Closing the Loop in Automated Driving: Dataset Design, Scenario Synthesis, and Deployed Perception

Università
di Genova

Luca Forneris

Supervisors: Prof. Francesco Bellotti
Dr. Changjae Oh

Department of Electrical, Electronic, Telecommunications Engineering and Naval Architecture (DITEN)
University of Genoa

This dissertation is submitted for the degree of
Doctor of Philosophy

Joint Doctorate in Interactive and
Cognitive Environments - Cycle 38

March 2026

Closing the Loop in Automated Driving: Dataset Design, Scenario Synthesis, and Deployed Perception

Luca Forneris

Joint Doctorate in Interactive and Cognitive Environments
JD-ICE

XXXVIII cycle

Acknowledgements

This PhD Thesis has been developed in the framework of, and according to, the rules of the Joint Doctorate in Interactive and Cognitive Environments JD-ICE with the cooperation of the following Universities:

Università degli Studi di Genova (UNIGE)

DITEN - Dept. of Electrical, Electronic, Telecommunications Engineering and Naval Architecture

ISIP40 - Information and Signal Processing for Cognitive Telecommunications

Primary Supervisor: Prof. Francesco Bellotti
Secondary Supervisor: Prof. Riccardo Berta

Queen Mary University of London (QMUL)

EECS - School of Electronic Engineering and Computer Science

CSI - Centre for Intelligent Sensing

Primary Supervisor: Dr. Changjae Oh
Secondary Supervisor: Prof. Andrea Cavallaro

As this journey comes to an end, I would like to express my heartfelt gratitude to all those who have been close to me over the past three years. First and foremost, I thank my family for their constant encouragement through times of difficulty and distance, and for their unwavering emotional and financial support. My deepest thanks go to Elena, my greatest supporter, who has lovingly accompanied me throughout this path, always ready to listen and to stand by me even in moments of despair. I am sincerely grateful to my supervisors, Prof. Riccardo Berta, Prof. Francesco Bellotti, Prof. Andrea Cavallaro, and Dr. Changjae Oh, for their invaluable theoretical and practical guidance, which has allowed me to grow both professionally and personally. I also wish to thank Luca, Marianna, Alessio, Matteo, Vafali, Hadi, Ossama, Ammar, and Lbaneen for creating a wonderful and joyful atmosphere in the laboratory. A special thank-you goes to Alessandro (Pigo), my companion through countless adventures, for sharing with me the experiences of my master's thesis, our time in the lab, and our life in London as flatmates. I am grateful to all my friends at the Box for helping me get through these demanding years, and to my teammates at TheBox66 for giving me the chance to unwind on the pitch.

Acknowledgements

The author would like to thank all partners within the Hi-Drive project for their cooperation and valuable contribution. [This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006664. The sole responsibility of this publication lies with the authors. Neither the European Commission nor CINEA – in its capacity of Granting Authority – can be made responsible for any use that may be made of the information this document contains.]

Abstract

This thesis investigates data-centric methodologies for automated driving and quantifies their impact from simulation to embedded deployment. First, we study decision transformer in offline reinforcement learning on a lightweight 2D platform, highlighting the limitations of high-level driving simulators. We hence transition to a more complex, reliable 3D pipeline, introducing a custom 60 km highway map engineered for long, reproducible runs, designed to overcome the limitations of the predefined driving environment within CARLA simulator.

On this infrastructure, two complementary problems are addressed under matched conditions. For lane change intention, human-in-the-loop sessions with 50 drivers yield >3,400 annotated maneuvers as deep-learning-ready time series. Formulating intention as time-to-lane-change regression enables anticipation up to 4 s; across 1D-CNN, LSTM, GRU, and Transformer baselines the best model attains RMSE = 0.51 s and MAE = 0.30 s, while a compact 1D-CNN offers favorable accuracy–efficiency trade-offs. For scenario understanding, an ASAM OpenSCENARIO 2.0-aligned generator produces labeled MP4 clips spanning nine highway classes; a 3D residual CNN establishes reference performance and highlights systematic confusions among visually similar behaviors.

Finally, we address robust traffic light detection on embedded hardware. A modular architecture comprising a frame checker, frame cleaner, takeover Request logic, automated driving revert logic, and a quantized YOLOv11 detector sustains near-nominal accuracy under sensor anomalies. On NVIDIA Jetson Orin Nano, the system achieves 84.6% mAP50 in clean conditions and 84.4–84.8% mAP50 under simulated rain and blur, operating in real time.

The contributions include: (i) a reproducible CARLA highway environment; (ii) a public lane change framework comprising a dataset and a data generation pipeline; (iii) a configurable scenario generator and benchmark; and (iv) an embedded, anomaly-aware traffic lights detection pipeline with explicit takeover request integration. Results demonstrate that rigorous dataset design and controlled simulation substantially improve evaluation fidelity and support deployable automated driving functions.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

1D One-Dimensional

2D Two-Dimensional

3D Three-Dimensional

AD Automated Driving

ADAS Advanced Driver Assistance Systems

ADAV Adversarial Defense for Autonomous Vehicles

ADF Automated Driving Function

ADRDM Automated Driving Revert Decision Maker

AP Average Precision

ASA Adaptive Square Attack

ASAM Association for Standardization of Automation and Measuring Systems

BSTLD Bosch Small Traffic Light Dataset

CARLA Car Learning to Act

CBAM Convolutional Block Attention Module

CDF Common Data Format

CL Context Length

CMOS Complementary Metal–Oxide–Semiconductor

CNN Convolutional Neural Network

CPU Central Processing Unit

DL Deep Learning

DNN Deep Neural Network

DQN Deep Q-Network

DRL Deep Reinforcement Learning

DT Decision Transformer

DTLD DriveU Traffic Light Dataset

EV Ego Vehicle

FCh Frame Checker

FCl Frame Cleaner

FLOP Floating Point Operation

FP16 16-bit Floating Point

FPS Frames Per Second

FR Free Ride

GAN Generative Adversarial Network

GPU Graphics Processing Unit

GRU Gated Recurrent Unit

HMI Human-Machine Interaction

HMM Hidden Markov Model

HNPV Hazardous Non-Player Vehicle

IoU Intersection over Union

IQA Image Quality Assessment

ISO International Organization for Standardization

KB Kilobyte

LC Lane Change

LIDAR Light Detection and Ranging

LK Lane Keeping

LLC Left Lane Change

LSH Logical Scenario Handler

LSTM Long Short-Term Memory

LV Leading Vehicle

mAP50 Mean Average Precision at 50% IoU

mAP Mean Average Precision

MB Megabyte

MDP Markov Decision Process

ML Machine Learning

MOBIL Minimizing Overall Braking Induced by Lane Changes

MP4 MPEG-4 Part 14

MSE Mean Squared Error

NDS Naturalistic Driving Study

NGSIM Next Generation Simulation

NHTSA National Highway Traffic Safety Administration

NMS Non-Maximum Suppression

NN Neural Network

ODD Operational Design Domain

PRC Precision-Recall Curve

PREVENTION PREdiction of VEhicles iNTentIONs

R3D 3D Residual Convolutional Network

RADAR Radio Detection and Ranging

RF Random Forest

RLC Right Lane Change

RL Reinforcement Learning

RMSE Root Mean Squared Error

RNN Recurrent Neural Network

ROC Receiver Operating Characteristic

RR Recovery Rate

SAPNet Segmentation-Aware Progressive Network

SE Squeeze-and-Excitation

SGD Stochastic Gradient Descent

SHRP2 Second Strategic Highway Research Program

SNPV Standard Non-Player Vehicle

SOTIF Safety Of The Intended Functionality

SPPF Spatial Pyramid Pooling – Fast

SVM Support Vector Machine

THW Time Headway

TLD Traffic Light Detection

TODM Takeover Decision Maker

TOR Takeover Request

TPSeNCE Triangular Probability Similarity and Semantic Noise Contrastive Estimation

TRPO Trust Region Policy Optimization

TTLC Time To Lane Change

TW Time Window

UI User Interface

VRAM Video Random Access Memory

VR Virtual Reality

YOLO11n YOLOv11 Nano

YOLOPv2 You Only Look Once for Panoptic Driving, version 2

YOLOvX You Only Look Once, version X

YOLO You Only Look Once

Chapter 1

Introduction

1.1 Research Questions and Objectives

The overarching research question of this thesis is:

> *How can data-centric design principles enable the development of robust, deployment-ready Automated Driving Functions across decision-making, maneuver recognition, scenario understanding, and safety-critical perception?*

This question is decomposed into four research objectives (ROs):

RO1. Scalable data infrastructure for highway driving. Design and release an open-source simulation framework that allows researchers to collect large-scale datasets of highway driving behavior without manual annotation effort, and validate it through human-in-the-loop experiments. After acknowledging the current lack of open-source frameworks and publicly available datasets for lane change prediction, this infrastructure will directly enable researchers to generate customized driving datasets tailored to specific operational design domains, facilitating rapid prototyping and validation of prediction models without the overhead of real-world data collection campaigns.

RO2. Early lane change intention recognition. Construct and publicly release a human-in-the-loop dataset of annotated lane change maneuvers collected under controlled yet realistic conditions, and benchmark sequence modeling architectures for Time-To-Lane-Change regression up to 4 s in advance. By leveraging the framework from RO1, this dataset will support the development of predictive advanced driving assistance systems functions that can provide earlier and more natural warnings to drivers, as well as enable smoother trajectory planning in automated lane change systems by anticipating maneuvers with sufficient time buffer.

RO3. Synthetic scenario generation and detection. Develop a public dataset of labeled highway driving scenarios spanning multiple classes, leveraging real-world statistical distributions, and evaluate its utility for both in-simulation classification and zero-shot transfer to real-world footage.

RO4. Robust embedded perception under sensor anomalies. Design a modular embedded architecture for traffic light detection that integrates anomaly detection, image recovery, and fail-safe takeover request logic, and validate it on resource-constrained hardware under representative sensor perturbations. By grounding validation on real-world image datasets with sensor noise patterns, this work aims to bridge the simulation-to-reality gap and ensure that robustness guarantees hold under actual deployment conditions.

Table 1.1 summarises the primary contributions of this thesis, mapping each to its research objective, the chapter where it is presented, and the associated publications.

Table 1.1 Summary of thesis contributions.

RO	Contribution	Description	Ch.	Publications
RO1	**Open-source CARLA highway framework**	First open-source simulation stack for Deep Learning-ready highway data collection: 60 km custom map, procedural traffic, first-person cockpit, standardised telemetry. Directly addresses the absence of stable, long-duration highway environments in CARLA.	3	[1, 2]
RO2	**DMIR dataset and time-to-lane-change regression baseline**	First open, human-in-the-loop lane change dataset collected in a physics-aware 3D simulator (50 drivers, $>$3 400 annotated maneuvers, $>$1 000 downloads). Benchmarks four sequence architectures for time-to-lane-change regression up to 4 s, achieving RMSE = 0.51 s.	4	[3, 2]
RO3	**Synthetic highway scenario dataset and generator**	First open dataset of labeled highway driving scenarios (9 classes, $\sim$3 M clips) grounded in real-world MOOVE distributions and compliant with ASAM OpenSCENARIO 2.0. Reference R3D-SE classifier achieves 91.9 % accuracy in simulation and 83.3 % zero-shot on PREVENTION.	4	[4–6]
RO4	**Modular embedded traffic light detection architecture with take over request**	First embedded architecture that combines frame anomaly detection, image recovery, and takeover request logic for robust traffic light detection. Quantized YOLOv11 on NVIDIA Jetson Orin Nano sustains 84.6 % mAP50 under perturbations at $\geq$10 FPS.	5	[7]
Instrumental contribution				
–	**Offline deep reinforcement learning with Decision Transformer in highway-env**	Empirical demonstration of the performance–safety trade-off of sequence modeling in 2D simulation, motivating the transition to physics-aware 3D environments for all subsequent data collection.	3	[8]

The four primary contributions address distinct but interconnected stages of a data-centric automated driving pipeline. RO1 provides the infrastructure; RO2 and RO3 exploit it to produce two complementary open resources targeting different temporal and semantic granularities of highway behavior; RO4 closes the loop by demonstrating that the same data-centric discipline carries through to deployment under real-world constraints. Although lane change intention recognition and traffic light detection address different functions, they share a common methodological challenge: the absence of deployment-ready, annotated datasets at scale. Both contributions respond to this gap by building purpose-designed data pipelines before training any model, placing data design at the center of the research rather than treating it as a preprocessing step.

1.2 Author's Publications and Contributions

The research presented in this thesis is based on the following publications. For each work, the author's specific contribution is outlined below.

1.2.1 Journal Articles

1. **L. Forneris**, R. Berta, M. Fresta, L. Lazzaroni, H. Rojhan, C. Oh, A. Pighetti, H. Ballout, F. Tango, and F. Bellotti, "A Deployment-Oriented Simulation Framework for Deep Learning-Based Lane Change Prediction," *IEEE Signal Processing Letters*, vol. 33, pp. 136–140, 2026. DOI:10.1109/LSP.2025.3638676

   Author's contribution: Led the design and implementation of the CARLA-based simulation framework, especially the custom highway map. Attended meeting with industrial partners, and coordinated internship students working on data collection, traffic management, and model training. Wrote the first full draft of the paper.

2. M. Cossu, R. Berta, L. Lazzaroni, **L. Forneris**, M. Fresta, A. Pighetti, A. Pilla, J.-L. Sauvaget, H. Touil, L. Durville, T. Griffon, G. B. Nejma, F. H. Selem, and F. Bellotti, "Generation of Synthetic Datasets of Highway Driving Scenarios," *IEEE Transactions on Intelligent Transportation Systems*, under review, manuscript no. T-ITS-25-12-6132, 2025.

   Author's contribution: Rebuilt the full scenario generation pipeline from scratch, coordinating thesis students throughout the process. Redesigned the dataset, and

produced the updated experimental results both for generation and detection pipeline. Wrote the corresponding sections of the manuscript.

3. R. Berta, L. Lazzaroni, A. Capello, M. Cossu, **L. Forneris**, A. Pighetti, and F. Bellotti, "Development of Deep-Learning-Based Autonomous Agents for Low-Speed Maneuvering in Unity," *Journal of Intelligent and Connected Vehicles*, vol. 7, no. 3, pp. 229–244, Sep. 2024. DOI:10.26599/JICV.2023.9210039

   Author's contribution: Contributed to design reward function and contributed to paper drafting.

4. L. Lazzaroni, F. Bellotti, A. Pighetti, **L. Forneris**, M. Fresta, ,H. Ballout, C. Oh, S. Pantawane, and R. Berta "An Embedded System Architecture for Robust Deep Learning-Based Traffic Light Detection," *IEEE Sensors Journal*, vol. 25, no. 24, pp. 44859–44874, Dec. 2025, DOI:10.1109/JSEN.2025.3627477

   Author's contribution: Contributed to defining the problem and planning the tasks to be carried out, explored literature. Created annotated datasets, and reported several results. Significantly contributed to the paper drafting.

5. **L. Forneris**, A. Pighetti, L. Lazzaroni, F. Bellotti, A. Capello, M. Cossu, and R. Berta, "Implementing Deep Reinforcement Learning (DRL)-Based Driving Styles for Non-Player Vehicles," *International Journal of Serious Games*, vol. 10, no. 4, pp. 153–170, 2023. DOI:10.17083/ijsg.v10i4.638

   Author's contribution: Implemented longitudinal and lateral controllers, collected results and significantly contributed to the drafting of the paper.

1.2.2 Conference Papers

6. **L. Forneris**, F. Bellotti, R. Berta, L. Lazzaroni, and C. Oh, "Exploring Decision Transformer for Highway Automated Driving," in *Proc. APPLEPIES 2024*, Lecture Notes in Electrical Engineering, vol. 1369, pp. 123–130, Springer, 2024. DOI:10.1007/978-3-031-84100-2_15

   Author's contribution: Designed the full experimental pipeline: offline dataset collection via a DQN behavior policy, Decision Transformer architecture implementation, hyperparameter configuration, and closed-loop evaluation. Performed comparative benchmarking against the DQN and a random agent. Wrote the manuscript.

7. **L. Forneris**, F. Bellotti, R. Berta, M. Cossu, M. Fresta, R. Rezaieamale, H. Rojhan, V. Soltanmuradov, F. Tango, and L. Lazzaroni, "Setting Up a Lightweight Simulation Environment for Automated Driving Dataset Collection," in *Proc. APPLEPIES 2024*, Lecture Notes in Electrical Engineering, vol. 1369, pp. 156–163, Springer, 2025. DOI:10.1007/978-3-031-84100-2_19

   Author's contribution: Conceived and built the custom 60 km highway map in RoadRunner/MATLAB, implemented the procedural NPV traffic management algorithm, coordinated internship students in configurating the cockpit and preliminary data collection. Wrote the manuscript.

8. A. Pighetti, **L. Forneris**, L. Lazzaroni, F. Bellotti, A. Capello, M. Cossu, A. De Gloria, and R. Berta, "High-Level Decision-Making Non-Player Vehicles," in *Proc. GALA 2022*, Lecture Notes in Computer Science, pp. 223–233, Springer, 2022. DOI:10.1007/978-3-031-22124-8_21

   Author's contribution: Ran several experiments and focused on the implementation of the longitudinal and lateral controllers. Significantly contributed to the paper drafting.

9. A. Capello, **L. Forneris**, A. Pighetti, F. Bellotti, L. Lazzaroni, M. Cossu, A. De Gloria, and R. Berta, "Investigating High-Level Decision Making for Automated Driving," in *Proc. APPLEPIES 2022*, Lecture Notes in Electrical Engineering, pp. 307–311, Springer, 2023. DOI:10.1007/978-3-031-30333-3_41

   Author's contribution: Ran several experiments and focused on the implementation of the longitudinal and lateral controllers. Significantly contributed to the paper drafting.

10. A. Pighetti, F. Bellotti, C. Oh, L. Lazzaroni, **L. Forneris**, M. Fresta, and R. Berta, "Investigating Adversarial Policy Learning for Robust Agents in Automated Driving Highway Simulations," in *Proc. APPLEPIES 2023*, Lecture Notes in Electrical Engineering, pp. 124–129, Springer, 2024. DOI:10.1007/978-3-031-48121-5_18

    Author's contribution: Contributed to experimental design and performance evaluation of victim and adversarial agents. Collected metrics and results. Contributed to paper drafting.

11. O. L. Audi, F. Bellotti, R. Berta, Z. Liu, L. Lazzaroni, A. Pilla, H. Rojhan, and **L. Forneris**, "Improving Lane Change Performance in CARLA Through Curvature-Adaptive B-Spline Planning," in *Proc. APPLEPIES 2025*, Springer Nature Switzerland, pp. 212–217, 2025. DOI:10.1007/978-3-032-17174-0_29

Author's contribution: Coordinated and supervised the first author throughout the development of the work, proposed the B-spline trajectory planning approach, designed the simulation evaluation protocol, and revised the majority of the methodology.

12. R. Berta, F. Bellotti, M. Cossu, **L. Forneris**, L. Lazzaroni, and A. Pighetti, "RL-Based Generation of a Synthetic Automotive Driving Scenario Dataset," in *Proc. APPLEPIES 2024*, Springer Nature Switzerland, pp. 395–402, 2024. DOI:10.1007/978-3-031-84100-2_47

    Author's contribution: Contributed to the development of the reinforcement learning component of the pipeline, including agent design, training, and evaluation for synthetic scenario generation.

13. A. Pighetti, **L. Forneris**, F. Bellotti, A. Capello, M. Cossu, G. Gioco, and R. Berta, "A Teacher-Configurable Scoring System for Serious Games," in *Proc. GALA 2023*, Springer Nature Switzerland, pp. 254–263, 2023. DOI:10.1007/978-3-031-49065-1_25

    Author's contribution: Participated in stakeholder and partner meetings to define system requirements, contributed to the design of the scoring algorithm, and co-wrote significant portions of the manuscript.

14. M. Cossu, R. Berta, **L. Forneris**, M. Fresta, L. Lazzaroni, J.-L. Sauvaget, and F. Bellotti, "YOLOP-Based Pre-Processing for Driving Scenario Detection," in *Proc. APPLEPIES 2023*, Springer Nature Switzerland, pp. 418–423, 2023. DOI:10.1007/978-3-031-48121-5_60

    Author's contribution: Contributed to the design of the YOLOP-based feature extraction pipeline and its integration as a pre-processing stage for the residual 3D convolutional network.

15. **L. Forneris**, R. Berta, A. Capello, M. Cossu, M. Fresta, F. Tango, and F. Bellotti, "A Synthetic Dataset Generator for Automotive Overtaking Maneuver Detection," in *Proc. APPLEPIES 2023*, Springer Nature Switzerland, pp. 364–369, 2023. DOI:10.1007/978-3-031-48121-5_52

    Author's contribution: Designed full development of the tool, including scenario generation logic, road curvature parametrization, and multi-format export pipeline using the MATLAB Automated Driving Toolbox. This work laid the groundwork for the custom highway map and the subsequent dataset generation pipeline developed in later publications.

1.3 Introduction

The development of safe and reliable automated driving (AD) systems is a multidisciplinary challenge that spans artificial intelligence (AI), robotics, and safety-critical engineering. While advances in perception, planning, and control are often credited as the main drivers of progress, this thesis is based on a complementary premise: the design, quality, and scope of datasets are equally decisive. Any data-driven model is bounded by the information on which it is trained and validated. Shortcomings in scale, diversity, realism, or annotation structure propagate into systematic bottlenecks across the AD lifecycle, from prototyping to verification and deployment.

Real-world data remains indispensable, yet it is structurally constrained. Large-scale acquisition is costly and slow, and hazardous or rare events, which are crucial for safety assessment, are seldom observed. Manual annotation in uncontrolled conditions is laborious and raises privacy and safety concerns, and uncontrolled sampling can leave important regions of the scenario space underrepresented. Simulation has therefore become a key complement. Modern simulators make it possible to generate extensive, precisely annotated datasets, to instantiate rare or hazardous scenarios in a controlled way, and to vary agents, sensors, and environmental conditions for coverage, reproducibility, and stress testing. In practice, the most robust results come from combining the two sources: real data for grounding physics, calibrating statistics, and validating assumptions, and simulation for expanding the long tail, exploring counterfactuals, and running scalable regression tests. Because domain shift is always a risk, sim-to-real techniques, sensor-accurate rendering, randomized assets and conditions, and adaptation methods are required to narrow the gap.

Deep Learning (DL) has accelerated Automated Driving Functions (ADFs) across context perception [9–11], predictive vehicle control [12–15], and decision-making (DM) [16, 17]. DM governs vehicle navigation [18] and maneuver execution [19], which requires adaptation to complex and dynamic environments under real-time constraints. Traditional DL often struggles in such interactive settings [20, 21]. Deep Reinforcement Learning (DRL) addresses this limitation by coupling representation learning with trial-and-error optimization [20, 22, 23]. Online DRL updates policies during interaction, which can be impractical when safety and data cost dominate. Offline DRL instead leverages fixed datasets [24, 25], a valid alternative for safety-critical domains. Recent progress in sequence modeling with attention, notably Transformers [26], has shown promise for RL because of improved long-term credit assignment and temporal context modeling [27], with gains in both offline [28] and online [29] regimes. These algorithmic advances, however, are only as useful as the data on which they are trained and evaluated, which brings the discussion back to the central bottleneck this thesis addresses.

Developing and evaluating DL-based ADFs requires simulation environments that can produce data at scale while remaining physically credible. High-level simulators enable rapid iteration for highway DM but limit safety, comfort, and transferability. Two-dimensional platforms typically omit vehicle dynamics, suspension and drivetrain effects, physically grounded sensing, and realistic weather or illumination. Under these simplifications, apparently good closed-loop behavior may reflect artifacts of idealized kinematics and perception, which makes it harder to probe edge cases such as wet pavement or complex junctions and to calibrate comfort and risk proxies based on jerk, slip, or braking envelopes [30–32]. Surveys of AD simulators converge on the need for physics-aware 3D environments when credible evaluation is the goal. Fidelity in dynamics, sensing, traffic, and rendering improves behavioral validity and measurement reliability, provided scenarios remain reproducible and computationally tractable [30, 32]. Evidence from vehicular-network and physics-centric studies shows that realistic geometry and physics materially change outcomes by coupling mobility, communication, occlusion, and environment [33]. Manipulating simulator fidelity yields measurable differences in presence and driving behavior, which further supports the use of richer simulators when research questions involve comfort, risk, or fine-grained control [34].

The thesis follows a data-centric trajectory in which each chapter addresses a distinct problem in the construction, validation, and exploitation of datasets for AD. The progression is deliberate: the work begins with algorithmic prototyping in a simplified 2D environment, where full control over the scenario allows rapid iteration but at the cost of physical realism; it then moves to human-in-the-loop data collection in a sensor-realistic 3D simulator, where richer signals and real driver behavior enable more meaningful modeling; it expands to multi-class scenario generation and classification, where complexity increases from individual maneuvers to structured traffic situations; and it culminates in a safety-critical perception function evaluated on real-world multi-dataset imagery under hardware constraints.

Each step in this progression is motivated by concrete limitations of the previous one. A 2D simulator such as highway-env [35] allows fast policy iteration and offline dataset collection, but the absence of vehicle dynamics, sensor noise, and physical continuity means that models trained or evaluated there cannot be trusted to generalise. This motivates the transition to CARLA [36], a physics-aware 3D simulator, where realistic sensing and dynamics make the data meaningful. However, CARLA's built-in highway maps suffer from collision anomalies, uneven pavement, junction artifacts, and limited road variability [37–39], which contaminate logs and interrupt sessions. This motivates the design of a custom 60 km highway map with balanced curvature, clean lane topology, minimal scenery for stable frame rates, and a standardised telemetry pipeline. Once a stable, instrumented environment

exists, collecting human driving data at scale becomes tractable, and the absence of open, DL-ready datasets of lane change (LC) maneuvers with ego-centric annotations makes this collection directly useful. Extending the same setup from individual maneuvers to structured traffic scenarios requires a generation layer on top of the simulator, grounded in real-world distributions, which produces an open dataset of labeled highway scenarios. Finally, a safety-critical perception function (i.e., traffic light detection (TLD) under sensor anomalies) closes the loop by demonstrating that the same data-centric discipline carries through to embedded deployment.

LC intention recognition, highway scenario detection, and TLD address different functions and different points in the AD stack. What they share is a data problem: in both cases, existing public datasets are either too narrow, too coarsely annotated, or structurally incompatible with the evaluation the task requires. Rather than adapting the research question to fit available data, both contributions start from the data gap and build upward, designing the collection or generation pipeline before choosing the model. This order of operations is the common thread across the experimental chapters.

Chapter 2 establishes the background on which the experimental work rests. It surveys simulation platforms from lightweight 2D environments to physics-aware 3D simulators, reviews existing datasets for LC prediction and scenario detection, and covers the core DL architectures and evaluation protocols used in subsequent chapters. It also characterises the limits of the state of the art, identifying the specific gaps that motivate each contribution.

Building on this groundwork, Chapter 3 first demonstrates empirically why 2D simulation is insufficient for credible data collection: an offline DRL experiment with a decision transformer (DT) in highway-env reveals a performance–safety trade-off that cannot be properly characterised without physical dynamics and sensor realism. This motivates the transition to CARLA, where the chapter introduces a custom 60 km highway map, a first-person cockpit, procedural traffic generation, and a standardised telemetry pipeline that exports DL-ready time series, namely the shared infrastructure for all subsequent data collection.

Chapter 4 exploits this infrastructure for two complementary problems. The first is LC intention recognition: 50 drivers collect over 3,400 annotated maneuvers on the custom map, and four sequence architectures are benchmarked on time-to-lane-change (TTLC) regression up to 4 s in advance. The second is highway scenario generation and detection: the same highway environment is used to synthesise labeled video clips for nine scenario classes grounded in real-world distributions, with a reference R3D classifier evaluated both in simulation and zero-shot on real-world PREVENTION footage.

Finally, Chapter 5 consolidates the accumulated experience in a safety-critical perception function: robust TLD under sensor anomalies. The focus is on deployment with real-world data and real-time constraints. The chapter introduces a modular embedded architecture that integrates anomaly detection, image recovery, and fail-safe protocols with take over request (TOR), and evaluates a quantized YOLOv11 detector [40, 41] under representative perturbations such as TPSeNCE rain [42] and Poltergeist blur [43]. On NVIDIA Jetson Orin Nano hardware, the system maintains near-nominal mAP50 and implements a controlled fallback when remediation is insufficient.

Three themes connect the chapters. First, careful data design is as decisive as model choice. The thesis contrasts signals and video, clarifies windowing and labeling strategies, and uses subject-independent splits to avoid optimistic estimates [44]. Second, holding geometry and tooling fixed enables comparable evaluation across tasks and faster iteration. The same 60 km highway map, cockpit, traffic generation, and logging pipeline support both maneuver-level and scenario-level studies, which reduces confounds due to heterogeneous assets. Third, robustness requires thinking beyond clean benchmarks. The work systematically injects variability, covers edge cases with controllable generators, and designs failure handling and TOR logic explicitly.

Together, these chapters argue for a data-centric path to AD in which reliable functions emerge from consistent choices across data design, simulation, modeling, and runtime safeguards.

Chapter 2

Background and Related Work

This chapter surveys the literature along four axes, each corresponding to one or more of the thesis contributions.

- Sequence modeling and offline reinforcement learning (Section 2.1): from online DRL to offline sequence models, culminating in the Decision Transformer (DT), whose application to highway driving is studied in Chapter 3.
- LC intention prediction (Section 2.2): from rule-based heuristics to deep sequence models, with a focus on prediction horizon and the regression-vs-classification dichotomy that motivates the LC recognition contribution of Chapter 4.
- Driving scenario generation and detection (Section 2.3): scenario formalisms (PEGASUS, OpenSCENARIO 2.0), real-world and synthetic datasets, and recognition architectures, culminating in Cossu's PhD work [45] as the primary benchmark for the scenario generation contribution of Chapter 4.
- Perception robustness and traffic light detection (Section 2.4): YOLO detector evolution, sensor vulnerability taxonomy, and image recovery methods, establishing the open problem addressed in Chapter 5.

For each topic the survey closes with a summary of open gaps. Section 2.5 consolidates all gaps in a single table that maps each to the chapter addressing it. Methodology choices (i.e., simulator configuration, dataset generation parameters, selection criteria, etc.) are deferred to the respective experimental chapters.

2.1 Sequence Modeling and Offline Reinforcement Learning

DM for AD requires agents to navigate complex, dynamic environments in real time. DRL addresses this by combining the representational power of DL with trial-and-error optimization, enabling agents to learn policies through direct interaction with the environment [20, 22, 23].

Within DRL, two paradigms exist. Online RL continuously updates policies from real-time feedback, which is effective but costly and unsafe for real-world vehicles [23]. Offline RL instead learns from fixed, pre-collected datasets [24, 25], avoiding live exploration and making it better suited to safety-critical domains. A known limitation of offline RL is distributional shift: the learned policy may encounter state-action pairs poorly represented in the dataset, leading to extrapolation errors [46].

Recent progress in sequence modeling with attention, notably Transformers [26], has opened a new perspective on offline RL. Rather than learning explicit value functions, the DT [28] frames RL as a sequence modeling problem: given a context of past returns, states, and actions, it autoregressively predicts the next action using a GPT-style causal decoder. The return-to-go, defined as:

$$\hat{G}_t = \sum_{t'=t}^{T} r_{t'} \tag{2.1}$$

serves as a conditioning signal, allowing the model to target desired performance levels at inference time. This approach has shown strong results in offline benchmarks such as Atari and MuJoCo [28], leveraging the long-range temporal credit assignment of the Transformer architecture.

Highway-env [47] is the standard lightweight 2D testbed for developing and evaluating DRL-based highway DM agents. It models the Ego Vehicle (EV) and surrounding traffic as agents in a discrete Markov decision process (MDP) [48]: the state encodes positions, velocities, and lane configurations; the EV selects actions (accelerate, brake, lane change) to maximize a reward balancing safety, efficiency, and comfort [49, 50]. Its Gym-compatible interface, transparent formulation, and low rendering cost make it suitable for large-scale offline dataset collection and rapid policy iteration. However, as a 2D abstraction, it omits vehicle dynamics, sensor noise, and physical continuity, which limits the credibility of safety and comfort evaluations and hinders generalization [30, 32]. This motivates the transition to a physics-aware 3D simulator described in Chapter 3.

Gap G0: the DT has been validated on Atari and MuJoCo benchmarks but its application to multi-agent highway scenarios with discrete action spaces and explicit safety constraints

remains underexplored. Furthermore, the performance-safety trade-off of offline sequence modeling in 2D simulation cannot be properly characterized without physical dynamics and sensor realism. This gap is addressed in Chapter 3.

2.2 Lane Change Intention Prediction

Predicting when and whether a driver will execute a LC is a core requirement for safe Advanced Driver Assistance Systems (ADAS) and AD. Given a time window of vehicular observations ending at time t, the goal is either to estimate the TTLC (regression formulation)

$$\tau = t_{\mathrm{LC}} - t \in [0, T] \tag{2.2}$$

where t_{LC} is the time at which the vehicle crosses the lane marking and t is the current observation time, or to assign a label $y \in \{\mathrm{LK}, \mathrm{LLC}, \mathrm{RLC}\}$ (classification formulation) at a prediction horizon h seconds before maneuver onset, where LK denotes lane keeping, LLC a left LC, and RLC a right LC. A prediction horizon $h = 0$ corresponds to the moment of lane marking crossing, while $h = T$ is the maximum anticipation horizon considered. Observations typically include EV kinematics (speed, acceleration, lateral position, indicator status, pedal activity), road geometry (lane boundary distances, curvature), and the kinematic state of neighboring vehicles [51–53].

This is a supervised sequence learning problem: a model is trained to infer driver intent from historical signal windows, without any assumption of optimizing a reward or controlling the vehicle. This distinguishes LC intention prediction from decision-making frameworks such as DRL, where the vehicle itself is the agent acting within an MDP [48]. The MDP formulation is relevant to the autonomous agent of Chapter 3, not to the intention recognition task surveyed here.

The survey is structured around three generations of approaches: rule-based (Section 2.2.1), classical Machine Learning (ML) (Section 2.2.2), and DL (Section 2.2.3). Existing datasets are reviewed in Section 2.2.4 and gaps are summarized in Section 2.2.5.

2.2.1 Rule-Based Approaches

Early methods for predicting LCs relied on explicit rules or driver behavior models derived from traffic theory [54–56]. These approaches typically apply predefined thresholds to infer LC intention on kinematic variables, checking whether the relative speed or gap to nearby vehicles exceeds certain limits. Agent-based decision models, game-theoretic frameworks, and logical conditions have all been applied in this manner [57].

Rule-based methods are valued for their interpretability, low computational complexity, and robustness in well-defined scenarios. However, they struggle to generalize to diverse drivers and traffic conditions, and they offer limited anticipation time. Khelfa and Tordeux [57] showed that a classic Minimizing Overall Braking Induced by Lane Changes (MOBIL) LC model [58] was significantly outperformed by a data-driven naive Bayes predictor, illustrating the need for learning-based techniques.

2.2.2 Classical Machine Learning Approaches

Classical ML methods treat LC intention as a data-driven inference problem over short time windows preceding the maneuver. Support Vector Machines (SVMs), Hidden Markov Models (HMM), Bayesian models, and shallow Neural Networks (NNs) deliver fast inference and reasonable accuracy for short horizons, typically 1–2 s before LC, but rely on careful feature engineering and degrade as the anticipation window grows [51, 59, 60, 52].

Mandalia and Salvucci [51] demonstrated that an SVM can detect imminent LC with high fidelity; a hybrid SVM-NN achieved approximately 94% accuracy for LK and 78% for LC detection on a short driving data window. While effective in the short term, these models struggle to capture the early subtle indicators of a LC several seconds ahead, since in real-world driving, drivers begin checking rear-view mirrors more than 4 s before a maneuver [61, 52].

2.2.3 Deep Learning Approaches

With DL, temporal sequence models learn richer, longer-range motion patterns. Recurrent architectures (LSTM, GRU, Bi-LSTM) have been widely adopted for LC prediction, achieving F1 scores of 0.82–0.90 at 2–3 s horizons on NGSIM or naturalistic logs [62, 63]. More complex architectures model interaction dynamics and routinely outperform classical classifiers on the same inputs [53].

Video-centric methods exploit Convolutional NNs (CNNs), 3D-CNNs, or CNN-Recurrent NN (RNN) hybrids to leverage appearance and motion cues, again mostly in classification settings. Zhang et al. [52] applied a multi-channel CNN to trajectory time-series data, achieving near state-of-the-art accuracy at around 3 s before maneuver, with claimed feasibility up to 7 s but with weaker reliability at that longer horizon. Ensemble methods on naturalistic datasets such as SHRP2 NDS report high accuracy when enriched with curated features [64], but still in classification settings.

In general, most DL classifiers are highly reliable in the 2–3 s range, with performance dropping off beyond 4 s. Virtually all existing DL methods remain classification-based:

continuous TTLC regression at horizons of 0–4 s is largely unexplored, limiting the utility of these methods for smooth trajectory planning and early warning systems.

2.2.4 Datasets for Lane Change Research

Tables 2.1 and 2.2 summarize proprietary and public datasets relevant to LC research.

Proprietary datasets

Toyota Highway [65], BMW NDS [66], and GAIA-1 [67] were collected or built for internal R&D and are not publicly accessible. GAIA-1, a generative model by Wayve trained on 4,700 h of driving data, does not release model weights or training data. Int2Planner [68] covers urban scenarios only. CRASH [69] contains only 50 scripted simulated LCs.

Public datasets

The NGSIM dataset [70] provides trajectories for specific highway stretches but lacks EV-centric features such as road geometry, lane boundary distances, pedal activity, and indicator status, and is affected by frame noise requiring filtering [71]. HighD [72] covers 16.5 h on German highways but misses up/downstream context and suffers from mandatory LC contamination [73]. PREVENTION [74] does not include EV maneuvers. HDD [75] covers mostly non-highway scenarios with LC events below 3% of the data. Brain4Cars [76], the only public dataset with LC annotations, contains only 274 LC events from 10 drivers. Waymo OMD [77] targets motion forecasting with 8 s horizons and does not provide ego-intention labels in a TTLC regression format.

A structural limitation shared by all public datasets is the absence of a generation engine and DL-ready time series: data requires preprocessing (video, point cloud, CAN logs), which is time-consuming. None provide continuous TTLC annotations in the $[0, 4]$ s range required by a regression formulation. A further consequence is class imbalance: LC events are rare in naturalistic recordings, making it difficult to train balanced models without distorting the natural distribution.

Table 2.1 Comparison of related datasets (Proprietary).

Dataset	Data Type & Format	Size and Scope	Intent. window	Real	Open	LC	Limitations
Toyota [65]	LIDAR, Radar, GPS	∼20 h US highways	3 s	✓	×	✓	Proprietary; limited diversity
GAIA-1 [67]	Multi-cam, CAN	4700 h London	N/A	✓	×	×	Weights/Data not released
Int2Planner [68]	Bird-eye tensors	∼680k scenarios	∼6 s	×	×	×	Synthetic; Urban only
BMW NDS [66]	CAN logs	500k LC events	2 s	✓	×	✓	Class imbalance; short window
CRASH [69]	Numeric vehicular	50 LCs (simulated)	4 s	×	×	✓	Scripted; very small scale
SHRP2 NDS [78]	6 cams, CAN, Radar	3100+ drivers	N/A	✓	○	✓	Restricted access; privacy terms
Ours [2]	Time-series; GT	∼1500 km; 3444 LCs	0–4 s	×	✓	✓	Simulated traffic; no weather

✓= Full support/Yes, ×= No support/Private, ○ = On request/Partial

Table 2.2 Comparison of related datasets (Public).

Dataset	Data Type & Format	Size and Scope	Intent. window	Real	Open	LC	Limitations
NGSIM [70]	Video and data trajectories	∼45 min/site; US highways	N/A	✓	✓	×	Frame noise; no EV-centric features
highD [72]	Drone trajectories	16.5 h, 110k vehicles	N/A	✓	✓	×	MLC contamination
PREVENTION [74]	2 cams, LIDAR, Radars	∼6 h, 540 km	N/A	✓	○	×	No EV maneuvers
HDD [75]	4 cams, LIDAR, CAN	104 h, 23k events	3 s	✓	✓	×	LC $< 3\%$; mostly non-highway
Brain4Cars [76]	Dual cam, GPS	274 LCs; 10 drivers	3 s	✓	✓	✓	Only 10 drivers; very small scale
Waymo OMD [77]	5 cams, LIDAR, HD map	∼103k scenarios	8 s	✓	✓	×	No ego-intention labels; 8s horizon
Ours [2]	Time-series	∼1500 km; 3444 LCs	0–4 s	×	✓	✓	Simulated; no weather; scripted traffic

✓= Full support/Yes, ×= No support, ○ = On request/Partial

2.2.5 Summary and Open Gaps

Table 2.3 compares the main LC prediction methods across model type, horizon, dataset, task formulation, and key limitation.

Table 2.3 LC intention prediction methods: model type, horizon, dataset, task, and key limitation.

Method	**Model type**	**Horizon**	**Dataset**	**Task**	**Key limitation**
Mandalia & Salvucci [51]	SVM + ANN	$\leq$1 s	Real	Classif.	Very short horizon
Dokania et al. [59]	HMM	$\sim$1 s	Real	Classif.	Markov assumption; no long-range
Khelfa & Tordeux [57]	Naive Bayes	$\sim$2 s	Sim.	Classif.	Limited horizon
wirthmuller et al. [62]	LSTM	3 s	NGSIM	Classif.	F1$\approx$0.82; no regression
Zhang et al. [52]	Multi-ch. CNN	3–7 s	NGSIM	Classif.	Weak at 4 s+; no regression
Gao et al. [53]	Dual-branch RNN	2–3 s	highD	Classif.	No continuous TTLC output
Xia et al. [61]	LSTM	$\leq$4 s	Real	Classif.	Binary only; no regression
Ours [2]	CNN / LSTM / Transformer	0–4 s	CARLA (synth.)	Regression	Simulated; no weather variation

Gap G1: no open, DL-ready dataset supports continuous TTLC regression for the EV in a controlled highway setting. Brain4Cars is the only public dataset with LC annotations but contains only 274 events from 10 drivers; NGSIM and highD lack EV-centric features; Waymo OMD provides 8 s forecasting horizons without EV-intention labels. No existing dataset is paired with a reproducible generation framework. This gap is addressed in Chapter 4 for the dataset and benchmark, and the data collection infrastructure is introduced in Chapter 3.

Gap G2: LC intention predictors are almost exclusively classifiers. Reliable regression of TTLC at 0–4 s is largely unexplored: the best DL classifiers degrade beyond 3 s, and no regression-based benchmark exists on a controlled, DL-ready dataset. This gap is addressed in Chapter 4.

2.3 Driving Scenario Generation and Detection

2.3.1 Safety Standards and Regulatory Frameworks

The development and validation of AD systems is increasingly governed by a layered set of safety standards that span functional safety, intended functionality, cybersecurity, and system-level assurance.

ISO 26262 [79] establishes the functional safety lifecycle for road vehicles, covering hazard analysis, safety requirements, and verification activities across hardware and software components. It addresses failures arising from malfunctions of electrical and electronic systems, but does not cover hazards arising from correct yet insufficient system behaviour.

ISO 21448 [80], known as SOTIF, complements ISO 26262 by addressing precisely this gap: hazards arising from functional insufficiencies such as insufficient sensor performance or algorithmic limitations, in the absence of hardware faults. Together, the two standards cover the full spectrum of safety-relevant failure modes relevant to perception-based ADFs.

UL 4600 [81], developed by Underwriters Laboratories and grounded in the work of Koopman et al. at Carnegie Mellon University [**?**], provides a framework for evaluating the safety case of fully autonomous products. Unlike ISO 26262, which follows a process-based V-model, UL 4600 is outcome-oriented: it requires the developer to construct and defend a structured safety argument covering machine learning components, edge cases, and the absence of a human operator in the loop. It is particularly relevant to the scenario-based testing and dataset design methodology developed in this thesis, as it explicitly addresses the question of when simulation-based evidence is sufficient to support a safety claim.

IEEE 7009 [82], defines fail-safe design methods for autonomous and semi-autonomous systems across robotics, vehicles, and industrial applications. It formalises the conditions under which a system should relinquish control or request human intervention, providing a normative basis for takeover request mechanisms in SAE L3/4 systems. The TOR architecture presented in Chapter 5 is aligned with the fail-safe principles this standard formalises.

ISO/SAE 21434 [83] addresses cybersecurity engineering across the road vehicle lifecycle, covering threat analysis, risk assessment, and secure development practices.

ISO/IEC 62443 [84] extends these principles to interconnected embedded architectures. Both are relevant to the sensor anomaly taxonomy discussed in Section 2.4.2, where camera perturbations such as Poltergeist represent hardware-level cyber threats.

2.3.2 Scenario Formalisms

The field of scenario-based development for ADS rests on a layered set of definitions that have been progressively refined from informal project-specific terminology toward mathematically founded, ontology-compatible frameworks. Scenarios concern a complex real-world design domain addressed with different terminological approaches. A widely adopted starting point is the work by Ulbrich et al., who define a scene as a snapshot of the environment at a given point in time, including static scenery, dynamic elements, all actors and observers, and the relationships between them [85]. A situation is the subject-specific (e.g., the EV) interpretation of one or more scenes, enriched with goals, risks, and possible evolutions, acting as a central data container for behavior planning and control. A scenario is then the temporal development between several scenes in a sequence driven by the actions and events of actors oriented towards their goals [85]. These definitions established a clear interface between perception, context modeling, and behavior planning, and have been widely reused and extended in later work.

Building on this conceptual basis, Andreotti et al. translate these verbal definitions into a set- and function-based mathematical framework for mixed-traffic simulations [86].

A scene from agent x's point of view at time t on space S is a 3-tuple:

$$E^x(S,t) = (V^x,\, C^x(V^x),\, f^x_{sa}(V^x)) \tag{2.3}$$

where V^x is the set of objects perceivable by x at time t (the x-object set); $C^x : V^x \to S$ is the x-position function, with $C^x(V^x)$ the corresponding x-position set; and $f^x_{sa} : V^x \to A^x$ is the x-states & attributes function, with $f^x_{sa}(V^x)$ the x-states & attributes set. The complete, omniscient scene is:

$$E(S,t) = (V,\, C(V),\, f_{sa}(V)) \tag{2.4}$$

The enlarged situation $\widetilde{N}^x(S,t)$ extends the scene with a goal-prediction component:

$$\widetilde{N}^x(S,t) = \left(V^x,\, C^x(V^x),\, f^x_{sa}(V^x),\, \hat{C}^x(V^x)\right) \tag{2.5}$$

where $\hat{C}^x : V^x \to S$ is the x-goals & values position function, mapping each object to its expected future position according to x. A relevance filter f^x_r reduces $\widetilde{N}^x$ to the situation from x's point of view:

$$N^x(S,t) = \left(V^x_r,\, C^x_r(V^x_r),\, f^{xr}_{sa}(V^x_r),\, \hat{C}^x_r(V^x_r)\right) \tag{2.6}$$

Finally, over a time set $T = \{t_0, \ldots, t_n\}$, a scenario from x's point of view is:

$$O^x_T(S) = \left(E^x_t(S),\, f^x_{gv},\, f^x_{ae}\right)_{t\in T} \tag{2.7}$$

where $f_{gv}^{x} : C^{x}(V^{x}) \to \hat{C}^{x}(V^{x}) \times T$ is the x-goals & values function, mapping current positions to expected goal positions and times; and $f_{ae}^{x} : C^{x}(V^{x}) \times T \to C^{x}(V^{x})$ is the x-actions & events function, describing actual motion over Δt.

Menzel et al. [87] connect scenario terminology to the ISO 26262 [88] safety lifecycle by introducing a hierarchy of abstraction levels. They distinguish functional scenarios (human-readable, qualitative descriptions suited for requirements engineering), logical scenarios (sets of scenarios with parameter ranges and optional probability distributions, bridging qualitative descriptions and test cases), and concrete scenarios (fully instantiated parameter sets executable reproducibly in simulation or on test tracks). This ladder allows scenarios to be systematically evolved along the V-model from high-level requirements to detailed verification activities.

Neurohr et al. [89] further refine these formalisms by giving precise linguistic and mathematical definitions for concrete, logical, and abstract scenarios, relating them to curves, mappings to sets of curves, and temporal logics, respectively. Their comparison of expressiveness, specification complexity, sampling, and monitoring clarifies the strengths and weaknesses of different scenario qualifications, bridging the gap between informal terminology and concrete scenario description languages. Kurian et al. [90] similarly propose self-contained mathematical definitions within a behavioral modeling framework tailored to formulating scenario-based testing (SBT) as an optimization problem under ISO 21448 [80].

Weber et al. [91] operationalize logical scenarios for safety assurance by proposing a framework that defines safety-relevant scenarios based on potential collisions, using the collision area on the subject vehicle and eight relative positions of the challenging object. Their catalog of base scenarios for controlled-access highways directly underpins the PEGASUS project [85]. Jeon et al. complement this with a simulation-based method that generates logical scenarios capturing both normal and hazardous situations, addressing cost, data bias, and limited hazard coverage in purely data-driven approaches [92].

Scenario definitions depend strongly on how the driving environment is modeled. Within the PEGASUS project, Scholtes et al. present the 6-Layer Model (6LM) as a structured description of traffic environments [93]. The 6LM organizes each scene into six layers: 6-layer model (6LM): (i) road geometry, (ii) road furniture and rules, (iii) temporary modifications, (iv) moving objects, (v) environmental conditions, and (vi) digital information [93]. This actor-independent model serves as a structured basis for scenario extraction, logical scenario description, ontology development, and perception-oriented analyses.

Ontology-based approaches offer complementary routes to precise scenario definition. De Gelder et al. develop an object-oriented framework in which scenarios and their building blocks are defined as classes with attributes, methods, and relationships, providing necessary

and sufficient characteristics for what qualifies as a scenario [94]. Bagschik et al. apply ontologies specifically to scene creation, using expert knowledge encoded in a knowledge base to automatically generate natural-language traffic scenes [95]. Both contributions underline the role of ontologies in bridging expert domain knowledge and systematic, tool-supported scenario generation.

Data-centric frameworks such as MetaScenario [96] address the practical problem of describing, storing, and indexing large volumes of scenario data. Chang et al. [96] introduce atom scenarios characterized via semantic graphs capturing the spatio-temporal evolution of interactions between traffic participants and the environment, enabling efficient scenario indexing for downstream machine learning and decision-making tasks. Pfeffer et al. [97] similarly propose a meta description of highway scenarios focused on the behavior of dynamic objects, using auto-encoders to analyze scenario uniqueness.

A key source of real-world statistical grounding for the scenario generation pipeline described in Chapter 4 is the MOOVE project [98, 99]. Launched in 2015 by the VEDECOM Institute in partnership with PSA, Renault, and Valeo, MOOVE aimed to build a large-scale reference database of real-world driving scenarios for the design and validation of automated driving functions. A fleet of nine instrumented vehicles, equipped with cameras, radars, and LiDARs, operated by human drivers, was deployed across 17 European countries, covering motorways, urban roads, rural roads, tunnels, and adverse weather conditions (rain, snow, fog). Over approximately 15,000 hours of driving, the fleet accumulated more than one million kilometres and generated 250 TB of raw sensor data. The raw logs were subsequently processed into a structured relational database of labeled scenarios, characterized by high-level parameters such as inter-vehicle distances, relative speeds, time headway distributions, and road geometry statistics. These empirical distributions, which cover scenario classes such as cut-in, cut-out, car-following, and free-ride, are used in Chapter 4 to parameterize the scenario generation pipeline.

2.3.3 Real-World Scenario Datasets

Real-world trajectory datasets are valuable for scenario extraction and motion forecasting. The Waymo Open Motion Dataset [77] contains 100,000 scenes of 20 s each at 10 Hz, covering interactive situations across six US cities. Argoverse 2 [100] provides three sub-datasets including 250,000 motion forecasting scenarios. The PREVENTION dataset [74, 101] covers 6 h of multi-sensor highway recordings with trajectory and LC annotations. SceNDD [102] provides 68 naturalistic driving sessions but without scenario category labels.

Drone-based datasets offer occlusion-free recordings suited for traffic behaviour analysis. ExiD [103] covers highway entries and exits. The INTERACTION dataset [104] targets

decision making in adversarial and cooperative scenarios. TrafficNet [105] is a library covering six critical driving categories.

The state of the art is rich in motion forecasting datasets but lacks publicly available datasets labeled for in-vehicle driving scenario detection, a gap addressed in Chapter 4.

2.3.4 Simulators and Synthetic Datasets

Simulators address the completeness dilemma in AD testing by enabling exhaustive and reproducible scenario coverage. Table 2.4 compares the main platforms across the criteria most relevant to this thesis.

Table 2.4 AD simulators comparison: physics fidelity, sensor suite, openness, highway map availability, DRL interface, and key limitation.

Simulator	**3D**	**Sensors**	**Open Source**	**Hwy**	**DRL Ready**	**Key Limitation**
highway-env [47]	×	×	✓	✓	✓	No physics; local scenarios only
SUMO RL [106]	×	×	✓	✓	✓	Network-level focus; no EV decision making
CommonRoad [107]	×	×	✓	✓	○	No native DRL; high setup overhead
Flow [108]	×	×	✓	✓	✓	Requires SUMO/CARLA backend
CARLA [36]	✓	✓	✓	○	○	Built-in highway maps have artifacts
AirSim [109]	✓	✓	✓	×	○	Limited traffic agents; no HD maps
NVIDIA DRIVE [110]	✓	✓	×	✓	×	Proprietary; no public access
Baidu Apollo [111]	✓	✓	○	×	×	Complex setup; limited community

✓= Full support, ×= No support, ○ = Partial support/effort required

Among open platforms, CARLA [36] is the leading simulator for developing, training, and validating AD systems [112, 113]. It provides high-fidelity urban environments, configurable traffic, and a broad sensor suite (RGB, depth, LIDAR, radar, GNSS, IMU) built on open assets and extensible APIs. Comparative studies consistently rank CARLA as state-of-the-art for end-to-end evaluation. However, its built-in highway maps (Town4, Town12) suffer from physics instabilities, invisible walls, and limited road geometry variety,

which contaminate logs and interrupt sessions [37, 38]. This motivates the custom 60 km highway map introduced in Chapter 3.

Surveys of AD simulators converge on the need for physics-aware 3D environments when credible evaluation is the goal [30, 32]. Realistic geometry and physics materially change outcomes by coupling mobility, communication, occlusion, and environment [33]. Manipulating simulator fidelity yields measurable differences in driving behavior, reinforcing the move to richer simulators when research questions involve comfort, risk, or fine-grained control [34].

Among synthetic datasets, Virtual KITTI 2 [114] targets object tracking and depth estimation. The DeepScenario platform [115] contains over 30,000 executable driving scenarios including 1,050 collision scenarios. For traffic flow simulation, SUMO [106] is widely used for microscopic modeling, while PTV Vissim integrates active traffic management [116].

Despite this breadth, no publicly available labeled dataset exists for in-vehicle highway scenario detection in either the real or synthetic domain.

2.3.5 Scenario Detection Approaches

Several approaches address detection and classification of driving scenarios from data.

Wang and Lin [117] propose a rule-based scenario classifier for three driving states. Montanari et al. [118] extract scenarios via unsupervised clustering of vehicular bus data.

Ding et al. [119] propose MIT-AVT, clustering common cases and detecting edge cases via DL over visual characteristics.

Chang et al. [120] introduce MetaScenario for scenario extraction using atomic semantic graphs.

Reichenbacher et al. [121] identify cut-in scenarios through knowledge-based meta-modeling.

Izquierdo et al. [74], Biparva et al. [122], and Liang et al. [123, 124] use PREVENTION to train DL recognizers of cut-in and cut-out maneuvers, with Biparva et al. focusing on early-phase prediction.

Kayatas and Bestle [125] classify cut-in, cut-out, and cut-through from radar-derived lateral distance signals using a Time Series Forest, achieving precision and recall above 94% on 10 h of real highway data. Despite strong performance on real measurements, the approach is limited to three kinematic classes and operates on scalar distance signals rather than video, making it unsuitable for onboard forward-camera deployment. Balasubramanian et al. [126] propose an open-world learning architecture that incrementally discovers new scenario classes from occupancy grids, combining open-set recognition, unsupervised clustering, and class-conditioned generative replay across 14 scenario categories derived from highD and

OpenTraffic. While this is the only approach that explicitly handles unknown scenario classes at deployment, it relies on existing datasets rather than a controllable generation pipeline, and the bird-eye-view occupancy representation is not compatible with onboard forward-camera systems. Sankaranarayanan et al. [127] propose a stacking ensemble for online cut-in prediction from kinematic features (range, range rate, lateral position), achieving 94.3% F1 on NGSIM, but address a single maneuver class without a scenario taxonomy or video modality.

The most comprehensive prior work directly comparable to the contributions of Chapter 4 is the framework by Cossu [45], which introduces an end-to-end pipeline for generating and classifying 17 functional highway scenarios from the MOOVE taxonomy, implemented in compliance with ASAM OpenSCENARIO 2.0. Scenarios are parameterized by EV speed (5–182 km/h), inter-vehicle distance (3–250 m), and duration (0.1–98 s). The classification stage uses a Res-3D CNN enhanced with SE and CBAM mechanisms, achieving up to 88.9% accuracy and 88.7% F1 at 3–6 ms per frame (37 GFLOPs). However, the generation pipeline discards approximately 30% of generated scenarios due to collisions and inconsistent trajectories, and still requires manual inspection. Furthermore, EV reactions are not modeled, causing systematic misclassification of behaviorally similar scenarios such as Brake vs. Following LV.

Table 2.5 consolidates these approaches across input modality, classifier architecture, number of scenario classes, presence of a generation pipeline, and real-world validation availability. Thorough analysis of Cossu's work are presented in Chapter 4.

Table 2.5 Highway scenario detection approaches: input modality, classifier, number of classes, generation pipeline, dataset, and key limitation.

Method	Input	Classifier	Cls.	Generator	Open	Real val.	Key limitation
Wang & Lin [117]	CAN logs	Rule-based	3	–	No	Yes	No DL; limited classes
Montanari et al. [118]	CAN logs	Unsup. clustering	N/A	–	No	Yes	No video; coarse labels
Kayatas & Bestle [125]	Lateral dist. $d(t)$	Time Series Forest	3	Hermite splines	No	Yes (10 h)	3 classes; scalar signal only
Balasubramanian et al. [126]	Occupancy grids	CNN + RF (open-world)	14	–	Yes	Yes (highD, OpenTraffic)	No generation; bird-eye only
Biparva et al. [122]	RGB video	Two-stream / SlowFast	2	–	No	Yes (PREVENTION)	Binary; surrounding vehicle only
Sankaranarayanan et al. [127]	Kinematics	Stacking ensemble	2	–	No	Yes (NGSIM)	Cut-in only; no video
Cossu [45]	RGB video	R3D-SE/CBAM	17	CARLA + ScenarioRunner	No	No	30% discard; no EV reaction
Ours [5]	RGB video	R3D-SE	9	CARLA + PyTree	Yes	Yes (PREVENTION)	Simulated; monocular only

Three observations consolidate the gaps motivating Chapter 4. First, all approaches except Cossu and ours address at most binary classification without a generation pipeline. Second, the only prior work combining generation and multi-class detection at scale is Cossu, but its pipeline suffers from a $\sim$30% discard rate, requires manual inspection, and does not model EV reactions, causing systematic misclassification of behaviorally similar scenarios. Third, no existing approach provides both an open labeled synthetic dataset and zero-shot validation on real-world footage.

2.3.6 Summary and Open Gaps

Gap G3: the state-of-the-art generation pipeline yields a $\sim$30% discard rate, requires manual inspection, and does not model EV reactions, causing systematic misclassification of behaviorally similar scenarios. This gap is addressed in Chapter 4.

Gap G4: no publicly available labeled dataset exists for in-vehicle highway driving scenario detection, in either real or synthetic form. Real-world datasets are rather designed for motion forecasting, than for scenario recognition. Synthetic platforms do not provide labeled scenario-detection datasets. This gap is addressed in Chapter 4.

2.4 Perception Robustness and Traffic Light Detection

The development of a robust TLD system capable of mitigating camera sensor anomalies requires understanding three areas: YOLO object detectors (Section 2.4.1); sensor vulnerabilities (Section 2.4.2); and image recovery techniques (Section 2.4.3). TLD datasets are surveyed in Section 2.4.4 and gaps are identified in Section 2.4.5.

2.4.1 YOLO Object Detectors

DL is currently the most employed technology for the TLD task [11]. Both two-stage approaches (Fast R-CNN [128], R-CNN [129]) and one-stage approaches (YOLO [130]) have been proposed, with the latter receiving more interest for real-time capability. We focus on the YOLO family for its state-of-the-art performance and popularity in AD research [131]. The YOLO algorithm divides the input image into a grid of cells, predicting bounding boxes at different scales and corresponding class probabilities. Since YOLOv10, NMS post-processing has been replaced by one-to-one mapping [132].

Terven et al. [133] provide a thorough review from YOLOv1 to YOLOv8. In the TLD context, Polley et al. [134] evaluate a YOLOv7/v8-based system on BSTLD and DTLD, achieving mAP50 of 83% and 74% on a three-class problem (red, yellow, green). Sharma et al. [40] compare YOLOv8–v11, showing YOLOv11 offers the fastest inference ($\approx$13.5 ms) with mAP50 $\approx$0.92. Khanam and Hussain [41] analyze the YOLOv11 architecture, highlighting C3k2, SPPF, and C2PSA components that improve feature extraction.

For embedded deployment, model quantization is a standard technique to reduce memory footprint and inference latency. FP16 quantization has been shown to preserve detection accuracy with minimal degradation [135], making it suitable for automotive platforms such as the NVIDIA Jetson family. Li et al. [136] performed the first quantitative analysis of sensor noise impact on YOLO and Faster R-CNN performance, considering YOLOv5. Our work updates this analysis to YOLOv11 and extends it to the TLD task on embedded hardware.

2.4.2 Sensor Vulnerabilities in Automated Driving

Anomalies in sensor signals require systematic analysis. Beyond functional implications, they raise cybersecurity concerns, as sensor channels can be exploited to inject false data or manipulate perception outputs. ISO/SAE 21434 [83] establishes cybersecurity engineering requirements across the road vehicle lifecycle. In parallel, ISO/IEC 62443 [84] extends these principles to interconnected embedded architectures.

Zhao et al. [137] propose a taxonomy of vehicular sensor anomaly sources across four groups: component faults; adaptability failures to external environments; cyberattacks; and sensor design deficiencies. Vargas et al. [138] examine how adverse weather conditions compromise sensor performance, offering attenuation models useful for simulation-based testing. Joshi and Kumar [139] examine sensor integration limitations that caused real accidents. Rastogi and Nygard [140] highlight the susceptibility of camera, LIDAR, and radar to cyberattacks and sensor manipulation.

For camera sensors, Ceccarelli and Secci [141] define failure modes under adverse lighting and weather, and introduce a robust lane-keeping control scheme with input-to-state stability. Zheng et al. [42] propose TPSeNCE, a rain generation method using a Triangular Probability Similarity constraint to create realistic rainy images. Ji et al. [43] identify an acoustic-based attack (Poltergeist) targeting image stabilization hardware, inducing blur that disrupts DL perception tasks, directly relevant to embedded automotive deployments. Li et al. [142] propose ASA, a black-box perturbation method targeting traffic sign recognition models.

2.4.3 Image Recovery Techniques

After anomaly detection, signal recovery is required to restore perceptual performance. Zhao et al. [137] review countermeasures ranging from diagnosis and fault-tolerant control to defense strategy implementation. Fang et al. [143] survey anomaly diagnosis methods across functional safety, SOTIF, and cybersecurity risk dimensions.

For camera sensors, Ye et al. [144] propose a reference-guided Transformer filter for rain removal. Zheng et al. [145] propose SAPNet, a lightweight segmentation-aware progressive network for image deraining that preserves semantic details for downstream perception. Kupyn et al. [146, 147] propose DeblurGAN and DeblurGAN-v2, achieving five-fold improvements in inference speed and deblurring quality; DeblurGAN-v2 uses a MobileNetV2 backbone, making it suitable for embedded platforms. GANs have also been used by Jin et al. [148] as a preprocessing step for perturbation removal.

Autoencoders have been studied for robust representation learning. Vincent et al. [149] propose Denoising Autoencoders for feature extraction. Hwang et al. [150] present PuVAE, which projects adversarial examples onto the class manifold to purify them. Diffusion models represent an emerging family for image restoration: Ozdenizci and Legenstein [151] achieve state-of-the-art multi-weather restoration, but diffusion-based methods suffer from high computational cost [152, 153], making them unsuitable for real-time embedded deployment.

2.4.4 Traffic Light Datasets

A factual survey of the main public TLD datasets is provided here; dataset selection criteria and preprocessing choices are discussed in Chapter 5.

The BSTLD [154] comprises 13,427 RGB images at 1280×720 px with 24,242 annotations from US urban areas, covering sunny to light-rain conditions. The DTLD [153] contains over 230,000 annotated traffic lights at 2048×1024 px across 11 German cities, with diverse illumination and weather conditions, and includes the red-yellow state relevant for cross-regional robustness. The LISA dataset [155] includes 43,007 frames from San Diego under day and night conditions at 1280×960 px with 113,888 annotations. Table 2.6 summarizes these datasets.

Table 2.6 Traffic light dataset characteristics.

Dataset	Images	Day/ Night	Resolution	Annotations	Cities	Red	Green	Yellow	Red Yellow	Off	Year
BSTLD [154]	13,427	D	1280×720	24,242	17	∼9.5K	∼13K	∼0.6K	absent	∼1.1K	2017
DTLD [153]	40,978	D	2048×1024	232,039	11	∼60K	∼80K	∼5K	∼2K	∼40K	2018
LISA [155]	43,007	D/N	1280×960	51,826	1	∼25K	∼24K	∼2K	absent	absent	2015

2.4.5 Summary and Open Gaps

Table 2.7 compares the main camera recovery methods reviewed above across anomaly type, recovery approach, real-time capability, TLD specificity, and TOR integration.

Table 2.7 Camera image recovery methods for AD: anomaly type, approach, real-time capability, TLD specificity, and TOR integration.

Method	Anomaly type	Recovery approach	Real-time	TLD-specific	TOR
Ye et al. [144]	Rain / drops	Transformer filter	Partial	×	×
SAPNet [145]	Rain streaks	Segm.-aware prog. net	✓	×	×
DeblurGAN-v2 [147]	Blur	GAN (MobileNetV2)	✓	×	×
Ozdenizci & Legenstein [151]	Multi-weather	Diffusion (patch)	×	×	×
PuVAE [150]	Adversarial	Autoencoder purif.	Partial	×	×
Li et al. [136]	Sensor noise	Quantitative analysis	–	Partial	×
Ours [7]	Blur + noise	Frame checker + recov.	✓	✓	✓

Gap G5: no existing work integrates camera anomaly detection, image recovery, and TOR into a single deployable AD function validated on embedded hardware. All recovery methods are standalone, none are validated end-to-end for TLD, and none include a TOR mechanism. No prior work evaluates a quantized FP16 YOLOv11 detector under realistic camera perturbations, nor proposes a training strategy for a frame-level anomaly checker that also leverages cleaned frames. This gap is addressed in Chapter 5.

2.5 Chapter Summary and Research Gaps

The four surveys above converge on five open problems, summarized in Table 2.8. Each gap identifies a specific limitation in the state of the art directly addressed by one of the experimental chapters.

Table 2.8 Research gaps identified in the literature and corresponding thesis contributions.

#	Gap	Evidence in literature	Addressed in
G0	DT applied to highway driving with discrete actions and safety constraints is unexplored; 2D simulation cannot characterize the performance-safety trade-off credibly	DT validated only on Atari/MuJoCo [28]; highway-env lacks physics and sensor noise [30]	Chapter 3
G1	No open, DL-ready dataset for EV LC intention with continuous TTLC labels (0–4 s regression) paired with a reproducible generation framework	Brain4Cars: 274 LCs / 10 drivers; NGSIM/highD: no EV-centric features; Waymo: 8 s horizon, no ego-intention labels; no dataset includes a generation engine	Chapters 3 and 4
G2	LC predictors are almost exclusively classifiers; TTLC regression beyond 3 s is unexplored	Zhang et al.: claimed 7 s but reliability drops; Gao et al., Xia et al.: classification only; no regression benchmark exists	Chapter 4
G3	Scenario generation pipelines discard ∼30% of output; not fully automated; EV reactions not modeled	Cossu (2025): 30% discard rate; Brake scenario misclassified as Following LV without EV reaction modeling	Chapter 4
G4	No public labeled dataset for in-vehicle highway scenario detection (real or synthetic)	Confirmed by survey of real-world (§2.3.3) and synthetic (§2.3.4) datasets	Chapter 4
G5	No integrated modular system combines camera anomaly detection, image recovery, and TOR on embedded hardware	All recovery methods are standalone; none validated end-to-end for TLD; no TOR mechanism; no FP16 YOLOv11 evaluation under perturbations	Chapter 5

These gaps directly motivate the methodological choices in the subsequent chapters: the DT experiment and CARLA framework (Chapter 3), the LC regression dataset and scenario generation pipeline (Chapter 4), and the anomaly-aware embedded TLD system (Chapter 5).

Chapter 3

From High- to Low-Level Simulation

3.1 Introduction

This chapter bridges the gap between abstract policy learning and realistic data acquisition, serving a dual purpose that underpins the contributions presented in Chapter 4. First, we validate sequence modeling for tactical DM in a controlled 2D setting using highway-env [35, 156], where we build an offline dataset, train a DT policy, and benchmark it against an online DQN. This lightweight environment enables rapid iteration and establishes proof-of-concept for offline decision-making.

However, the inherent limitations of 2D abstractions (e.g., safety-critical nuances, comfort metrics, sim-to-real transferability) motivate our transition to a sensor-realistic 3D pipeline. We therefore introduce our CARLA simulation framework, featuring a custom highway map and lightweight data-collection infrastructure designed for long, stable runs and human-in-the-loop data acquisition. This 3D environment provides the necessary fidelity for large-scale data collection and subsequent model evaluation in later chapters.

3.2 Methodology

In this section, we describe the methodologies used in our research to explore the application of DT within the AD domain. We begin by outlining the creation of an offline RL dataset, followed by the description of the training process of the DT model. Lastly, we present the evaluation metrics and results obtained from comparing our DT model with a traditional DQN) [157] agent.

3.2.1 Environment

Experiments are conducted in highway-env (Fig. 3.1), a lightweight 2D open-source simulator tailored to tactical decision making (DM) in multi-lane highway traffic. The platform provides a controlled yet realistic setting where autonomous driving (AD) agents, particularly DRL-based approaches [8, 158–160, 49], can be trained, evaluated, and analyzed.

The environment reproduces typical highway maneuvers, including lane keeping, overtaking, and LCs, within dynamic traffic scenarios. Its Gym-compatible interface enables seamless integration with standard DRL libraries and rapid prototyping, thanks to the python framework which easily allows to gather informations (i.e. speed, lane position, agent's observation, reward, etc.) . Compared to SUMO RL [106], which targets network-level traffic flow rather than single-agent DM, and CommonRoad [107], which requires substantial setup overhead and lacks native DRL support, highway-env offers a minimal yet sufficient abstraction for offline sequence modeling experiments at the scale required here. In addition, the lightweight rendering allows large-scale training and offline dataset generation without GPU bottlenecks, while the transparent and well-documented formulation supports controlled ablations and reproducibility [49, 50].

Formally, highway-env models the EV and surrounding traffic as agents in a MDP [48]. The state encodes positions, velocities, and lane configurations; the EV selects actions such as acceleration, braking, or lane changes to maximize a reward function balancing safety, efficiency, and comfort. This structure makes the simulator suitable both for benchmarking new decision-making strategies and for analyzing the behavior of trained policies.

Because of its open-source nature and extensibility, highway-env has become a standard testbed in the AD research community for developing interpretable, robust, and safe driving policies in complex highway scenarios.

Fig. 3.1 Snapshot from highway-env. The EV, namely the learning agent, is colored in green; NPVs are light-blue.

Observation

Observations are kinematic and structured as a matrix of size $V \times F$, where V is the number of observed vehicles and F the number of features per vehicle. Each observation comprises $F = 7$ features per vehicle, in the following order: presence indicator, horizontal coordinate, vertical coordinate, longitudinal speed, lateral speed, and two trigonometric headings (cosine and sine). We set $V = 10$ observed vehicles out of the 15 total present in each episode. Hence, each observation is a 10×7 matrix.

The empirical observation window of V=10 vehicles was chosen to balance contextual richness and input dimensionality. Including too few vehicles risks missing relevant interactions during dense traffic; including all 15 increases input size without proportional benefit, as distant vehicles rarely influence immediate decisions [49]. The 7 features per vehicle, adopted also in other works (i.e., [49]) extend the standard 5-feature kinematic vector to capture vehicle orientation, which is relevant for anticipating lateral maneuvers.

Rewards

Reward design follows three components: collision avoidance, lane discipline, and speed. A sparse penalty of -1 is applied upon collision, immediately ending the episode. Dense rewards encourage the EV to remain in the right-most lane whenever feasible and to maintain speeds within a target range $[30, 36]\,\mathrm{m/s}$. Table 3.1 summarizes the employed signals. Reward weights are rebalanced to reduce the bias toward aggressive acceleration and to align better with safety considerations.

Table 3.1 Employed rewards in highway-env.

Reward	Description	Type	Weight
Collision	Assigned when the EV collides with another vehicle	Sparse	-1
Right-most lane	Encourages the EV to remain in the right-most lane when possible	Dense	0.3
High-speed	Active only in $[30, 36]\,\mathrm{m/s}$ to promote adequate speed	Dense	0.3

More specifically, the reward function can be defined by Eq. 3.1

$$R = c_{\text{coll}} \cdot R_{\text{coll}} + c_{rml} \cdot R_{rml} + c_{hs} \cdot R_{hs}, \tag{3.1}$$

where R_{coll} , R_{rml}, and R_{hs} represent the three primary reward functions already described in Table 3.1. Such rewards help shape the EV decision-making during training. The equal weighting of lane discipline and speed rewards reflects the dual objective of maintaining traffic-compliant positioning while sustaining highway-appropriate velocities. The collision penalty dominates by design, as a single crash terminates the episode. These values are consistent with reward structures reported in related highway-env studies [49, 50], where balanced dense rewards with a heavy sparse collision penalty have been shown to produce stable training.

Collision The collision reward R_{coll} (Eq. 3.2) penalizes the agent when a collision occurs. Specifically, it assigns a reward of -1 if the EV crashes into another vehicle, triggering the end of the episode to prevent the accumulation of positive rewards. Conversely, if the EV completes the simulation without crashing, the reward remains 0.

$$R_{\text{coll}} = \begin{cases} -1, & \text{if the vehicle crashes} \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

Lane Discipline The right-most lane reward R_{rml} (Eq. 3.3) incentivizes the EV to maintain its position in the right-most lane whenever possible. The highest reward is achieved when the EV remains in the right-most lane, and the reward linearly decreases as the EV moves towards the left-most lane, where it reaches zero. The equation is defined as:

$$R_{\text{rml}} = \frac{\text{Current EV lane index}}{N} \tag{3.3}$$

where N represents the total number of lanes on the road. Unlike the collision reward, this reward is dense, meaning it is continuously provided to the agent at each simulation step.

Speed The high-speed reward R_{hs} (Eq. 3.4) encourages the EV to maintain a high speed. It is also a dense reward, meaning it is frequently given during training. The highest reward is achieved when the EV cruising speed remains within a specific range, which in our case is 30 to 36 m/s (108 to 130 km/h). The reward function is defined as:

$$R_{\text{hs}} = \text{clip}(\text{scaled speed}, 0, 1) \tag{3.4}$$

where scaled speed is a normalized value obtained through linear interpolation:

$$\text{scaled speed} = \left(\frac{v - l_{\min}}{l_{\max} - l_{\min}} \right) \tag{3.5}$$

where:

- v is the EV current speed,
- $l_{\min}$ and $l_{\max}$ are the minimum and maximum thresholds of the speed range where the agent receives a positive reward.

To ensure stability in the DRL process, the values of the rewards are mapped between $[0, 0.1]$ before being fed into the algorithm. This normalization step is done to prevent an excessive increase in the total reward magnitude, which could destabilize the learning process.

Action Space

We adopt the following discrete action space:

$$\mathscr{A}_{\text{LL}} = \{\texttt{faster}, \texttt{slower}, \texttt{left}, \texttt{right}, \texttt{idle}\},$$

with cardinality $|\mathscr{A}_{\text{LL}}| = 5$. Each action corresponds to a one-step primitive that directly affects the kinematic state of the EV within the simulator. This compact set captures acceleration and deceleration in the longitudinal axis, lateral lane changes, and the option to maintain the current trajectory. Table 3.2 reports the semantics of each primitive.

Table 3.2 Low-Level discrete action space in highway-env.

Action	Description
`faster`	Increases the EV longitudinal velocity by applying a discrete positive acceleration increment.
`slower`	Decreases the EV longitudinal velocity by applying a discrete negative acceleration increment.
`left`	Initiates a lane change maneuver to the immediate left lane, if available.
`right`	Initiates a lane change maneuver to the immediate right lane, if available.
`idle`	Maintains the current velocity and lane, with no change in control.

3.2.2 Data Collection

Offline RL dataset must be composed of MDP trajectories, (i.e. states, actions, and rewards) to reconstruct RL transitions. To build the offline dataset, an online DQN agent was trained

in the highway environment for 250,000 steps. Each episode lasted at most 30 seconds and included 15 vehicles. After training, the policy was executed to collect 150,000 episodes, totaling approximately 3.86 million trajectories. Table 3.3 summarizes the dataset statistics. Episode durations are in seconds (s). Min and max returns refer to the lowest and highest episode-level cumulative rewards (dimensionless) observed across the full dataset.

Table 3.3 Dataset statistics.

Min return	Max return	Avg return	Trajectories	Episodes	Min duration (s)	Avg duration (s)
7.6	23.2	11.32	3.86M	150K	10	23.06

We acknowledge that collecting data from a single DQN policy introduces coverage limitations inherent to offline RL [46]. Despite using ε-greedy exploration during data collection, introducing stochastic action selection that diversifies the visited state-action space, the dataset may not cover all relevant state-action regions, particularly rare or suboptimal transitions. However, the scale of the dataset (3.86M trajectories across 150K episodes) and the stochastic nature of the traffic provide meaningful behavioral diversity. This setup is consistent with standard offline DRL benchmarks [161], where medium-quality single-policy datasets remain a common and accepted baseline for evaluating sequence models.

3.2.3 Decision Transformer Architecture

In accordance with prior research [28, 162], we adopt return-conditioned upside-down RL via a cross-entropy loss to predict subsequent actions in an autoregressive manner. While these works demonstrate the effectiveness of DT in Atari and MuJoCo benchmarks, their application to multi-agent highway scenarios with discrete action spaces and safety constraints remains underexplored.

Given an offline trajectory $\tau = \{(s_1, a_1, r_1), \ldots, (s_T, a_T, r_T)\}$, we compute the return-to-go as

$$\hat{G}_t = \sum_{t'=t}^{T} r_{t'} \tag{3.6}$$

and train an autoregressive policy that predicts a_t conditioned on a fixed-length history of returns, states, and past actions. Concretely, at each training step we feed the model the last K timesteps (context length, CL) of the triplets $(\hat{G}_t, s_t, a_{t-1})$ and minimize the next-action cross-entropy:

$$\mathscr{L} = -\sum_{t=t_0}^{t_0+K-1} \log \pi_\theta \left(a_t \mid \hat{G}_{1:t}, s_{1:t}, a_{1:t-1} \right). \tag{3.7}$$

CL specifies how many past observations and actions the model can attend to when producing a_t.

Tokenization follows the DT design [28]: each timestep contributes three tokens (i.e., return-to-go, state, and action) so that one episodic timestep corresponds to a length-3 token block. We use an episodic timestep embedding shared by the three tokens of the same timestep, rather than a single position per token as in the standard Transformer setup [26]. This preserves the semantic separation between the conditioning signal (return-to-go), the observation (state), and the decision (action), allowing the model to attend selectively to each component across timesteps. This design choice is supported by ablation studies in [28] showing that fused embeddings degrade performance. The resulting token sequence is processed by a causal, GPT-style decoder with masked self-attention, and the model outputs action logits autoregressively. Training uses a standard cross-entropy objective over the discrete action vocabulary. Our backbone is a GPT-2–like decoder-only architecture [163], adapted to the $(\hat{G}, s, a)$ tokenization described above. The choice of a GPT-2–style decoder, despite the low dimensionality of the input tokens, follows the original DT formulation [28], which demonstrated that autoregressive sequence modeling over return-state-action triplets is effective even in low-dimensional control tasks.

Figure 3.2 illustrates the data flow: tokens derived from the dataset are embedded, combined with timestep encoding, and passed through the transformer to produce action logits. The model is trained with cross-entropy loss over the discrete action space.

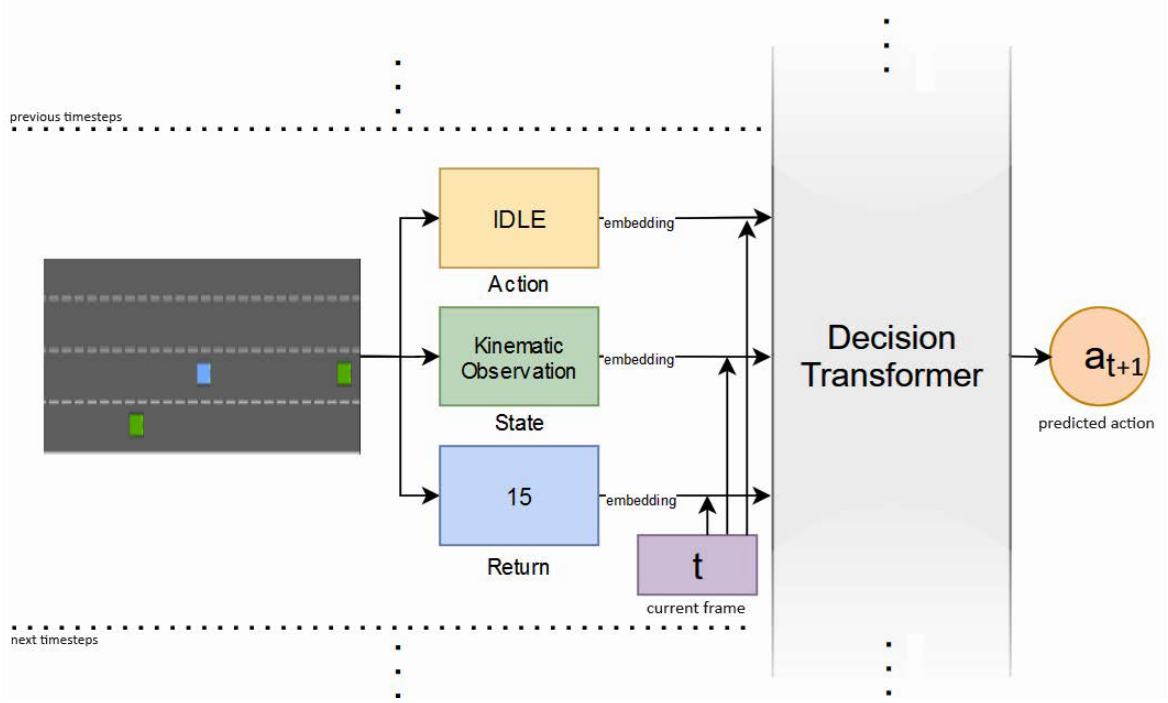


Fig. 3.2 Diagram of the DT architecture used in highway-env. Each timestep contributes return, state, and action tokens, which are embedded and processed with causal attention.

3.2.4 Training setup

The DT was trained for 500 epochs. We chose the DT hyperparameters to use according to Bhargava et al. [162]. The values that differ from this set are the used attention heads (i.e. 4), steps per iteration (i.e. 10,000), and batch size (64). The training was performed on a computer equipped with an Intel i9-14900K CPU, 32 GB of RAM, and an NVIDIA GeForce RTX 4080.

3.3 Results

To evaluate the trained DT, we compared it against the DQN agent used to generate the dataset and against a random agent that uniformly sampled actions from the discrete set. Evaluation consisted of 300 closed-loop episodes in the same highway environment. The target return at test time was set to the maximum value observed in the dataset, to encourage the DT toward high-return behaviors.

Table 3.4 summarizes the results. DT achieves the highest average return (14.01), outperforming the DQN (11.35) and the random baseline (6.41). It also sustains slightly longer average episodes (22.66 seconds) compared to DQN (22.25). The DT maintains an

average speed of 22.56 m/s, faster than DQN (21.95) but slightly slower than the random agent (24.02). However, collision rate reveals a safety trade-off: DT suffers a 51% collision rate, higher than the 40% observed for DQN, while the random baseline, unsurprisingly, reaches 96%.

Table 3.4 Obtained results after 300 episodes for each agent.

DM Agent	Type	Avg return	Avg episode length (s)	Avg speed (m/s)	Collision rate
DQN	Online	11.35	22.25	21.95	**40%**
DT	Offline	**14.01**	**22.66**	22.56	51%
Random	Heuristic	6.41	13.03	**24.02**	96%

The obtained results indicate that DT, when trained on a large offline dataset, is capable of surpassing the behavior policy in terms of return and maintaining longer episodes, which reflects improved DM performance. However, this comes with a clear cost: the collision rate is significantly higher than that of DQN. This performance–safety trade-off highlights the limitations of optimizing only for return without explicit safety constraints.

The DT attains higher returns than the behavior policy but at a higher collision rate, a known offline-RL trade-off. Because 2D abstractions limit comfort and risk assessment, we now transition to a 3D, physics-aware pipeline. The next part replaces highway-env with CARLA and introduces a custom highway map and a lightweight data-collection stack to test DM under realistic dynamics and sensing.

3.4 The Importance of Increasing Realism

In this work, we tackled the application of DRL via sequence modeling within the AD domain. We selected highway-env as our AD environment, which, despite its high level of abstraction, serves as an effective starting point for evaluating ML models, assessing AD scenarios [164], and investigating impact of various driving factors [160]. The results obtained are promising, demonstrating how sequence modeling can be effectively applied for DM in AD. We recognize that the DT collision rate is not yet optimal. However, the performance of the DQN and DT models is inherently interrelated. It is believed that with a more comprehensive and higher-quality dataset, along with a pre-trained model exhibiting a higher success rate, the DT performances can be significantly enhanced. Improved data collection efforts will provide the necessary foundation for more effective training and results. While the DQN, a well-established DRL algorithm in highway-env, resulted in fewer

collisions than our model, it yielded a lower average return, leading to more conservative driving behavior in terms of speed and collisions. We acknowledge that replacing low-level actions with more realistic, structured, higher-level actions could enhance the realism and safety of the model [18, 160, 159].

Furthermore, we stress the importance of the data quality, as it significantly impacts the performance and reliability of ML and DRL models. The quality and variety of the data ensure optimal learning and generalization capabilities for AD scenarios. In light of the fact that fine-tuning to learn new tasks is applicable in offline DRL, opposite to online RL, it is imperative to consider the prevention of catastrophic forgetting when dealing with previously unencountered tasks. Recent advancements in the literature suggest that Multi-Domain DT may be employed to learn more effective representations of trajectories, while avoiding catastrophic forgetting [165].

To address more realistic situations, it is crucial to employ more sophisticated driving simulators, such as CARLA [36], which takes into account vehicle physics, 3D models, and many other factors that highway-env and other driving simulators do not focus on. Crucially, the trade-off between safety and performance observed in our DT experiments cannot be properly characterized or mitigated in a 2D abstraction: comfort margins, sensor noise, and spatial continuity require a physics-aware setting, as well as a wide coverage of socio-behavioral factors, such as driver prudence and driving style variability, which are essential to accurately reflect human-robot interaction in mixed-traffic scenarios. The CARLA framework introduced in the following sections addresses this gap, while also serving as the data collection infrastructure that enables the LC intention recognition and scenario generation tasks presented in the subsequent chapters of this thesis.

3.5 From 2D Prototyping to 3D High-Fidelity Simulation

Results from DT on highway-env exposed performance–safety trade-offs under simplified dynamics and sensing. Rather than porting those experiments as-is, we focus here on a realistic pipeline tailored to long, stable highway runs.

We developed a lightweight yet stable highway environment in CARLA [1]. The contribution is twofold: a custom highway map engineered for stability and controlled variety (long straights, broad direction-balanced curves) and a data–collection stack that turns the simulator into a reliable pipeline. The framework [1, 2] offers a first–person driving interface (mirrors, indicators, gauges, RPM–synchronized audio), procedural traffic that maintains interaction density without excessive GPU load, and a standardized telemetry pipeline that exports sessions as analysis–ready time series for DL models.

3.6 Framework Design

This section illustrates the framework structure. It starts with the map design, featuring description and statistics. It then follows with the traffic modeling that had been chosen to limit GPU overload and to make the framework lightweight.

The scientific contribution of this framework is threefold. First, it provides a reusable and open-source[1] [2], simulation stack that can be adapted to different viewpoints, modalities, and experimental designs without rebuilding from scratch. Second, it addresses a concrete methodological gap: the absence of stable, long-duration highway environments in CARLA suitable for human-in-the-loop data collection at scale, as existing maps (Town4, Town12) suffer from physics instabilities and limited highway geometry. Third, it produces structured, annotation-ready time series of LC maneuvers, directly enabling the supervised learning benchmarks presented in Chapter 4. In this sense, the framework is not an end in itself but a methodological enabler for the data-driven contributions that follow.

3.6.1 Map

At the current version, CARLA offers 15 towns, with Town12 and Town13 featuring large highway systems, but these maps have issues like invisible walls and uneven pavement. To address this, we created a custom map using RoadRunner tool by MATLAB [166], shown in Fig. 3.3. The new map represents a 60 km one–way highway track with two 3.75 m–wide lanes, featuring both straight and carefully balanced left/right curves, with different radii, to ensure behavioral variety and prevent road geometry biases in the collected data. Map parameters were designed to reflect real European highway standards. Specifically, the 3.75 m lane width aligns with Italian highway norms (3.50–3.75 m per lane [167]). The 60 km circuit length ensures that data collection sessions are not artificially constrained by track boundaries, avoiding behavioral artifacts from route familiarity. We prioritized simulation efficiency by minimizing environmental assets and avoiding computationally expensive elements (e.g., vegetation, water effects, or particle systems). This design choice reduces GPU load, enabling longer uninterrupted simulation runs and higher frame rates, which are essential for stable and consistent data logging. Moreover, it isolates the driving task from visually irrelevant elements, ensuring that the agent's behavior is shaped primarily by traffic dynamics rather than by perceptual distractions unrelated to LC decisions. This is consistent with findings that task-relevant cues such as lane geometry and nearby vehicles are more important for driving behavior than background scenery, and that simplified visuals can still support valid driving performance measures when core cues are preserved [168, 34, 169].

[1] https://elios-lab.github.io/LC-Intention-Framework/

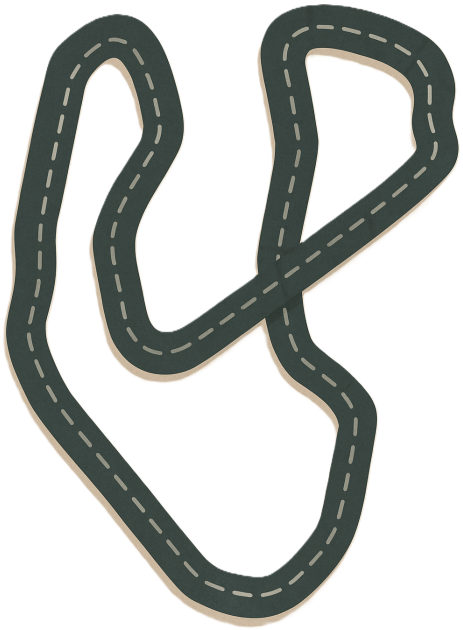


Fig. 3.3 Top view of the map.

For completeness, Table 3.5 summarizes the main specifications of our highway track against commonly used alternatives in CARLA-based studies; detailed map and traffic design choices are documented in the following sections.

Table 3.5 Comparison of available highway maps.

Map	**Lane width (m)**	**Total length (km)**	**Map area (km^2)**	**# Curves**	**Min/Max radius (m)**	**Min/Max straight length (km)**
NGSIM rep. [70]	3.70	1	0.1	0	N/A	1/1
CARLA Town4 [36]	3.50	4	0.8	6	54/500	0.1/0.9
Ours	3.75	60	147	30	232/697	1/4.4

As shown in the table, while the minimum radius of 232 m is below the 339 m recommended for Italian extra-urban highways (designed for 90 km/h), it remains above the minimum for lower-speed or urban highway sections (45 m) [167]. This range was deliberately selected to introduce geometrically challenging segments that elicit a broader range of driver behaviors, including speed adaptation and increased lateral attention during LC maneu-

vers. While this implies lateral accelerations that exceed typical highway comfort thresholds at maximum speed, drivers naturally self-regulate their speed on tighter curves [170], as confirmed by the observed speed variability across participants [2]. The primary design objective was behavioral diversity in the collected dataset rather than strict adherence to infrastructure standards, ensuring physically plausible and safe simulation dynamics.

3.6.2 Driver View

Vehicle dynamics rely on CARLA's PhysX–based model (nonlinear tire lateral forces with slip), with a maximum EV speed set to 140 km/h to span typical highway limits while retaining a buffer for safe scenario elicitation. To support human–in–the–loop data collection and to standardize the first–person viewpoint, we augmented the cockpit with back and left rear–view mirrors, indicators status, tachometer, and odometer overlays, as shown in Fig. 3.4; engine sound was tied to RPM to stabilize speed control by the driver.

Fig. 3.4 EV view.

Cockpit augmentations were included to provide an ecologically plausible driving context, following established human-machine interaction and simulator validity literature. Speed feedback improves speed maintenance accuracy and reduces attentional cost; mirror configuration systematically influences visual scanning and LC decision times; and RPM-synchronized audio, as part of multimodal feedback, has been linked to improved reaction times and more realistic speed perception in simulator environments. Indicator overlays are standard practice in medium-fidelity HMI evaluation. A formal controlled comparison with and without these elements was not conducted and is left as future work [171–176].

3.6.3 Traffic Modeling

Following our two previous studies [160, 159], adding NPVs to the scene was needed to increase realism, but on the other hand a strategy to keep the simulation smooth and lightweight to avoid GPU overload was needed.

We implemented two types of NPVs: SNPVs and HNPVs. SNPVs populate the right lane at a 100 km/h constant speed, to naturally induce LC scenarios. To optimize computational efficiency in our vast environment, we developed a procedural traffic generation system that dynamically spawns SNPVs ahead of the EV and recycles them once overtaken and no longer visible in the rearview mirrors. This approach maintains realistic traffic density while preventing GPU overload and frame rate degradation. HNPV serves as a stochastic yet realistic risk element. With a maximum speed of 180 km/h, HNPVs are initialized in the left lane to create challenging scenarios. Following each encounter, the HNPV is systematically repositioned behind the EV through a teleportation mechanism. The traffic management system is formalized in Algorithm 1, while examples of HNPV and SNPVs are illustrated on Fig 3.5.

Algorithm 1 NPVs Traffic Management System

```
1: min_dist ← 300; max_dist ← 400; offsetS ← 300; offsetH ← 800
2: while true do
3:     for i = 0 to N − 1 do
4:         if min_dist < EV.pos − SNPVs[i].pos < max_dist then
5:             while true do
6:                 if EV.pos + offsetS is a free location then
7:                     teleport SNPVs[i] at EV.pos + offsetS
8:                     offsetS ← 300
9:                     break
10:                else
11:                    offsetS ← offsetS + 300
12:                end if
13:            end while
14:        end if
15:    end for
16:    if min_dist < HNPV.pos − EV.pos < max_dist then
17:        teleport HNPV at EV.pos − offsetH
18:    end if
19: end while
```

(a) HNPV (visible in the rear-view mirrors) approaching from behind.

(b) The HNPV has overtaken the EV, and it is visible in the left lane.

(c) EV approaching a SNPV, starting the LC maneuver.

(d) EV after overtaking the SNPV (rear-view mirror).

Fig. 3.5 Scenario with HNPV and SNPV: (a) HNPV pre-overtake; (b) HNPV post overtake; (c) A SNPV is overtaken by the EV; (d) the EV has overtaken the SNPV.

3.7 Results

3.7.1 Data Collection Involving Human Participants

To test the simulation framework, we gathered vehicular data related to LC maneuvers performed by real human drivers. To do so, we chose the Logitech G920 driving system, widely referenced in driving simulator development (e.g., [177]). The setup included a firmly fixed steering wheel and pedals, a 29” ultra-wide curved screen for a broad field of view, and a dedicated driving seat. An external camera recorded the test sessions and user behavior. A supervisor was present during the whole experiment and annotated any unforeseen issues, such as collisions or technical malfunctions.

To conduct the experiments, a computer equipped with an Intel i9-14900K CPU, 32 GB of RAM, and an NVIDIA GeForce RTX 4080 was used. Four participants of different ages, genders, and driving abilities took part in the test. In accordance with the General Data Protection Regulation (GDPR), this research adhered to the requisite standards to ensure the confidentiality of participants’ data and information. Before testing, participants are requested to sign the GDPR agreement and provide personal information (e.g. age, sex, nationality, driving license, driving proficiency. etc.) by filling in a survey. The classification

of participants includes different perspectives and abilities. To ensure some driving skill, participants are required to possess a driving license for a minimum of two years and a minimum of 1,500 km of driving experience. The supervisor is responsible for providing a comprehensive briefing on key aspects, including the rationale for the test, driving rules and data collection procedures. Upon the conclusion of the analysis, any personal data have been deleted.

Each participant drove for 15 minutes, and the whole driving session was recorded. During the experiment 10 vehicles (excluding the EV) were on the road, and the drivers had to perform LC maneuvers whenever they felt it safe to do. Simulation stability was monitored throughout all data collection sessions. The framework sustained a consistent frame rate above 30 FPS on the test hardware (Intel i9-14900K, RTX 4080, 32 GB RAM) across sessions of up to 15 minutes with 10 active NPVs. No physics anomalies were observed (e.g., vehicle tunneling, abnormal collision geometry, kinematic discontinuities occurred during logged runs, etc.). Episode resets were triggered exclusively by intentional collision events or manual interruption, with no simulator crashes recorded across the full data collection campaign. Table 3.6 resumes the results obtained after the data collection phase.

Table 3.6 Preliminary results.

Users	**Speed (km/h)**	**LCs**	**Traveled distance (m)**
User1	140	150	2100
User2	120	100	1800
User3	120	80	1800
User4	110	82	1650

As illustrated in the table, the number of LCs ranges from 80 to 150, depending on the driver's average speed and confidence in overtaking SNPVs. Each participant completed a 15-minute driving session, as the primary objective was to generate a dataset for LC intention recognition rather than to estimate realistic traffic statistics. These preliminary results confirm that the simulator produces LC events for data collection purposes. However, the limited sample size and short exposure duration prevent any generalizable conclusions regarding typical LC rates or overall simulator realism. The observed average of approximately six LCs per minute should therefore be interpreted as scenario-specific and exploratory.

This preliminary session was designed as a pilot study to validate the simulator setup, identify usability issues, and estimate LC elicitation rates before the full-scale data collection

campaign described in Chapter 4, which involved 50 participants. Results are therefore not intended to be statistically generalizable, but to inform simulator refinements and experimental design.

3.7.2 Participants Feedback

User feedback received upon test conclusion included comments pertaining to various aspects of the simulation. These aspects included the perceived width of the road, the traffic simulation, and the GUI displayed on the dashboard. In particular, from two users it was observed that the road was perceived as narrow, although the map width adheres to the standards. This discrepancy may be imputed to the relatively narrow field of view of the main camera, set at 105°, which, on the one hand prevents the driver from experiencing the sensation of driving too fast compared to reality, but on the other hand, results in a road view that is somewhat narrow in comparison to reality. One potential solution to the problem might be employing of two additional cameras, which would provide enhanced side visibility. However, this would result in a higher computational load.

All the received feedbacks had been taken into consideration to improve the simulator flow to make it suitable to better driving sessions.

3.8 Conclusion

This chapter combined low-level prototyping in highway-env with a high-fidelty, realistic 3D pipeline in CARLA. In highway-env, offline DT training on a large dataset yielded higher returns than the behavior policy but at a higher collision rate, exposing the safety–performance trade-off under simplified dynamics and sensing. These findings justified the move to a physics-aware setting where comfort, risk, and failure modes can be measured credibly.

The CARLA framework addressed the main blockers observed in 2D: it enabled long, uninterrupted runs, providing stable logs and supporting controlled variability through a custom highway map and procedural traffic. The first-person cockpit and standardized telemetry turned driving sessions into analysis-ready time series suitable for downstream learning tasks.

Initial human-in-the-loop experiments, though limited in sample size, consistently elicited LC maneuvers at a high rate and produced structured annotations with low overhead. Together with the DT results in highway-env, this establishes a path from rapid policy iteration to credible data collection: fast hypothesis testing in 2D to shape models and rewards, then

3D acquisition to quantify safety, comfort, and transferability under realistic sensing and dynamics.

Practically, the resulting asset is not a one-off prototype but a reusable product. Its modular map, traffic, cockpit, and logging stack generalize across projects and modalities. It scales to larger campaigns without excessive computational cost and supports both learning from demonstrations and closed-loop policy evaluation.

This work underpins two projects later in the thesis: one on LC intention recognition and one on driving scenario generation and detection, both leveraging the same map and logging stack while addressing distinct learning problems (see Chapter 4).

Chapter 4

Lane Change Intention Recognition and Scenario Detection on Highways

4.1 Introduction

This chapter merges together two problems on the same simulation setup: early recognition of LC intention and detection of highway driving scenarios. Building on the CARLA-based framework and custom highway map introduced earlier, we use a coherent stack to study both a fine-grained maneuver task (TTLC-oriented LC intention regression, trained on telemetry/time-series) and a higher-level context task (video-based scenario classification, rendered as MP4 clips without telemetry). For the scenario part we reuse the same highway map to ensure geometric consistency across tasks.

The LC part leverages human-in-the-loop sessions collected on the custom highway track to model driver intent as a TTLC regression problem and to benchmark alternative architectures under realistic interaction density. Our LC intention recognition framework [2][1] provides a ready-made dataset of annotated maneuvers and the tools to reproduce and extend it. Being built on CARLA, researchers can modify scenarios (road layout, traffic density, driver behaviors) and regenerate data to suit new experiments. The open release of both data and code enables reproducible research and collaboration. A graphical overview is shown in Fig. 4.1.

We then address driving scenario generation and detection: using the same highway map, we synthesize labeled video clips for multiple highway scenarios and export them as MP4s (no telemetry/time-series). Generation follows configurable templates (traffic density, relative speeds, gaps) and renders controllable variability while keeping geometry and markings

[1] https://elios-lab.github.io/LC-Intention-Framework/

consistent. A reference classifier trained on these clips provides baseline performance and error analyses for scenario recognition under the same road layout used for LC.

In sum, this chapter uses one map and two data modalities to study LC and scenario understanding under consistent conditions. We detail the data generation pipelines (vehicular time-series for LC, clips for scenarios), the annotation protocols, and the model setups for TTLC regression and video classification. We then report a unified evaluation with common splits and metrics, analyze failure modes, and discuss how insights at maneuver level inform scenario recognition and vice versa.

4.2 Lane Change Data Collection

This section details the steps to introduce the open-source framework designed for LC intention recognition in highway driving settings. The framework combines a large-scale highway map, an experimental protocol involving 50 drivers, and a dataset comprising more than 3,400 annotated LC maneuvers. The 60 km map integrates variable curvature radii and straight segment lengths, ensuring diversity in driving conditions while preserving consistency and reproducibility.

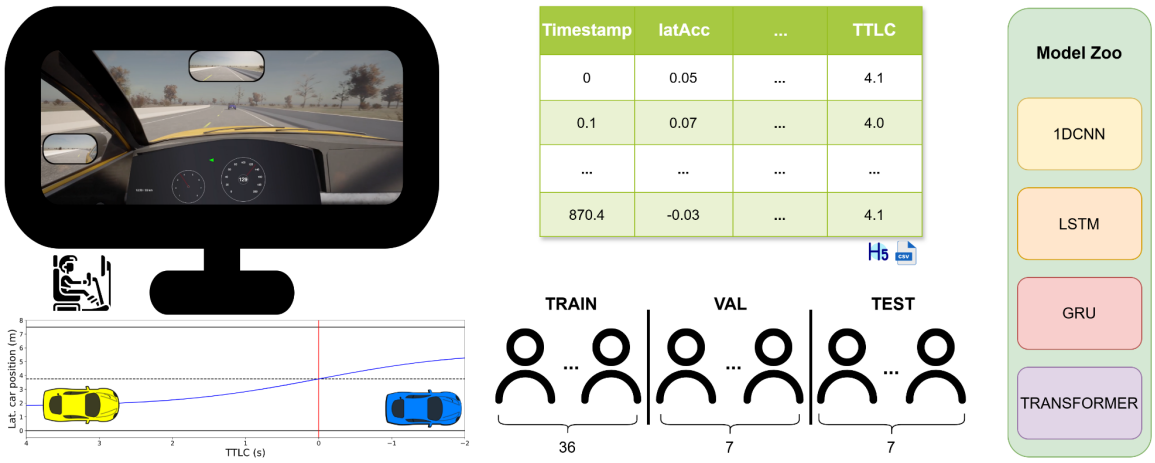


Fig. 4.1 Pipeline of driving simulation, data collection, and model training.

4.2.1 Participants and Procedure for Data Collection

To ensure a robust representation of driving behaviors, we engaged 50 participants in a controlled experiment designed to elicit naturalistic responses under standardized conditions. All participants provided informed consent. The cohort was heterogeneous in age (19–39 years, average 27), nationality (Italy, Iran, Algeria, Russia, Ethiopia, Lebanon), sex

(14 females, 36 males), and driving experience. Building on the principles established in Section 3.7.1, Chapter 3, this chapter serves as an expansion and confirms that the research adhered to the requisite standards to ensure the confidentiality of participants' data and information.

Each session consisted of a 30 km drive, with starting positions varied systematically; an unconstrained familiarization preceded every session. Each driver session averaged 15 minutes, totaling $\sim$12.5 hours of recorded data. Hardware included a Logitech G920 force–feedback system, a 29" ultra–wide curved display (106° horizontal FOV), and an ergonomic seat; a 1080p camera recorded driver behavior and a supervisor documented critical events.

4.2.2 Logging and Dataset Preparation

During each session, the CARLA log was sampled at 10 Hz and converted into a tabular CSV. Each file contains 31 features, including timestamp, EV dynamics (acceleration, speed, indicator status, pedal activity), road geometry (curvature, distance to lane boundaries), and details about the nearest non-player vehicles (NPVs), either hazardous (HNPVs) and standard (SNPVs). Features were selected for real-world relevance and validated experimentally.

To capture LC intention, the target is TTLC. Unlike prior work that frames intention as binary classification [178, 179], we regress TTLC in $[0,4]$ s with 0.1 s granularity. Each frame is labeled with its TTLC to the next LC; frames beyond 4 s are labeled 4.1 to indicate absence of LC intention. A 1.5 s window after each LC is excluded to remove ambiguous dynamics. Time series are segmented into 5 s windows (50 frames at 10 Hz), and each sample inherits the TTLC of its last frame.

LC dynamics across participants show large variability in frequency (Fig. 4.2). The trajectory analysis (Fig. 4.3) indicates that lateral motion typically starts between 2.5 and 1.5 s before LC; beyond $\sim$2.5 s the model must rely more on scenario context than on kinematic precursors. Distance/time headway to the leading vehicle (LV) at LC varies across drivers, which impacts both model design and evaluation.

To avoid subject leakage [44], participants are split into disjoint sets: 36 users (72%) for training, 7 (14%) for validation, and 7 (14%) for testing.

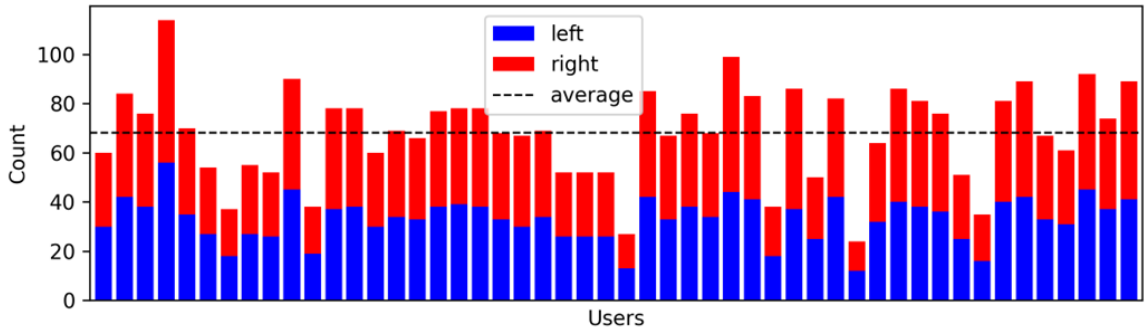


Fig. 4.2 LC occurrences across 50 participants, split by right/left LCs.

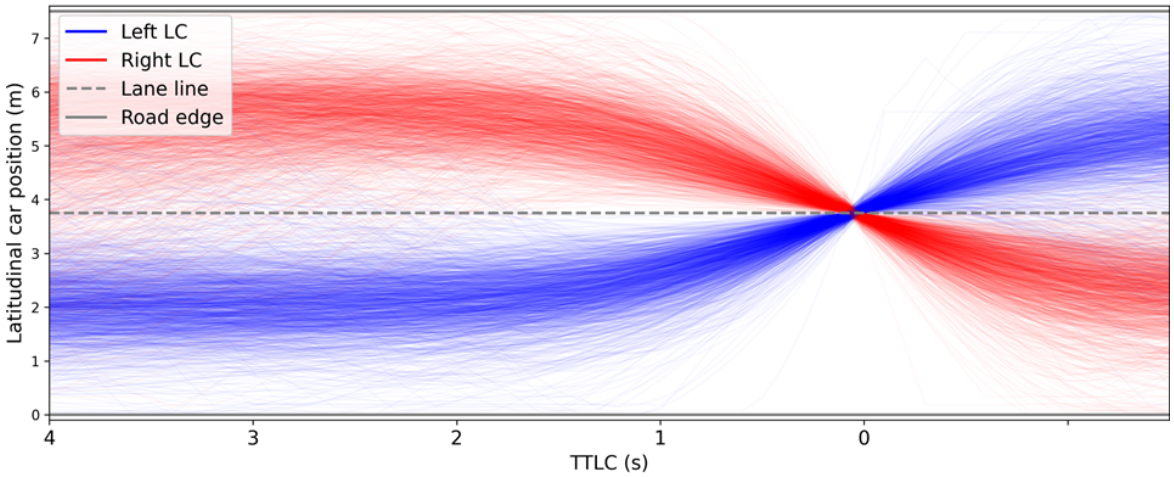


Fig. 4.3 LC trajectories across 50 participants.

4.3 Experimental Assessment

4.3.1 Evaluation Protocol

We report results from an LC prediction experiment conducted in Hi–Drive [180]. The test included all windows labeled in $[0, 4]$ s (LC). To balance LC and Free ride, windows labeled 4.1 s were sub-sampled to match the LC count. The resulting dataset consisted of 165,182 training, 36,602 validation, and 36,100 test samples.

4.3.2 Results and Deployment Implications

We evaluated four time-series architectures: 1D CNN, LSTM, GRU, Transformer. A 50–trial Bayesian optimization tuned each model on the validation set. RMSE, MSE, and MAE are computed over the TTLC regression target, expressed in seconds. MAE is defined in Eq. **??**

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{4.1}$$

Table 4.1 TTLC regression on the held-out test set (seconds).

Model	RMSE [s]	MSE [s]	MAE [s]	Model size [KB]	Inference time [ms]	Power [mW]	Energy [mJ]
1DCNN	0.544	0.296	0.310	**331**	**9.9**	**70**	**0.7**
LSTM	0.549	0.301	0.302	2,039	217.3	65	14.1
GRU	0.568	0.323	0.299	1,897	193.4	65	12.6
Transformer	**0.510**	**0.260**	**0.298**	799	115	80	9.2

The four architectures were selected to span the main families of sequence models used in driving intention recognition literature: CNN as a lightweight convolutional baseline [181, 182], LSTM and GRU as established recurrent architectures widely adopted for time-series prediction in AD [62, 183], and Transformer as the current state-of-the-art sequence model for long-range temporal dependencies [3]. This selection enables a systematic comparison of computational efficiency against predictive accuracy across model families.

The best Transformer model consists of a positional encoding applied to the input, followed by two encoder blocks with single–head attention (head size of 128) and feed–forward layers of equal size. The regression head consists of a global average pooling layer, a dense layer with 160 units and dropout of 0.1, and a final linear output. The model uses Adam with a learning rate of 0.001.

The optimal 1D CNN includes two convolutional layers with 64 and 32 filters (kernel size 3), followed by dropout of 0.2 and 0.4. A dense layer with 32 units and dropout of 0.2 precedes the linear output. Training uses Adam with 0.001 learning rate.

The selected LSTM model consists of a single LSTM layer with 192 units and a dropout of 0.4, followed by a dense linear output. It is trained using Adam with a 0.001 learning rate.

The best GRU model features three stacked GRU layers with 192, 32, and 32 units, respectively, each one followed by dropout of 0.2. The last layer is mapped to a linear output, and the model is trained with Adam at a 0.001 learning rate.

Transformer yields the lowest error (RMSE 0.510, MAE 0.298) at higher complexity (799 KB size, 80 mW power consumption, 9.2 mJ energy consumption). 1D CNN is the most compact (331 KB) with competitive accuracy (RMSE 0.544), offering a favorable trade-off

for constrained deployments. Overall, LC intentions are recognized up to 4 s in advance with average error below 0.6 s across diverse models.

Comparison With Existing Works

Table 4.2 resumes comparison between our work and other significant researches. Rule-based and computational driver models (e.g., [184]) showed that LC anticipation is feasible but reliable mainly at onset (1–2 s) and under hand-crafted assumptions. Data-driven classifiers improved short-horizon performance: LSTM/Bi-LSTM on NGSIM or naturalistic logs reach F1/Acc. $\sim$0.8–0.9 at 2–3 s [62, 63], and classical ensembles on SHRP2 NDS report high accuracy when enriched with curated features [64]. CNN variants push farther (up to 7 s) but degrade with horizon length [52], reflecting the difficulty of inferring intent before lateral motion cues become visible.

Compared to this line, our study differs in three aspects. First, we cast LC intention as TTLC regression rather than (bi/multi-class) classification, providing a temporally calibrated signal that is directly usable in DM and arbitration. Second, we evaluate up to 4 s with subject-independent splits on a controlled yet heterogeneous CARLA corpus (50 drivers, 3.4k LCs), showing sub-0.6 s average error (RMSE 0.51 s, MAE 0.30 s) where many classifiers either switch to coarse labels or lose reliability. Third, we analyze accuracy–efficiency trade-offs: a compact 1D-CNN offers strong deployment characteristics, while a Transformer attains the best error, informing choices under compute constraints.

Overall, prior work excels at short horizons or with rich engineered features; our results indicate that TTLC regression on L time-series can sustain competitive performance deeper into the anticipation window, narrowing the gap between maneuver prediction and its integration in real-time ADF stacks.

Furthermore, with the release of the framework and dataset, we significantly contribute to the current research.

Table 4.2 LC intention recognition: comparison with representative works.

Paper	Task	Window	Model	Dataset	Key results
[184]	Detection	1–2 s	Rule-based + driver model	Simulator + instr. vehicle	$\sim$85% detection; reliable only at LC onset.
[62]	Classification	3 s	LSTM	NGSIM	F1 $\approx$0.82 at short horizons.
[63]	Classification	2 s	Bi-LSTM	Naturalistic (20k seq.)	Acc. $\approx$90%.
[64]	Classification	1–5 s	RF, SVM, ANN, XGBoost	SHRP2 NDS	Up to 95.9% with rich features.
[52]	Classification	up to 7 s	Multi-channel CNN	NGSIM	$\sim$99% at 3 s; $\sim$95% at 4 s; $\sim$87% at 7 s.
Ours	Regression	up to 4 s	Transformer, 1D-CNN	Custom CARLA (50 drivers)	RMSE = 0.51 s, MAE = 0.30 s; direct TTLC regression.

Temporal Error Analysis

Fig. 4.4 shows predicted vs. ground-truth TTLC distributions. For 0–1.2 s the trend is close to ideal. In 1.2–3 s the model tends to overestimate; for 3–4 s it regresses toward the mean. Free ride cases (4.1) are well separated near the upper bound. For ADF design, a practical pairing is a binary intention detector (0–5 s) plus a regressor focused on 0–3.5 s.

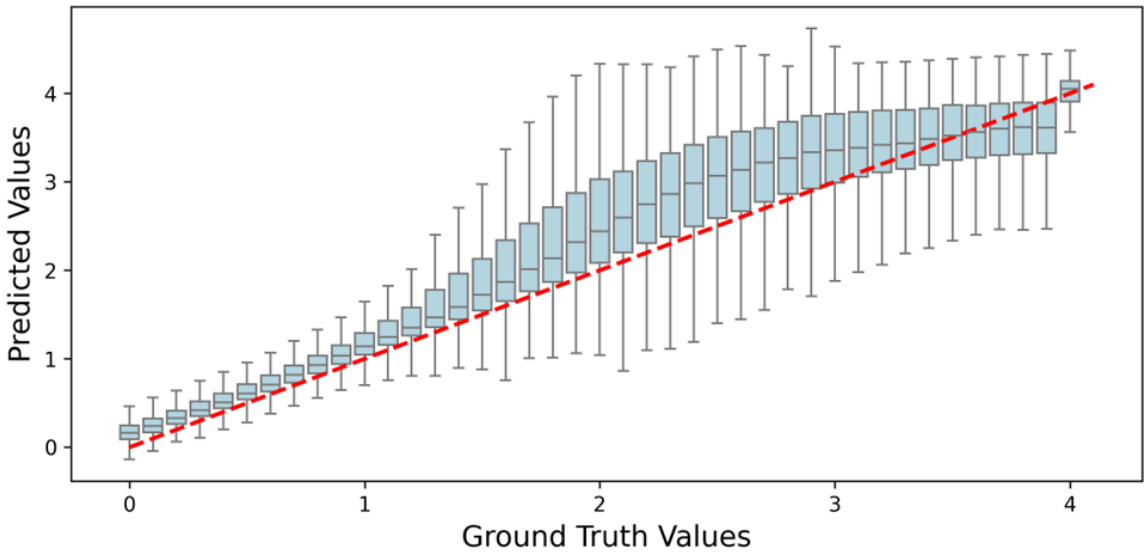


Fig. 4.4 Predicted vs. ground-truth TTLC distributions (0–4 s). The dashed red line indicates $y = x$.

4.4 Generation of Synthetic Datasets of Highway Driving Scenarios

The LC results above show that beyond ∼2.5 s, anticipation depends increasingly on the visual scene and interaction context (Fig. 4.3 and 4.4). We therefore expand from maneuver prediction to scenario generation and detection, reusing the same highway map to render scenario classes and exporting MP4 clips (no telemetry).

A key link is the simulation environment. Among the road maps employed, one is our highway introduced earlier, used here alongside a standard CARLA town [185]. Its 60 km length supports long uninterrupted runs; the mix of long straights (up to 4.4 km) and balanced left/right curves (radii 232–697 m) creates varied relative-speed and occlusion patterns (Cut-in/Cut-out, Approach/Follow, LC). The junction-free layout and minimal scenery keep physics stable and frame rates consistent, improving reproducibility; lane geometry and markings simplify annotation.

We formalize a methodology to create synthetic highway scenario datasets grounded in MOOVE distributions and rendered via an enhanced CARLA/ScenarioRunner stack [98, 185]. The main goal is to address the current shortage of publicly available driving scenario-labeled datasets. We present a dataset spanning nine scenario classes (Brake, free ride, Cut-in, Cut-out, Ego LC, Approaching LV, Pulling away LV, Following LV, Following distant LV) together with a reference classification baseline. The focus here is scenario classification over synthetic video, complementing the LC intention regression built on human trajectories. To validate the sim-to-real transferability of the proposed pipeline, the trained model is evaluated in a zero-shot setting on a real-world highway dataset (PREVENTION [186], without any fine-tuning. To support open research in the field, the whole dataset has been publicly released [2] [4].

4.4.1 Data Preparation Methodology

We adopt the PEGASUS functional–logical– concrete hierarchy [87] and the 6LM environment model [93] as the organizational backbone of the scenario generation pipeline. Its taxonomy is explicitly compatible with ASAM OpenSCENARIO 2.0 [187], the interchange format adopted in our scenario generator, ensuring interoperability with industrial toolchains and coherence with the statistical distributions from the MOOVE project [98, 99].

Table 4.3 summarizes the match between the abstraction levels and our process.

A key tenet, also for development of VR-based tools, is centrality of real-world data. To this end, a fundamental aspect consists in the specification of the parameter values of the logical scenarios. To support realism and user need centrality, our generator system (described in the next section) has been designed to allow the user to specify the distributions as a configuration file. In this work, we focus on a single dataset implementation, exploiting a subset (for confidentiality reasons) of data from the MOOVE project. Table 4.4 lists functional scenarios. The considered classes are illustrated in Fig. 4.5.

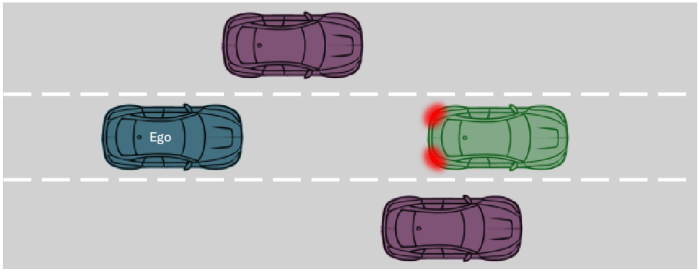
[2] https://zenodo.org/records/17652045

Table 4.3 OpenSCENARIO 2.0 abstraction levels vs. our dataset process.

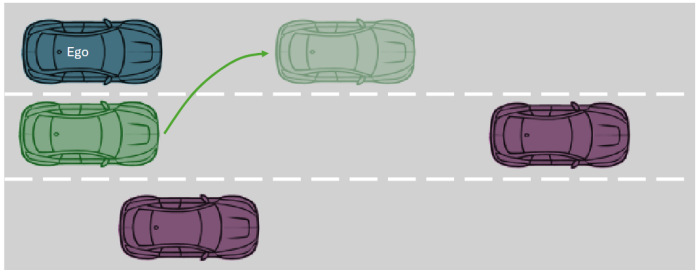
Scenario-abstraction level	**Implementation**
Functional	Based on MOOVE data [98] and experience in L3Pilot [188] and Hi-Drive [180], we defined the target scenarios (Table 4.4). E.g., Cut-in: "A vehicle changes lane and enters the lane in front of the EV."
Abstract	Dependencies and relations between attributes and behaviors are specified via constraints on the scenario actors and their interactions. Cut-in: EV speed > 0; cutter starts in adjacent lane and ends in EV lane.
Logical	Each class is formalized in Python and parameterized over road geometry, traffic dynamics, and environmental conditions. Users specify probability histograms per parameter (Table 4.5). E.g., Cut-in: initial EV speed $\in [8, 171]$ km/h; relative cutter speed $\in [-118, 71]$ km/h; initial distance $\in [3, 200]$ m; direction: left or right.
Concrete	Each sample instantiates a logical scenario by sampling from the distributions. E.g., Cut-in: initial distance $= 45$ m; cutter speed $= 87$ km/h; direction: left; rainy weather; medium traffic.

Table 4.4 Functional scenarios.

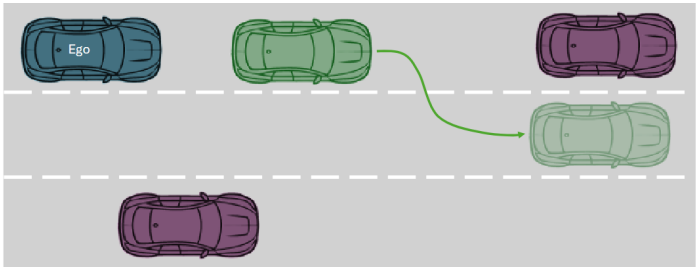
Scenario name	Description
Cut-in	A vehicle changes lane and enters the lane in front of the EV.
Cut-out	A vehicle in front of the EV leaves its lane.
Ego LC	The EV changes lane.
Approaching LV	The EV gets closer to the leading vehicle.
Pulling away LV	The EV gets farther away from the leading vehicle.
Following LV	The EV maintains a speed similar to the leading vehicle.
Follow distant LV	The EV follows the LV at > 50 m.
Free ride	The LV, if present, is > 200 m away.
Brake	The LV brakes in front of the EV.



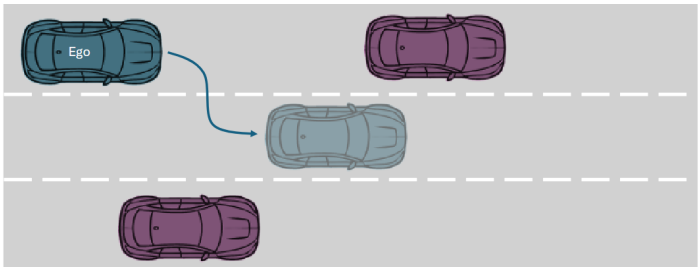
(a) Brake scenario. The LV brakes.



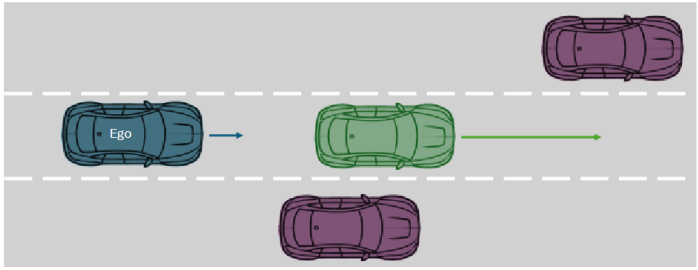
(b) Cut-in scenario. a vehicle drives up and positions itself in front of the EV.



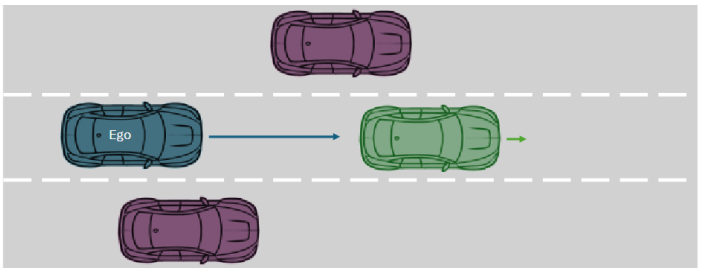
(c) Cut-out scenario. The LV moves lanes and no longer occupies the same lane as the EV.



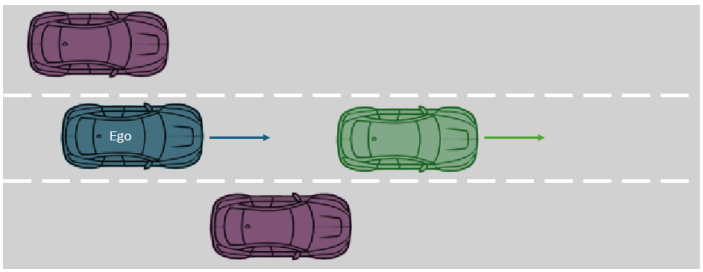
(d) Ego LC scenario. The EV performs a LC.



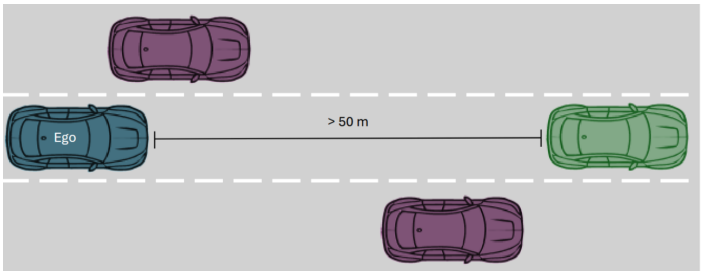
(e) Pulling away from LV scenario. The LV increases the distance with the EV.



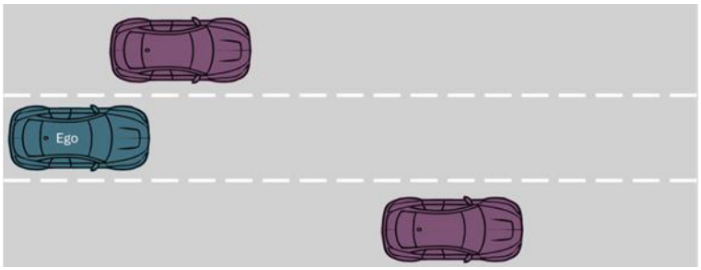
(f) Approaching LV scenario. The EV narrows the distance with the LV.



(g) Following LV scenario. The EV follows the LV. No significative increase in the distance between them.



(h) Following distant LV scenario. The EV follows a distant (> 50m) LV.



(i) Free ride scenario. The LV, if present, is > 200 m away.

Fig. 4.5 Scenario classes.

Table 4.5 reports, for each scenario class, its relevant scenario-specific parameters and the value ranges extracted by MOOVE. Other significant variability elements for any functional scenarios are represented by generic context parameters, such as environment, illumination [189] weather conditions, and context traffic (also in the opposite direction). Table 4.6 reports the generic parameters we defined for our generation system and the value ranges we considered for the case study at hand. For simplicity, and given the lack of relevant informa-

tion, we consider the scenario-specific and generic parameter distributions as independent of each other.

Table 4.5 Scenario-specific parameters and their value ranges.

Scenario name	**Parameters and Value Ranges**
Brake	Initial ego speed ($[6,137]$ km/h); Relative lead speed ($[-33,19]$ km/h); Delta lead speed ($[-57,7]$ km/h); Duration ($[0.1,9.4]$ s); Initial distance ($[5,87]$ m).
Cut-in	Initial ego speed ($[8,171]$ km/h); Relative cutter speed ($[-118,71]$ km/h); Delta cutter speed ($[-62,40]$ km/h); Duration ($[0.3,40.6]$ s); Initial distance ($[3,200]$ m); Direction (Left or Right).
Cut-out	Initial ego speed ($[7,173]$ km/h); Relative cutter speed ($[-123,71]$ km/h); Delta cutter speed ($[-74,57]$ km/h); Duration ($[0.1,50.9]$ s); Initial distance ($[3,200]$ m); Direction (Left or Right).
EV LC	Initial ego speed ($[1,182]$ km/h); Delta ego speed ($[-55,127]$ km/h); Duration ($[0.1,18.7]$ s); Direction (Left or Right).
Following LV	Initial ego speed ($[7,170]$ km/h); Relative lead speed ($[-53,47]$ km/h); Delta lead speed ($[-57,48]$ km/h); Duration ($[0.1,81.3]$ s); Initial distance ($[6,143]$ m).
Free ride	Initial ego speed ($[10,136]$ km/h); Relative lead speed ($[-41,41]$ km/h); Delta lead speed ($[-43,32]$ km/h); Duration ($[0.9,98.4]$ s); Initial distance ($[204,247]$ m).

Table 4.6 Generic parameter value ranges.

Parameter name	Value range
Weather	Clear; cloudy (four intensity levels); rain (four intensity levels).
Time of day	Dawn, morning, noon, evening, sunset, night.
Fog	Yes or no.
Opposite-direction traffic	None, low traffic, or heavy traffic.
Same-direction traffic	Free ride, low traffic, medium traffic, or high traffic.
Type of traffic vehicle	Car, truck, or bus.

4.4.2 System Architecture

Fig. 4.6 sketches the architecture for generating scenario instances based on user-specified scenario types, parameter distributions, and sensors. The Configurator initializes the LSH with distributions and sensor settings; the LSH creates the requested number of Concrete Scenarios by sampling parameters, and initializes a Validator monitoring execution and recording.

Each Concrete Scenario executes simulation with weather/time-of-day, while the Validator performs online checks; offline validation inspects the CARLA log. The pipeline produces variable-length MP4 files at 1280×720 and includes conversion to L3Pilot CDF [190].

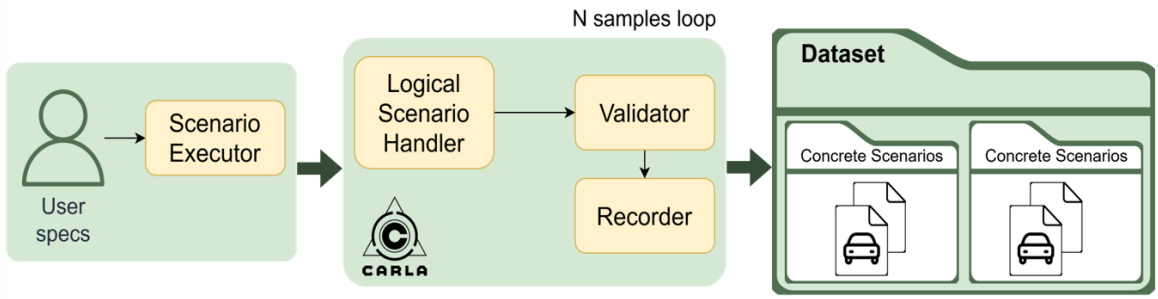


Fig. 4.6 System architecture.

Logical Scenario

We designed the Logical Scenario class to define, step by step, the behavior of each actor in the scenario including the protagonists (i.e., the EV and, if relevant to the scenario

type, the leading/rear vehicle) and the non-protagonists, such as the traffic components. An OpenSCENARIO 2.0 logical scenario consists of a hierarchical composition of phases (composites) specifying the behavior of one or more actors. Events, triggered by triggers, cause the execution of actions. For instance, proximity to a car on the same lane triggers a left lane change maneuver We formalized logical scenarios as trees, that are implemented through PyTree, a python library for tree data structures. We map OpenSCENARIO 2.0 composition to PyTree composites, and OpenSCENARIO 2.0 actions and triggers to PyTree behaviors.

PyTree composites serve as a structural backbone for managing and executing child behaviors, that can be sequential or parallel. The description of each scenario in our Generator (Fig. 4.7) is a sequential composite, containing three parallel composites: (i) traffic initialization (vehicle spawning and application of the initial speeds), (ii) recording start and scenario action execution, (iii) recording end. We highlight that the recording starts in the second block, as soon as all the target parameter values are reached and provided that the other protagonist than the EV (if any) is not occluded, which requires few seconds since the initial speed application is not instantaneous.

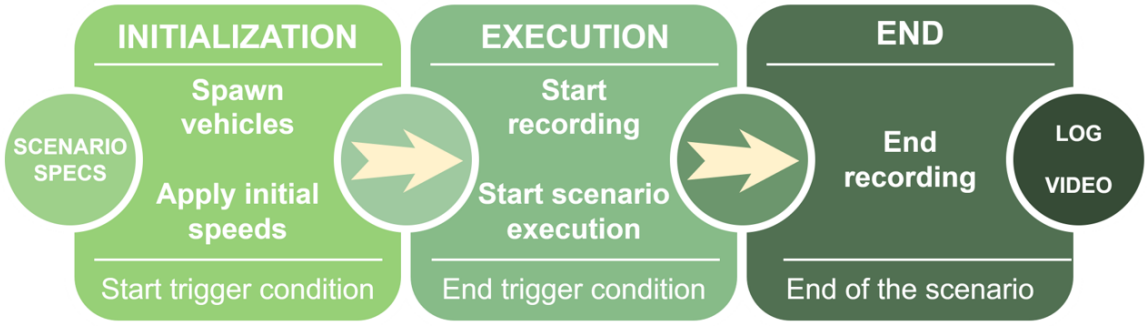


Fig. 4.7 Generic workflow of a logical scenario.

We added specific rules for traffic generation. We set four traffic intensity clusters: freeway, low traffic (1 to 10 vehicles around the EV), medium traffic (10 to 30), heavy traffic (30 to 50). Then we based the traffic density group on the EV speed (ground truth data), choosing medium/high traffic if the EV speed is low and no/low traffic if speed is high. Also, we set the distance between vehicles based on the THW (i.e., the time gap between the front bumper of the EV and the front bumper of the LV) distributions from a real-world traffic analysis [191]. For instance, for the cut-in scenarios, we set a THW between 1 and 2 seconds for medium/high traffic scenarios, and a THW of 2 to 5 seconds for low traffic scenarios. We also added few cases with one or more traffic vehicles having lower THW (until 0.7 seconds), to account for rare hazardous conditions.

Algorithm 2 provides the pseudo-code example of a logic scenario implementation, consisting of four phases. The first one spawns all the specified vehicles. The second applies the initial speed to all the vehicles and ends as soon as the target distance between the EV and the protagonist is reached. The third one starts the recording and includes behavior nodes for the protagonist (execute a LC maneuver) and for all the others (proceed in their lane). The ending trigger checks whether the protagonist has reached the end of its path. The last phase ends the recording.

Algorithm 2

```
SCENARIO CUT_IN:
  vehicle ego_vehicle, vehicle cutter
  do serial:
    drive_start: parallel
      ego_vehicle_drive(path) with speed([25kph,...,80kph]), lane(2)
      cutter_drive(path) with speed([25kph,...,80kph]), lane(left_of: ego_vehicle)
    change_lane: parallel(duration: [1s,...,15s])
      cutter executes lane change into EV lane
    drive_end: parallel(duration: 1ms)
SCENARIO TOP: path.set_map("TownX"); cut_in()
```

Scenario Execution, Recording and Validation

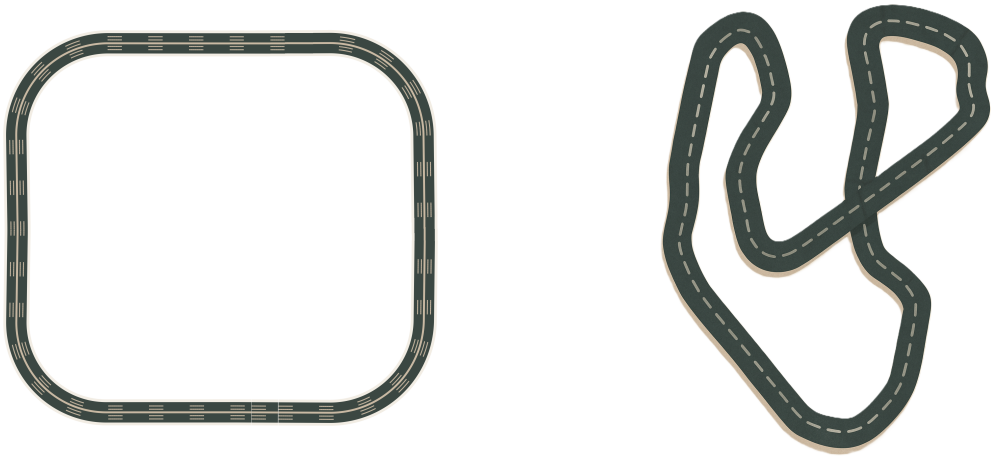
Each dataset sample is created by instantiating a Concrete Scenario class through the sampling of the parameters' probability distributions and of specific trigger, traffic and action values. The Concrete Scenario executes all the actions defined by the Logic Scenario, also checking absence of collisions.

Validation is performed online and offline, at the end of the generation, inspecting the CARLA log file, which reports position and orientation of each vehicle at every simulation tick. The check concerns the following: (i) absence of off-roads and crashes; (ii) a-posteriori check of the thresholds (and consequent class assignment) among similar scenarios: free ride, following LV, following distant LV; (iii) check that the non-EV protagonist is visible in the video clip. If a violation of type (i) or (ii) is detected, the sample is deleted. Concerning the (iii) check, if the non-EV protagonist is not visible at the beginning of the scenario, the recording starts as soon as it becomes visible, while if this happens after the scenario recording start, the simulation is stopped and the partial clip stored.

4.4.3 Experimental Results

Dataset Distributions

As anticipated, the generated synthetic dataset is based on the traffic data distributions provided by the MOOVE project. The virtual environment context is given by one town available in the CARLA package (Fig. 4.8a) and our custom map (Fig. 4.8b), which allows to augment the dataset variety and to match the MOOVE specifications in terms of curve radii, number of lanes and length of straight sections (Table 4.7).



(a) External Ring of CARLA Town5.

(b) Our custom highway map.

Table 4.7 Maps used in the dataset.

Map	Area (km^2)	Path length (km)	No. lanes	Min/Max radius (m)	Min/Max straight (m)
Fig. 4.8a	0.16	2.04	3	97.2–500	6.40–139.39
Fig. 4.8b	147	60	2	232–697	1000–4400

Table 4.8 reports the distribution of scenario classes in the training, validation, and test sets. While validation and test sets follow the the same distribution provided by MOOVE, reflecting occurrences of driving scenarios in the real world, the training set distributions have been rebalanced multiple times to improve the model performances, as the default data distributions did not yield satisfactory results.

Table 4.9 provides an overview of the high-level parameters characterizing each scenario class in the training set, including average and standard deviation (computed per video-clip) of the instance durations, and of vehicular information such as speed and acceleration of

the EV and of other traffic vehicles, number of surrounding vehicles (within a 100 meter radius). The training set is quite balanced in terms of types of generated video-clips, but their time-lengths differ significantly, with Cut-in being the class having (on average) the shortest clips and free ride the longest (some samples are longer than 60 seconds). Results in the other columns show the large variability of the produced samples. Variability is within scenarios (as can be observed by the standard deviation values) and between scenarios. For instance, we see that low-traffic scenarios (e.g., free ride) are characterized by higher speeds. Data about speed and number of vehicles clearly show that the majority of the original distribution datapoints concern highways with medium-high traffic.

Table 4.8 Scenario distributions for training, test, and validation sets.

Scenario Class	**Train**			**Val**			**Test**		
	# videos	**# samples**	**% samples**	**# videos**	**# samples**	**% samples**	**# videos**	**# samples**	**% samples**
Brake	13,990	292,822	10.4%	1,021	20,673	7.6%	959	20,699	7.6%
Cut-in	14,720	297,555	10.6%	499	9,421	3.4%	497	9,352	3.5%
Cut-out	14,539	297,605	10.6%	362	7,515	2.7%	315	7,548	2.8%
EV lane change	9,545	302,063	10.7%	639	20,157	7.4%	639	20,141	7.4%
EV pulling away from LV	9,285	285,295	10.1%	633	17,721	6.5%	881	17,582	6.5%
Approaching LV	10,982	290,554	10.3%	747	17,521	6.4%	755	16,835	6.2%
Following LV	11,910	446,624	15.8%	1,296	43,896	16.0%	1,296	42,245	15.6%
Following distant LV	6,937	297,743	10.6%	1,078	39,333	14.4%	900	38,835	14.3%
Free ride	4,694	308,989	11.0%	1,460	97,680	35.7%	1,420	97,457	36.0%

Table 4.9 High-level characterization of the scenario classes in the training set.

Scenario	Mean duration (std) [s]	Mean EV speed (std) [km/h]	Mean EV acc. (std) [m/s^2]	Mean traffic speed (std) [km/h]	Mean traffic acc. (std) [m/s^2]	Mean # traffic vehicles (std)
Brake	3.09 (1.82)	40 (25)	0.00 (0.16)	47 (23)	-0.01 (0.16)	18 (6)
Cut-in	3.01 (2.37)	65 (32)	-0.02 (0.35)	55 (26)	0.00 (0.02)	8 (4)
Cut-out	3.05 (2.35)	64 (31)	0.01 (0.22)	53 (26)	0.00 (0.02)	8 (4)
EV LC	4.16 (1.34)	81 (31)	0.01 (0.21)	57 (33)	0.01 (0.03)	3 (3)
EV pulling away from LV	4.09 (3.66)	56 (34)	0.01 (0.11)	43 (27)	0.00 (0.02)	8 (4)
Approaching LV	3.62 (2.29)	48 (31)	0.00 (0.16)	45 (24)	0.00 (0.02)	6 (3)
Following LV	4.79 (4.57)	50 (31)	0.00 (0.15)	44 (25)	0.00 (0.02)	7 (4)
Following distant LV	5.18 (4.79)	95 (28)	0.00 (0.13)	91 (26)	0.00 (0.02)	4 (2)
Free ride	7.44 (9.42)	110 (26)	0.00 (0.04)	109 (30)	0.00 (0.02)	1 (1)

Fig. 4.9 shows the scenario-wise breakdown of traffic density in road segments around the EV. The figure clearly shows the difference between low-traffic scenarios (free ride, but also EV LC and Following distant LV) and medium/high traffic scenarios (the three other types of Following LV, but also the Cut-in/out and Brake). The large standard deviation values show the intra-scenario variability, which is significant particularly in high-traffic.

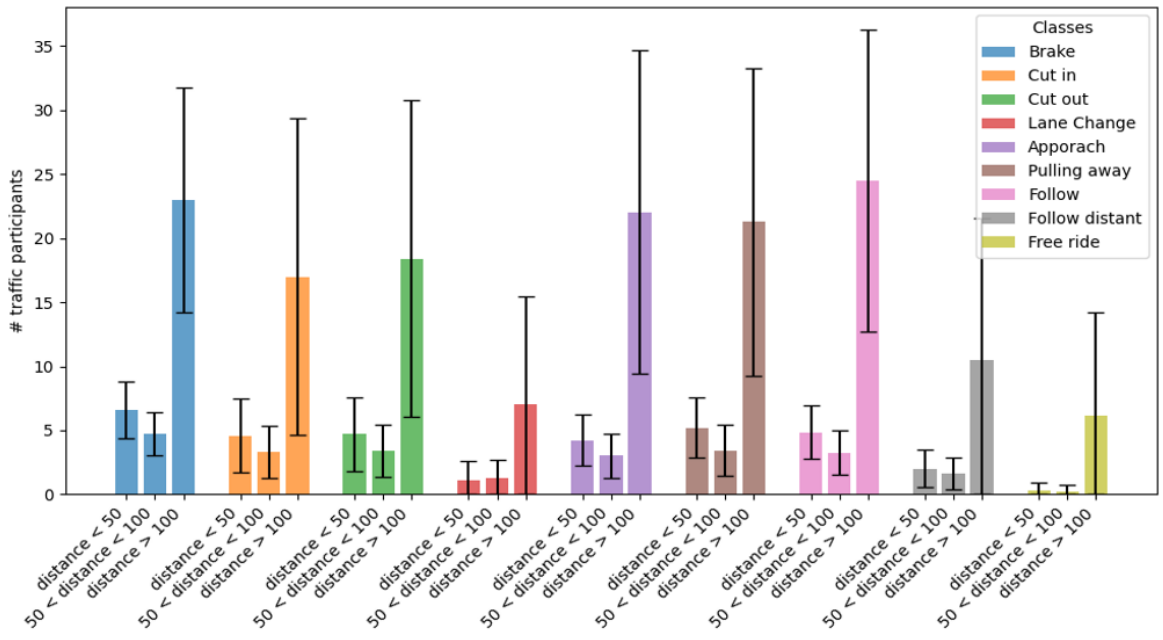


Fig. 4.9 Scenario-wise breakdown of traffic density around the EV.

Another important aspect of the dataset characterization concerns the distribution of the weather, time of day and visibility (presence of fog) conditions, reported in Fig. 4.10.

Rare conditions are over-represented to improve robustness. For the surrounding traffic, vehicles are randomly instantiated out of twenty-one different car models (each one is further differentiated in terms of colors). The presented distributions were considered suitable by the Hi-Drive analysis experts. Examples of scenarios with different weather conditions are depicted in Fig. 4.11.

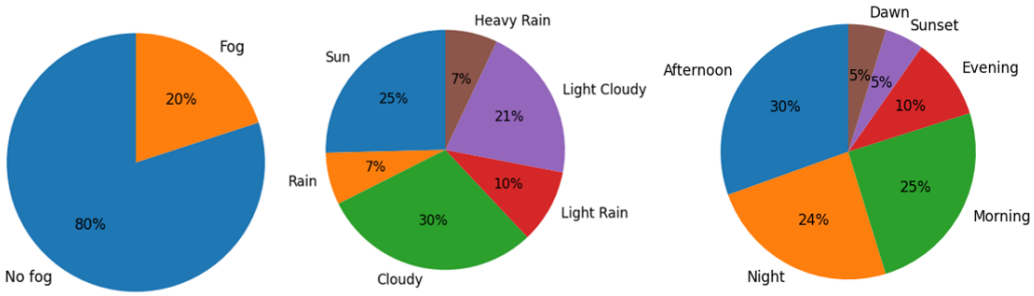


Fig. 4.10 Considered weather distributions.

(a) Brake scenario in the rain.

(b) Approaching LV scenario in clear weather at sunset.

(c) Cut-in scenario in cloudy weather with fog.

(d) Cut-out scenario in clear weather at noon.

Fig. 4.11 Examples of different scenario with different weather distributions, according to Fig. 4.10.

While the environmental conditions shown in Fig. 4.10 cover a broad range of weather and illumination scenarios, CARLA rendering engine (Unreal Engine [192]) approximates

these conditions through parametric models. As acknowledged in the CARLA documentation and related literature [36, 37, 193], the simulator does not fully replicate second-order optical artifacts present in real camera footage, such as lens distortion, sensor noise, and spatially inhomogeneous light scattering. These factors contribute to the domain gap observed during zero-shot testing on the PREVENTION dataset [74] (Section 4.4.5), where uncontrolled illumination and camera artifacts were identified as primary sources of performance degradation.

Scenario Generation Performance

A major goal of the proposed system is to efficiently generate datasets based on input configuration. Table 4.10 reports, for each class, the correct-instance generation rates, the error rate (as said, some clips may contain errors, typically long-lasting occlusions, and must be removed), and the simulation rate (number of simulation minutes generated per hour). Results show a significant variance among the different classes, that we essentially relate to the density of proximity traffic (cf. Fig. 4.9). Notably, production rates strongly depend on the length of scenario types (e.g., brake scenarios are much shorter than the others). Variance within classes is more limited (and similar across classes), and reflects the different conditions (e.g., weather and particularly context traffic) that may happen in each scenario type. Averaging over all scenario classes, the production time is 148 clips per hour and the simulation rate is about 17.5 min per simulation hour. Also averaging over simulation hours, the average simulation rate is similar ($\sim$18 min per simulation hour). Rates are measured on a workstation with an NVIDIA RTX 4080 GPU with 16 GB of VRAM and a CPU with 21 processors at 3.0GHz.

The Brake scenario achieves the highest number of correct instances rate (240), yet not achieving the lowest error rate. This is due to the intrinsic dynamic of the scenario, which led to several collisions to discard. Also, considering the short duration of such a scenario, the simulation rate is the lowest one among the scenario classes.

The Cut-in/out scenarios achieve overall high instance generation rates (about 190 per hour), moderate simulation rates (17–18 minutes/hour), and low error rates (8–10%), while the EV lane change has moderate generation rates (155 instances/hour, 23 minutes/hour) and relatively high error rates.

The near Following LV scenarios (Pulling away, Approaching, Following) produce fewer instances (115–120 per hour) and have a moderate simulation output (14–15 minutes/hour), but suffer from higher error rates (28%–32%), essentially because of the difficulty in maintaining proper following distances and behaviors during the whole simulation span.

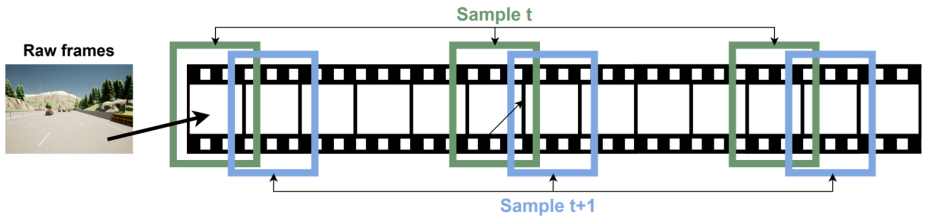
The Following distant LV scenario has the lowest clip generation rate (85 per hour) and the second lowest simulation generation low (13 minutes per hour), essentially because of the highest error rate (52%) and the difficulty of generating simulations keeping distant vehicles visible without occlusions.

Table 4.10 Scenario–generation rates.

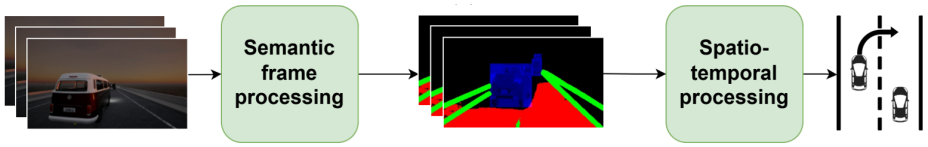
Scenario	**Correct inst. rate (std) [inst/h]**	**Error rate (std) [%]**	**Simulation rate (std) [min/h]**
Brake	240 (16)	22 (7)	10 (1)
Cut-in	190 (13)	8 (4)	23 (11)
Cut-out	188 (11)	9 (3)	23 (8)
EV LC	155 (31)	36 (16)	23 (7)
Pulling away LV	118 (29)	35 (16)	18 (16)
Approaching LV	115 (38)	30 (19)	13 (10)
Following LV	120 (29)	32 (18)	14 (9)
Following distant LV	85 (44)	52 (20)	13 (11)
Free ride	115 (23)	23 (11)	37 (22)

Scenario Classification Performance

A last, but key assessment of the system concerned the classification ability of a state of the art NN trained on the generated dataset. Coherently with the dataset design (single-scenario clips), we developed scenario recognition as a classification task of overlapping TWs of 1 second length (Fig. 4.12a). Each TW is generated with a stride of 0.1 seconds (so, the system response frequence can be up to 10 Hz) and consists of three equally spaced frames. The proposed pipeline involves frame-level semantic pre-processing followed by overall spatio-temporal processing (Fig. 4.12b). All the experiments we present in the following were conducted on an NVIDIA RTX 4080 GPU, with 16 GB of VRAM.



(a)



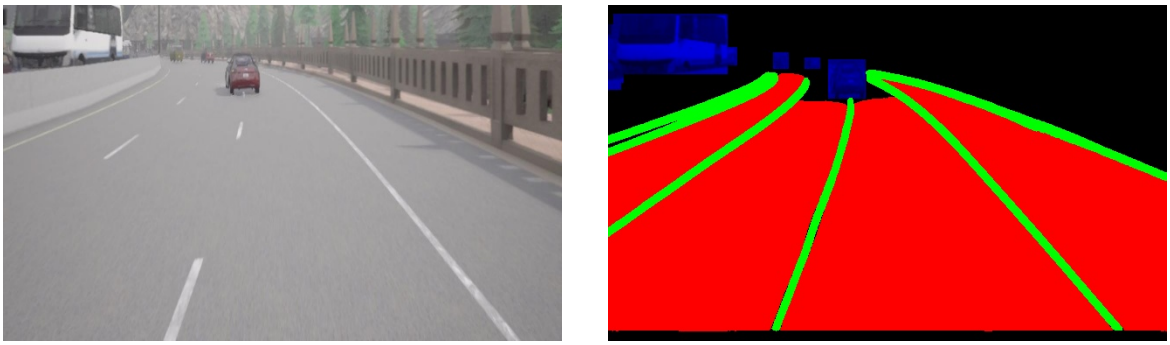
(b)

Fig. 4.12 Time windows 3-frame sampling (a) and scenario classification pipeline (b).

We compare three R3D CNN architectures [194]: the baseline model, one enhanced with the CBAM [195], and one enhanced with the SE module [196]. We chose this family of models, after comparing with other state of the art solutions (e.g., CNN+LSTM, Timesformer [197]), for their superior performance and simple structure, leading to a smooth training process. Each model was trained on the synthetic dataset described in the previous sections, comprising approximately three million one-second TWs. Specifically, the training set (Table 4.8, Train column) contains about 2.8 million samples, while the validation and test sets include around 270,000 samples each. To avoid overfitting, the training, validation and test sets contain samples taken from distinct videoclips.

Before feeding the network, each 1280×720 frame is processed by YoloPv2 [198] to extract high-level semantic information such as the vehicles bounding boxes, the drivable area and the lane line area (Fig. 4.13), implicitly encoding spatial relationships without requiring explicit depth maps. We acknowledge that depth ambiguity remains a limitation, particularly for distance-dependent classes such as Following distant LV, and that future work could incorporate stereo or LiDAR inputs to improve robustness.

These three types of outputs are combined into a single three-channel representation, including also the original frame, in order to highlight the most informative features. The frame is cropped to (640×384) pixels, focusing on the driving scene, and finally resized to 224×224.



(a) Original frame. (b) YoloPv2-processed frame.

Fig. 4.13 Semantic preprocessing with YoloPv2.

The pipeline keeps a real-time performance, with a latency of 17 ms (8 ms of generation time and 9 ms for detection). Removing background noise and highlighting semantic information improves the ML metrics by about 2-5% [199, 200]. We also verified that this solution outperforms a motion-based preprocessing approach (i.e., optical flow [201]) in terms of both training time and ML performance [200]. We trained the R3D CNNs with a mini-batch size of 32 for 5 epochs for a total time of $\sim$200 hours. We used an Adam optimizer with a learning rate of 10^{-3}, momentum parameters ($\beta 1$ and $\beta 2$) set to 0.9 and 0.98 respectively, and an ε value of 10^{-8}. Training was performed using a weighted cross-entropy loss function, where a sample-specific class weight was applied to mitigate imbalances in the sample's dataset distribution. Particularly classes with fewer instances were given higher weights to reduce the overrepresentation of more common categories and thus prevent biased convergence towards majority classes. This setup ensured a more balanced learning process across all classes.

Table 4.11 reports the aggregate classification metrics for each considered model. The best results are obtained by the R3D CNN+SE module, outperforming standard and CBAM-enhanced R3D networks.

Table 4.11 Classification performances.

Model	Accuracy	Precision	Recall	F1 score
R3D	91.1%	91.5%	91.1%	91.2%
R3D CBAM	91.2%	91.4%	91.2%	91.3%
R3D SE	**91.9%**	**92.1%**	**91.9%**	**91.9%**

Classification results across the considered scenario classes are presented in the confusion matrix in Fig. 4.14. The model (we detail best model results only) achieves high accuracy in

classes such as Brake (98.75%), EV lane change (98.08%), Free-ride (97.36%), and Cut-in (92.48%).

On the other hand, several misclassifications, occur between behaviorally-similar classes. For instance, Approaching LV is misclassified as Following LV in 8.78% of the cases, and as Brake in 3.54%. Pulling away from LV is misclassified as Following (8.07%) and Following distant (6.28) classes. Pulling away LV is confused with Following LV in the 8.30% of the samples. In 7.45% and 5.56% of the samples, Following LV is misclassified as Pulling away from and Approaching LV respectively.

Cut-out is taken as Cut-in 3.78% of the times (2.04% for the vice-versa). This highlights a limit of the model generalization capability, since the adversarial movement for the two scenarios is the same, the discriminant being the reference to the EV lane (but different positions in lane make some samples more ambiguous, particularly on curved road segments).

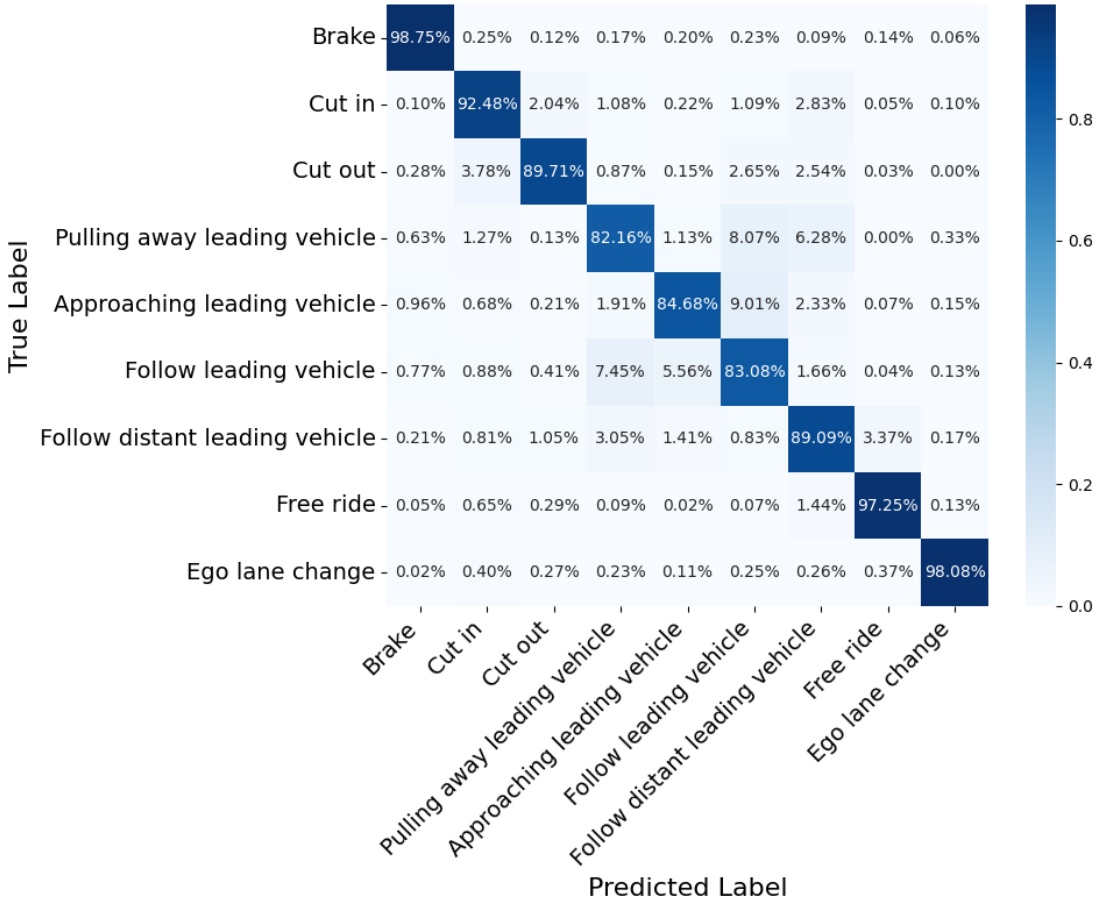


Fig. 4.14 Confusion matrix on the synthetic scenario dataset.

Given the highly imbalanced test set (Table 4.8, Test column) we also analyze the PRC, which is more informative than ROC [202] when dealing with highly imbalanced data [203]. The multi-class PRCs in Fig. 4.15 show that the model achieves consistently high performance across all scenario classes, with AP values ranging from 0.88 to 1.00. Specifically, Brake, Free ride, and EV LC classes reach perfect precision and recall (AP=1.00), while slightly lower values are observed for visually similar classes (Following LV-like classes), suggesting higher inter-class ambiguity.

The micro-average (black dashed line) aggregates all true positives, false positives, and false negatives across classes, providing a global estimate of detection performance (AP=0.98). This metric is particularly meaningful under class imbalance, as it weighs each sample equally regardless of its class. Therefore, the micro-average better reflects the overall quality of the model compared to the macro-average, which would overemphasize less likely driving scenarios according to the distributions provided by MOOVE.

Overall, these results suggest that the model maintains strong precision–recall characteristics even in imbalanced conditions, yet leaving room for improvement in the discrimination of maneuvers with similar spatial-temporal dynamics.

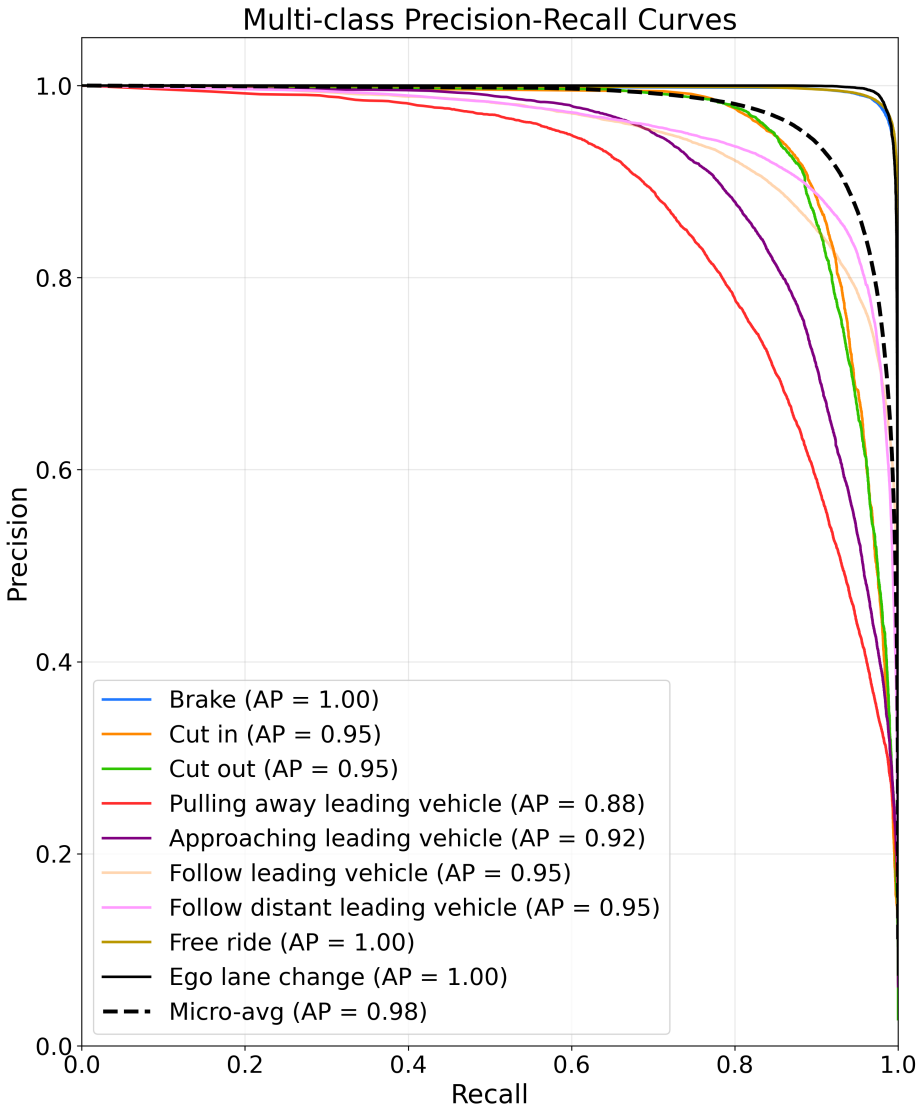


Fig. 4.15 PRC of each considered scenario class.

The obtained results collectively indicate some criticalities in identifying - using a single camera, with very short time spans - the nuances in the spatial and temporal features between visually-similar driving scenarios.

Beyond temporal constraints, another critical factor is the environmental complexity. To address the potential implications of traffic density on the model accuracy, we analyzed the relationship between classification performance and the environmental complexity described in Section 4.4.3. As illustrated in the scenario-wise breakdown (Fig. 4.9), the dataset exhibits significant variance in the number of traffic participants across different maneuvers and distance ranges.

The empirical results from the confusion matrix and PRC reveal several key insights regarding the robustness of the architecture:

- **High-Density Robustness:** Despite the high traffic density observed in the Brake scenario, averaging 18 vehicles (std = 6) as shown in Table 4.9, the model achieves a near-perfect AP of 1.00. This suggests that the SE blocks effectively recalibrate the feature maps, allowing the network to prioritize relevant spatio-temporal cues and ignore environmental noise even when the visual field is cluttered. This robustness is further supported by the explicit modeling of EV reactions, which proved fundamental in disambiguating dense scenarios.

- **Performance in Sparse Environments:** In low-density scenarios such as Free ride and EV LC, which typically involve only 1 to 3 vehicles, the model maintains maximum accuracy ($AP = 1.00$). This confirms that the architecture is inherently stable and does not suffer from performance degradation in the absence of complex interactions.

- **Semantic Ambiguity vs. Traffic Volume:** The slight performance drop for the Pulling away class ($AP = 0.88$, accuracy = 82.16%) is not strictly correlated with traffic density. As indicated by the Confusion Matrix, the primary source of error is the semantic overlap with the Following LV classes. This suggests that the challenge for the R3D kernels lies in distinguishing subtle longitudinal transitions (kinematic similarity) rather than the absolute number of actors in the scene.

In conclusion, the integration of 3D convolutions with channel-wise attention ensures that classification metrics remain consistent across the diverse traffic regimes identified in our scenario breakdown. These results demonstrate that the model's high overall performance is not merely an artifact of "easier" low-traffic cases, but a direct consequence of effective feature selection in complex driving environments.

The goal of this analysis is to give an assessment of the proposed dataset, providing a quantitative characterization. A thorough model optimization - investigating new types of models, and minimizing deployability metrics such as memory footprint, power and energy consumption, and computational cost - is out of the scope of this article. Thus, we did not optimize further the model performance and leave this as future research, supporting it through the open-source release of the dataset.

Here we stress that the DL-based assessment has been particularly useful for assessing improvements implemented in scenario generation. For instance, we achieved a key classification performance burst by incorporating in the scenario representation the EV reaction. Before that modification, the Brake scenario was often misinterpreted as Approaching LV,

with a classification rate of 43% only. By explicitly modeling the EV driver dynamic in the scenario definition, the model was able to better capture the actual dynamic, which led to a dramatic performance improvement (current classification rate: 98.75%).

Results from a classification pipeline that we built exploiting state of the art DL models, also including semantic pre-processing, indicate that the task is challenging and should stimulate research on new models, better able to capture the nuances among the patterns characterizing the different classes.

4.4.4 Comparison with Prior Work

The methodology outlined above was built upon the foundational work of Cossu [45], which already implemented synthetic generation of highway driving scenarios in CARLA based on parameter distributions derived from the MOOVE project.

That pipeline represented a reasonable balance between statistical realism and process automation, but it also presented several structural issues. Some maneuvers were recorded with excessively long time windows. For instance, the Brake scenario was often ambiguous, as the EV did not always react consistently to the LV deceleration; the Cut-in, Cut-out and EV LC classes contained trajectory segments that overlapped with simple following LV behaviors.

Several improvements and modifications have been carried out to enhance the results both of scenario generation and detection modules.

Scenario Generation

To address the scenario generation weaknesses, the first and most substantial improvement lies in the complete rewriting of the generation logic. This change allows precise control over the timing of maneuver triggering, recording duration, and synchronization between ego and leading vehicles. The Brake scenario provides a clear example: in the previous dataset, the LV decelerated, but the EV reaction was often absent, producing video clips in which the braking event appeared diluted and easily confused with Following LV classes. In the current version, the EV reaction is implemented and consistent.

The second area of revision involves lateral maneuvers, particularly Cut-in, Cut-out, and EV LC. In the original dataset, these categories suffered from contextual noise—video windows often contained frames preceding or following the actual maneuver, or even periods of steady-state driving. In highway contexts, where visual cues can be highly similar across scenarios, a few irrelevant frames are enough to blur the semantic boundary between classes. The updated generation logic solves this by refining the trigger conditions: recording

begins only when the lateral maneuver starts and ends once the LC is completed. This refinement produces two desirable outcomes. First, instances of the same class become more homogeneous, reducing intra-class variance. Second, the separation between kinematically similar classes becomes clearer, since the temporal noise that masked lateral movement is now removed.

For instance, after modeling the Brake scenario, the instance generation and simulation rates nearly doubled (from 118 to 240 instances per hour, and from 5s to 10s respectively), keeping the the same error rate ($\sim$ 22-23%), mostly due to unavoidable stochastic traffic events. In large-scale DL pipelines, especially with models requiring thousands of data per class, this improvement has a direct, measurable impact.

Table 4.12 shows in detail the same scenario generation metrics reported by [45].

Table 4.12 Previous Scenario–generation rates. In bold the scenario classes that we re-implemented and improved (cf Table 4.10).

Scenario	**Correct inst. [inst./h]**	**Error rate [%]**	**Simulation rate [min/h]**
Brake	**118**	**23**	**5**
Cut-in	**168**	**8.7**	**19**
Cut-out	**169**	**6.4**	**18**
Pulling away LV	118	28.2	18
Approaching LV	136	28.1	16
Following LV	119	33	20
Following distant LV	77	53.2	13
Free ride	111	23.4	39
EV LC	**136**	**37.5**	**20**

Scenario Detection

Several differences occur also on the detection side, leading to notable improvements of the previous work [45]. Both ours and the approach proposed by Cossu employ the same DL network to detect scenarios. Fig. 4.16 shows the results obtained by the best model in [45], while Table 4.13 shows the aggregate metrics.

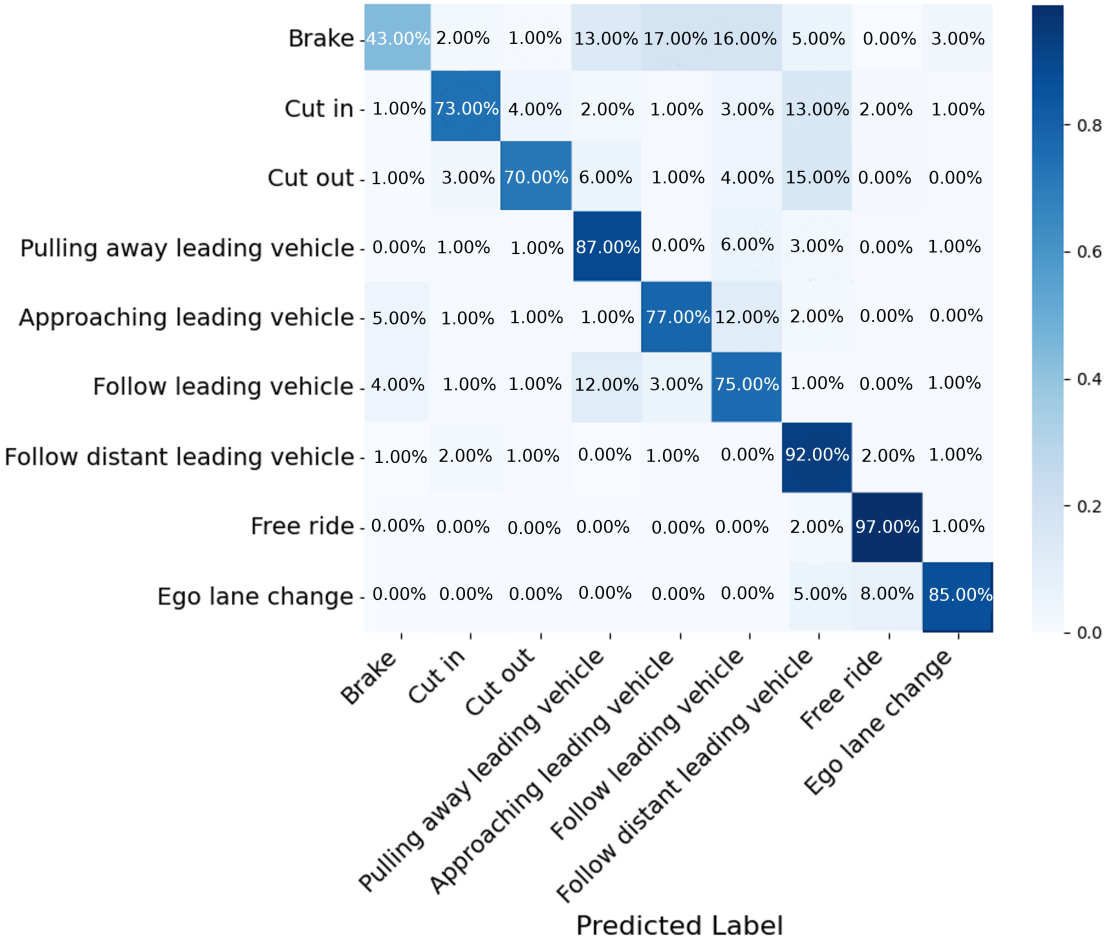


Fig. 4.16 Previous scenario classification results. Comparing with Fig. 4.14, the differences in terms of accuracy per class are striking.

Table 4.13 Previous aggregate classification metrics for each model. Best results in bold (cf. Table 4.11).

Model	Accuracy	Precision	Recall	F1 score
R3D	83.3%	84.5%	83.31%	83.2%
R3D CBAM	86.2%	86.7%	86.2%	86.2%
R3D SE	**88.9%**	**89.0%**	**88.9%**	**88.7%**

By comparing the confusion matrices in Fig. 4.14 and 4.16, it is possible to note that our new scenario implementations have drastically improved results in detection. For example, implementing the EV reactions has introduced visual cues, allowing the models to clearly differentiate Brake from Following LV, which was not feasible in [45]. Furthermore, the

revised Cut-in/out and EV LC increase detection accuracy, as they only capture lateral movement without longitudinal driving sessions that may be misclassified as Following LV-like classes, (Fig. 4.16).

Specifically, accuracy has increased by 55.75% for Brake, 19.48 for Cut-in, 19.71% for Cut-out, and 13.08% for EV LC. As a consequence, misclassifications have decreased drastically (cf Fig. 4.14).

4.4.5 Real-World Zero-Shot Learning

We finally assessed the real-world transfer learning ability of the system as a validation of our simulation-based approach. We performed zero-shot testing evaluating performance of the trained R3D SE system directly on a small random selection of excerpts taken from all the records (1-5) of the Prevention dataset [74], with no fine-tuning. As per our ODD, the selections involve 2/3 lane highway sections, with no lateral merges. We manually annotated such excerpts, amounting to a total of 3 minutes and 59 seconds, corresponding to 2,390 samples. The resulting scenario distribution (Table 4.14) is quite different from the synthetic test set extracted from the MOOVE distributions (Table 4.8), representing an additional challenge. Detection results are reported in Table 4.15 and Fig. 4.17. The R3D-SE model exhibits a performance degradation compared to the synthetic test set (Table IX), Overall accuracy and F1-score decrease from 0.919 to 0.828 and 0.848, respectively – a relative drop of approximately 10%.

This decline is primarily attributable to the domain shift between the controlled virtual environment and the natural variability of real-world footage. Prevention introduces two main challenging conditions not present in the synthetic data: uncontrolled illumination and geometric instabilities. On the one hand, camera glare creates saturated regions and bright streaks that can resemble lane markings. Abrupt intensity changes occur in shadows cast by bridges or during high-contrast transitions at tunnel entrances and exits. On the other, road-surface irregularities and vehicle suspension oscillations induce low-frequency viewpoint jitter, disrupting temporal coherence across frames. Although CARLA simulator achieves high photorealism in weather, lighting, and material rendering, such second-order physical artifacts are only partially represented, likely explaining the performance drop.

Table 4.14 PREVENTION test set driving scenario distribution.

Scenario	# samples	% samples
Brake	4	0.14
Cut-in	553	18.75
Cut-out	87	2.95
Pulling away from LV	175	5.93
Approaching LV	12	0.41
Following LV	1901	64.46
Following distant LV	74	2.51
Free ride	16	0.54
EV LC	127	4.31

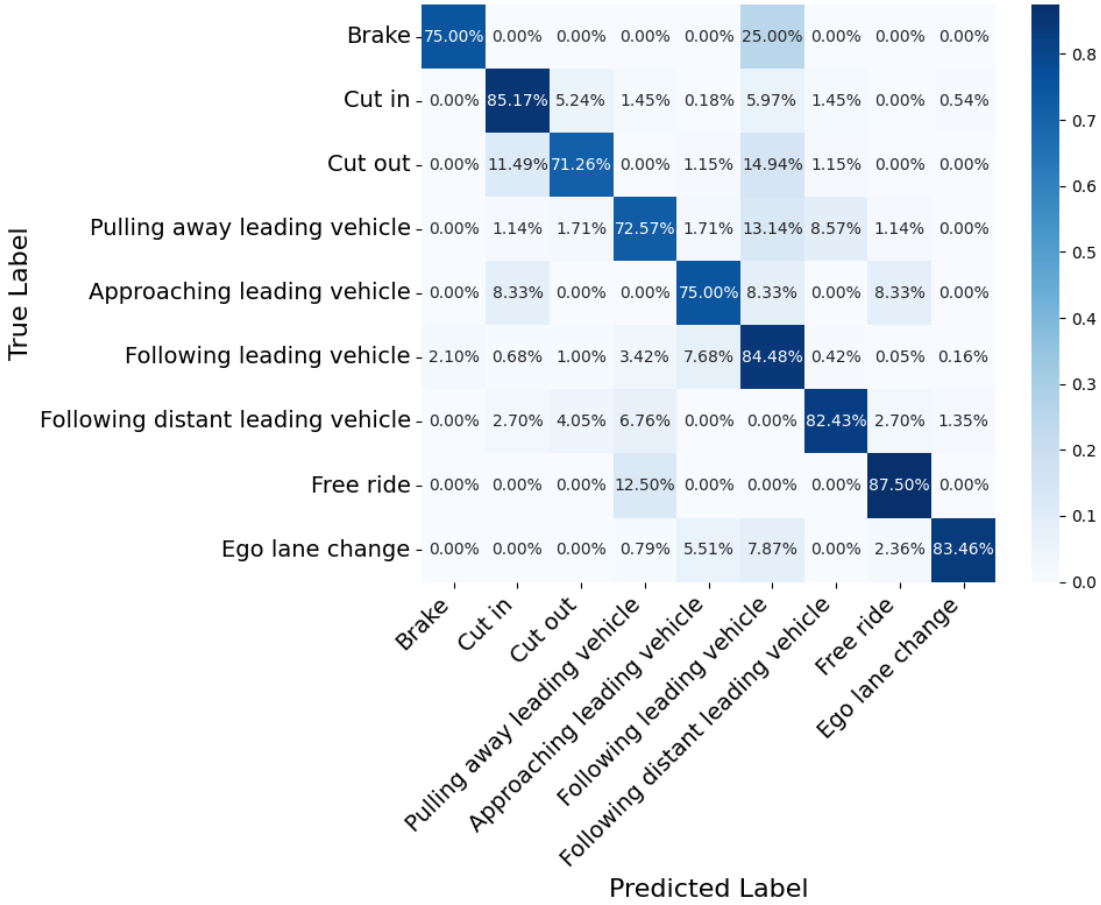


Fig. 4.17 Confusion matrix on the PREVENTION real-world subset.

Table 4.15 Aggregate classification metrics on the PREVENTION subset.

Accuracy	Precision	Recall	F1 Score
83.3%	89.0%	83.0%	85.0%

Despite this domain shift, the model demonstrates a robust understanding of structural scenario cues, as indicated by its maintained precision (0.883) and the preservation of dominant inter-class relationships (Fig. 4.14 vs Fig. 4.17). Misclassifications are mostly confined to semantically related categories (e.g., Brake, Approaching, Following), confirming that the learned features capture underlying motion logic while remaining sensitive to visual distortions and real-world noise.

Notably, although the PREVENTION dataset was acquired with a multi-sensor setup, only the forward-facing camera stream was used in this evaluation, consistently with our monocular pipeline. This alignment ensures that the observed performance degradation is attributable to the sim-to-real domain gap rather than to sensor modality differences.

These preliminary results underscore the value of the synthetic dataset for developing a pre-trained model, which shows significant promise for fine-tuning with limited real-world data. We thus argue that the proposed synthetic framework provides a scalable and cost-effective pre-training source that effectively captures core scene dynamics and facilitates subsequent domain adaptation.

4.4.6 Conclusion

This chapter presented a unified study of LC intention recognition and highway scenario detection on a shared map, using complementary data modalities and consistent evaluation. On the maneuver side, TTLC regression on human-in-the-loop time series anticipates LC up to 4 s, with a clear horizon-dependent behavior: below $\sim$2.5 s, kinematics dominate; beyond that, scene context becomes more relevant. This supports a practical design pairing a lightweight intention detector for long horizons with a focused regressor for short ones, balancing accuracy and computational footprint (1D-CNN vs. Transformer).

On the scenario side, a configurable generator aligned with OpenSCENARIO 2.0 and grounded in MOOVE distributions was introduced, producing labeled MP4 clips over nine highway classes on the same map used for LC. A reference R3D-SE model with semantic preprocessing reached 93% accuracy in simulation, with confusions mainly among visually similar following patterns. The generated data were further validated on a real-world subset from the PREVENTION dataset, where the same model achieved 83.3% accuracy, 89%

precision, and 85% F1-score without any fine-tuning. Despite the strong class imbalance inherent to natural driving data, the model maintained consistent performance across dominant scenarios, confirming the robustness and transferability of the synthetic-to-real pipeline.

The open-world limitation identified by Balasubramanian et al. [**?**], which involves the recognition of genuinely novel scenarios at deployment, remains an open direction for future work.

The next chapter extends this work from simulated and real-world validation to embedded deployment of a safety-critical perception function under sensor degradations, completing the loop from data design to on-device operation.

Chapter 5

Robust Traffic Light Detection under Sensor Failures

5.1 Introduction

This final chapter turns the data-centric insights of this thesis into a real deployment problem: robust TLD under sensor anomalies. Building on the lessons of the previous chapters, we ask whether that accumulated know-how (i.e., data design, simulator realism, scenario coverage, and embedded constraints) can be composed into a system that works in practice. Concretely, we reuse and adapt those pieces to design, implement, and assess a TLD pipeline that remains reliable when the camera stream is degraded.

Reliability matters because DL perception hinges on sensor quality: corrupted inputs can cascade into unsafe behavior [137, 204, 205]. TLD is central for SAE L3/4 operation [206] and must integrate cleanly with TOR policies that handle conditions outside the ODD [207]. Rather than revisiting broad adversarial robustness, we focus on realistic degradations and hardware-like failures that emerge in deployment [42, 43] and are known to impair vision systems [208, 209, 141].

Our contribution is a modular embedded architecture that (i) detects anomaly-affected frames, (ii) restores images when possible, (iii) filters unrecoverable inputs, and (iv) triggers a TOR if reliability falls below a threshold—then monitors for recovery. We study a concrete use case: an RGB-camera TLD stack built around a quantized (16-bit) YOLOv11 detector [40, 41] on an NVIDIA Jetson Orin Nano, evaluated with two black-box, deployment-plausible perturbations, TPSeNCE (rain) [42] and Poltergeist (stabilizer-like blur) [43], that require no prior system knowledge [140]. The design choices reflect embedded constraints on compute,

power, and latency typical of in-vehicle platforms [210] and aim at a practical path to robust operation.

Section 5.1.1 describes the datasets, Section 5.2 details the architecture, Section 5.3 reports results, and Section 5.4 outlines takeaways and future directions.

5.1.1 Dataset Selection and Post-Processing

Based on Table 2.6, we chose Bosch, DTLD, and Valeo for developing our system. The Bosch dataset ensures variety across cities and a substantial number of annotations. DTLD offers very high-quality annotations and includes the red-yellow state, which is essential for cross-regional robustness. The LISA dataset was excluded, aligning with [153], due to missing annotations, imprecise bounding boxes, and the absence of the off label.

Since the datasets cover different states with different labels, we mapped them into five unified classes: off, green, red, yellow, and red-yellow.

A post-processing step consisted of injecting perturbations into both training and testing sets for the FCh and the test set of the complete system. The dataset was divided into three splits: one perturbed with rain noise through TPSeNCE [42], one with blur noise through Poltergeist [43], and one left clean. Poltergeist perturbations were injected at three intensity levels: weak, medium, and strong. Parameters were set following [43]: weak $\delta = 0.0128, L = 0$; medium $\delta = 0.0256, L = 5$; strong $\delta = 0.05, L = 10$, with $\beta = 0.1$ for rotation. For TPSeNCE, intensity levels could not be varied due to the lack of annotated datasets with controlled rain conditions.

5.1.2 Noise Selection Rationale and Scope

TPSeNCE and Poltergeist were selected because they represent two structurally distinct classes of camera anomaly relevant to automotive deployment. TPSeNCE models adverse weather conditions, specifically rain, which degrades image quality through light scattering, streak artifacts, and contrast reduction, a naturally occurring and well-documented source of perception degradation in outdoor driving scenarios [138, 141]. Poltergeist instead models a hardware-level failure: it exploits acoustic resonance to induce unintended motion compensation in the camera stabilizer, producing blur artifacts that are independent of weather and can occur under any lighting or environmental condition [43]. Together, the two perturbations cover one environmental and one hardware/cyber threat vector, providing complementary coverage of the anomaly space without requiring prior knowledge of the target system. Both are black-box, reproducible, and parametrizable at controlled intensity levels, which enables systematic evaluation across a defined severity range.

The use of synthetic perturbations rather than real-world noisy data is a deliberate methodological choice. No public dataset provides TLD images with controlled, annotated camera anomalies at multiple severity levels. Synthetic injection via TPSeNCE and Poltergeist is therefore the only approach that guarantees reproducibility, parametric control over intensity, and ground-truth knowledge of the anomaly type.

The architecture's robustness is intentionally scoped to the two selected perturbation classes. A FCh trained on three classes (clean, TPSeNCE, Poltergeist) will not generalise to unseen anomalies such as low light, lens flare, or fog without retraining. This is a limitation. However, the modular design of the architecture directly addresses it: extending coverage to a new anomaly type requires only retraining the FCh with additional samples and integrating a corresponding cleaner module, without modifying the core detector. This extensibility is a deliberate architectural property, not a workaround.

5.2 System Architecture

The proposed system architecture implements a comprehensive reliability workflow designed to maintain operational integrity during sensor failures. At its core, the framework integrates four principal components working in concert. The frame checker (FCh) module is responsible for real-time detection of sensor perturbations, while the frame cleaner (FCl) attempts to mitigate corrupted input data. The system incorporates two critical DM modules: the takeover decision maker (TODM), which evaluates when driver intervention becomes necessary and issues a TOR, and the AD Revert Decision Maker (ADRDM), which determines when AD operation can safely resume. Central to this architecture is the YOLOv11-based TLD system, which serves as the foundational perception component.

The workflow comprises two execution modes: an AD mode and a manual driving mode, which is activated when the ADF is disabled. The system consists of five modules: a Frame Checker for classifying the sensor failure (no anomaly, TPSeNCE, or Poltergeist); a Frame Cleaner for mitigating image corruption; a Traffic Light Detector, a decision-maker to trigger a TOR in case of persisting perturbations; and a second decision-maker responsible for re-activating the ADF once the checker verifies the absence of anomalies.

5.2.1 Target Workflow

The proposed workflow (Fig. 5.1) incorporates a comprehensive safety protocol that includes both TOR initiation and, when conditions permit, ADF resumption. This dual-mode operation mirrors the fundamental safety paradigm of autonomous systems, where TORs

are triggered either when the ODD boundaries are violated [211] or in response to system malfunctions [212].

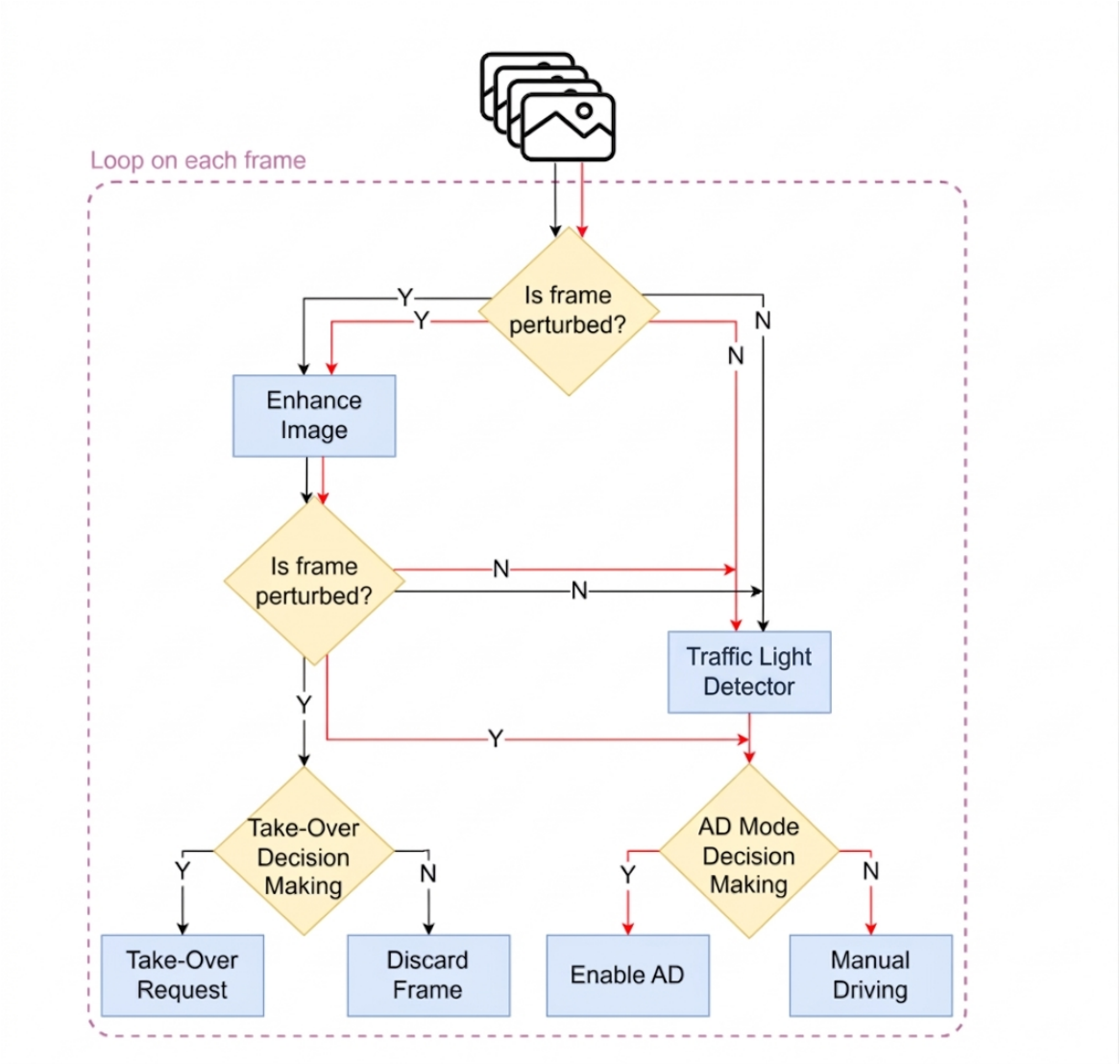


Fig. 5.1 Proposed TLD workflow. Black arrows indicate AD mode execution, red arrows indicate the manual driving mode.

A fundamental design decision involves implementing a modular architecture with dedicated subsystems for object detection and anomaly detection. While previous studies have explored noise-robust neural networks [136, 213], our partitioned solution offers distinct advantages. In our framework, the object detector is trained exclusively on clean (non-corrupted) images, while system robustness is achieved through two specialized components: (1) an anomaly detection module and (2) anomaly-specific frame-enhancement modules. This separation provides scalability and extensibility, as adapting to new anomaly types

only requires updating the anomaly detection component and adding a corresponding frame-enhancement module, without modifying the core TLD system.

5.2.2 Frame Checker

The FCh is a classifier that processes the TLD input and outputs the estimated failure type or "clean" if no failure is detected. It is trained using the TLD dataset with perturbations injected as described in Section 5.1.1. The dual training workflow (object detection and anomaly detection) is sketched in Fig. 5.2.

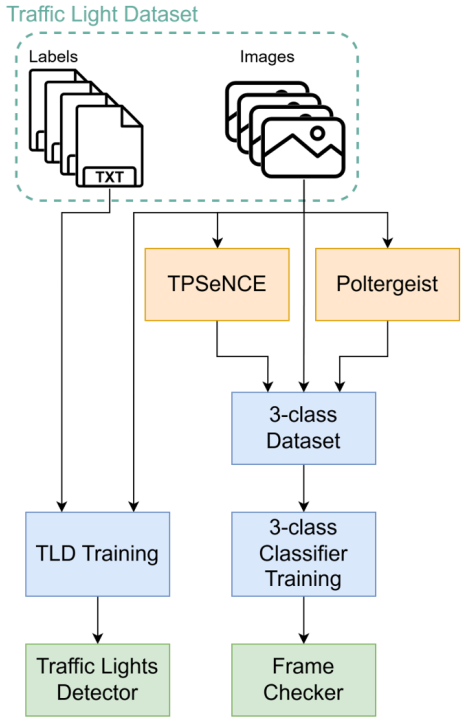


Fig. 5.2 Dual training workflow for traffic lights and anomaly detectors.

As the basis for the FCh, we use MobileNetV2 [214] pre-trained on the ImageNet dataset. MobileNetV2 is a model optimized for mobile deployment, thus suited also for an automotive environment in terms of memory footprint and computational demand. We fine-tune this model on a 3-class task ('clean', 'tpsence', and 'poltergeist'). We classify as 'tpsence' the rainy images and as 'poltergeist' the samples affected by the relevant adversarial attack or camera's stabilizer malfunction. The MobileNetV2 classifier head is thus replaced by an output layer consisting of a 3 fully-connected neurons with softmax activation. In addition, to enhance the model's ability to capture task-specific patterns, the four deepest layers of the MobileNetV2 are unfrozen for fine-tuning, while the rest of the layers remain frozen to

preserve the robust feature extraction capabilities learned on ImageNet, and favor training convergence. These include the input layer size, which has been kept at $224\times224\times3$.

For the training phase, we employed the ADAM optimizer over the Cross-Entropy loss, with a weight decay of 10^{-5} and learning rate 0.001. We trained the classifier for 12 epochs, using Early Stopping with patience 5 and batch size 256.

5.2.3 Frame Cleaner

Whenever a frame is classified as perturbed, the FCl module is in charge of enhancing it. We decided to apply two different image recovery techniques given the significant difference between the two types of sensor failures considered.

To restore TPSeNCE-perturbed frames, we employed SAPNet [145], a Segmentation-Aware Progressive Network which utilizes progressive dilated units to clear multi-scale rain streaks while preserving semantic details via unsupervised background segmentation.

To restore images generated by Poltergeist, we opted for a more complex method, as per the more sophisticated nature of the perturbation, which involves both linear and rotational motion blurs. So, we used a specific GAN (i.e., DeblurGAN-v2) purposely designed for image deblurring [215]. Also in this case, we employ a MobileNetV2 backbone [214], keeping into account the limitations of the target execution environment.

The TLD module is run only on images that are deemed as clean by the FCh. Thus, frames that cannot be effectively cleaned (i.e., are still classified as anomalies after the mitigation attempt) are discarded.

5.2.4 Takeover Decision Maker

The TODM is in charge of deciding whether to execute or not a TOR if an attack/sensor failure is detected. For our baseline implementation, we employ a sliding-window detection approach where we:

1. Count the number of anomalous frames (A) within a window of N consecutive frames.

2. Normalize this value to obtain the anomaly percentage:

$$\tilde{A} = \frac{A}{N} \cdot 100 \tag{5.1}$$

3. Compare against a threshold A_{thres} using the decision function:

$$D(\tilde{A}) = \begin{cases} 1 & \text{if } \tilde{A} > A_{\text{thres}} \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

where $D(\tilde{A}) = 1$ triggers a TOR, while $D(\tilde{A}) = 0$ maintains normal operation. The most conservative $\tilde{A} = 1/N$ threshold initiates a TOR upon detection of even a single anomalous frame, prioritizing maximum safety at the potential cost of increased false positives.

Alternative approaches to change point detection, including statistical methods based cumulative sum control chart [216] or pruned exact linear time [217], or AI-based solutions such as autoencoders and LSTM-based anomaly detectors, could offer improved sensitivity and adaptability. However, for the purposes of this work, the sliding-window approach was selected for its simplicity, low computational overhead, and ease of interpretability, leaving the exploration of more advanced methods to future investigations.

5.2.5 Automated Driving Revert Decision Maker

Once the TOR has been issued, the driver takes control of the vehicle, and the system toggles its working mode. However, the ADF can be resumed if the system judges that the sensor anomaly issue has been overcome. This decision is delegated to the ADRDM. For this module, we propose a dual logic of the TODM (Eq. 5.1 and Eq. 5.2), with a different TW length N and decision threshold $\tilde{A}$. A discussion on the setting of such parameters is given in the experimental section of this article.

5.2.6 Traffic Light Detector

At the core of our system, we have chosen the YOLOv11 model [40, 41] as the solution to detect traffic lights at various scales and in real time.

As the YOLOv11 architecture comes in different sizes (from nano to extra-large), we selected the nano size, to be able to deploy the trained model on an embedded computing platform with reasonable inference times. The nano architecture involves 2.6M parameters and 6.5B FLOPs. We trained the model for 1,000 epochs, with batch size 32, and enabled early stopping with patience of 50 epochs.

The dataset was partitioned into three sets: 80% for training, 10% for validation, and 10% for testing. To properly assess generalization, the dataset was split by cities. The `imgsz` parameter was set to 640×480 pixels, which was chosen as a trade-off between computational efficiency and detectable object size. We disabled the `rect` setting, to ensure the mosaic data

augmentation, which is used to create composite images from multiple inputs to enhance model generalization. The SGD optimizer was employed, which is the default for YOLOv11.

5.2.7 Overall System Architecture

Fig. 5.3 provides a high-level overview of the in-vehicle deployed system, highlighting the two different impact points of the considered perturbations, the computing devices, and the final UI.

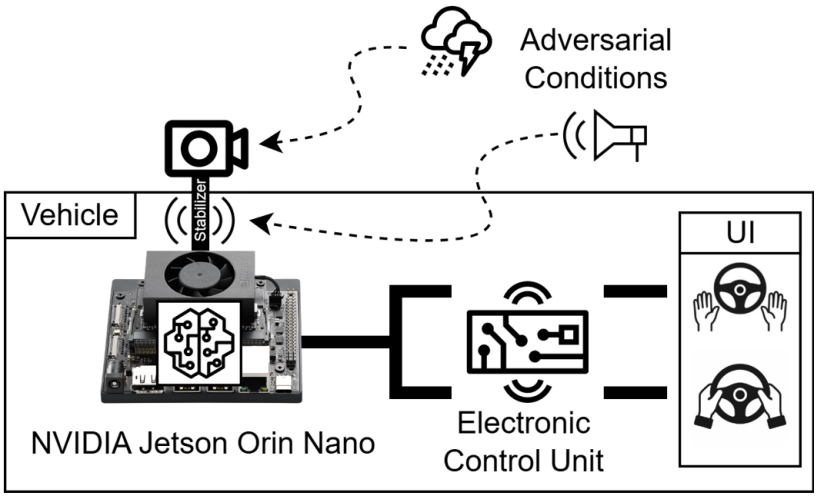


Fig. 5.3 High-level overview of the system architecture

As the core hardware platform, we selected the NVIDIA Jetson Orin Nano [218], which is well suited for DL-centric automotive applications because of its compact dimensions (100 mm $\times$ 80 mm $\times$ 29 mm), robust computing capabilities, and powerful GPU. This platform features a quad-core ARM Cortex-A57 MPCore Processor CPU and a 128-CUDA core NVIDIA Maxwell GPU running at 1.43 GHz. It also features a CPU-GPU shared memory of 4 GB of 64-bit LPDDR4 RAM and is equipped with 16 GB of eMMC 5.1 storage.

Fig. 5.4 details the overall workflow supported by the proposed system architecture.

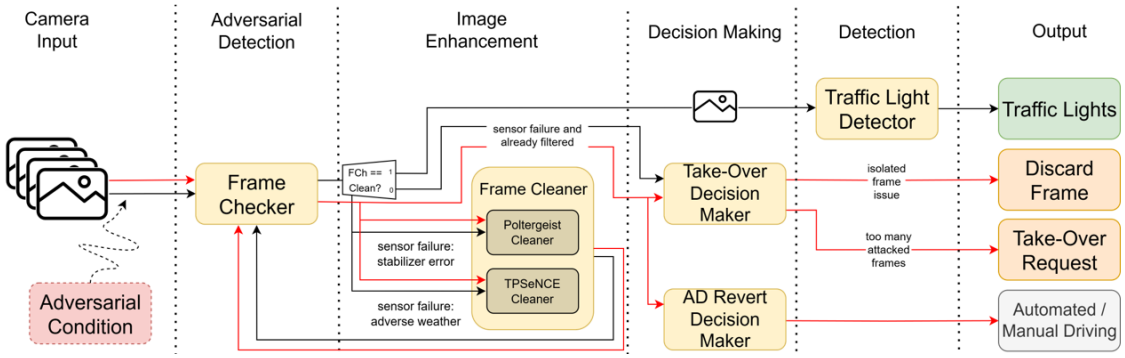


Fig. 5.4 Functional workflow supported by the proposed system architecture. Black arrows indicate execution in AD mode, while red arrows indicate execution in manual driving mode.

The system receives as input a stream of frames. Each frame is first verified by the FCh. If it is classified as non-perturbed, it is processed by YOLOv11 to detect traffic lights. Otherwise, the frame is enhanced with the FCl and fed again to the FCh. If it is again classified as anomalous, it is discarded, and the TODM decides whether to give the control back to the driver through a TOR. In this case, the system execution continues (dashed red arrows) to check if other attacks are coming. The decision on whether to re-enable the AD mode rests with the ADRDM. If the second check is passed, the cleaned frame is instead sent to YOLOv11.

The system's modular architecture ensures independent operation of each component, providing two key advantages: (i) generalization capability beyond TLD applications, and (ii) straightforward updatability of individual modules (e.g., seamless integration of improved object detection algorithms as they become available).

5.3 Experimental Results

5.3.1 YOLO11 TLD Performance on Clean Dataset

The YOLO11 TLD model trained and tested on the original (i.e., unperturbed) dataset described in Sec. 5.1.1 achieves a precision of 86.4%, a recall of 77.7%, and an F1-score of 81.8% on the test set. We also report a mAP50 of 84.7%. We report mAP50 rather than mAP50-95 because traffic light bounding boxes are small and tightly annotated across datasets; mAP50-95 penalizes localization error more heavily, while mAP50 intuitively reflects overall performance especially in small object detection scenarios [219]. We argue that performance is negatively influenced by the class imbalance present across the three datasets used. Specifically, the majority of samples belong to the 'green' and 'red' classes,

while the 'yellow' class contains an order of magnitude fewer samples. The 'red-yellow' and 'off' classes are severely under-represented (two orders of magnitude fewer samples than the 'green' and 'red' classes) and less accurately detected. The class imbalance directly impacts model performance, as evidenced by the lower mAP50 scores, which are averaged per class. mAP50 values are 92.9%, 87.6%, and 87.5% for the 'green', 'red', and 'yellow' classes, respectively. But values for 'red-yellow' and 'off' are 84.3% and 71.2%, respectively.

To deploy the model on the Jetson Orin Nano target platform, we exported it to the TensorRT format [220] with the FP16 quantization (floating point half precision), which is the maximum quantization supported by the board. The quantized model achieves a similar performance as the original one, with a precision of 86.1%, a recall of 77.6%, an F1-score of 81.6%, and a mAP50 of 84.6%. While the preservation of model performance under quantization has been previously established in the literature (e.g., [135]), our work provides the first empirical validation of this effect specifically for TLD tasks. The quantized model's inference time is 59 ms and size 9.5 MB.

Polley et al. [134] present a TLD application based on four datasets, each one with a different set of target classes. For each dataset, they report performance for the three-class problem (green, yellow, and red). Thus, for the sake of comparison, we performed a similar analysis considering the three-class problem and the BSTLD and DTLD datasets separately. Results in Table 5.1 show that our FP16-quantized model far outperforms their full-precision model. A broader comparison covering additional metrics (precision, recall, inference time, model size) is reported in Table 5.2 and Table 5.4 for our full five-class system, which operates under more challenging conditions than the three-class baselines.

Table 5.1 mAP50 for the 3-class task on BSTLD and DTLD datasets

Model	**BSTLD**	**DTLD**
YOLOv7 [134]	83%	74%
YOLO11n (ours)	**93%**	**95%**

5.3.2 YOLO11 Robustness Analysis

This section analyzes the performance of the YOLO11 model in scenarios with TPSeNCE and Poltergeist camera sensor anomalies, using the perturbed dataset presented in Sec. 5.1.1. Table 5.2 (YOLO11 column group) shows the performance of the YOLO11 at the three perturbation intensity levels.

Table 5.2 Performance against frame perturbations

Perturbation	Intensity	YOLO11				Re-trained YOLO11				Proposed System			
		Prec	Rec	F1	mAP50	Prec	Rec	F1	mAP50	Prec	Rec	F1	mAP50
No	-	86.1%	77.6%	81.6%	84.6%	82.6%	76.9%	79.6%	80.5%	86.1%	77.6%	81.6%	84.6%
TPεNCE [42]	-	80.2%	75.1%	77.6%	79.0%	81.8%	75.5%	78.5%	80.8%	86.2%	77.6%	81.7%	84.6%
Poltergeist [43]	weak	88.2%	75.3%	81.2%	84.3%	85.5%	78.8%	82.0%	84.1%	86.0%	77.5%	81.5%	84.4%
	medium	78.8%	75.0%	76.9%	77.7%	79.1%	74.6%	76.8%	78.2%	85.9%	77.5%	81.5%	84.4%
	strong	69.0%	42.1%	52.3%	49.4%	76.3%	55.9%	64.5%	66.3%	85.9%	77.9%	81.7%	84.8%

Results show that both TPSeNCE and Poltergeist significantly affect performance, under every considered metric. With TPSeNCE, mAP50 drops from 84.6% (original) to 79.0% (strong perturbation), and degradation is more pronounced for Poltergeist (49.4% with strong perturbation). Fig. 5.5 gives an example of the impact on YOLO11 inference, showing TLD detections on the original frame, with TPSeNCE noise and three levels of Poltergeist.

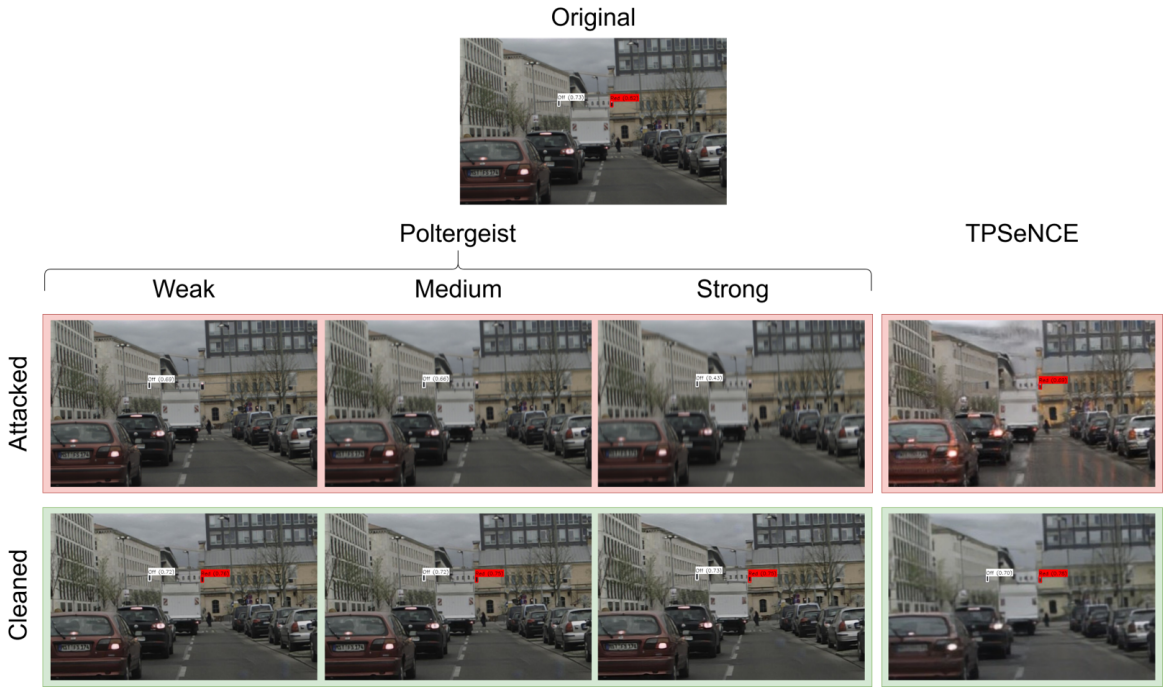


Fig. 5.5 YOLO11 performance visualized. Inference of TLD model before (in red) and after (in green) FCl intervention for both cases.

The different performance in the two anomaly types indicates that YOLO11 is quite robust to adverse weather (TPSeNCE) perturbations, but very weak to camera malfunctions (Poltergeist), in the TLD task. We thus tried to re-train the YOLO11 detector with a dataset augmented with TPSeNCE- and Poltergeist-perturbed frames at different intensity levels. Results (Table 5.2, Re-trained YOLO11 vs YOLO11 column groups) show that mAP50

increases for perturbed images (particularly, strong Poltergeist 66.3% vs 49.4%) but falls in the more common, clean case (80.5% vs 84.6%).

These findings - representing, to our knowledge, the first comprehensive perturbation analysis of YOLOv11 - have motivated our development of a novel system architecture specifically designed for robust real-world deployment.

The proposed system architecture emphasizes modularity and extensibility to accommodate new perturbation types. To enhance the robustness against novel noise patterns, the framework requires only: (i) retraining the FCh with additional perturbed samples, and (ii) integrating a new dedicated cleaner module. Notably, this approach preserves the original YOLO detector, which remains optimized for processing clean images - the predominant case in normal operation.

5.3.3 Frame Checker

The FCh is intended to identify perturbed frames. In our implementation, it recognizes three classes: ‘clean’, ‘tpsence’, and ‘poltergeist’. The training configuration reported in Sec. 5.2.2 achieves 99.8% accuracy, 99.6% precision, 99.6% recall, and 99.6% F1-score. As per our target workflow , frames flagged as perturbed are submitted to the FCl for a cleaning attempt and then processed again by the FCh, to check the outcome of the attempt.

A common issue for deployment concerns the false alarm rate, which may undermine the user trust in a system causing stress or distraction. In this view, it is important to report that the FCh flags 0% of clean frames as perturbed. On the other hand, there is 0.27% of false negatives (0.25% due to TPSeNCE-perturbed images misclassified as ‘clean’, 0.02% to weak Poltergeist).

5.3.4 Frame Cleaner

Table 5.3 (column group: Final version) illustrates the recovery rate (RR) achieved by the proposed system, segmented by perturbation type and intensity.

Table 5.3 FCH's RR and system's mAP50 per attack type.

Attack Type	**Intensity**	**Early version**		**Final version**	
		RR	**mAP50**	**RR**	**mAP50**
TPεNCE	-	90.7%	84.6%	90.7%	84.6%
Poltergeist	weak	99.8%	84.3%	99.3%	84.4%
	medium	59.7%	84.2%	56.6%	84.4%
	strong	96.9%	52.8%	29.9%	84.8%

To calculate such values, we process each attacked test-set image through the whole system. RR is as follows Eq. 5.3:

$$RR = \frac{CC2}{PC1} \tag{5.3}$$

where CC2 is the number of images that are classified as clean by the FCh after the FCl processing, and PC1 is the total number of images sent to the FCl (i.e., the input images flagged as perturbed by FCh).

The RR per se is not a performance indicator for the overall system. Results from an early version of the system (Table 5.3, Early version columns) show an average RR of 86.8% and a final mAP50 of 76.5%, with a performance drop of about 8% with respect to the unperturbed case (Table 5.2, YOLO11). In this early version, we trained the FCh with clean and perturbed images only.

In this work, mAP50 is adopted as the sole quality criterion for the FCl output. This choice reflects the perspective of the downstream detector rather than human visual perception: what matters for the system is whether the restored frame allows YOLOv11 to detect traffic lights correctly, not whether the frame looks clean to a human observer. This distinction is supported by recent work showing that conventional IQA metrics such as SSIM and PSNR, being designed to align with human perceptual judgements, are weak predictors of DNN performance in detection and classification tasks [221, 222].

We acknowledge that this choice has a limitation: mAP50 measures the aggregate effect of image quality on detection, but does not provide insight into which aspects of the restored image (sharpness, contrast, artefact level) drive the recovery. A complementary analysis correlating standard or task-oriented IQA scores with mAP50 on the restored frames would help characterise the FCl behaviour more completely, and is left as future work.

To improve the overall system performance, we then extended the FCh dataset with samples representative of its second iteration, particularly from strong Poltergeist, which had

high RR (96.9%) but low mAP (52.8%). Such additional samples are perturbed images that underwent the FCl but were not cleaned well enough (i.e., achieved a mAP50 value lower than 85%). This fine tuning led to a reduction in RR (69.1%), but a surge in final mAP50 (84.6%) (Table 5.3, Final version column). The implemented system demonstrates enhanced capability to identify critical scenarios, with the TODM dynamically evaluating whether to trigger a TOR based on anomaly frequency. Our approach maintains the original clean-image detection accuracy (84.6 mAP50, Table III), though with a measurable frame rate reduction due to the conservative frame-discarding policy.

As shown in the last group of columns of Table 5.2, the defended YOLO11 maintains consistent performance across all tested perturbation types and intensity levels, sustaining nearly 85% mAP50. This represents a significant improvement over the baseline unprotected YOLO11 implementation (which degraded up to 49.4% mAP50 under same conditions), validating the effectiveness of the proposed defense architecture.

5.3.5 Resource Analysis

We evaluated the computational performance of the proposed architecture on the Jetson Orin Nano platform in terms of both inference time and memory footprint. The software stack is built on Python 3.8, PyTorch 2.0, and the Ultralytics library for YOLO. Inference time was computed within the python code.

Table 5.4 presents the breakdown of the inference times for the various components of the system using the Jetson Orin Nano platform.

Table 5.4 System architecture components' inference time.

Component	**Inference Time [ms]**	
	TPSeNCE	**Poltergeist**
FCh	16	16
FCI	27	28
YOLO11 Detector	23	23
Total – perturbation filtered out	82	83
Total – perturbation not filtered	59	60
Total – no perturbation detected	39	39

The YOLOv11 detector has a latency of 23 ms. The overhead introduced by the FCh is 16 ms. The normal (i.e., no-attack detected) system latency is thus 39 ms. The overhead posed by the FCl is 27 ms in the case of TPSeNCE perturbation, and 28 ms in Poltergeist scenarios.

When a perturbed frame is detected, another FCh run is needed, leading to a total latency of 83 ms (82 ms for TPSeNCE). This is the maximum latency. The system demonstrates three distinct operational modes with varying frame rates:

1. Attack mitigation mode: 12 FPS (during active perturbation filtering)
2. Normal operation mode: 25 FPS (attack-free conditions)
3. Frame discard mode: 16 FPS (when perturbations cannot be filtered)

For context, at a vehicle speed of 50 km/h – a common urban speed limit with traffic lights – under the worst-case latency (12 FPS), the system processes frames approximately every 1 meter of travel distance. While our datasets lack precise distance metadata, manual inspection reveals that labeled traffic lights typically appear at 70–80 meter distances in sample images. This detection range provides margin for the system's operational latency, even in worst-case scenarios.

Table 5.5 reports the parameter counts and VRAM consumption of the system architecture's core components. Hardware resource analysis was conducted using jtop, allowing to monitor GPU and CPU utilization.

Table 5.5 Hardware resources usage.

Component	**# Parameters**		**Memory [MB]**	
	TPSeNCE	**Poltergeist**	**TPSeNCE**	**Poltergeist**
YOLO11 Detector	2.6M	2.6M	60	60
FCh	2.2M	2.2M	10	10
FCI	0.1M	4.3M	62	65

The YOLOv11 detector requires 2.6M parameters and ∼60 MB, while the FCl adds 62 MB for TPSeNCE and 65 MB for Poltergeist. The FCh remains lightweight, with 2.2M parameters and 10 MB, also thanks to the reduced input size ($224\times224\times3$ vs $640\times480\times3$). Thus, when the system processes non-perturbed frames, the peak memory usage is 60 MB; when perturbed frames require cleaning, the peak rises to 62 MB for TPSeNCE and 65 MB for Poltergeist.

Overall, the memory footprint of the system remains well within the 4 GB VRAM capacity of the Jetson Orin Nano, confirming the feasibility of real-time deployment on embedded automotive hardware.

5.3.6 Decision Maker (TODM and ADRDM) settings

To design the TODM, we first determined an optimal window length N for anomaly monitoring. Given a frame rate of 12 FPS (the operational rate when the TLD system fails to filter perturbations), a 5-frame window spans approximately 0.4 seconds, corresponding to a traversed distance of 5.6 meters at a velocity of 50 km/h. This window duration strikes a balance between rapid detection of sustained perception degradation and sufficient smoothing of transient noise, which may intermittently arise under normal operating conditions—particularly in urban environments.

Within the 5-frame window, we evaluated different threshold values A_{thres} of the maximum acceptable number of discarded frames before issuing a TOR. The goal is to keep the attack-less mAP50 value ($\geq$84%), while accepting a FPS degradation from 25 FPS until a real-time performance threshold ($\geq$10 FPS [223]). FPS reduction may be due to two factors: the increased latency due to the cleaning attempt (the worst-case execution time occurs when an attack is cleaned) and the loss of discarded unrecoverable frames.

With $A_{\text{thres}} = 0\%$ (where a single discarded frame triggers a TOR), the worst-case latency is $\sim$0.06 s. Combined with a distracted driver's average reaction time of 2.5 s [207], this yields a total takeover time of 2.56 s, corresponding to $\sim$35 m at 50 km/h. While this setting preserves a minimum frame rate of 12 FPS, its high sensitivity to isolated disturbances (e.g., minor vibrations from uneven pavement) increases false positives.

Increasing A_{thres} to 40% (requiring $\geq$3 perturbed frames in the window) delays TOR issuance to a maximum of 0.18 s (3 consecutive frames), extending the total response time to $\sim$2.68 s ($\sim$37 m). This threshold risks sub-real-time performance: if two frames are discarded and the remaining three are filtered by the FCl, no TOR is issued, but the frame rate drops to 8 FPS.

To balance reliability and responsiveness, we selected $A_{\text{thres}} = 20\%$, requiring $\geq$2 perturbed frames per window. This results in:

- Maximum TOR latency: 0.12 s.
- Total takeover time of $\sim$ 2.62 s ($\sim$36 m).
- Worst-case FPS: 10 (one frame is discarded and four are filtered), still meeting real-time requirements [223].

When the TLD system is used by a highly vigilant driver (reaction time 0.7 s [223]), the total latency drops to 0.82 s, with the vehicle traveling only $\sim$11 m before control is regained.

Differently from the TODM, the ADRDM module—responsible for resuming AD—requires a longer-term assessment. Specifically, we decide to re-enable AD only after a 25 s

interval without any detected perturbation (i.e., $N = 300$ and $A_{\text{thres}} = 0\%$). This conservative design strategy ensures that automated driving functionality is restored only after sustained perception stability is confirmed, thereby mitigating the risk of premature re-engagement under uncertain operational conditions.

The selected threshold value of $A_{\text{thres}} = 20\%$ was empirically determined from our dataset and experimental setup to balance responsiveness and reliability. In fact, both the threshold values and window lengths are tunable parameters that can be optimized for specific vehicle platforms ODDs, driver preferences or regulatory requirements.

5.3.7 Trip-Segment Analysis

The final experiment evaluates the system under an authentic application scenario simulating live video stream processing. Given the significant impact of Poltergeist perturbations (as demonstrated in Table 5.2 and Table 5.3) we simulate an acoustic-based camera attack scenario, where the sensor is compromised by Poltergeist-generated noise patterns. These perturbations were injected at controlled intensity levels directly into the Valeo test set frames (which are temporal sequences), ensuring flexible, reproducible conditions for testing.

Prior literature provides very few examples of attack pattern generation with some similarity to our use case ([224–226]). In [224], only a few frames at the end of each video are altered, which is not adaptable to an AD case, in which the vehicle may be attacked at any time. In [225], a white-box approach is adopted: the adversary is assumed to have access to the training data and model parameters of the video classifier. This setting allows carefully crafted perturbations but does not align with our use case, where we model sensor anomalies that occur without such privileged knowledge. In [84], adversarial patches are applied to frame sequences from BDD100K sampled at 30 FPS. The patch is inserted at a random time between seconds 1 and 10 and then moved across the image for 0.2–2 seconds before disappearing. This produces short bursts of corruption within the video stream and is suited to model only external attacks, not also realistic sensor anomalies. A patch attack is a localized perturbation applied to part of the image to mislead the detector. Unlike sensor anomalies such as rain or blur, patch attacks are digitally crafted and do not originate from intrinsic failures of the sensing or computing pipeline.

To address this gap, we developed two distinct attack test cases: Sporadic Attack (i.e., intermittent perturbations) and Massive Attack (i.e., sustained perturbations). Each scenario comprises a 600-sample time series (50 s video). In both cases, the initial anomaly threshold ($\tilde{A}$) is set to 0, reflecting the assumption of no prior attacks in the pre-analysis frames.

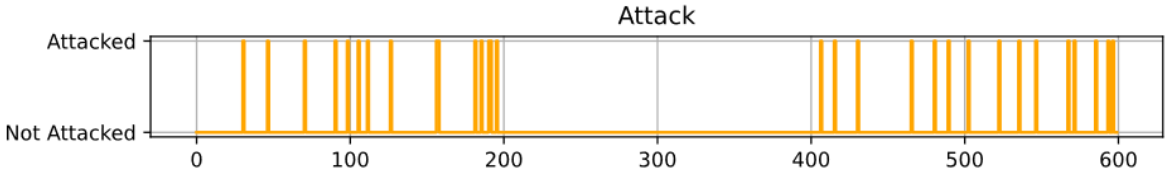
Sporadic Scenario

The sporadic attack scenario features randomly distributed perturbations concentrated in specific intervals (Fig. 5.6a): frames 1-100: 5 'weak' Poltergeist attacks; frames 100-200: 10 'weak' attacks; frames 200-400: no attacks; frames 400-600: 15 'medium' attacks. The 30 resulting attacked frames are the following: 31, 47, 71, 91, 99, 106, 112, 127, 157, 158, 182, 186, 191, 192, 196, 407, 416, 431, 466, 481, 490, 503, 523, 536, 547, 568, 572, 586, 594, 597. This configuration simulates imperfect camera stabilization due to poor road conditions (e.g., uneven surfaces), resulting in minor image oscillations. The scenario enables evaluation of the system's ability to maintain AD mode during low-intensity noise events.

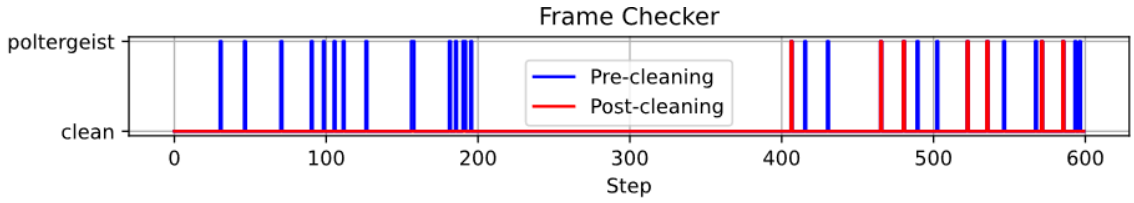
In this scenario, we evaluated four system configurations:

1. Baseline: $A_{\text{thres}} = 0\%$, FCl disabled (Fig. 5.6c)
2. FCl-enabled:$A_{\text{thres}} = 0\%$, FCl active (Fig. 5.6d)
3. Threshold-adjusted: $A_{\text{thres}} = 20\%$, FCl disabled (Fig. 5.6e).
4. Optimized $A_{\text{thres}} = 20\%$, FCl active (Fig. 5.6f)

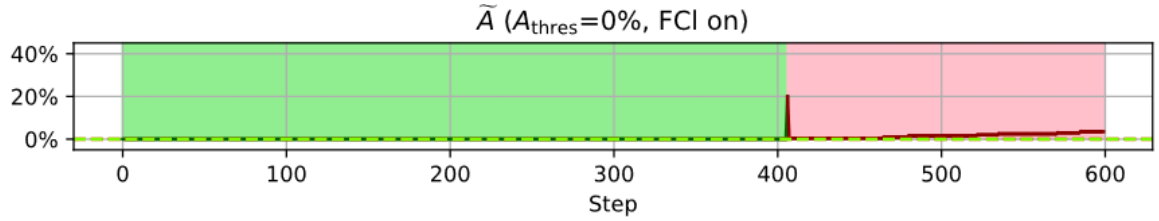
Fig. 5.6 shows an example of sporadic attack during a trip. It depicts the attack scenario, the FCh result, the $\tilde{A}$ metric evolution and the TOR threshold.



(a)



(b)



(c)

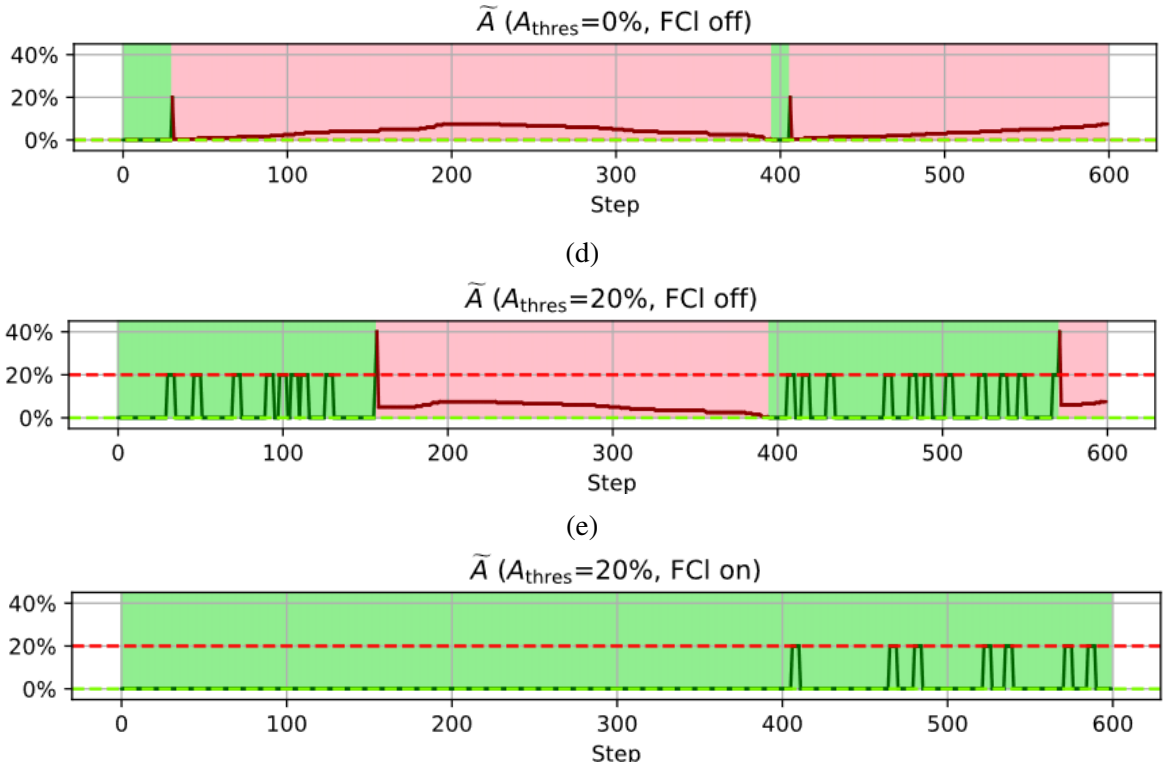


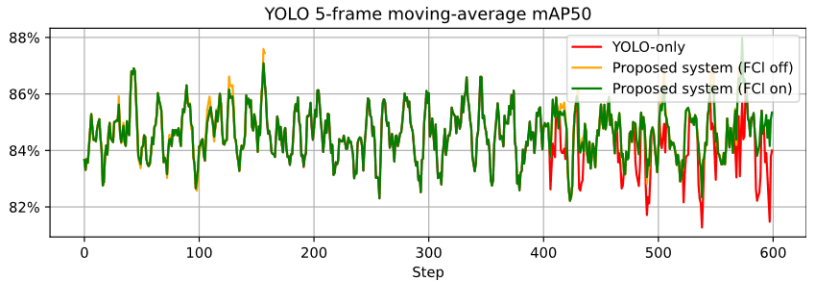
Fig. 5.6 Trip-segment analysis, sporadic attack. (a) depicts the attack scenario; (b) represents the FCh results, blue for pre-cleaning FCh classification and red for post-cleaning FCh classification; (c–f) show the $\tilde{A}$ metric evolution with the two thresholds described in IV.D and IV.E represented by red and green dashed lines, respectively. In (c) and (d), the take-over threshold (red line) is set to 0%, in (e) and (f) to 20%. In (c) and (e) FCl is disabled.

The FCh activity in the scenario is reported in Fig. 5.6b.
We can observe the following:

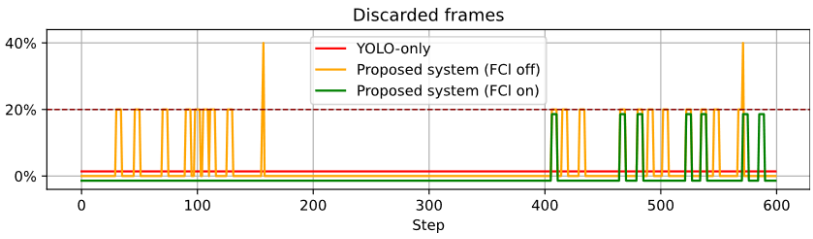
- FCh effectiveness: the FCh is able to detect 100% of the perturbed frames
- FCl effectiveness: the FCl successfully cleans all 'weak' perturbations and 56.6% of 'medium' perturbations (FCl recovery rate: 59.7%), preventing premature TOR triggers (which occurred at frame 31 in configuration 1 and frame 157 in configuration 2).
- Threshold Optimization: the $A_{\text{thres}} = 20\%$ setting maintained AD mode throughout the entire sequence (Configuration 4). While some frames (400-600) were discarded due to 'medium' perturbations, the cumulative anomaly threshold A_{thres} approached but did not exceed the 20% limit.

- System Robustness: the combined $FCl/A_{\text{thres}} = 20\%$ configuration effectively mitigated sporadic noise impacts, demonstrating stable AD maintenance despite intermittent perturbations.

The trip quality assessment is presented in Fig. 5.7 through two metrics: (a) 5-frame moving average mAP50, and (b) count of discarded frames.



(a)



(b)

Fig. 5.7 System performance with and without FCh and FCl in the sporadic scenario. (a) shows the YOLO performance, (b) the discarded frames in the TW.

In the initial trip segment (frames 0-200), we observe that: (i) weak perturbations show negligible impact on mAP50 (consistent with Table 5.2 findings); the FCh-only configuration (with no FCl - orange line) triggers a TOR at frame 158. In the final segment (frames 400-600) we observe that: (i) medium-intensity perturbations reduce baseline mAP50 by 7% (Table 5.2); (ii) the baseline system response shows a minor mAP50 degradation (the effect is actually mitigated in the plotted average), while the employment of FCh+FCl allowed maintaining the target performance (84.6% mAP50), also preventing a TOR at frame 572, which instead occurred in FCl-disabled configuration. Thus, the integrated system (with $A_{\text{thres}} = 20\%$ threshold) demonstrates effective noise compensation across perturbation intensities, preserving YOLO's nominal performance (84.6% mAP50), eliminating two critical TOR events (frames 158 and 572), and keeping an FPS ≥ 10.

Massive Scenario

The massive attack scenario (Fig. 5.8a) features a first phase (frames 1-300), with 30 random 'strong' Poltergeist attacks (10% concentration) and second phase (frames 301-600), with no attacks (recovery phase). The resulting attacked frames are the following: 8, 18, 32, 56, 69, 79, 94, 103, 114, 134, 140, 145, 161, 166, 180, 185, 198, 202, 208, 218, 228, 229, 245, 253, 255, 257, 270, 273, 283, 294.

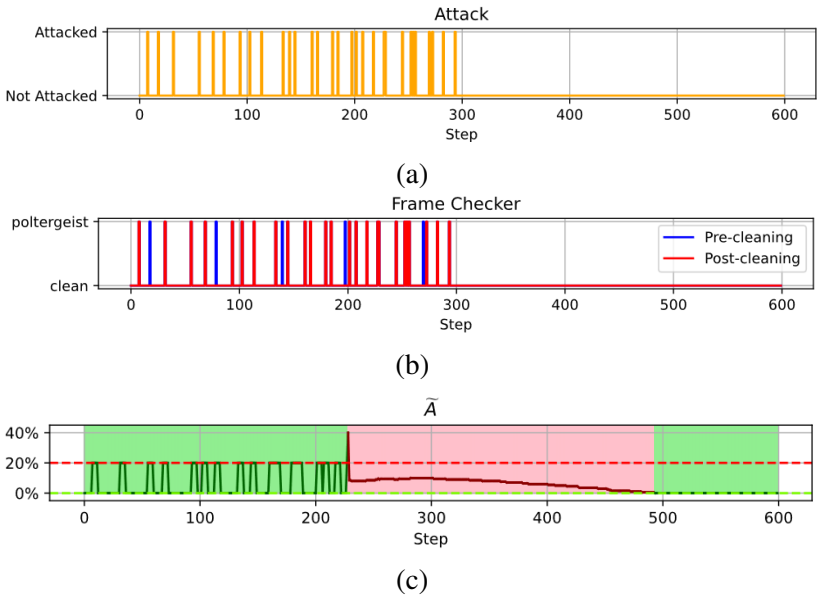


Fig. 5.8 Trip-segment analysis, massive attack. (a) is the attack scenario, (b) the FCh results, and (c) the $\tilde{A}$ evolution.

This represents a critical adversarial acoustic attack scenario, as 'strong' Poltergeist perturbations demonstrate the lowest recovery rates (Table 5.3, final version). The anomaly threshold ($\tilde{A}$) is initialized at 0, assuming no prior attacks.

Fig. 5.8c shows that some attacked frames are correctly cleaned (e.g., frames 18 and 78), whereas some others cannot be recovered (e.g., frames 8, 32, and 56). Also in this scenario, the recorded FCh detection rate is 100%. The system is able to keep the ADF enabled until step 228, when the critical threshold is overcome. The ADF is enabled again at step 493, because of the cessation of attacks at step 300.

The system quality assessment under a massive attack scenario (Fig. 5.9) focuses on the first 300 frames, as the latter trip segment was attack-free.

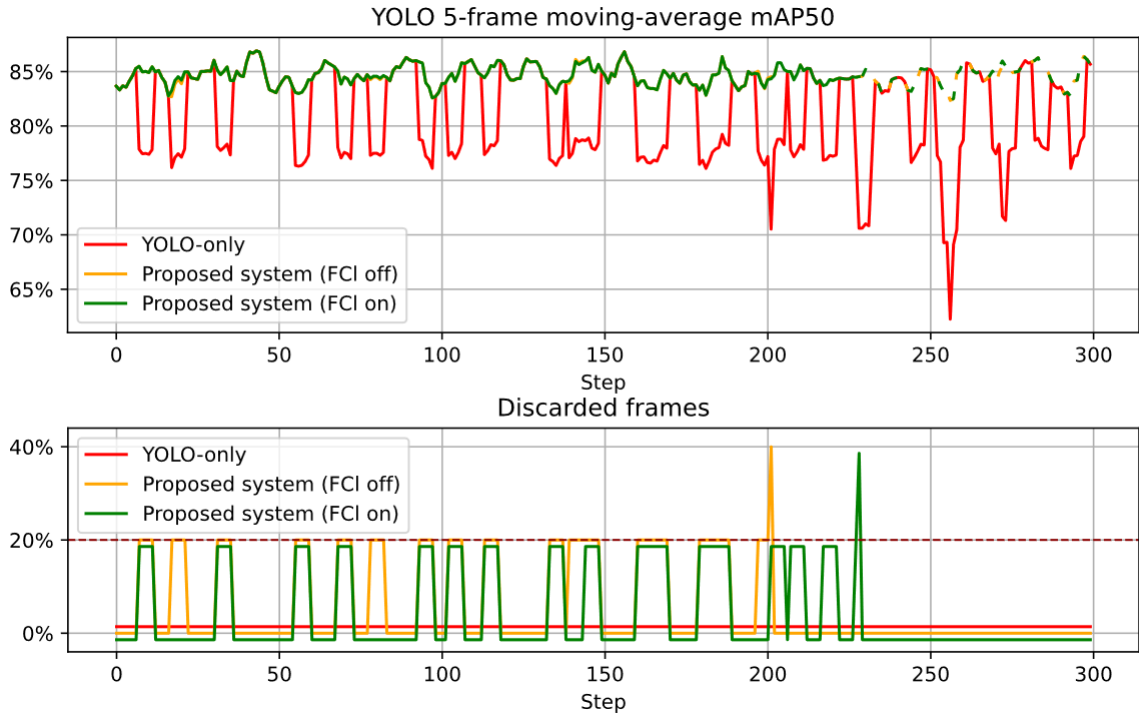


Fig. 5.9 System performance with and without FCh and FCl in the massive scenario, attack phase (frames 1–300). Top shows the YOLO performance, (b) the discarded frames in the TW. Dashed lines in bottom indicate traveling in manual driving mode.

The baseline ADF's mAP50 drops to 64% under intense attacks, whereas enabling the FCh and FCl maintains the target mAP50 (84%) at the cost of discarding some corrupted frames. TOR triggers earlier without FCl (frame 202 vs. frame 229 with FCl), demonstrating FCl's attack-dampening effect and its role in extending user reaction time, critical consideration for HMI design.

Extended testing (3,600 frames, 5 minutes) reveals no TOR in sporadic attacks but three in massive attacks, highlighting the need for robust cleaning strategies and balanced TOR thresholds to ensure system responsiveness and safety under sustained threats.

5.3.8 Comparison with State-of-the-Art Defense Systems

While the TOR-including complete defense system is completely original in literature, we can compare the performance of the proposed TLD pipeline with two highly representative recent systems for defense against perturbations in object detection: [226] and [227].

Mu [226] proposes ADAV, which combines temporal consistency checks with attribution-based masking to counter object-vanishing patches in AD scenarios. Their system improves adversarial mAP from 22% to 36% (clean = 44%) while sustaining 56 FPS in clean conditions and 20 FPS under defense. However, also in this case, no embedded deployment is considered.

Yu et al. [227] address adversarial patch attacks on person detection, introducing a Universal Defensive Frame. The proposed method reduces attack effectiveness substantially, raising mAP from $\sim$18–24% under patch attacks to $\sim$47–51%. Experiments are conducted on a desktop GPU (NVIDIA GeForce GTX 1080Ti), but no embedded deployment or system-level integration is presented.

Performance comparison is summarized in Table 5.6. Overall, compared to these state-of-the-art defenses, our proposed approach stands out for:

1. focusing on realistic sensor-degradation anomalies instead of digital adversarial patch manipulations,

2. working on an embedded platform while still maintaining real-time performance, and

3. providing a modular defense architecture that ensures robust performance through high recovery rates (mAP50 drops from 84.6% to 49.4% under attack, but our FCl is able to restore a mAP50 close to the original one, differently from the compared systems).

Table 5.6 State-of-the-art defense systems comparison.

		[226]	[227]	Ours
Task		Object detection in AD	Person detection	Traffic light detection
Perturbation		Adversarial vanishing patches	Adversarial patches	Sensor anomalies
Checker		✓	×	✓
Defense		Saliency map masking	Universal Defensive Frame	SAPNet, DeblurGAN-v2
Detector		YOLOv5s	YOLOv2	YOLO11n
Fail-safe management		×	×	✓
mAP50	no atk	44%	89.63%	84.6%
	atk (no def)	22%	18.80–24.32%	49.4–84.3%
	atk (w. def)	36%	47.22–51.96%	84.4–84.8%
FPS	no def	63	43	43
	w. def	20	35	12–25
Embedded		×	×	✓
Platform		Workstation	Workstation	Jetson Orin Nano
GPU		Tesla T4	GTX 1080Ti	GM20B

This is achieved by introducing the two TOR management modules (TODM and ADRDM) which supervise the ADF and guarantee a safety exit through human intervention when anomalies cannot be adequately recovered. This form of fail-safe management is absent in [226, 227], whose objective is to clean adversarially perturbed frames and restore acceptable mAP and FPS, without considering a real-world deployment.

5.4 Conclusion

To evaluate real-world viability, we rigorously tested the robustness of our TLD system against state-of-the-art camera sensor vulnerabilities. Our assessment of a cutting-edge YOLO11 detector revealed substantial performance degradation across all standard metrics, with the most severe failure case – simulated using Poltergeist noise injection to emulate stabilizer malfunction – driving mAP50 down to just 49.4%. These findings highlight critical vulnerabilities that must be addressed before deployment in safety-critical automated systems.

To mitigate these vulnerabilities, we propose a novel modular architecture comprising five components: Frame Checker, Frame Cleaner, TOR Decision Maker, AD Revert Decision Maker, and Traffic Light Detector. This design adopts an anti-virus inspired paradigm that decouples noise mitigation from core detection tasks, enhancing both performance and system extensibility. Notably, this represents the first documented architecture incorporating explicit TOR protocols for camera sensor failure scenarios.

Experimental validation demonstrates robust performance across operational conditions. In nominal operation, the system achieves 86.1% precision (77.6% recall, 84.6% mAP50). The FCh achieves 0% false alarms with only 0.3% false negatives. In sensor failure (strong Poltergeist), the system achieves 85.9% precision (77.9% recall, 84.8% mAP50), at a cost of a processing speed degradation until the minimum real-time performance threshold of 10 FPS. The system implements a fail-safe protocol when the FCl cannot adequately remediate corrupted frames, triggering TOR initiation.

We contend that the proposed system architecture and its associated evaluation represent a substantial step forward in enabling trustworthy TLD systems. Addressing sensor failures remains a critical challenge, particularly due to the constrained TOR reaction time in scenarios where driver vigilance is insufficient. Currently, no straightforward solution exists for this issue.

The presented work opens significant perspectives for future research. The system should be tested under different sensor failure scenarios, including more complex and rarer failures (e.g., fog, ice) [141]. The current evaluation considers single-perturbation scenarios, which represent a necessary baseline for systematic assessment. Real automotive deployments, however, may expose the camera to multiple simultaneous anomalies, such as rain combined with low-light conditions or hardware failures under adverse weather. Extending the architecture to handle compound perturbations, including the design of a multi-class FCh and combined cleaning strategies, represents a natural direction for future work.

More sophisticated DM modules for TOR and AD resume could be investigated as well. The system timing performance should also be improved, via hardware and/or software. For instance, stronger quantization techniques could be explored (e.g., binarization [228]) to reduce latency and make the system architecture deployable also on microcontrollers. An extensive human-factor analysis should be performed as well, through simulation and eventually real-world field tests. Although our analysis concentrated on sensor-level anomalies, computing devices are another significant source of faults, representing a critical area for future system-level investigation.

Chapter 6

Final Conclusion

This thesis followed a data–centric path to AD, moving from controlled prototyping to human–in–the–loop collection and, finally, to deployment under sensor anomalies. Chapter 3 combined DT with high–level simulation to probe offline RL for highway DM, and introduced a stable CARLA highway map and tooling to overcome practical limitations of built–in towns. The outcome was twofold: algorithmic insight into sequence modeling for DM, and an instrumented environment that removes friction for long, reproducible runs.

Chapter 4 then used the same highway map to study two complementary perception and decision-making problems under consistent conditions. First, LC intention was reframed as TTLC regression on DL-ready time series collected from 50 drivers, resulting in a public dataset and baselines that anticipate maneuvers up to 4 s ahead with a mean error below 0.6 s. Second, scenario generation and detection scaled the viewpoint from individual maneuvers to contextual understanding: a configurable pipeline rendered labeled MP4 clips across nine highway scenarios, trained a reference R3D classifier, and analyzed separability and confusions among classes. The synthetic models were further validated on real-world data from the PREVENTION dataset, achieving 83.3% accuracy without fine-tuning, thus confirming the robustness of the synthetic-to-real transfer and the realism of the generated distributions. Overall, these results demonstrate how shared geometry, logging, and annotation design enable consistent evaluation across temporal and semantic levels while providing a reproducible foundation for cross-domain generalization studies in highway perception.

Chapter 5 translated the accumulated experience into an embedded, safety–oriented use case: robust TLD on resource–constrained hardware. A modular architecture integrated anomaly detection, image recovery, and TOR logic around a quantized YOLOv11, preserving near–nominal mAP50 under representative perturbations (TPSeNCE, Poltergeist) in real time. This demonstrates how simulation–grounded design choices and dataset discipline can carry into deployable perception functions.

Across chapters, three themes recur. First, careful data design (signals vs. video, windowing, labels) is as decisive as model choice. Second, holding maps and tooling fixed enables fair comparisons and faster iteration. Third, robustness requires thinking beyond clean benchmarks: variability, edge cases, and failure handling must be designed in from the start.

The study's primary limitations stem from the absence of on-vehicle experiments and explicit out-of-distribution evaluations. The degree of realism exhibited in the text is subject to variation across chapters, attributable to the heterogeneity of the ODDs. The deployment of a DRL-based DM agent on a vehicle prototype entails costs, safety constraints, and assurance activities that differ categorically from those involved in maneuver recognition or scenario classification. These are not merely speculative concerns; rather, they are fundamental to the practice of responsible translation.

Nevertheless, a consistent methodology underlies the work: simulate from empirically grounded distributions to achieve coverage; use controllable, scalable simulation to stress the components that matter; define targets and labels that enable actionable predictions; and enforce real-time constraints wherever code is expected to run. The frameworks, datasets, and models developed here operationalize this philosophy and provide a practical basis for controlled on-vehicle trials, enabling fair comparisons under shared conditions.

In summary, the thesis proposes a coherent methodology that can be applied from the initial design of data principles to the development of executable systems. The argument posits that dependable AD is contingent upon three factors: (i) the selection of appropriate prediction metrics and their rigorous measurement, (ii) the curation and integration of data that accentuate the behaviors of interest, and (iii) the enforcement of runtime constraints that are consistent with the vehicle platform.

References

[1] L. Forneris, F. Bellotti, R. Berta, M. Cossu, M. Fresta, R. Rezaieamale, H. Rojhan, V. Soltanmuradov, F. Tango, and L. Lazzaroni, "Setting Up a Lightweight Simulation Environment for Automated Driving Dataset Collection," in *Applications in Electronics Pervading Industry, Environment and Society*. Springer Nature Switzerland, 2025, pp. 156–163.

[2] L. Forneris, R. Berta, M. Fresta, L. Lazzaroni, H. Rojhan, C. Oh, A. Pighetti, F. Tango, and F. Bellotti, "Lane change intention recognition dataset," Aug. 2025. [Online]. Available: https://doi.org/10.5281/zenodo.16686054

[3] L. Forneris, R. Berta, M. Fresta, L. Lazzaroni, H. Rojhan, C. Oh, A. Pighetti, H. Ballout, F. Tango, and F. Bellotti, "A deployment-oriented simulation framework for deep learning-based lane change prediction," *IEEE Signal Processing Letters*, vol. 33, pp. 136–140, 2026.

[4] M. Cossu, R. Berta, L. Lazzaroni, L. Forneris, M. Fresta, A. Pighetti, A. Pilla, J.-L. Sauvaget, H. Touil, L. Durville, T. Griffon, G. Ben Nejma, H. S. Fouad, and F. Bellotti, "Synthetic highway driving scenarios dataset," Dec. 2025. [Online]. Available: https://doi.org/10.5281/zenodo.17652045

[5] M. Cossu, R. Berta, L. Lazzaroni, L. Forneris, M. Fresta, A. Pighetti, A. Pilla, J.-L. Sauvaget, H. Touil, L. Durville, T. Griffon, G. Ben Nejma, F. Hadj Selem, and F. Bellotti, "Generation of synthetic datasets of highway driving scenarios," *IEEE Transactions on Intelligent Transportation Systems*, 2025, under review, manuscript no. T-ITS-25-12-6132.

[6] O. L. Audi, F. Bellotti, R. Berta, Z. Liu, L. Lazzaroni, A. Pilla, H. Rojhan, and L. Forneris, "Improving lane change performance in carla through curvature-adaptive b-spline planning," in *Applications in Electronics Pervading Industry, Environment and Society*, M. Ruo Roch, A. Gómez, L. Lazzaroni, M. Martina, P. Motto Ros, N. Pazos Escudero, and R. Berta, Eds. Cham: Springer Nature Switzerland, 2026, pp. 212–217.

[7] L. Lazzaroni, F. Bellotti, A. Pighetti, L. Forneris, M. Fresta, H. Ballout, C. Oh, S. Pantawane, and R. Berta, "An embedded system architecture for robust deep learning-based traffic light detection," *IEEE Sensors Journal*, vol. 25, no. 24, pp. 44 859–44 874, 2025.

[8] L. Forneris, F. Bellotti, R. Berta, L. Lazzaroni, and C. Oh, "Exploring decision transformer for highway automated driving," in *Applications in Electronics Pervading*

Industry, Environment and Society - APPLEPIES 2024, Turin, Italy, 19-20 September 2024, ser. Lecture Notes in Electrical Engineering, M. R. Roch, F. Bellotti, R. Berta, M. Martina, and P. M. Ros, Eds., vol. 1369. Springer, 2024, pp. 123–130. [Online]. Available: https://doi.org/10.1007/978-3-031-84100-2_15

[9] S. Devi, P. Malarvezhi, R. Dayana, and K. Vadivukkarasi, "A comprehensive survey on autonomous driving cars: A perspective view," *Wirel. Pers. Commun.*, vol. 114, no. 3, p. 2121–2133, Oct. 2020, Available: https://doi.org/10.1007/s11277-020-07468-y. [Online]. Available: https://doi.org/10.1007/s11277-020-07468-y

[10] C. Gianoglio, A. Rizik, E. Tavanti, D. D. Caviglia, and A. Randazzo, "On the edge recurrent neural network approach for ground moving fmcw radar target classification," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 522–534, 2024.

[11] S. Gautam and A. Kumar, "Image-based automatic traffic lights detection system for autonomous cars: a review," *Multimedia Tools and Applications*, vol. 82, no. 17, pp. 26 135–26 182, jul 1 2023.

[12] A. Pighetti, F. Bellotti, C. Oh, L. Lazzaroni, L. Forneris, M. Fresta, and R. Berta, "Investigating adversarial policy learning for robust agents in automated driving highway simulations," in *Applications in Electronics Pervading Industry, Environment and Society*, F. Bellotti, M. D. Grammatikakis, A. Mansour, M. Ruo Roch, R. Seepold, A. Solanas, and R. Berta, Eds. Cham: Springer Nature Switzerland, 2024, pp. 124–129.

[13] L. Lazzaroni, F. Bellotti, A. Capello, M. Cossu, A. De Gloria, and R. Berta, "Deep reinforcement learning for automated car parking," in *Applications in Electronics Pervading Industry, Environment and Society*, R. Berta and A. De Gloria, Eds. Cham: Springer Nature Switzerland, 2023, pp. 125–130, Available: https://doi.org/10.1007/978-3-031-30333-3$_1$6.

[14] L. Bai, F. Zheng, K. Hou, X. Liu, L. Lu, and C. Liu, "Longitudinal Control of Automated Vehicles: A Novel Approach by Integrating Deep Reinforcement Learning With Intelligent Driver Model," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 8, pp. 11 014–11 028, 8 2024, event-title: IEEE Transactions on Vehicular Technology.

[15] M. Fresta, F. Bellotti, I. Bochenko, L. Lazzaroni, G. Merlhiot, F. Tango, and R. Berta, "Deep Learning-based Real-Time Driver Cognitive Distraction Detection," *IEEE Access*, pp. 1–1, 2025, event-title: IEEE Access.

[16] S. Wang, D. Jia, and X. Weng, "Deep reinforcement learning for autonomous driving," 2019. [Online]. Available: https://arxiv.org/abs/1811.11329

[17] R. Berta, L. Lazzaroni, A. Capello, M. Cossu, L. Forneris, A. Pighetti, and F. Bellotti, "Development of Deep-Learning-Based Autonomous Agents for Low-Speed Maneuvering in Unity," *Journal of Intelligent and Connected Vehicles*, vol. 7, no. 3, pp. 229–244, 9 2024, event-title: Journal of Intelligent and Connected Vehicles.

[18] A. Capello, L. Forneris, A. Pighetti, F. Bellotti, L. Lazzaroni, M. Cossu, A. De Gloria, and R. Berta, "Investigating high-level decision making for automated driving," in *Applications in Electronics Pervading Industry, Environment and Society*, R. Berta and A. De Gloria, Eds. Cham: Springer Nature Switzerland, 2023, pp. 307–311.

[19] R. Berta, L. Lazzaroni, A. Capello, M. Cossu, L. Forneris, A. Pighetti, and F. Bellotti, "Development of deep-learning-based autonomous agents for low-speed maneuvering in unity," *Journal of Intelligent and Connected Vehicles*, vol. 7, no. 3, pp. 229–244, 2024.

[20] S. Wang, X. Liang, D. Zhao, J. Huang, X. Xu, B. Dai, and Q. Miao, "Deep reinforcement learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, pp. 5064–5078, 2022.

[21] K. Arulkumaran, M. Deisenroth, M. Brundage, and A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, pp. 26–38, 2017.

[22] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Perez, "Deep Reinforcement Learning for Autonomous Driving: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9351818/

[23] R. Sutton and A. Barto, "Reinforcement Learning: An Introduction," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 1054–1054, Sep. 1998. [Online]. Available: https://ieeexplore.ieee.org/document/712192/

[24] R. Figueiredo Prudencio, M. R. O. A. Maximo, and E. L. Colombini, "A survey on offline reinforcement learning: Taxonomy, review, and open problems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 8, pp. 10 237–10 257, 2024.

[25] T. Seno and M. Imai, "d3rlpy: An offline deep reinforcement learning library," *J. Mach. Learn. Res.*, vol. 23, pp. 315:1–315:20, 2021.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[27] R. Sutton and A. Barto, "Reinforcement learning: An introduction," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 1054–1054, 1998, Available: https://doi.org/10.1109/TNN.1998.712192.

[28] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: reinforcement learning via sequence modeling," in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NIPS '21. Red Hook, NY, USA: Curran Associates Inc., 2021.

[29] E. Parisotto, H. F. Song, J. W. Rae, R. Pascanu, C. Gulcehre, S. M. Jayakumar, M. Jaderberg, R. L. Kaufman, A. Clark, S. Noury, M. M. Botvinick, N. Heess, and R. Hadsell, "Stabilizing transformers for reinforcement learning," in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML'20. JMLR.org, 2020.

[30] L. Bruck, B. Haycock, and A. Emadi, "A review of driving simulation technology and applications," *IEEE Open Journal of Vehicular Technology*, vol. 2, pp. 1–16, 2021.

[31] F. Silva, V. Alves, P. Santos, N. Antunes, N. Cruz, and J. Almeida, "Realistic 3d simulators for automotive: A review of main applications and features," *Sensors*, vol. 24, no. 18, p. 5880, 2024.

[32] Y. Li, W. Yuan, S. Zhang, W. Yan, Q. Shen, C. Wang, and M. Yang, "Choose your simulator wisely: A review on open-source simulators for autonomous driving," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2311.11056

[33] E. Egea-Lopez, F. Losilla, J. Pascual-García, and J. M. Molina-García-Pardo, "Vehicular networks simulation with realistic physics," *IEEE Access*, vol. 7, pp. 44 021–44 036, 2019.

[34] C. Himmels, J. Venrooij, A. Parduzi, M. Peller, and A. Riener, "The bigger the better? investigating the effects of driving simulator fidelity on driving behavior and perception," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 101, pp. 250–266, 2024.

[35] E. Leurent, "An environment for autonomous driving decision-making," https://github.com/eleurent/highway-env, 2018, accessed: 2025-09-26.

[36] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. López, and V. Koltun, "CARLA: an open urban driving simulator," in *1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA, November 13-15, 2017, Proceedings*, ser. Proceedings of Machine Learning Research, vol. 78. PMLR, 2017, pp. 1–16. [Online]. Available: http://proceedings.mlr.press/v78/dosovitskiy17a.html

[37] S. Malik, M. A. Khan, Aadam, H. El-Sayed, F. Iqbal, J. Khan, and O. Ullah, "Carla+: An evolution of the carla simulator for complex environment using a probabilistic graphical model," *Drones*, 2023.

[38] M. M. Chaar, J. Raiyn, and G. Weidl, "Improve bounding box in carla simulator," pp. 267–275, 2024.

[39] J.-E. Deschaud, D. Duque, J. P. Richa, S. Velasco-Forero, B. Marcotegui, , and F. Goulette, "Paris-carla-3d: A real and synthetic outdoor point cloud dataset for challenging tasks in 3d mapping," *Remote. Sens.*, vol. 13, p. 4713, 2021.

[40] A. Sharma, V. Kumar, and L. Longchamps, "Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN models for detection of multiple weed species," *Smart Agricultural Technology*, vol. 9, p. 100648, 12 2024.

[41] R. Khanam and M. Hussain, "Yolov11: An Overview of the Key Architectural Enhancements," 2024, version: 1. [Online]. Available: https://arxiv.org/abs/2410.17725

[42] S. Zheng, C. Lu, and S. G. Narasimhan, "Tpsence: Towards Artifact-Free Realistic Rain Generation for Deraining and Object Detection in Rain," in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 1 2024, pp. 5382–5391, iSSN: 2642-9381. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10484120

[43] X. Ji, Y. Cheng, Y. Zhang, K. Wang, C. Yan, W. Xu, and K. Fu, "Poltergeist: Acoustic Adversarial Machine Learning against Cameras and Computer Vision," in *2021 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, 5 2021, pp. 160–175, [Online; accessed 2024-12-11]. [Online]. Available: https://ieeexplore.ieee.org/document/9519394/

[44] M. Fresta *et al.*, "Deep learning-based real-time driver cognitive distraction detection," *IEEE Access*, vol. 13, pp. 26 589–26 607, 2025.

[45] M. Cossu, "A simulation-based framework for highway driving scenario recognition," Ph.D. dissertation, University of Genoa, Italy, 2025. [Online]. Available: https://hdl.handle.net/11567/1241978

[46] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *CoRR*, vol. abs/2005.01643, 2020. [Online]. Available: https://arxiv.org/abs/2005.01643

[47] E. Leurent, "An environment for autonomous driving decision-making," https://github.com/eleurent/highway-env, 2018.

[48] M. L. Puterman, "Markov decision processes," *Handbooks in operations research and management science*, vol. 2, pp. 331–434, 1990.

[49] F. Bellotti, L. Lazzaroni, A. Capello, M. Cossu, A. D. Gloria, and R. Berta, "Explaining a deep reinforcement learning (drl)-based automated driving agent in highway simulations," *IEEE Access*, vol. 11, pp. 28 522–28 550, 2023.

[50] A. Capello, L. Forneris, A. Pighetti, F. Bellotti, L. Lazzaroni, M. Cossu, A. De Gloria, and R. Berta, "Investigating High-Level Decision Making for Automated Driving," in *Applications in Electronics Pervading Industry, Environment and Society*, ser. Lecture Notes in Electrical Engineering, R. Berta and A. De Gloria, Eds. Cham: Springer Nature Switzerland, 2023, pp. 307–311.

[51] H. M. Mandalia and M. D. D. Salvucci, "Using support vector machines for lane-change detection," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 49, no. 22, pp. 1965–1969, 2005. [Online]. Available: https://doi.org/10.1177/154193120504902217

[52] Y. Zhang, S. Zhang, and R. Luo, "Lane change intent prediction based on multi-channel cnn considering vehicle time-series trajectory," 10 2022, pp. 287–292.

[53] K. Gao, X. Li, B. Chen, L. Hu, J. Liu, R. Du, and Y. Li, "Dual transformer based prediction for lane change intentions and trajectories in mixed traffic environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, pp. 6203–6216, 2023.

[54] J. Nilsson, J. Fredriksson, and E. Coelingh, "Rule-based highway maneuver intention recognition," *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 950–955, 2015.

[55] C.-J. Jin, V. Knoop, D. Li, L. Meng, and H. Wang, "Discretionary lane-changing behavior: empirical validation for one realistic rule-based model," *Transportmetrica A: Transport Science*, vol. 15, pp. 244 – 262, 2019.

[56] K. Shengjie, K. Jiang, Y. Weiguang, R. Yan, W. Zhou, M. Yang, and D. Yang, "Lane change intention recognition for intelligent connected vehicle using trajectory prediction," pp. 389–400, 2020.

[57] B. Khelfa, I. Ba, and A. Tordeux, "Predicting highway lane-changing maneuvers: A benchmark analysis of machine and ensemble learning algorithms," *Physica A: Statistical Mechanics and its Applications*, vol. 612, p. 128471, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437123000262

[58] A. Kesting, M. Treiber, and D. Helbing, "General Lane-Changing Model MOBIL for Car-Following Models," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1999, no. 1, pp. 86–94, Jan. 2007. [Online]. Available: https://journals.sagepub.com/doi/10.3141/1999-10

[59] P. Dokania, M. Perrollaz, S. Lefevre, and C. Laugier, "Learning-based approach for online lane change intention prediction," 06 2013.

[60] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 4363–4369.

[61] Y. Xia, Z. Qu, Z. Sun, and Z. Li, "A human-like model to understand surrounding vehicles' lane changing intentions for autonomous driving," *IEEE Transactions on Vehicular Technology*, vol. 70, pp. 4178–4189, 2021.

[62] F. Wirthmüller, M. Klimke, J. Schlechtriemen, J. Hipp, and M. Reichert, "Predicting the time until a vehicle changes the lane using lstm-based recurrent neural networks," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2357–2364, 2021.

[63] Y. Jeong, "Predictive lane change decision making using bidirectional long shot-term memory for autonomous driving on highways," *IEEE Access*, vol. 9, pp. 144 985–144 998, 2021, publisher Copyright: © 2021 Institute of Electrical and Electronics Engineers Inc.. All rights reserved.

[64] A. Das, M. N. Khan, and M. M. Ahmed, "Detecting lane change maneuvers using shrp2 naturalistic driving data: A comparative study machine learning techniques," *Accident Analysis Prevention*, vol. 142, p. 105578, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457519315751

[65] S. Patel, B. Griffin, K. Kusano, and J. J. Corso, "Predicting future lane changes of other highway vehicles using rnn-based deep models," 2019. [Online]. Available: https://arxiv.org/abs/1801.04340

[66] F. Kruber, J. Wurst, M. Botsch, and S. Chakraborty, *Unsupervised Random Forest Learning for Traffic Scenario Categorization*, 09 2023, pp. 565–590.

[67] A. Hu, L. Russell, H. Yeo, Z. Murez, G. Fedoseev, A. Kendall, J. Shotton, and G. Corrado, "Gaia-1: A generative world model for autonomous driving," 2023. [Online]. Available: https://arxiv.org/abs/2309.17080

[68] X. Chen, J. Yan, W. Liao, T. He, and P. Peng, "Int2planner: an intention-based multi-modal motion planner for integrated prediction and planning," in *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. [Online]. Available: https://doi.org/10.1609/aaai.v39i14.33595

[69] M. Manzour, A. Ballardini, R. Izquierdo, and M. Sotelo, "Explainable lane change prediction for near-crash scenarios using knowledge graph embeddings and retrieval augmented generation," 2025. [Online]. Available: https://arxiv.org/abs/2501.11560

[70] U.S. Department Of Transportation Federal Highway Administration, "Next Generation Simulation (NGSIM) Vehicle Trajectories and Supporting Data," 2017. [Online]. Available: https://data.transportation.gov/d/8ect-6jqj

[71] B. Coifman and L. Li, "A critical evaluation of the next generation simulation (ngsim) vehicle trajectory dataset," *Transportation Research Part B: Methodological*, vol. 105, pp. 362–377, 11 2017.

[72] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. Maui, HI: IEEE, 11 2018, pp. 2118–2125, [Online; accessed 2025-07-04]. [Online]. Available: https://ieeexplore.ieee.org/document/8569552/

[73] Z. Wang and Y. Gao, "Lane change inconsistencies in the highd dataset," *Transport Findings*, vol. 2025, Mar. 2025.

[74] R. Izquierdo, A. Quintanar, I. Parra, D. Fernández-Llorca, and M. A. Sotelo, "The PREVENTION dataset: a novel benchmark for PREdiction of VEhicles iNTentIONs," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 10 2019, pp. 3114–3121.

[75] V. Ramanishka, Y.-T. Chen, T. Misu, and K. Saenko, "Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[76] A. Jain, H. S. Koppula, S. Soh, B. Raghavan, A. Singh, and A. Saxena, "Brain4cars: Car that knows before you do via sensory-fusion deep learning architecture," 2016. [Online]. Available: https://arxiv.org/abs/1601.00740

[77] S. Ettinger *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9690–9699.

[78] Federal Highway Administration, "SHRP2 annual report 2017," U.S. Department of Transportation, Washington, DC, USA, Technical Report, 2017, accessed: 2026. [Online]. Available: https://www.fhwa.dot.gov/goshrp2/

[79] "Iso 26262," https://www.vector.com/int/en/lp/us/iso-26262/, [Online; accessed 2023-12-07]. [Online]. Available: https://www.vector.com/int/en/lp/us/iso-26262/

[80] International Organization for Standardization, "Iso 21448:2022 — safety of the intended functionality (sotif)," 2022, accessed: Dec. 07, 2023. [Online]. Available: https://www.iso.org/standard/77490.html

[81] *Evaluation of Autonomous Products, UL Standard 4600*, 1st ed., Underwriters Laboratories (UL), Northbrook, IL, USA, 2022. [Online]. Available: https://standardscatalog.ul.com/standards/en/standard_4600_1

[82] M. Farrell, M. Luckcuck, L. Pullum, M. Fisher, A. Hessami, D. Gal, Z. Murahwi, and K. Wallace, "Evolution of the IEEE P7009 Standard: Towards Fail-Safe Design of Autonomous Systems," in *2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, Oct. 2021, pp. 401–406. [Online]. Available: https://ieeexplore.ieee.org/document/9700402

[83] International Organization for Standardization, "Road vehicles — cybersecurity engineering," 2018, accessed: Dec. 07, 2023. [Online]. Available: https://www.iso.org/standard/70918.html

[84] "IEC 62443-4-2: Security for industrial automation and control systems — technical security requirements for iacs components," International Electrotechnical Commission, Tech. Rep., 2019, available at https://www.iec.ch/.

[85] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, "Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 9 2015, pp. 982–988, iSSN: 2153-0017. [Online]. Available: https://ieeexplore.ieee.org/document/7313256

[86] E. Andreotti, P. B. Baykas, and S. Selpi, "Mathematical definitions of scene and scenario for analysis of automated driving systems in mixed-traffic simulations," *IEEE Transactions on Intelligent Vehicles*, vol. 6, pp. 366–375, 2021.

[87] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for Development, Test and Validation of Automated Vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 6 2018, pp. 1821–1827, iSSN: 1931-0587. [Online]. Available: https://ieeexplore.ieee.org/document/8500406

[88] International Organization for Standardization, "Iso 26262 — road vehicles — functional safety," 2018, accessed: Dec. 07, 2023. [Online]. Available: https://www.vector.com/int/en/lp/us/iso-26262/

[89] C. Neurohr, L. Westhofen, T. Koopmann, E. Möhlmann, E. Böde, and A. Hahn, "On scenario formalisms for automated driving," *ArXiv*, vol. abs/2504.04868, 2025.

[90] K. T. Kurian, N. Rajesh, E. Lefeber, J. Ploeg, N. Van De Wouw, I. Besselink, and M. Alirezaei, "Formalizing vocabulary for scenario-based testing of automated driving systems: From semantics to mathematics," in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, 2024, pp. 2679–2686.

[91] H. Weber, J. Bock, J. Klimke, C. Roesener, J. Hiller, R. Krajewski, A. Zlocki, and L. Eckstein, "A framework for definition of logical scenarios for safety assurance of automated driving," *Traffic Injury Prevention*, vol. 20, no. sup1, pp. S65–S70, jun 12 2019, number: sup1 publisher: Taylor & Francis _eprint: https://doi.org/10.1080/15389588.2019.1630827 PMID: 31381437.

[92] J. Jeon, J. Yoo, T. Oh, and J. Yoo, "Simulation-based logical scenario generation and analysis methodology for evaluation of autonomous driving systems," *IEEE Access*, vol. 13, pp. 43 338–43 359, 2025.

[93] M. Scholtes, L. Westhofen, L. R. Turner, K. Lotto, M. Schuldes, H. Weber, N. Wagener, C. Neurohr, M. H. Bollmann, F. Körtke, J. Hiller, M. Hoss, J. Bock, and L. Eckstein, "6-Layer Model for a Structured Description and Categorization of Urban Traffic and Environment," *IEEE Access*, vol. 9, pp. 59 131–59 147, 2021, event-title: IEEE Access.

[94] E. de Gelder, J.-P. Paardekooper, A. K. Saberi, H. Elrofai, O. O. d. Camp, S. Kraines, J. Ploeg, and B. De Schutter, "Towards an ontology for scenario definition for the assessment of automated vehicles: An object-oriented framework," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, p. 300–314, Jun. 2022. [Online]. Available: http://dx.doi.org/10.1109/TIV.2022.3144803

[95] G. Bagschik, T. Menzel, and M. Maurer, "Ontology based scene creation for the development of automated vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1813–1820.

[96] C. Chang, D. Cao, L. Chen, K. Su, K. Su, Y. Su, F.-Y. Wang, J. Wang, P. Wang, J. Wei, G. Wu, X. Wu, H. Xu, N. Zheng, and L. Li, "Metascenario: A framework for driving scenario data description, storage and indexing," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1156–1175, 2023.

[97] R. Pfeffer, J. He, and E. Sax, "Development and implementation of a concept for the meta description of highway driving scenarios with focus on interactions of road users," 02 2020.

[98] A. Bracquemond and G. Thiolon, "Moove Project: Recognition of Road Scenes by the Data Collected at the Output of the Sensors of the Autonomous Vehicle," in *Proc.s of 8th Aachen Colloquium Automobile and Engine Technology*, Aachen, 2019.

[99] G. Ben Nejma and L. Durville, "Focusing on critical scenarios for the safety of autonomous vehicles," in *Proceedings, 21rst Driving Simulation & Virtual Reality Conference (DSC 2022)*, Strasbourg, France, 9 2022.

[100] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays, "Argoverse 2: Next Generation Datasets for Self-Driving Perception and Forecasting," jan 1 2023, arXiv:2301.00493 [cs]. [Online]. Available: http://arxiv.org/abs/2301.00493

[101] I. García-Aguilar, J. García-González, D. Medina, R. M. Luque-Baena, E. Domínguez, and E. López-Rubio, "Detection of dangerously approaching vehicles over onboard cameras by speed estimation from apparent size," *Neurocomputing*, vol. 567, p. 127057, jan 28 2024.

[102] A. Prabu, N. Ranjan, L. Li, R. Tian, S. Chien, Y. Chen, and R. Sherony, "Scendd: A Scenario-based Naturalistic Driving Dataset," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 10 2022, pp. 4363–4368, [Online; accessed 2023-12-01]. [Online]. Available: https://ieeexplore.ieee.org/document/9921953

[103] T. Moers, L. Vater, R. Krajewski, J. Bock, A. Zlocki, and L. Eckstein, "The exiD Dataset: A Real-World Trajectory Dataset of Highly Interactive Highway Scenarios in Germany," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 6 2022, pp. 958–964, [Online; accessed 2023-12-02]. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9827305

[104] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle, and M. Tomizuka, "Interaction Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps," sep 30 2019, arXiv:1910.03088 [cs, eess]. [Online]. Available: http://arxiv.org/abs/1910.03088

[105] D. Zhao, Y. Guo, and Y. J. Jia, "Trafficnet: An open naturalistic driving scenario library," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 10 2017, pp. 1–8, iSSN: 2153-0017. [Online]. Available: https://ieeexplore.ieee.org/document/8317860

[106] L. N. Alegre, "SUMO-RL," https://github.com/LucasAlegre/sumo-rl, 2019.

[107] M. Althoff, M. Koschi, and S. Manzinger, "Commonroad: Composable benchmarks for motion planning on roads," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 719–726.

[108] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: Architecture and benchmarking for reinforcement learning in traffic control," *CoRR*, vol. abs/1710.05465, 2017. [Online]. Available: http://arxiv.org/abs/1710.05465

[109] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," in *Field and Service Robotics*, ser. Springer Proceedings in Advanced Robotics, M. Hutter and R. Siegwart, Eds. Cham: Springer International Publishing, 2018, pp. 621–635.

[110] "Nvidia TensorRT," https://developer.nvidia.com/tensorrt, [Online; accessed 2024-12-10]. [Online]. Available: https://developer.nvidia.com/tensorrt

[111] F. Zhu, L. Ma, X. Xu, D. Guo, X. Cui, and Q. Kong, "Baidu Apollo Auto-Calibration System - An Industry-Level Data-Driven and Learning based Vehicle Longitude Dynamic Calibrating Algorithm," aug 30 2018, arXiv:1808.10134 [cs]. [Online]. Available: http://arxiv.org/abs/1808.10134

[112] S. Malik, M. A. Khan, and H. El-Sayed, "Carla: Car learning to act - an inside out," pp. 742–749, 2021.

[113] P. Kaur, S. Taghavi, Z. Tian, and W. Shi, "A survey on simulators for testing self-driving cars," *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*, pp. 62–70, 2021.

[114] Y. Cabon, N. Murray, and M. Humenberger, "Virtual KITTI 2," jan 29 2020, arXiv:2001.10773 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2001.10773

[115] C. Lu, T. Yue, and S. Ali, "Deepscenario: An Open Driving Scenario Dataset for Autonomous Driving System Testing," in *2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR)*, 5 2023, pp. 52–56, iSSN: 2574-3864. [Online]. Available: https://ieeexplore.ieee.org/document/10174023

[116] Y. Kang, H. Yin, and C. Berger, "Test Your Self-Driving Algorithm: An Overview of Publicly Available Driving Datasets and Virtual Testing Environments," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 171–185, 6 2019, event-title: IEEE Transactions on Intelligent Vehicles.

[117] S. Wang and X. Lin, "Eco-driving control of connected and automated hybrid vehicles in mixed driving scenarios," *Applied Energy*, vol. 271, p. 115233, aug 1 2020.

[118] F. Montanari, R. German, and A. Djanatliev, "Pattern Recognition for Driving Scenario Detection in Real Driving Data," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 10 2020, pp. 590–597, iSSN: 2642-7214. [Online]. Available: https://ieeexplore.ieee.org/document/9304560

[119] L. Ding, M. Glazer, M. Wang, B. Mehler, B. Reimer, and L. Fridman, "Mit-AVT Clustered Driving Scene Dataset: Evaluating Perception Systems in Real-World Naturalistic Driving Scenarios," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 10 2020, pp. 232–237, iSSN: 2642-7214. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9304677

[120] C. Chang, D. Cao, L. Chen, K. Su, K. Su, Y. Su, F.-Y. Wang, J. Wang, P. Wang, J. Wei, G. Wu, X. Wu, H. Xu, N. Zheng, and L. Li, "Metascenario: A Framework for Driving Scenario Data Description, Storage and Indexing," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1156–1175, 2 2023, event-title: IEEE Transactions on Intelligent Vehicles.

[121] C. Reichenbächer, M. Rasch, Z. Kayatas, F. Wirthmüller, J. Hipp, T. Dang, and O. Bringmann, "Identifying Scenarios in Field Data to Enable Validation of Highly Automated Driving Systems," in *Proceedings of the 8th International Conference on Vehicle Technology and Intelligent Transport Systems*, 2022, pp. 134–142, arXiv:2203.03515 [cs]. [Online]. Available: http://arxiv.org/abs/2203.03515

[122] M. Biparva, D. Fernández-Llorca, R. I. Gonzalo, and J. K. Tsotsos, "Video Action Recognition for Lane-Change Classification and Prediction of Surrounding Vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 569–578, 9 2022, event-title: IEEE Transactions on Intelligent Vehicles.

[123] K. Liang, J. Wang, and A. Bhalerao, "Lane Change Classification and Prediction with Action Recognition Networks," in *Computer Vision – ECCV 2022 Workshops*, ser. Lecture Notes in Computer Science, L. Karlinsky, T. Michaeli, and K. Nishino, Eds. Cham: Springer Nature Switzerland, 2023, pp. 617–632.

[124] ——, "Lane change classification and prediction with action recognition networks," 2024.

[125] Z. Kayatas and D. Bestle, "Scenario identification and classification to support the assessment of advanced driver assistance systems," *Applied Mechanics*, 2024.

[126] L. Balasubramanian, J. Wurst, M. Botsch, and K. Deng, "Open-world learning for traffic scenarios categorisation," *IEEE Transactions on Intelligent Vehicles*, vol. 8, pp. 3506–3521, 2023.

[127] P. Sankaranarayanan, A. Ramanujam, S. Sathy, and R. Jayaprakash, "Predicut - a machine learning model for online prediction of cut-in manoeuvre for autonomous vehicles," *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, pp. 1–5, 2023.

[128] R. Girshick, "Fast R-CNN," 2015, pp. 1440–1448, [Online; accessed 2024-04-15]. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2015/html/Girshick_Fast_R-CNN_ICCV_2015_paper.html

[129] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 6 2014, pp. 580–587, iSSN: 1063-6919. [Online]. Available: https://ieeexplore.ieee.org/document/6909475

[130] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2016, pp. 779–788, iSSN: 1063-6919. [Online]. Available: https://ieeexplore.ieee.org/document/7780460

[131] S. A. Babu Parisapogu, N. Narla, A. Juryala, and S. Ramavath, "Yolo based Object Detection Techniques for Autonomous Driving," in *2024 Second International Conference on Inventive Computing and Informatics (ICICI)*, 6 2024, pp. 249–256, [Online; accessed 2024-12-24]. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10675605?casa_token=dW4jz-7SjbgAAAAA:V83aiTBWKEZ4SmCkA_AEfSUSykN9F3bQq1m96YfQKIzdUgyqbX86I9brEm9c8aVsASuDCEqPY2c

[132] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "Yolov10: Real-Time End-to-End Object Detection," oct 30 2024, arXiv:2405.14458 [cs]. [Online]. Available: http://arxiv.org/abs/2405.14458

[133] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, nov 20 2023.

[134] N. Polley, S. Pavlitska, Y. Boualili, P. Rohrbeck, P. Stiller, A. K. Bangaru, and J. M. Zöllner, "Tld-READY: Traffic Light Detection – Relevance Estimation and Deployment Analysis," sep 11 2024, arXiv:2409.07284 [cs]. [Online]. Available: http://arxiv.org/abs/2409.07284

[135] B. Rokh, A. Azarpeyvand, and A. Khanteymoori, "A Comprehensive Survey on Model Quantization for Deep Neural Networks in Image Classification," *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 6, pp. 97:1–97:50, nov 14 2023.

[136] B. Li, P. H. Chan, G. Baris, M. D. Higgins, and V. Donzella, "Analysis of Automotive Camera Sensor Noise Factors and Impact on Object Detection," *IEEE Sensors Journal*, vol. 22, no. 22, pp. 22 210–22 219, 11 2022, event-title: IEEE Sensors Journal.

[137] X. Zhao, Y. Fang, H. Min, X. Wu, W. Wang, and R. Teixeira, "Potential sources of sensor data anomalies for autonomous vehicles: An overview from road vehicle safety perspective," *Expert Systems with Applications*, vol. 236, p. 121358, feb 1 2024.

[138] J. Vargas, S. Alsweiss, O. Toker, R. Razdan, and J. Santos, "An Overview of Autonomous Vehicles Sensors and Their Vulnerability to Weather Conditions," *Sensors*, vol. 21, no. 16, p. 5397, aug 10 2021.

[139] H. C. Joshi and S. Kumar, "Artificial Intelligence Failures in Autonomous Vehicles: Causes, Implications, and Prevention," *Computer*, vol. 57, no. 11, pp. 18–30, 11 2024.

[140] A. Rastogi and K. Nygard, "Threats and Alert Analytics in Autonomous Vehicles," pp. 48–35, [Online; accessed 2024-11-25]. [Online]. Available: https://easychair.org/publications/paper/sllG

[141] A. Ceccarelli and F. Secci, "Rgb Cameras Failures and Their Effects in Autonomous Driving Applications," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 4, pp. 2731–2745, 7 2023, event-title: IEEE Transactions on Dependable and Secure Computing.

[142] Y. Li, X. Xu, J. Xiao, S. Li, and H. T. Shen, "Adaptive Square Attack: Fooling Autonomous Cars With Adversarial Traffic Signs," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6337–6347, 4 2021, event-title: IEEE Internet of Things Journal.

[143] M. Everett, B. Lütjens, and J. P. How, "Certifiable Robustness to Adversarial State Uncertainty in Deep Reinforcement Learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 4184–4198, 9 2022, event-title: IEEE Transactions on Neural Networks and Learning Systems.

[144] Z. Ye, J. Cho, and C. Oh, "Improving Image De-Raining Using Reference-Guided Transformers," in *2024 IEEE International Conference on Image Processing (ICIP)*, 10 2024, pp. 1629–1634, iSSN: 2381-8549. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10648113

[145] S. Zheng, C. Lu, Y. Wu, and G. Gupta, "Sapnet: Segmentation-Aware Progressive Network for Perceptual Contrastive Deraining," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 1 2022, pp. 52–62, iSSN: 2690-621X. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9707514

[146] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "Deblurgan: Blind Motion Deblurring Using Conditional Adversarial Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, 6 2018, pp. 8183–8192, [Online; accessed 2024-12-08]. [Online]. Available: https://ieeexplore.ieee.org/document/8578952/

[147] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "Deblurgan-v2: Deblurring (Orders-of-Magnitude) Faster and Better," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, 10 2019, pp. 8877–8886, [Online; accessed 2024-12-08]. [Online]. Available: https://ieeexplore.ieee.org/document/9008540/

[148] W. Guilluy, L. Oudre, and A. Beghdadi, "Video stabilization: Overview, challenges and perspectives," *Signal Processing: Image Communication*, vol. 90, p. 116015, jan 1 2021.

[149] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning - ICML '08*. Helsinki, Finland: ACM Press, 2008, pp. 1096–1103, [Online; accessed 2024-04-16]. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1390156.1390294

[150] U. Hwang, J. Park, H. Jang, S. Yoon, and N. I. Cho, "Puvae: A Variational Autoencoder to Purify Adversarial Examples," *IEEE Access*, vol. 7, pp. 126 582–126 593, 2019.

[151] O. Özdenizci and R. Legenstein, "Restoring Vision in Adverse Weather Conditions With Patch-Based Denoising Diffusion Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 8, pp. 10 346–10 357, 8 2023.

[152] C. Zhang, P. Benz, A. Karjauv, J. W. Cho, K. Zhang, and I. S. Kweon, "Investigating Top-k White-Box and Transferable Black-box Attack," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2022, pp. 15 064–15 073, iSSN: 2575-7075. [Online]. Available: https://ieeexplore.ieee.org/document/9879035

[153] Y. Zhang, A. Carballo, H. Yang, and K. Takeda, "Perception and sensing for autonomous vehicles under adverse weather conditions: A survey," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 196, pp. 146–177, feb 1 2023.

[154] K. Behrendt, L. Novak, and R. Botros, "A deep learning approach to traffic lights: Detection, tracking, and classification," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 5 2017, pp. 1370–1377, [Online; accessed 2023-12-08]. [Online]. Available: https://ieeexplore.ieee.org/document/7989163?denied=

[155] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Vision for Looking at Traffic Lights: Issues, Survey, and Perspectives," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1800–1815, 7 2016, event-title: IEEE Transactions on Intelligent Transportation Systems.

[156] Farama Foundation, "Gymnasium," https://github.com/Farama-Foundation/Gymnasium/blob/main/CITATION.cff, 2024, accessed: 2024-05-31.

[157] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013. [Online]. Available: https://arxiv.org/abs/1312.5602

[158] A. Pighetti, F. Bellotti, C. Oh, L. Lazzaroni, L. Forneris, M. Fresta, and R. Berta, "Investigating Adversarial Policy Learning for Robust Agents in Automated Driving Highway Simulations," in *Applications in Electronics Pervading Industry, Environment and Society*, ser. Lecture Notes in Electrical Engineering, F. Bellotti, M. D. Grammatikakis, A. Mansour, M. Ruo Roch, R. Seepold, A. Solanas, and R. Berta, Eds. Cham: Springer Nature Switzerland, 2024, pp. 124–129.

[159] A. Pighetti, L. Forneris, L. Lazzaroni, F. Bellotti, A. Capello, M. Cossu, A. De Gloria, and R. Berta, "High-level decision-making non-player vehicles," in *Games and Learning Alliance*, K. Kiili, K. Antti, F. de Rosa, M. Dindar, M. Kickmeier-Rust, and F. Bellotti, Eds. Cham: Springer International Publishing, 2022, pp. 223–233.

[160] *International Journal of Serious Games*, vol. 10, no. 4, p. 153–170, Nov. 2023. [Online]. Available: https://journal.seriousgamessociety.org/index.php/IJSG/article/view/638

[161] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, "D4RL: datasets for deep data-driven reinforcement learning," *CoRR*, vol. abs/2004.07219, 2020. [Online]. Available: https://arxiv.org/abs/2004.07219

[162] Y. Omori, Z. Dong, and K. Ross, "Should we ever prefer decision transformer for offline reinforcement learning?" 07 2025.

[163] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[164] N. Mahajan and Q. Zhang, "Intent-aware autonomous driving: A case study on highway merging scenarios," 2023. [Online]. Available: https://arxiv.org/abs/2309.13206

[165] T. Schmied, M. Hofmarcher, F. Paischer, R. Pascanu, and S. Hochreiter, "Learning to modulate pre-trained models in rl," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23. Red Hook, NY, USA: Curran Associates Inc., 2023.

[166] T. M. Inc., "Matlab version: 9.13.0 (r2022b)," Natick, Massachusetts, United States, 2022. [Online]. Available: https://www.mathworks.com

[167] "Ministerial Decree No. 6792 of November 5, 2001 | Ministry of Infrastructure and Transport." [Online]. Available: https://www.mit.gov.it/normativa/decreto-ministeriale-numero-6792-del-05112001

[168] P. M. van Leeuwen, C. G. i Subils, A. R. Jimenez, R. Happee, and J. D. de Winter, "Effects of visual fidelity on curve negotiation, gaze behaviour and simulator discomfort," *Ergonomics*, vol. 58, pp. 1347 – 1364, 2015.

[169] X. Zhao and W. Sarasua, "How to use driving simulators properly: Impacts of human sensory and perceptual capabilities on visual fidelity," *Transportation Research Part C: Emerging Technologies*, 2018.

[170] G. Reymond, A. Kemeny, J. Droulez, and A. Berthoz, "Role of lateral acceleration in curve driving: Driver model and experiments on a real vehicle and a driving simulator," *Human Factors: The Journal of Human Factors and Ergonomics Society*, vol. 43, pp. 483 – 495, 2001.

[171] R. Sojourner and J. Antin, "The effects of a simulated head-up display speedometer on perceptual task performance," *Human Factors: The Journal of Human Factors and Ergonomics Society*, vol. 32, pp. 329 – 339, 1990.

[172] M. M. Ahmed, G. Yang, and S. M. Gaweesh, "Assessment of drivers' perceptions of connected vehicle–human machine interface for driving under adverse weather conditions: Preliminary findings from wyoming," *Frontiers in Psychology*, vol. 11, 2020.

[173] Q. Hussain, W. K. M. Alhajyaseen, A. Pirdavani, N. Reinolsmann, K. Brijs, and T. Brijs, "Speed perception and actual speed in a driving simulator and real-world: A validation study," *Transportation Research Part F: Traffic Psychology and Behaviour*, 2019.

[174] M. Gupta and N. Velaga, "Validation of motorized two-wheeler virtual environment: Influence of perceived realism and simulator fidelity," *Transportation Research Part F: Traffic Psychology and Behaviour*, 2025.

[175] D. Large, E. Crundall, G. Burnett, C. Harvey, and P. Konstantopoulos, "Driving without wings: The effect of different digital mirror locations on the visual behaviour, performance and opinions of drivers." *Applied ergonomics*, vol. 55, pp. 138–148, 2016.

[176] Y. Wang, B. Mehler, B. Reimer, V. Lammers, L. D'Ambrosio, and J. Coughlin, "The validity of driving simulation for assessing differences between in-vehicle informational interfaces: A comparison with field testing," *Ergonomics*, vol. 53, pp. 404 – 420, 2010.

[177] S. Ansari, F. Naghdy, H. Du, and Y. N. Pahnwar, "Driver mental fatigue detection based on head posture using new modified relu-bilstm deep neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 957–10 969, 2022.

[178] Y. Jeong, "Predictive lane change decision making using bidirectional long short-term memory for autonomous driving on highways," *IEEE Access*, vol. 9, pp. 144 985–144 998, 2021.

[179] M. Manzour, A. Ballardini, R. Izquierdo, and M. A. Sotelo, "Vehicle lane change prediction based on knowledge graph embeddings and bayesian inference," in *IEEE Intelligent Vehicles Symposium (IV)*, 2024.

[180] A. Capello, M. Fresta, F. Bellotti, H. Haghighi, J. Hiller, S. Mozaffari, and R. Berta, "Exploiting Big Data for Experiment Reporting: The Hi-Drive Collaborative Research Project Case," *Sensors*, vol. 23, no. 18, p. 7866, 1 2023, number: 18 publisher: Multidisciplinary Digital Publishing Institute.

[181] S. Mozaffari, E. Arnold, M. Dianati, and S. Fallah, "Early lane change prediction for automated driving systems using multi-task attention-based convolutional neural networks," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 758–770, 2022.

[182] A. Cura, H. Küçük, E. Ergen, and İsmail Burak Öksüzoğlu, "Driver profiling using long short term memory (lstm) and convolutional neural network (cnn) methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 6572–6582, 2021.

[183] G. Xie, A. Shangguan, R. Fei, W. Ji, W. Ma, and X. Hei, "Motion trajectory prediction based on a cnn-lstm sequential model," *Science China Information Sciences*, vol. 63, 2020.

[184] D. Salvucci, H. Mandalia, N. Kuge, and T. Yamamura, "Lane-change detection using a computational driver model," *Human factors*, vol. 49, pp. 532–42, 07 2007.

[185] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An Open Urban Driving Simulator," 2017, version: 1. [Online]. Available: https://arxiv.org/abs/1711.03938

[186] R. Izquierdo, A. Quintanar, I. Parra, D. Fernández-Llorca, and M. Sotelo, "The prevention dataset: a novel benchmark for prediction of vehicles intentions," in *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3114–3121.

[187] ASAM e.V., "Asam openscenario 2.0.0," 2023, accessed: Dec. 07, 2023. [Online]. Available: https://www.asam.net/standards/detail/openscenario/v200/

[188] "l3pilot/l3pilot-cdf," aug 15 2023, original-date: 2019-09-10T08:07:33Z. [Online]. Available: https://github.com/l3pilot/l3pilot-cdf

[189] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza, "Dsec: A Stereo Event Camera Dataset for Driving Scenarios," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4947–4954, 7 2021.

[190] l3pilot, "L3pilot common data format (cdf)," https://github.com/l3pilot/l3pilot-cdf, 2023, accessed: Mar. 04, 2024.

[191] T. Ayres, L. Li, D. Schleuning, and D. Young, "Preferred time-headway of highway drivers," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*. Oakland, CA, USA: IEEE, 2001, pp. 826–829, [Online; accessed 2023-05-29]. [Online]. Available: http://ieeexplore.ieee.org/document/948767/

[192] Epic Games, "Unreal engine." [Online]. Available: https://www.unrealengine.com

[193] D. Yang, Z. Liu, W. Jiang, G. Yan, X. Gao, B. Shi, S. Liu, and X. Cai, "Realistic rainy weather simulation for lidars in carla simulator," *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 951–957, 2023.

[194] K. Hara, H. Kataoka, and Y. Satoh, "Learning Spatio-Temporal Features with 3d Residual Networks for Action Recognition," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. Venice: IEEE, 10 2017, pp. 3154–3160, [Online; accessed 2025-01-09]. [Online]. Available: http://ieeexplore.ieee.org/document/8265584/

[195] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: convolutional block attention module," *CoRR*, vol. abs/1807.06521, 2018. [Online]. Available: http://arxiv.org/abs/1807.06521

[196] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *CoRR*, vol. abs/1709.01507, 2017. [Online]. Available: http://arxiv.org/abs/1709.01507

[197] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 2021, pp. 813–824. [Online]. Available: http://proceedings.mlr.press/v139/bertasius21a.html

[198] C. Han, Q. Zhao, S. Zhang, Y. Chen, Z. Zhang, and J. Yuan, "Yolopv2: Better, Faster, Stronger for Panoptic Driving Perception," aug 24 2022, arXiv:2208.11434 [cs]. [Online]. Available: http://arxiv.org/abs/2208.11434

[199] M. Cossu, R. Berta, L. Forneris, M. Fresta, L. Lazzaroni, J.-L. Sauvaget, and F. Bellotti, *YoloP-Based Pre-processing for Driving Scenario Detection*. Cham: Springer Nature Switzerland, 2024, vol. 1110, pp. 418–423, collection-title: Lecture Notes in Electrical Engineering DOI: 10.1007/978-3-031-48121-5_60. [Online]. Available: https://link.springer.com/10.1007/978-3-031-48121-5_60

[200] M. Cossu, R. Berta, M.A. Bagheri Orumi, L. Lazzaroni, J.L. Sauvaget, H. Touil, and F. Bellotti, "Comparing Detection-Based and Motion-Based Preprocessing of Video Frames for Automated Driving Scenario Detection," in *Proceedings of Applications in Electronics Pervading Industry, Environment and Society (Applepies) 2024*, 2024.

[201] Z. Teed and J. Deng, "Raft: Recurrent All-Pairs Field Transforms for Optical Flow," aug 25 2020, arXiv:2003.12039 [cs]. [Online]. Available: http://arxiv.org/abs/2003.12039

[202] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006. [Online]. Available: https://doi.org/10.1016/j.patrec.2005.10.010

[203] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, p. e0118432, 03 2015.

[204] P. F. Suawa, A. Halbinger, M. Jongmanns, and M. Reichenbach, "Noise-Robust Machine Learning Models for Predictive Maintenance Applications," *IEEE Sensors Journal*, vol. 23, no. 13, pp. 15 081–15 092, 7 2023, event-title: IEEE Sensors Journal.

[205] NHTSA, "Automated Driving Systems: A Vision for Safety 2.0," https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/13069a-ads2.0_090617_v9a_tag.pdf. [Online]. Available: https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/13069a-ads2.0_090617_v9a_tag.pdf

[206] "J3016_201401: Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems - SAE International," https://www.sae.org/standards/content/j3016_201401/, [Online; accessed 2023-12-07]. [Online]. Available: https://www.sae.org/standards/content/j3016_201401/

[207] S. Soares, A. Lobo, S. Ferreira, L. Cunha, and A. Couto, "Takeover performance evaluation using driving simulation: a systematic review and meta-analysis," *European Transport Research Review*, vol. 13, no. 1, p. 47, sep 3 2021.

[208] A. Lotfi, S. Hukerikar, K. Balasubramanian, P. Racunas, N. Saxena, R. Bramley, and Y. Huang, "Resiliency of automotive object detection networks on GPU architectures," in *2019 IEEE International Test Conference (ITC)*. Washington, DC, USA: IEEE, 11 2019, pp. 1–9, [Online; accessed 2025-09-03]. [Online]. Available: https://ieeexplore.ieee.org/document/9000150/

[209] E. Ibe, H. Taniguchi, Y. Yahagi, K.-i. Shimbo, and T. Toba, "Impact of Scaling on Neutron-Induced Soft Error in SRAMs From a 250 nm to a 22 nm Design Rule," *IEEE Transactions on Electron Devices*, vol. 57, no. 7, pp. 1527–1538, 7 2010.

[210] C. Ravikumar, "Industrial Practices in Low-Power Robust Design," in *2020 IEEE 26th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 7 2020, pp. 1–4, iSSN: 1942-9401. [Online]. Available: https://ieeexplore.ieee.org/document/9159736

[211] N.T.S.H.A., "Investigation: Pe 21-020," https://static.nhtsa.gov/odi/inv/2021/INOA-PE21020-1893.PDF, aug 13 2021. [Online]. Available: https://static.nhtsa.gov/odi/inv/2021/INOA-PE21020-1893.PDF

[212] N. H. T. S. Administration (NHTSA), "Engineering Analysis EA22-002: Autopilot & First Responder Scenes," https://static.nhtsa.gov/odi/inv/2022/INOA-EA22002-3184.PDF, 6 2022. [Online]. Available: https://static.nhtsa.gov/odi/inv/2022/INOA-EA22002-3184.PDF

[213] H. Bakır and R. Bakır, "Evaluating the Robustness Of YOLO Object Detection Algorithm in terms of Detecting Objects in Noisy Environment," *Journal of Scientific Reports-A*, no. 054, pp. 1–25, sep 30 2023, number: 054 publisher: Kütahya Dumlupinar University.

[214] C. W. Lee, N. Nayeer, D. E. Garcia, A. Agrawal, and B. Liu, "Identifying the Operational Design Domain for an Automated Driving System through Assessed Risk," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 10 2020, pp. 1317–1322, iSSN: 2642-7214. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9304552?

casa_token=Rt8FDceFgCoAAAAA:AmA2PTRKlvtlKXIOSDnw4e-UEAN6Y_PSyaoe3I3eA0z0JOSSxOJZCQWhQpnBpgF-mITkJqA7zc4

[215] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "Deblurgan-v2: Deblurring (Orders-of-Magnitude) Faster and Better," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, 10 2019, pp. 8877–8886, [Online; accessed 2024-12-08]. [Online]. Available: https://ieeexplore.ieee.org/document/9008540/

[216] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954. [Online]. Available: http://www.jstor.org/stable/2333009

[217] R. Killick, P. Fearnhead, and I. A. Eckley, "Optimal detection of changepoints with a linear computational cost," *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1590–1598, 2012. [Online]. Available: https://doi.org/10.1080/01621459.2012.737745

[218] "Jetson Orin Nano Developer Kit Getting Started Guide," https://developer.nvidia.com/embedded/learn/get-started-jetson-orin-nano-devkit, [Online; accessed 2025-07-01]. [Online]. Available: https://developer.nvidia.com/embedded/learn/get-started-jetson-orin-nano-devkit

[219] Y. Zhang and Q. Yang, "An overview of multi-task learning," *National Science Review*, vol. 5, no. 1, pp. 30–43, Jan. 2018. [Online]. Available: https://academic.oup.com/nsr/article/5/1/30/4101432

[220] "Nvidia TensorRT," https://developer.nvidia.com/tensorrt, [Online; accessed 2024-12-10]. [Online]. Available: https://developer.nvidia.com/tensorrt

[221] N. Drenkow and M. Unberath, "A causal framework for aligning image quality metrics and deep neural network robustness," *npj Artificial Intelligence*, vol. 1, 2025.

[222] C. Li, Y. Tian, X. Ling, Z. Zhang, H. Duan, H. Wu, Z. Jia, X. Liu, X. Min, G. Lu, W. Lin, and G. Zhai, "Image quality assessment: From human to machine preference," *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7570–7581, 2025.

[223] Z. Ouyang, J. Niu, Y. Liu, and M. Guizani, "Deep CNN-Based Real-Time Traffic Light Detector for Self-Driving Vehicles," *IEEE Transactions on Mobile Computing*, vol. 19, no. 2, pp. 300–313, 2 2020.

[224] Z. Chen, L. Xie, S. Pang, Y. He, and Q. Tian, "Appending Adversarial Frames for Universal Video Attack," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1 2021, pp. 3198–3207, iSSN: 2642-9381. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9423258

[225] S. Li, A. Neupane, S. Paul, C. Song, S. V. Krishnamurthy, A. K. R. Chowdhury, and A. Swami, "Adversarial Perturbations Against Real-Time Video Classification Systems," in *Proceedings 2019 Network and Distributed System Security Symposium*, 2019, arXiv:1807.00458 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1807.00458

[226] J. Mu, "Practitioner Paper: A Real-Time Defense Against Object Vanishing Adversarial Patch Attacks for Object Detection in Autonomous Vehicles," in *Security and Privacy in Cyber-Physical Systems and Smart Vehicles*, X. Hei, L. Garcia, T. Kim, and K. Kim, Eds. Cham: Springer Nature Switzerland, 2025, pp. 296–307.

[227] Y. Yu, H. J. Lee, H. Lee, and Y. M. Ro, "Defending Person Detection Against Adversarial Patch Attack by Using Universal Defensive Frame," *IEEE Transactions on Image Processing*, vol. 31, pp. 6976–6990, 2022.

[228] F. Sakr, R. Berta, J. Doyle, A. Capello, A. Dabbous, L. Lazzaroni, and F. Bellotti, "Cbin-NN: An Inference Engine for Binarized Neural Networks," *Electronics*, vol. 13, no. 9, p. 1624, 1 2024, number: 9 publisher: Multidisciplinary Digital Publishing Institute.