies for dataset engineering in automotive AI applications. In *Design, Automation and Test in Europe Conference (DATE)*, 2022.

[185] Georg Hess, Johan Jaxing, Elias Svensson, David Hagerman, Christoffer Petersson, and Lennart Svensson. Masked Autoencoder for Self-Supervised Pre-training on Lidar Point Clouds. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2023.

[186] Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining Improvements in Deep Reinforcement Learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[187] Irina Higgins and Mihaela Rosca. Frontiers in Deep Learning: Unsupervised Representation Learning. https://storage.googleapis.com/deepmind-media/UCLxDeepMind_2020/L10%20-%20UCLxDeepMind%20DL2020.pdf, 2020.

[188] Hideitsu Hino and Shinto Eguchi. Active Learning by Query by Committee with Robust Divergences. *Information Geometry*, 2023.

[189] Carl-Johan Hoel, Krister Wolff, and Leo Laine. Automated Speed and Lane Change Decision Making Using Deep Reinforcement Learning. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[190] Anthony Hu. *Neural World Models for Computer Vision*. PhD thesis, Wolfson College, 2023.

[191] Anthony Hu, Gianluca Corrado, Nicolas Griffiths, Zak Murez, Corina Gurau, Hudson Yeo, Alex Kendall, Roberto Cipolla, and Jamie Shotton. Model-Based Imitation Learning for Urban Driving. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[192] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A Generative World Model for Autonomous Driving. *arXiv:2309.17080*, 2023.

[193] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A Generative World Model for Autonomous Driving, 2023.

[194] Yeping Hu, Alireza Nakhaei, Masayoshi Tomizuka, and Kikuo Fujimura. Interaction-aware Decision Making with Adaptive Strategies under Merging Scenarios. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[195] Zibo Hu, Kun Gao, Xiaodian Zhang, Junwei Wang, Hong Wang, and Jiawei Han. Probability Differential-Based Class Label Noise Purification for Object Detection in Aerial Images. *IEEE Geoscience and Remote Sensing Letters (GRSL)*, 2022.

[196] Keli Huang, Botian Shi, Xiang Li, Xin Li, Siyuan Huang, and Yikang Li. Multi-Modal Sensor Fusion for Auto Driving Perception: A Survey. *arXiv:2202.02703*, 2022.

[197] Lu Huang, Huawei Liang, Biao Yu, Bichun Li, and Hui Zhu. Ontology-Based Driving Scene Modeling, Situation Assessment and Decision Making for Autonomous Vehicles. In *Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, 2019.

[198] Po-Yu Huang, Wan-Ting Hsu, Chun-Yueh Chiu, Ting-Fan Wu, and Min Sun. Efficient Uncertainty Estimation for Semantic Segmentation in Videos. In *European Conference on Computer Vision (ECCV)*, 2018.

[199] Eyke Hüllermeier and Willem Waegeman. Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods. *Machine Learning*, 2021.

[200] Michael Hülsen, J. Marius Zöllner, and Christian Weiss. Traffic intersection situation description ontology for advanced driver assistance. In *IEEE Intelligent Vehicles Symposium (IV)*, 2011.

[201] Britta Hummel. *Description Logic for Scene Understanding at the Example of Urban Road Intersections*. PhD thesis, Karlsruhe Institute of Technology (KIT), 2009.

[202] Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, Yin Zhou, James Guo, Dragomir Anguelov, and Mingxing Tan. EMMA: End-to-End Multimodal Model for Autonomous Driving. *arXiv:2410.23262*, 2024.

[203] Jeeho Hyun, Sangyun Kim, Giyoung Jeon, Seung Hwan Kim, Kyunghoon Bae, and Byung Jun Kang. ReConPatch: Contrastive patch representation learning for industrial anomaly detection. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2024.

[204] INSA Rouen Normandie. CornerSet. https://nuage.insa-rouen.fr/index.php/s/wWkLy8gB7SgwF2N?path=CornerSet_Object_level_06_11_2023, 2023. Accessed: 2025-03-16.

[205] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive Mixtures of Local Experts. *Neural Computation*, 1991.

[206] Maximilian Jaritz, Raoul de Charette, Marin Toromanoff, Etienne Perot, and Fawzi Nashashibi. End-to-End Race Driving with Deep Reinforcement Learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[207] Tianchen Ji, Sri Theja Vuppala, Girish Chowdhary, and Katherine Driggs-Campbell. Multi-Modal Anomaly Detection for Unstructured and Uncertain Environments. In *Conference on Robot Learning (CoRL)*, 2020.

[208] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think Twice before Driving: Towards Scalable Decoders for End-to-End Autonomous Driving. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2023.

[209] Jielin Jiang, Jiale Zhu, Muhammad Bilal, Yan Cui, Neeraj Kumar, Ruihan Dou, Feng Su, and Xiaolong Xu. Masked Swin Transformer Unet for Industrial Anomaly Detection. *IEEE Transactions on Industrial Informatics*, 2022.

[210] Longlong Jing and Yingli Tian. Self-Supervised Visual Feature Learning With Deep Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

[211] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *European Conference on Computer Vision (ECCV)*, 2016.

[212] Michael Jordan. *I Can't Accept Not Trying: Michael Jordan on the Pursuit of Excellence*. Harper San Francisco, 1994.

[213] Sanghun Jung, Jungsoo Lee, Daehoon Gwak, Sungha Choi, and Jaegul Choo. Standardized Max Logits: A Simple yet Effective Approach for Identifying Unexpected Road Obstacles in Urban-Scene Segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[214] Gemb Kaljavesi, Xiyan Su, and Frank Diermeyer. Integrating End-to-End and Modular Driving Approaches for Online Corner Case Detection in Autonomous Driving. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2024.

[215] Andrej Karpathy. Tesla Autonomy Day. https://youtu.be/Ucp0TTmvqOE?t=8671, 2019. Accessed: 2022-06-15.

[216] Andrej Karpathy. Tesla Keynote: CVPR 2021 Workshop on Autonomous Driving. https://www.youtube.com/watch?v=g6bOwQdCJrc, 2021. Accessed: 2024-06-13.

[217] Dhanoop Karunakaran, Stewart Worrall, and Eduardo Nebot. Efficient statistical validation with edge cases to evaluate Highly Automated Vehicles. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2020.

[218] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. In *British Machine Vision Conference (BMVC)*, 2017.

[219] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[220] Tarasha Khurana, Peiyun Hu, David Held, and Deva Ramanan. Point Cloud Forecasting as a Proxy for 4D Occupancy Forecasting. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2023.

[221] Hoon Kim, Kangwook Lee, Gyeongjo Hwang, and Changho Suh. Crash to Not Crash: Learn to Identify Dangerous Vehicles Using a Simulator. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[222] Seulbae Kim. Multiple issues in the Behavior Agent. https://github.com/carla-simulator/carla/issues/5398, 2022. Accessed: 2024-03-07.

[223] Seung Wook Kim, , Jonah Philion, Antonio Torralba, and Sanja Fidler. DriveGAN: Towards a Controllable High-Quality Neural Simulation. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2021.

[224] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[225] Yair Kittenplon, Yonina C. Eldar, and Dan Raviv. FlowStep3D: Model Unrolling for Self-Supervised Scene Flow Estimation. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2021.

[226] Marius Kloetzer and Calin Belta. A Fully Automated Framework for Control of Linear Systems from Temporal Logic Specifications. *IEEE Transactions on Automatic Control*, 2008.

[227] Florian Klueck, Yihao Li, Mihai Nica, Jianbo Tao, and Franz Wotawa. Using Ontologies for Test Suites Generation for Automated and Autonomous Driving Functions. In *IEEE International Symposium on Software Reliability Engineering (ISSRE) Workshops*, 2018.

[228] Elizabeth Knowles. *Oxford Dictionary of Quotations*. Oxford University Press, 2009.

[229] W. Bradley Knox, Alessandro Allievi, Holger Banzhaf, Felix Schmitt, and Peter Stone. Reward (Mis)design for autonomous driving. *Artificial Intelligence*, 2023.

[230] Lingdong Kong, Youquan Liu, Runnan Chen, Yuexin Ma, Xinge Zhu, Yikang Li, Yuenan Hou, Yu Qiao, and Ziwei Liu. Rethinking Range View Representation for LiDAR Segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[231] Philip Koopman. Anatomy of a Robotaxi Crash: Lessons from the Cruise Pedestrian Dragging Mishap. In *International Conference on Computer Safety, Reliability and Security (SafeComp)*, 2024.

[232] Philip Koopman. Linkedin post. https://www.linkedin.com/posts/philip-koopman-0631a4116_curse-of-rarity-for-autonomous-vehicles-activity-7204930563672002562-6uO5/, 2024. Accessed: 2025-03-14.

[233] Philip Koopman. Time to Formally Define Level 2+ Vehicle Automation. https://philkoopman.substack.com/p/time-to-formally-define-level-2-vehicle-196, 2024. Accessed: 2025-03-28.

[234] Philip Koopman, Aaron Kane, and Jen Black. Credible Autonomy Safety Argumentation. In *Safety-Critical Systems Symposium (SSS)*, 2019.

[235] Michael Kösel, Marcel Schreiber, Michael Ulrich, Claudius Gläser, and Klaus Dietmayer. Revisiting Out-of-Distribution Detection in LiDAR-based 3D Object Detection. In *IEEE Intelligent Vehicles Symposium (IV)*, 2024.

[236] Kamil Kowol, Matthias Rottmann, Stefan Bracke, and Hanno Gottschalk. YOdar: Uncertainty-based Sensor Fusion for Vehicle Detection with Camera and Radar Sensors. In *International Conference on Agents and Artificial Intelligence (ICAART)*, 2021.

[237] Paul J. Krassnig and Dieter P. Gruber. ISP-AD: A Large-Scale Real-World Dataset for Advancing Industrial Anomaly Detection with Synthetic and Real Defects. *arXiv:2503.04997*, 2025.

[238] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf, 2009. Accessed: 2025-03-07.

[239] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2012.

[240] Punit Kumar and Atul Gupta. Active Learning Query Strategies for Classification, Regression, and Clustering: A Survey. *Journal of Computer Science and Technology (JCST)*, 2020.

[241] Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 2003.

[242] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[243] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast Encoders for Object Detection From Point Clouds. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2019.

[244] Zakaria Laskar, Tomas Vojir, Matej Grcic, Iaroslav Melekhov, Shankar Gangisettye, Juho Kannala, Jiri Matas, Giorgos Tolias, and C. V. Jawahar. A Dataset for Semantic Segmentation in the Presence of Unknowns. *arXiv:2503.22309*, 2025.

[245] Yann LeCun. A Path Towards Autonomous Machine Intelligence. *OpenReview:BZ5a1r-kVsf*, 2022.

[246] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-Based Learning Applied to Document Recognition. In *Proceedings of the IEEE*, 1998.

[247] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[248] Ted Lentsch, Holger Caesar, and Dariu M Gavrila. UNION: Unsupervised 3D Object Detection using Object Appearance-based Pseudo-Classes. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[249] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle Detection from 3D Lidar Using Fully Convolutional Network. In *Robotics: Science and Systems (RSS)*, 2016.

[250] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. CutPaste: Self-Supervised Learning for Anomaly Detection and Localization. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2021.

[251] Hongyang Li, Yang Li, Huijie Wang, Jia Zeng, Huilin Xu, Pinlong Cai, Li Chen, Junchi Yan, Feng Xu, Lu Xiong, Jingdong Wang, Futang Zhu, Chunjing Xu, Tiancai Wang, Fei Xia, Beipeng Mu, Zhihui Peng, Dahua Lin, and Yu Qiao. Open-sourced Data Ecosystem in Autonomous Driving: the Present and Future. *Scientia Sinica Informationis (SSI)*, 2024.

[252] Jianan Li and Qiulei Dong. Open-Set Semantic Segmentation for Point Clouds via Adversarial Prototype Framework. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2023.

[253] Kaican Li, Kai Chen, Haoyu Wang, Lanqing Hong, Chaoqiang Ye, Jianhua Han, Yukuai Chen, Wei Zhang, Chunjing Xu, Dit-Yan Yeung, Xiaodan Liang, Zhenguo Li, and Hang Xu. CODA: A Real-World Road Corner Case Dataset for Object Detection in Autonomous Driving. In *European Conference on Computer Vision (ECCV)*, 2022.

[254] Shengze Li, Jianjian Cao, Peng Ye, Yuhan Ding, Chongjun Tu, and Tao Chen. ClipSAM: CLIP and SAM Collaboration for Zero-Shot Anomaly Segmentation. *Neurocomputing*, 2024.

[255] Xin Li, Tao Ma, Yuenan Hou, Botian Shi, Yuchen Yang, Youquan Liu, Xingjiao Wu, Qin Chen, Yikang Li, Yu Qiao, and Liang He. LoGoNet: Towards Accurate 3D Object Detection with Local-to-Global Cross- Modal Fusion. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2023.

[256] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying Voxel-based Representation with Transformer for 3D Object Detection. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[257] Yihao Li, Jianbo Tao, and Franz Wotawa. Ontology-based test generation for automated and autonomous driving functions. *Information and Software Technology*, 2020.

[258] Yiming Li, Zhiheng Li, Nuo Chen, Moonjun Gong, Zonglin Lyu, Zehong Wang, Peili Jiang, and Chen Feng. Multiagent Multitraversal Multimodal Self-Driving: Open MARS Dataset. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2024.

[259] Yueyuan Li, Wei Yuan, Songan Zhang, Weihao Yan, Qiyuan Shen, Chunxiang Wang, and Ming Yang. Choose Your Simulator Wisely: A Review on Open-Source Simulators for Autonomous Driving. *IEEE Transactions on Intelligent Vehicles (T-IV)*, 2024.

[260] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. BEVFusion: A Simple and Robust LiDAR-Camera Fusion Framework. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[261] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022.

[262] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV)*, 2014.

[263] Krzysztof Lis, Sina Honari, Pascal Fua, and Mathieu Salzmann. Detecting Road Obstacles by Erasing Them. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2024.

[264] Krzysztof Lis, Krishna Kanth Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the Unexpected via Image Resynthesis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[265] Fangchen Liu, Hao Liu, Aditya Grover, and Pieter Abbeel. Masked Autoencoding for Scalable and Generalizable Decision Making. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[266] Haisong Liu, Tao Lu, Yihui Xu, Jia Liu, Wenjie Li, and Lijun Chen. CamLiFlow: Bidirectional Camera-LiDAR Fusion for Joint Optical Flow and Scene Flow Estimation. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2022.

[267] Henry X. Liu, Zhong Cao, Xintao Yan, Shuo Feng, and Qiujing Lu. Autonomous Vehicles: A Critical Review (2004-2024) and a Vision for the Future. *TechRxiv*, 2025.

[268] Henry X. Liu and Shuo Feng. Curse of Rarity for Autonomous Vehicles. *Nature Communications*, 2024.

[269] Mingyu Liu, Ekim Yurtsever, Jonathan Fossaert, Xingcheng Zhou, Walter Zimmer, Yuning Cui, Bare Luka Zagar, and Alois C. Knoll. A Survey on Autonomous Driving Datasets: Statistics, Annotation Quality, and a Future Outlook. *IEEE Transactions on Intelligent Vehicles (T-IV)*, 2024.

[270] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. In *European Conference on Computer Vision (ECCV)*, 2024.

[271] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future Frame Prediction for Anomaly Detection – A New Baseline. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2018.

[272] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-Supervised Learning: Generative or Contrastive. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2021.

[273] Xinhao Liu, Moonjun Gong, Qi Fang, Haoyu Xie, Yiming Li, Hang Zhao, and Chen Feng. LiDAR-based 4D Occupancy Completion and Forecasting. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[274] Yuyuan Liu, Choubo Ding, Yu Tian, Guansong Pang, Vasileios Belagiannis, Ian Reid, and Gustavo Carneiro. Residual Pattern Learning for Pixel-wise Out-of-Distribution Detection in Semantic Segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[275] Zhian Liu, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. A Hybrid Video Anomaly Detection Framework via Memory-Augmented Flow Reconstruction and Flow-Guided Frame Prediction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[276] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L. Rus, and Song Han. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[277] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2019.

[278] Jonas Lohdefink, Justin Fehrling, Marvin Klingner, Fabian Huger, Peter Schlicht, Nico M. Schmidt, and Tim Fingscheidt. Self-Supervised Domain Mismatch Estimation for Autonomous Perception. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR) Workshops*, 2020.

[279] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[280] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal Event Detection at 150 FPS in MATLAB. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2013.

[281] Jiachen Lu, Ze Huang, Zeyu Yang, Jiahui Zhang, and Li Zhang. WoVoGen: World Volume-Aware Diffusion for Controllable Multi-camera Driving Scene Generation. In *European Conference on Computer Vision (ECCV)*, 2025.

[282] Xinyang Lu, Flint Xiaofeng Fan, and Tianying Wang. Action and Trajectory Planning for Urban Autonomous Driving with Hierarchical Reinforcement Learning. In *International Conference on Machine Learning (ICML) Workshops*, 2023.

[283] Weixin Luo, Wen Liu, and Shenghua Gao. A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[284] Weixin Luo, Wen Liu, and Shenghua Gao. A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[285] Enhui Ma, Lijun Zhou, Tao Tang, Zhan Zhang, Dong Han, Junpeng Jiang, Kun Zhan, Peng Jia, Xianpeng Lang, Haiyang Sun, Di Lin, and Kaicheng Yu. Unleashing Generalization of End-to-End Autonomous Driving with Controllable Long Video Generation. *arXiv:2406.01349*, 2024.

[286] Yunsheng Ma, Wenqian Ye, Can Cui, Haiming Zhang, Shuo Xing, Fucai Ke, Jinhong Wang, Chenglin Miao, Jintai Chen, Hamid Rezatofighi, Zhen Li, Guangtao Zheng, Chao Zheng, Tianjiao He, Manmohan Chandraker, Burhaneddin Yaman, Xin Ye, Hang Zhao, and Xu Cao. Position: Prospective of Autonomous Driving - Multimodal LLMs World Models Embodied Intelligence AI Alignment and Mamba. In *IEEE Winter Conference on Applications of Computer Vision (WACV) Workshops*, 2025.

[287] Kira Maag, Robin Chan, Svenja Uhlemeyer, Kamil Kowol, and Hanno Gottschalk. Two Video Data Sets for Tracking and Retrieval of Out of Distribution Objects. In *Asian Conference on Computer Vision (ACCV)*, 2023.

[288] Mad1 Minute. Trees Falling On Road. https://www.youtube.com/watch?v=3VsLeUtXvxk&ab_channel=Mad1Minute, 2017. Accessed: 2022-07-21.

[289] Madecu. CARLA 0.9.7 release. https://carla.org/2019/12/11/release-0.9.7/, 2019. Accessed: 2025-05-12.

[290] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly Detection in Crowded Scenes. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2010.

[291] Sebastian Maierhofer, Anna-Katharina Rettinger, Eva Charlotte Mayer, and Matthias Althoff. Formalization of Interstate Traffic Rules in Temporal Logic. In *IEEE Intelligent Vehicles Symposium (IV)*, 2020.

[292] Andrey Malinin and Mark Gales. Predictive Uncertainty Estimation via Prior Networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[293] Yunze Man, Liang-Yan Gui, and Yu-Xiong Wang. BEV-Guided Multi-Modality Fusion for Driving Perception. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2023.

[294] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Hanxue Liang, Jingheng Chen, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Jie Yu, Hang Xu, and Chunjing Xu. One Million Scenes for Autonomous Driving: ONCE Dataset. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[295] Zhenjiang Mao, Dong-You Jhong, Ao Wang, and Ivan Ruchkin. Language-Enhanced Latent Representations for Out-of-Distribution Detection in Autonomous Driving. In *IEEE International Conference on Robotics and Automation (ICRA) Workshops*, 2024.

[296] Mana Masuda, Ryo Hachiuma, Ryo Fujii, Hideo Saito, and Yusuke Sekikawa. Toward Unsupervised 3d Point Cloud Anomaly Detection Using Variational Autoencoder. In *IEEE International Conference on Image Processing (ICIP)*, 2021.

[297] MATH+. Hanno Gottschalk (TU Berlin) on Automated Driving Using AI. https://mathplus.de/news/hanno-gottschalk-on-automated-driving-using-ai/, 2024. Accessed: 2024-07-23.

[298] Tekedra Mawakana and Dmitri Dolgov. Investing to bring the Waymo Driver to more riders. https://waymo.com/blog/2024/10/investing-to-bring-the-waymo-driver-to-more-riders, 2024. Accessed: 2025-03-14.

[299] Jakob Mayr, Christian Unger, and Federico Tombari. Self-Supervised Learning of the Drivable Area for Autonomous Vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[300] Hendrik Alexander Mehrtens, Camila González, and Anirban Mukhopadhyay. Improving Robustness and Calibration in Ensembles with Diversity Regularization. In *German Conference on Pattern Recognition (GCPR)*, 2022.

[301] Sachin Mehta and Mohammad Rastegari. Separable Self-attention for Mobile Vision Transformers. *Transactions on Machine Learning Research (TMLR)*, 2022.

[302] Till Menzel, Gerrit Bagschik, Leon Isensee, Andre Schomburg, and Markus Maurer. From Functional to Logical Scenarios: Detailing a Keyword-Based Scenario Description for Execution in a Simulation Environment. In *IEEE Intelligent Vehicles Symposium (IV)*, 2019.

[303] Mercedes-Benz. DRIVE PILOT - Support speed of up to 95 km/h on German motorways. https://group.mercedes-benz.com/innovations/product-innovation/autonomous-driving/drive-pilot-95-kmh.html, 2024. Accessed: 2025-03-28.

[304] Robert K. Merton. *On The Shoulders of Giants - A Shandean Postscript*. The Free Press, New York, 1965.

[305] MinyiLin. Roboflow Universe: CARLA Computer Vision Project. https://universe.roboflow.com/minyilin-3nzak/carla-f4l16, 2022. Accessed: 2924-06-07.

[306] Himangi Mittal, Brian Okorn, and David Held. Just Go With the Flow: Self-Supervised Scene Flow Estimation. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2020.

[307] Majid Moghadam, Ali Alizadeh, Engin Tekin, and Gabriel Hugh Elkaim. A Deep Reinforcement Learning Approach for Long-term Short-term Planning on Frenet Frame. In *IEEE International Conference on Automation Science and Engineering (CASE)*, 2021.

[308] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, 2018.

[309] Jeremy Morse, Dejanira Araiza-Illan, Jonathan Lawry, Arthur Richards, and Kerstin Eder. A Fuzzy Approach to Qualification in Design Exploration for Autonomous Robots and Systems. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017.

[310] Bálint Mucsányi, Michael Kirchhof, and Seong Joon Oh. Benchmarking uncertainty disentanglement: Specialized uncertainties for specialized tasks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[311] Subramanya Nageshrao, H. Eric Tseng, and Dimitar Filev. Autonomous Highway Driving using Deep Reinforcement Learning. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2019.

[312] Mahyar Najibi, Jingwei Ji, Yin Zhou, Charles R. Qi, Xinchen Yan, Scott Ettinger, and Dragomir Anguelov. Motion Inspired Unsupervised Perception and Prediction in Autonomous Driving. In *European Conference on Computer Vision (ECCV)*, 2022.

[313] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. Curriculum Learning for Reinforcement Learning Domains: A Framework and Survey. *Journal of Machine Learning Research (JMLR)*, 2020.

[314] Keith Naughton and Monica Raymunt. Ford and Volkswagen pull the plug on robocar unit Argo AI in major setback to their self-driving plans. https://fortune.com/europe/2022/10/27/ford-volkswagen-pull-plug-robocar-unit-argo-ai-major-setback-self-driving-plans/, 2022. Accessed: 2024-07-19.

[315] Sergio Naval Marimont, Vasilis Siomos, Matthew Baugh, Christos Tzelepis, Bernhard Kainz, and Giacomo Tarroni. Ensembled Cold-Diffusion Restorations for Unsupervised Anomaly Detection. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2024.

[316] Nazir Nayal, Mısra Yavuz, João F. Henriques, and Fatma Güney. RbA: Segmenting Unknown Regions Rejected by All. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[317] Alexey Nekrasov, Malcolm Burdorf, Stewart Worrall, Bastian Leibe, and Julie Stephany Berrio Perez. Spotting the Unexpected (STU): A 3D LiDAR Dataset for Anomaly Segmentation in Autonomous Driving. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2025.

[318] Alexey Nekrasov, Alexander Hermans, Lars Kuhnert, and Bastian Leibe. UGainS: Uncertainty Guided Anomaly Instance Segmentation. In *German Conference on Pattern Recognition (GCPR)*, 2023.

[319] Alexey Nekrasov, Rui Zhou, Miriam Ackermann, Alexander Hermans, Bastian Leibe, and Matthias Rottmann. OoDIS: Anomaly Instance Segmentation Benchmark. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR) Workshops*, 2024.

[320] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[321] Isaac Newton. Letter to Robert Hooke. https://discover.hsp.org/Record/dc-9792, 1675. Accessed: 2025-04-22.

[322] Ming Nie, Xinyue Cai, Hang Xu, and Li Zhang. LaneCorrect: Self-supervised Lane Detection. *arXiv: 2404.14671*, 2024.

[323] David Nister, Hon-Leung Lee, Julia Ng, and Yizhou Wang. An Introduction to the Safety Force Field. Technical report, Nvidia, 2019.

[324] Julia Nitsch, Masha Itkina, Ransalu Senanayake, Juan Nieto, Max Schmidt, Roland Siegwart, Mykel J. Kochenderfer, and Cesar Cadena. Out-of-Distribution Detection for Automotive Perception. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2021.

[325] Dantong Niu, Xudong Wang, Xinyang Han, Long Lian, Roei Herzig, and Trevor Darrell. Unsupervised Universal Image Segmentation. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2024.

[326] Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident Learning: Estimating Uncertainty in Dataset Labels. *Journal of Artificial Intelligence Research (JAIR)*, 2021.

[327] N. Noy and Deborah Mcguinness. Ontology Development 101: A Guide to Creating Your First Ontology. Technical report, Stanford Knowledge Systems Laboratory, 2001.

[328] Julian Nubert, Shehryar Khattak, and Marco Hutter. Self-supervised Learning of LiDAR Odometry for Robotic Applications. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[329] Lucas Nunes, Xieyuanli Chen, Rodrigo Marcuzzi, Aljosa Osep, Laura Leal-Taixé, Cyrill Stachniss, and Jens Behley. Unsupervised Class-Agnostic Instance Segmentation of 3D LiDAR Data for Autonomous Vehicles. *IEEE Robotics and Automation Letters (RA-L)*, 2022.

[330] Toshiaki Ohgushi, Kenji Horiguchi, and Masao Yamanaka. Road Obstacle Detection Method Based on an Autoencoder with Semantic Segmentation. In *Asian Conference on Computer Vision (ACCV)*, 2021.

[331] Madalina Olteanu, Fabrice Rossi, and Florian Yger. Meta-Survey on Outlier and Anomaly Detection. *Neurocomputing*, 2023.

[332] On-Road Automated Driving Committee. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. Standard J3016-202104, SAE International, 2021.

[333] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research (TMLR)*, 2023.

[334] Tinghui Ouyang, Vicent Sanz Marco, Yoshinao Isobe, Hideki Asoh, Yutaka Oiwa, and Yoshiki Seo. Corner Case Data Description and Detection. In *IEEE/ACM Workshop on AI Engineering - Software Engineering for AI (WAIN)*, 2021.

[335] Matteo Pagliardini, Martin Jaggi, François Fleuret, and Sai Praneeth Karimireddy. Agree to Disagree: Diversity through Disagreement for Better Transferability. In *International Conference on Learning Representations (ICLR)*, 2023.

[336] Anshul Paigwar, Ozgur Erkent, David Sierra-Gonzalez, and Christian 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.

[337] Su Pang, Daniel Morris, and Hayder Radha. Fast-CLOCs: Fast Camera-LiDAR Object Candidates Fusion for 3D Object Detection. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2022.

[338] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving Adversarial Robustness via Promoting Ensemble Diversity. In *International Conference on Machine Learning (ICML)*, 2019.

[339] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning Memory-guided Normality for Anomaly Detection. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2020.

[340] Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer, Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Hadsell, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A Large-Scale Foundation World Model. https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/, 2024. Accessed: 2025-04-24.

[341] Jay Patrikar, Sushant Veer, Apoorva Sharma, Marco Pavone, and Sebastian Scherer. RuleFuser: An Evidential Bayes Approach for Rule Injection in Imitation Learned Planners and Predictors for Robustness under Distribution Shifts. In *International Symposium of Robotics Research (ISRR)*, 2024.

[342] Svetlana Pavlitskaya, Christian Hubschneider, and Michael Weber. *Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety*, chapter Evaluating Mixture-of-Experts Architectures for Network Aggregation. Springer, 2022.

[343] Svetlana Pavlitskaya, Christian Hubschneider, Michael Weber, Ruby Moritz, Fabian Huger, Peter Schlicht, and J. Marius Zollner. Using Mixture of Expert

Models to Gain Insights into Semantic Segmentation. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR) Workshops*, 2020.

[344] Ru Peng, Heming Zou, Haobo Wang, Yawen Zeng, Zenan Huang, and Junbo Zhao. Energy-Based Automated Model Evaluation. In *International Conference on Learning Representations (ICLR)*, 2024.

[345] Jerg Pfeil, Jochen Wieland, Thomas Michalke, and Andreas Theissler. On Why the System Makes the Corner Case: AI-based Holistic Anomaly Detection for Autonomous Driving. In *IEEE Intelligent Vehicles Symposium (IV)*, 2022.

[346] Jonah Philion and Sanja Fidler. Lift, Splat, Shoot: Encoding Images from Arbitrary Camera Rigs by Implicitly Unprojecting to 3D. In *European Conference on Computer Vision (ECCV)*, 2020.

[347] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and Found: Detecting Small Road Hazards for Self-Driving Vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[348] Aldi Piroli, Vinzenz Dallabetta, Johannes Kopp, Marc Walessa, Daniel Meissner, and Klaus Dietmayer. LS-VOS: Identifying Outliers in 3D Object Detections Using Latent Space Virtual Outlier Synthesis. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2023.

[349] Alexander Popov, Alperen Degirmenci, David Wehr, Shashank Hegde, Ryan Oldja, Alexey Kamenev, Bertrand Douillard, David Nistér, Urs Muller, Ruchi Bhargava, Stan Birchfield, and Nikolai Smolyanskiy. Mitigating Covariate Shift in Imitation Learning for Autonomous Vehicles Using Latent Space Generative World Models. *arXiv:2409.16663*, 2024.

[350] Janis Postels, Mattia Segu, Tao Sun, Luca Sieber, Luc Van Gool, Fisher Yu, and Federico Tombari. On the Practicality of Deterministic Epistemic Uncertainty. In *International Conference on Machine Learning (ICML)*, 2022.

[351] pyoscx. scenariogeneration. https://github.com/pyoscx/scenariogeneration, 2022. Accessed: 2022-07-20.

[352] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[353] Yiran Qin, Chaoqun Wang, Zijian Kang, Ningning Ma, Zhen Li, and Ruimao Zhang. SupFusion: Supervised LiDAR-Camera Fusion for 3D Object Detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[354] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models

From Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[355] Saeed Rahmani, Sabine Rieder, Erwin de Gelder, Marcel Sonntag, Jorge Lorente Mallada, Sytze Kalisvaart, Vahid Hashemi, and Simeon C. Calvert. A Systematic Review of Edge Case Detection in Automated Driving: Methods, Challenges and Future Directions. *arXiv:2410.08491*, 2024.

[356] Shyam Nandan Rai, Fabio Cermelli, Dario Fontanel, Carlo Masone, and Barbara Caputo. Unmasking Anomalies in Road-Scene Segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[357] Yousra Regaya, Fodil Fadli, and Abbes Amira. Point-Denoise: Unsupervised Outlier Detection for 3D Point Clouds Enhancement. *Multimedia Tools and Applications*, 2021.

[358] Raymond Reiter. *Logic and Data Bases*, chapter On Closed World Data Bases. Springer, 1978.

[359] Davis Rempe, Jonah Philion, Leonidas J. Guibas, Sanja Fidler, and Or Litany. Generating Useful Accident-Prone Driving Scenarios via a Learned Traffic Prior. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2022.

[360] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

[361] Albert Rizaldi, Jonas Keinholz, Monika Huber, Jochen Feldle, Fabian Immler, Matthias Althoff, Eric Hilgendorf, and Tobias Nipkow. Formalising and Monitoring Traffic Rules for Autonomous Vehicles in Isabelle/HOL. In *International Conference on Integrated Formal Methods (IFM)*, 2017.

[362] Robotics Knowledgebase. Trajectory Planning in the Frenet Space. https://roboticsknowledgebase.com/wiki/planning/frenet-frame-planning/, 2022. Accessed: 2024-02-06.

[363] Alina Roitberg, Ziad Al-Halah, and Rainer Stiefelhagen. Informed Democracy: Voting-based Novelty Detection for Action Recognition. In *British Machine Vision Conference (BMVC)*, 2018.

[364] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2022.

[365] O. Ronneberger, T. Brox, and P. Fischer. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[366] Kevin Rösch, Florian Heidecker, Julian Truetsch, Kamil Kowol, Clemens Schicktanz, Maarten Bieshaar, Bernhard Sick, and Christoph Stiller. Space, Time, and Interaction: A Taxonomy of Corner Cases in Trajectory Datasets for Automated Driving. In *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2022.

[367] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations (ICLR)*, 2020.

[368] Lukas Rummelhard, Amaury Negre, and Christian Laugier. Conditional Monte Carlo Dense Occupancy Tracker. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2015.

[369] Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. GAIA-2: A Controllable Multi-View Generative World Model for Autonomous Driving. *arXiv:2503.20523*, 2025.

[370] SAE International. SAE J3016 Levels of Driving Automation. https://www.sae.org/binaries/content/assets/cm/content/blog/sae-j3016-visual-chart_5.3.21.pdf, 2021. Accessed: 2025-03-28.

[371] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The Adverse Conditions Dataset with Correspondences for Semantic Driving Scene Understanding. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[372] Mohammadreza Salehi, Hossein Mirzaei, Dan Hendrycks, Yixuan Li, Mohammad Hossein Rohban, and Mohammad Sabokrou. A Unified Survey on Anomaly, Novelty, Open-Set, and Out of-Distribution Detection: Solutions and Future Challenges. *Transactions on Machine Learning Research (TMLR)*, 2022.

[373] Nayel Fabian Salem, Marcus Nolte, Veronica Haber, Till Menzel, Hans Steege, Robert Graubohm, and Markus Maurer. An Ontology-based Approach Towards Traceable Behavior Specifications in Automated Driving. *IEEE Access*, 2024.

[374] Ahmad El Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. End-to-End Deep Reinforcement Learning for Lane Keeping Assist. In *Conference on Neural Information Processing Systems (NeurIPS) Workshops*, 2016.

[375] Jules Sanchez. recoverKITTI360label. https://github.com/JulesSanchez/recoverKITTI360label/pull/3, 2022. Accessed: 2024-06-26.

[376] Jeff Schneider. Self Driving Cars & AI: Transforming our Cities and our Lives. https://www.youtube.com/watch?v=jTio_MPQRYc, 2019.

[377] Hans-Peter Schoener and Jens Mazzega. Introduction to PEGASUS. In *China Autonomous Driving Testing Technology Innovation Conference (ADTTI)*, 2018.

[378] Christopher Schröder, Andreas Niekler, and Martin Potthast. Revisiting Uncertainty-based Query Strategies for Active Learning with Transformers. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

[379] Marius Schubert, Tobias Riedlinger, Karsten Kahl, Daniel Kröll, Sebastian Schoenen, Siniša Šegvić, and Matthias Rottmann. Identifying Label Errors in Object Detection Datasets by Loss Inspection. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2024.

[380] Eli Schwartz, Assaf Arbelle, Leonid Karlinsky, Sivan Harary, Florian Scheidegger, Sivan Doveh, and Raja Giryes. MAEDAY: MAE for few-and zero-shot AnomalY-Detection. *Computer Vision and Image Understanding*, 2024.

[381] Francesco Secci and Andrea Ceccarelli. On failures of RGB cameras and their effects in autonomous driving applications. In *International Symposium on Software Reliability Engineering (ISSRE)*, 2020.

[382] Florian Seligmann, Philipp Becker, Michael Volpp, and Gerhard Neumann. Beyond Deep Ensembles: A Large-Scale Evaluation of Bayesian Deep Learning under Distribution Shift. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[383] Raul Sena Ferreira, Joris Guérin, Jérémie Guiochet, and Hélène Waeselynck. SiMOOD: Evolutionary Testing Simulation with Out-Of-Distribution Images. In *IEEE Pacific Rim International Symposium on Dependable Computing (PRDC)*, 2022.

[384] Burr Settles. Active Learning Literature Survey. University of Wisconsin-Madison. Computer Sciences Technical Report., 2010.

[385] Alireza Shafaei, Mark Schmidt, and James J. Little. A Less Biased Evaluation of Out-of-distribution Sample Detectors. In *British Machine Vision Conference (BMVC)*, 2019.

[386] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a Formal Model of Safe and Scalable Self-driving Cars. *arXiv:1708.06374*, 2018.

[387] Hao Shao, Letian Wang, RuoBing Chen, Hongsheng Li, and Yu Liu. Safety-Enhanced Autonomous Driving Using Interpretable Sensor Fusion Transformer. In *Conference on Robot Learning (CoRL)*, 2022.

[388] Wenbo Shao, Boqi Li, Wenhao Yu, Jiahui Xu, and Hong Wang. When Is It Likely to Fail? Performance Monitor for Black-Box Trajectory Prediction Model. *IEEE Transactions on Automation Science and Engineering (T-ASE)*, 2024.

[389] Amnon Shashua. Intel Newsroom: CES 2021: Under the Hood. https://www.youtube.com/watch?v=B7YNj66GxRA, 2021. Accessed: 2024-06-13.

[390] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2019.

[391] Youssef Shoeb, Azarm Nowzad, and Hanno Gottschalk. Out-of-Distribution Segmentation in Autonomous Driving: Problems and State of the Art. *arXiv:2503.08695*, 2025.

[392] Elena Shrestha, Chetan Reddy, Hanxi Wan, Yulun Zhuang, and Ram Vasudevan. Sense, Imagine, Act: Multimodal Perception Improves Model-Based Reinforcement Learning for Head-to-Head Autonomous Racing. *arXiv:2305.04750*, 2023.

[393] Chonghao Sima, Wenwen Tong, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, and Hongyang Li. Scene as Occupancy. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[394] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[395] Rohan Sinha, Amine Elhafsi, Christopher Agia, Matthew Foutter, Edward Schmerling, and Marco Pavone. Real-Time Anomaly Detection and Reactive Planning with Large Language Models. In *Robotics: Science and Systems (RSS)*, 2024.

[396] John Roar Ventura Solaas, Enrico Mariconti, and Nilufer Tuptuk. Systematic Literature Review: Anomaly Detection in Connected and Autonomous Vehicles. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 2025.

[397] Andrea Stocco, Michael Weiss, Marco Calzana, and Paolo Tonella. Misbehaviour Prediction for Autonomous Driving Systems. In *IEEE/ACM International Conference on Software Engineering (ICSE)*, 2020.

[398] Lei Sun, Kailun Yang, Xinxin Hu, Weijian Hu, and Kaiwei Wang. Real-Time Fusion Network for RGB-D Semantic Segmentation Incorporating Unexpected Obstacle Detection for Road-driving Images. *IEEE Robotics and Automation Letters (RA-L)*, 2020.

[399] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2020.

[400] Tao Sun, Mattia Segu, Janis Postels, Yuxuan Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. SHIFT: a synthetic driving dataset for continuous multi-task domain adaptation. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2022.

[401] Richard S. Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2020.

[402] Zaid Tahir and Rob Alexander. Intersection Focused Situation Coverage-Based Verification and Validation Framework for Autonomous Vehicles Implemented in CARLA. In *Modelling and Simulation for Autonomous Systems (MESAS)*, 2022.

[403] Jacopo Talamini, Alberto Bartoli, Andrea De Lorenzo, and Eric Medvet. On the Impact of the Rules on Autonomous Drive Learning. *Applied Sciences*, 2020.

[404] Jianbo Tao, Yihao Li, Franz Wotawa, Hermann Felbinger, and Mihai Nica. On the Industrial Application of Combinatorial Testing for Autonomous Driving Functions. In *IEEE International Conference on Software Testing, Verification and Validation (ICST) Workshops*, 2019.

[405] Brad Templeton. Robotaxis — Mostly Waymo — Are Giving 1.3 Million Rides A Month. Why? https://www.forbes.com/sites/bradtempleton/2025/03/07/robotaxis-mostly-waymo-are-giving-13-million-ridesmonth--why/, 2025. Accessed: 2025-03-13.

[406] Tesla. Full Self-Driving (Supervised). https://www.tesla.com/ownersmanual/modely/en_us/GUID-2CB60804-9CEA-4F4B-8B04-09B991368DC5.html, 2025. Accessed: 2025-03-28.

[407] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, and Pamela Mahoney. Stanley: The Robot that Won the DARPA Grand Challenge. *Journal of Field Robotics (JFR)*, 2006.

[408] Beiwen Tian, Huan ang Gao, Leiyao Cui, Yupeng Zheng, Lan Luo, Baofeng Wang, Rong Zhi, Guyue Zhou, and Hao Zhao. Latency-aware Road Anomaly Segmentation in Videos: A Photorealistic Dataset and New Metrics. *arXiv:2401.04942*, 2024.

[409] Beiwen Tian, Mingdao Liu, Huan-ang Gao, Pengfei Li, Hao Zhao, and Guyue Zhou. Unsupervised Road Anomaly Detection with Language Anchors. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[410] Thomas Tian, Boyi Li, Xinshuo Weng, Yuxiao Chen, Edward Schmerling, Yue Wang, Boris Ivanovic, and Marco Pavone. Tokenize the World into Object-level Knowledge to Address Long-tail Events in Autonomous Driving. In *Conference on Robot Learning (CoRL)*, 2025.

[411] Yu Tian, Yuyuan Liu, Guansong Pang, Fengbei Liu, Yuanhong Chen, and Gustavo Carneiro. Pixel-Wise Energy-biased Abstention Learning for Anomaly Segmentation on Complex Urban Driving Scenes. In *European Conference on Computer Vision (ECCV)*, 2022.

[412] Ivan Tishchenko, Sandro Lombardi, Martin R. Oswald, and Marc Pollefeys. Self-Supervised Learning of Non-Rigid Residual Flow and Ego-Motion. In *International Conference on 3D Vision (3DV)*, 2020.

[413] Maxime Tremblay, Shirsendu Sukanta Halder, Raoul de Charette, and Jean-François Lalonde. Rain Rendering for Evaluating and Improving Robustness to Bad Weather. *International Journal of Computer Vision (IJCV)*, 2021.

[414] Sifan Tu, Xin Zhou, Dingkang Liang, Xingyu Jiang, Yumeng Zhang, Xiaofan Li, and Xiang Bai. The Role of World Models in Shaping Autonomous Driving: A Comprehensive Survey. *arXiv:2502.10498*, 2025.

[415] Yuanpeng Tu, Yuxi Li, Boshen Zhang, Liang Liu, Jiangning Zhang, Yabiao Wang, and Cai Rong Zhao. Self-Supervised Likelihood Estimation with Energy Guidance for Anomaly Segmentation in Urban Scenes. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

[416] Cumhur Erkan Tuncali, Georgios Fainekos, Danil Prokhorov, Hisahiro Ito, and James Kapinski. Requirements-Driven Test Generation for Autonomous Vehicles With Machine Learning Components. *IEEE Transactions on Intelligent Vehicles (T-IV)*, 2020.

[417] Udacity. Final Leaderboard of Udacity Challenge 2. https://github.com/udacity/self-driving-car, 2016.

[418] Svenja Uhlemeyer. *Unsupervised Open World Recognition in Computer Vision*. PhD thesis, University of Wuppertal, 2023.

[419] Svenja Uhlemeyer, Julian Lienen, Eyke Hüllermeier, and Hanno Gottschalk. Detecting Novelties with Empty Classes. *arXiv:2305.00983*, 2023.

[420] Svenja Uhlemeyer, Matthias Rottmann, and Hanno Gottschalk. Towards Unsupervised Open World Semantic Segmentation. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2022.

[421] UK Parliament. Debate on the address - HC Deb 04 November 1952 vol 507 cc7-134 - [FIRST DAY] - 2.40 p.m. https://api.parliament.uk/historic-hansard/commons/1952/nov/04/debate-on-the-address, 1952. Accessed: 2025-04-22.

[422] Simon Ulbrich, Till Menzel, Andreas Reschka, Fabian Schuldt, and Markus Maurer. Defining and substantiating the terms scene, situation, and scenario for automated driving. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2015.

[423] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, MN Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert, Thomas M. Howard, Sascha Kolski, Alonzo Kelly, Maxim Likhachev, Matt McNaughton, Nick Miller, Kevin Peterson, Brian Pilnick, Raj Rajkumar, Paul Rybski, Bryan Salesky, Young-Woo Seo, Sanjiv Singh, Jarrod Snider, Anthony Stentz, William "Red" Whittaker, Ziv Wolkowicki, Jason Ziglar, Hong Bae, Thomas Brown, Daniel Demitrish, Bakhtiar Litkouhi, Jim Nickolaou, Varsha Sadekar, Wende Zhang, Joshua Struble, Michael Taylor, Michael Darms, and Dave Ferguson. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics (JFR)*, 2008.

[424] Raquel Urtasun. Interpretable Neural Motion Planner. https://youtube.com/watch?v=PSZ2Px9PrHg, 2021. Accessed: 2024-03-01.

[425] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural Discrete Representation Learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[426] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[427] Lennart Vater, Marcel Sonntag, Johannes Hiller, Philipp Schaudt, and Lutz Eckstein. A Systematic Approach Towards the Definition of the Terms Edge Case and Corner Case for Automated Driving. In *IEEE International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 2023.

[428] Sushant Veer, Karen Leung, Ryan K Cosner, Yuxiao Chen, Peter Karkus, and Marco Pavone. Receding horizon planning with rule hierarchies for autonomous vehicles. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[429] Tomas Vojir, Tomáš Šipka, Rahaf Aljundi, Nikolay Chumerin, Daniel Olmeda Reino, and Jiri Matas. Road Anomaly Detection by Partial Image Reconstruction with Segmentation Coupling. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[430] Laura Von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, et al. Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2021.

[431] Ha Son Vu, Daisuke Ueta, Kiyoshi Hashimoto, Kazuki Maeno, Sugiri Pranata, and Sheng Mei Shen. Anomaly Detection with Adversarial Dual Autoencoders. *arXiv:1902.06924*, 2019.

[432] Waabi. Introducing the Waabi Driver. https://waabi.ai/introducing-the-waabi-driver/, 2022. Accessed: 2025-04-24.

[433] Zhexiong Wan, Yuxin Mao, Jing Zhang, and Yuchao Dai. RPEFlow: Multimodal Fusion of RGB-PointCloud-Event for Joint Optical Flow and Scene Flow Estimation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[434] Hang Wang, Xin Ye, Feng Tao, Chenbin Pan, Abhirup Mallik, Burhaneddin Yaman, Liu Ren, and Junshan Zhang. AdaWM: Adaptive World Model based Planning for Autonomous Driving. In *International Conference on Learning Representations (ICLR)*, 2025.

[435] Junpeng Wang, Liang Wang, Yan Zheng, Chin-Chia Michael Yeh, Shubham Jain, and Wei Zhang. Learning-From-Disagreement: A Model Comparison and Visual Analytics Framework. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2023.

[436] Lening Wang, Wenzhao Zheng, Yilong Ren, Han Jiang, Zhiyong Cui, Haiyang Yu, and Jiwen Lu. OccSora: 4D Occupancy Generation Models as World Simulators for Autonomous Driving. *arXiv:2405.20337*, 2024.

[437] Liang Wang, Junpeng Wang, Yan Zheng, Shubham Jain, Chin-Chia Michael Yeh, Zhongfang Zhuang, Javid Ebrahimi, and Wei Zhang. Learning from Disagreement for Event Detection. In *IEEE International Conference on Big Data (IEEE BigData)*, 2022.

[438] Lu Wang, Dongkai Zhang, Jiahao Guo, and Yuexing Han. Image Anomaly Detection Using Normal Data Only by Latent Space Resampling. *Applied Sciences*, 2020.

[439] Shuguang Wang, Qian Zhou, Kui Wu, Jinghuai Deng, Dapeng Wu, Wei-Bin Lee, and Jianping Wang. Interventional Root Cause Analysis of Failures in Multi-Sensor Fusion Perception Systems. In *Network and Distributed System Security Symposium (NDSS)*, 2025.

[440] Tianqi Wang, Sukmin Kim, Wenxuan Ji, Enze Xie, Chongjian Ge, Junsong Chen, Zhenguo Li, and Ping Luo. DeepAccident: A Motion and Accident Prediction Benchmark for V2X Autonomous Driving. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

[441] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, and Jiwen Lu. DriveDreamer: Towards Real-world-driven World Models for Autonomous Driving. In *European Conference on Computer Vision (ECCV)*, 2024.

[442] Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2024.

[443] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004.

[444] Ziyu Wang, Jing Ma, and Edmund M-K Lai. A Survey of Scenario Generation for Automated Vehicle Testing and Validation. *Future Internet*, 2024.

[445] Waymo. Introducing the 5th-generation Waymo Driver. https://waymo.com/blog/2020/03/introducing-5th-generation-waymo-driver.html, 2020. Accessed: 2023-06-01.

[446] Waymo. Waymo One. https://waymo.com/waymo-one/, 2025. Accessed: 2025-03-13.

[447] Waymo LLC. Part 573 Safety Recall Report 24E-013. https://static.nhtsa.gov/odi/rcl/2024/RCLRPT-24E013-4528.PDF, 2024. Accessed: 2025-03-14.

[448] Wayve. Wayve Raises Over $1 Billion Led by SoftBank to Develop Embodied AI Products for Automated Driving. https://wayve.ai/press/series-c/, 2024. Accessed: 2025-03-14.

[449] Wayve. Wayve's AV2.0 Approach. https://wayve.ai/technology/#AV2.0, 2025. Accessed: 2025-04-24.

[450] Andy Weedman. A billboard tricked my Tesla into stopping! . https://www.youtube.com/watch?v=-OdOmU58zOw, 2021. Accessed: 2025-03-14.

[451] Moritz Werling, Sören Kammel, Julius Ziegler, and Lutz Gröll. Optimal Trajectories for Time-Critical Street Scenarios Using Discretized Terminal Manifolds. *International Journal of Robotics Research (IJRR)*, 2012.

[452] Walt Whitman. *Leaves of Grass*. University of Chicago, facsimile edition edition, 1855. Accessed: 2025-04-22.

[453] Julian Wiederer, Arij Bouazizi, Marco Troina, Ulrich Kressel, and Vasileios Belagiannis. Anomaly detection in multi-agent trajectories for automated driving. In *Conference on Robot Learning (CoRL)*, 2022.

[454] Julian Wiederer, Julian Schmidt, Ulrich Kressel, Klaus Dietmayer, and Vasileios Belagiannis. A Benchmark for Unsupervised Anomaly Detection in Multi-Agent Trajectories. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2022.

[455] Hansen Wijanarko, Evelyne Calista, Li-Fen Chen, and Yong-Sheng Chen. Tri-VAE: Triplet Variational Autoencoder for Unsupervised Anomaly Detection in Brain Tumor MRI. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR) Workshops*, 2024.

[456] Andrew Gordon Wilson. The Case for Bayesian Deep Learning. *arXiv:2001.10995*, 2020.

[457] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next Generation Datasets for Self-driving Perception and Forecasting. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[458] Peter Wolf, Karl Kurzer, Tobias Wingert, Florian Kuhnt, and J. Marius Zöllner. Adaptive Behavior Generation for Autonomous Driving Using Deep Reinforcement Learning with Compact Semantic States. In *IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[459] Kelvin Wong, Shenlong Wang, Mengye Ren, Ming Liang, and Raquel Urtasun. Identifying Unknown Instances for Autonomous Driving. In *Conference on Robot Learning (CoRL)*, 2020.

[460] Franz Wotawa and Yihao Li. From Ontologies to Input Models for Combinatorial Testing. In *International Conference on Testing Software and Systems (ICTSS)*, 2018.

[461] Di Wu, Shicai Fan, Xue Zhou, Li Yu, Yuzhong Deng, Jianxiao Zou, and Baihong Lin. Unsupervised Anomaly Detection via Masked Diffusion Posterior Sampling. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2024.

[462] Jianzong Wu, Xiangtai Li, Shilin Xu, Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, et al. Towards open vocabulary learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2024.

[463] Peng Wu, Xuerong Zhou, Guansong Pang, Yujia Sun, Jing Liu, Peng Wang, and Yanning Zhang. Open-vocabulary video anomaly detection. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2024.

[464] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. DayDreamer: World Models for Physical Robot Learning. In *Conference on Robot Learning (CoRL)*, 2022.

[465] Philipp Wu, Arjun Majumdar, Kevin Stone, Yixin Lin, Igor Mordatch, Pieter Abbeel, and Aravind Rajeswaran. Masked Trajectory Models for Prediction, Representation, and Control. In *International Conference on Machine Learning (ICML)*, 2023.

[466] Wei Wu. Unsupervised Learning. University of Tübingen. Seminar: Introduction to Machine Learning, 2022.

[467] Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. PointPWC-Net: Cost Volume on Point Clouds for (Self-) Supervised Scene Flow Estimation. In *European Conference on Computer Vision (ECCV)*, 2020.

[468] Zehuan Wu, Jingcheng Ni, Xiaodong Wang, Yuxin Guo, Rui Chen, Lewei Lu, Jifeng Dai, and Yuwen Xiong. HoloDrive: Holistic 2D-3D Multi-Modal Street Scene Generation for Autonomous Driving. *arXiv:2412.01407*, 2024.

[469] Yingda Xia, Yi Zhang, Fengze Liu, Wei Shen, and Alan L. Yuille. Synthesize Then Compare: Detecting Failures and Anomalies for Semantic Segmentation. In *European Conference on Computer Vision (ECCV)*, 2020.

[470] Zhongyu Xia, Jishuo Li, Zhiwei Lin, Xinhao Wang, Yongtao Wang, and Ming-Hsuan Yang. OpenAD: Open-World Autonomous Driving Benchmark for 3D Object Detection. *arXiv:2411.17761*, 2024.

[471] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-Shot Learning—A Comprehensive Evaluation of the Good, the Bad and the Ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.

[472] Yichen Xie, Chenfeng Xu, Marie-Julie Rakotosaona, Patrick Rim, Federico Tombari, Kurt Keutzer, Masayoshi Tomizuka, and Wei Zhan. SparseFusion: Fusing Multi-Modal Sparse Representations for Multi-Sensor 3D Object Detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[473] Shaocong Xu, Pengfei Li, Qianpu Sun, Xinyu Liu, Yang Li, Shihui Guo, Zhen Wang, Bo Jiang, Rui Wang, Kehua Sheng, Bo Zhang, Li Jiang, Hao Zhao, and Yilun Chen. LiON: Learning Point-wise Abstaining Penalty for LiDAR Outlier DetectioN Using Diverse Synthetic Data. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

[474] Shenjie Xu and Leilani H. Gilpin. DANGER: A Framework of Danger-Aware Novel Dataset Generator Extension for Robustness Test of Machine Learning. In *Bay Area Machine Learning Symposium (BayLearn)*, 2022.

[475] Xin Xu, Lei Zuo, Xin Li, Lilin Qian, Junkai Ren, and Zhenping Sun. A Reinforcement Learning Approach to Autonomous Decision Making of Intelligent Vehicles on Highways. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[476] Feng Xue, Anlong Ming, Menghan Zhou, and Yu Zhou. A Novel Multi-layer Framework for Tiny Obstacle Discovery. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[477] Salisu Wada Yahaya, Ahmad Lotfi, and Mufti Mahmud. A consensus novelty detection ensemble approach for anomaly detection in activities of daily living. *Applied Soft Computing*, 2019.

[478] Junjie Yan, Yingfei Liu, Jianjian Sun, Fan Jia, Shuailin Li, Tiancai Wang, and Xiangyu Zhang. Cross Modal Transformer: Towards Fast and Robust 3D Object Detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[479] Ziyang Yan, Wenzhen Dong, Yihua Shao, Yuhang Lu, Liu Haiyang, Jingwen Liu, Haozhe Wang, Zhe Wang, Yan Wang, Fabio Remondino, and Yuexin Ma. RenderWorld: World Model with Self-Supervised 3D Label. *arXiv:2409.11356*, 2025.

[480] Chule Yang, Alessandro Renzaglia, Anshul Paigwar, Christian Laugier, and Danwei Wang. Driving Behavior Assessment and Anomaly Detection for Intelligent Vehicles. In *IEEE International Conference on Cybernetics and Intelligent Systems and Robotics, Automation and Mechatronics (CIS-RAM)*, 2019.

[481] Honghui Yang, Sha Zhang, Di Huang, Xiaoyang Wu, Haoyi Zhu, Tong He, Shixiang Tang, Hengshuang Zhao, Qibo Qiu, Binbin Lin, Xiaofei He, and Wanli Ouyang. UniPAD: A Universal Pre-training Paradigm for Autonomous Driving. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2024.

[482] Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, Jun Zhang, Andreas Geiger, Yu Qiao, and Hongyang Li. Generalized Predictive Model for Autonomous Driving. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2024.

[483] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision (IJCV)*, 2024.

[484] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. FoldingNet: Point Cloud Auto-Encoder via Deep Grid Deformation. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2018.

[485] Yu Yang, Jianbiao Mei, Yukai Ma, Siliang Du, Wenqing Chen, Yijie Qian, Yuxiang Feng, and Yong Liu. Driving in the Occupancy World: Vision-Centric 4D Occupancy Forecasting and Planning via World Models for Autonomous Driving. *AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

[486] Zetong Yang, Li Chen, Yanan Sun, and Hongyang Li. Visual point cloud forecasting enables scalable autonomous driving. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2024.

[487] Zeyu Yang, Jiaqi Chen, Zhenwei Miao, Wei Li, Xiatian Zhu, and Li Zhang. DeepInteraction: 3D Object Detection via Modality Interaction. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[488] Hang Yao, Ming Liu, Haolin Wang, Zhicun Yin, Zifei Yan, Xiaopeng Hong, and Wangmeng Zuo. GLAD: Towards Better Reconstruction with Global and Local Adaptive Diffusion Models for Unsupervised Anomaly Detection. In *European Conference on Computer Vision (ECCV)*, 2024.

[489] Fei Ye, Xuxin Cheng, Pin Wang, Ching-Yao Chan, and Jiucai Zhang. Automated Lane Change Strategy using Proximal Policy Optimization-based Deep Reinforcement Learning. In *IEEE Intelligent Vehicles Symposium (IV)*, 2020.

[490] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-Based 3D Object Detection and Tracking. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2021.

[491] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv:1805.04687v1*, 2018.

[492] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2020.

[493] Lingli Yu, Xuanya Shao, Yadong Wei, and Kaijun Zhou. Intelligent Land-Vehicle Model Transfer Trajectory Planning Method Based on Deep Reinforcement Learning. *Sensors*, 2018.

[494] Yaodong Yu, Zitong Yang, Alexander Wei, Yi Ma, and Jacob Steinhardt. Predicting Out-of-Distribution Error with the Projection Norm. In *International Conference on Machine Learning (ICML)*, 2022.

[495] Tahir Zaid. Intersection focused Situation Coverage-based Verification and Validation Framework for Autonomous Vehicles Implemented in CARLA. https://web.archive.org/web/20220903201212/https://github.com/zaidtahirbutt/Situation-Coverage-based-AV-Testing-Framework-in-CARLA, 2022. Accessed: 2025-03-07.

[496] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Reconstruction by inpainting for visual anomaly detection. *Pattern Recognition*, 2021.

[497] Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernandez Dominguez. WildDash - Creating Hazard-Aware Benchmarks. In *European Conference on Computer Vision (ECCV)*, 2018.

[498] Yihan Zeng, Da Zhang, Chunwei Wang, Zhenwei Miao, Ting Liu, Xin Zhan, Dayang Hao, and Chao Ma. LIFT: Learning 4D LiDAR Image Fusion Transformer for 3D Object Detection. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2022.

[499] Chen Zhang, Zefan Huang, Marcelo H. Ang, and Daniela Rus. LiDAR Degradation Quantification for Autonomous Driving in Rain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[500] Chi Zhang, Kais Kacem, Gereon Hinz, and Alois Knoll. Safe and Rule-Aware Deep Reinforcement Learning for Autonomous Driving at Intersections. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2022.

[501] Haiming Zhang, Ying Xue, Xu Yan, Jiacheng Zhang, Weichao Qiu, Dongfeng Bai, Bingbing Liu, Shuguang Cui, and Zhen Li. An Efficient Occupancy World Model via Decoupled Dynamic Flow and Image-assisted Training. *arXiv:2412.13772*, 2024.

[502] Haiming Zhang, Ying Xue, Xu Yan, Jiacheng Zhang, Weichao Qiu, Dongfeng Bai, Bingbing Liu, Shuguang Cui, and Zhen Li. An Efficient Occupancy World Model via Decoupled Dynamic Flow and Image-assisted Training. *arXiv:2412.13772*, 2024.

[503] Lunjun Zhang, Yuwen Xiong, Ze Yang, Sergio Casas, Rui Hu, and Raquel Urtasun. Learning Unsupervised World Models for Autonomous Driving via Discrete Diffusion. In *International Conference on Learning Representations (ICLR)*, 2024.

[504] Mengshi Zhang, Yuqun Zhang, Lingming Zhang, Cong Liu, and Sarfraz Khurshid. DeepRoad: GAN-based Metamorphic Testing and Input Validation Framework for Autonomous Driving Systems. In *ACM/IEEE International Conference on Automated Software Engineering (ASE)*, 2018.

[505] Shuo Zhang, Yupeng Zhai, Jilin Mei, and Yu Hu. FusionOcc: Multi-Modal Fusion for 3D Occupancy Prediction. In *ACM International Conference on Multimedia (MM)*, 2024.

[506] Yumeng Zhang, Shi Gong, Kaixin Xiong, Xiaoqing Ye, Xiao Tan, Fan Wang, Jizhou Huang, Hua Wu, and Haifeng Wang. BEVWorld: A Multimodal World Model for Autonomous Driving via Unified BEV Latent Space. *arXiv:2407.05679*, 2024.

[507] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-End Urban Driving by Imitating a Reinforcement Learning Coach. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[508] Zhiwei Zhang, Zhizhong Zhang, Qian Yu, Ran Yi, Yuan Xie, and Lizhuang Ma. LiDAR-Camera Panoptic Segmentation via Geometry-Consistent and Semantic-Aware Alignment. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[509] Lihua Zhao, Ryutaro Ichise, Zheng Liu, Seiichi Mita, and Yutaka Sasaki. Ontology-Based Driving Decision Making: A Feasibility Study at Uncontrolled Intersections. *IEICE Transactions on Information and Systems*, 2017.

[510] Wenjie Zhao, Jia Li, Xin Dong, Yu Xiang, and Yunhui Guo. Segment Every Out-of-Distribution Object. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2024.

[511] Jingxing Zhou and Jürgen Beyerer. Corner Cases in Data-Driven Automated Driving: Definitions, Properties and Solutions. In *IEEE Intelligent Vehicles Symposium (IV)*, 2023.

[512] Kang Zhou, Yuting Xiao, Jianlong Yang, Jun Cheng, Wen Liu, Weixin Luo, Zaiwang Gu, Jiang Liu, and Shenghua Gao. Encoding Structure-Texture Relation with P-Net for Anomaly Detection in Retinal Images. In *European Conference on Computer Vision (ECCV)*, 2020.

[513] Xin Zhou, Dingkang Liang, Sifan Tu, Xiwu Chen, Yikang Ding, Dingyuan Zhang, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. HERMES: A Unified Self-Driving World Model for Simultaneous 3D Scene Understanding and Generation. *arXiv:2501.14729*, 2025.

[514] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2018.

[515] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR Segmentation. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2021.

[516] Adrian Zlocki, Joachim G. Taiber, Stephan Hinze, Christian Rösener, John Tintinalli, Thomas Bock, and Simon Rößner. Terms and Definitions Related to Testing of Automated Vehicle Technologies. DIN-SAE-Specification 91381, German Institute for Standardization and SAE International, 2019.

[517] Vlas Zyrianov, Henry Che, Zhijian Liu, and Shenlong Wang. LidarDM: Generative LiDAR Simulation in a Generated World. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.