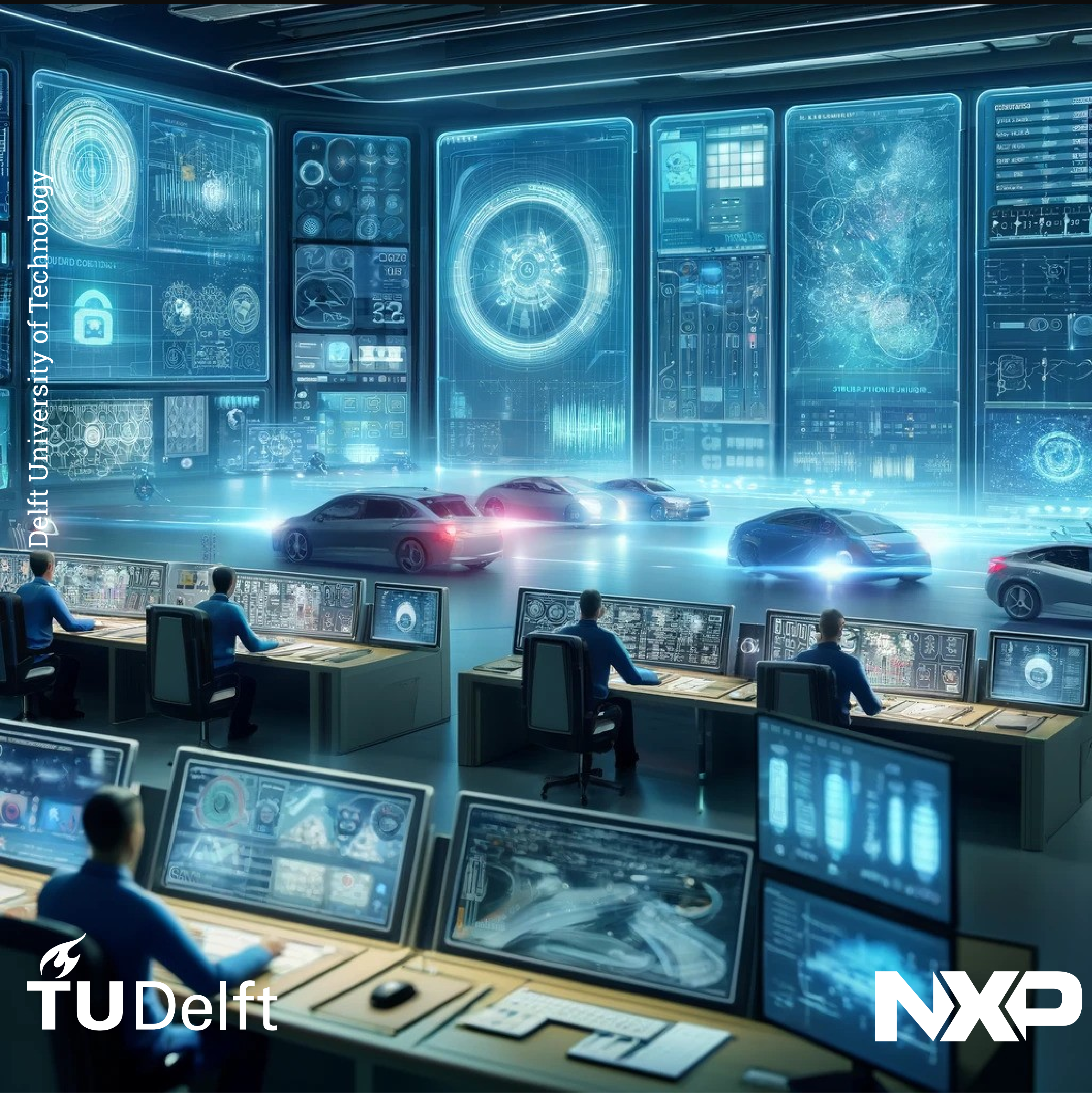
Continuous Improvement of Driving Automation

Using Safety Performance Indicators and Hazardous Scenario Identification

Murali Manohar Selva Kumar



Delft University of Technology

Continuous Improvement of Driving Automation

Using Safety Performance Indicators and Hazardous Scenario Identification

by

Murali Manohar Selva Kumar

Student number:	5559871
Project duration:	November 6, 2023 – June 27, 2024

Thesis Committee

Dr. Ranga Rao Venkatesha Prasad	TU Delft, Supervisor
Dr. Burcu Özkan	TU Delft, External Committee Member
Dr. Andrei Terechko	NXP Semiconductors

An electronic version of this thesis is available at `http://repository.tudelft.nl/`

Cover image created using OpenAI's DALL-E

Quote

"Discovery consists of seeing what everybody has seen and thinking what nobody has thought."
– Albert Szent-Györgyi

Acknowledgements

From my experience, I firmly believe that no one is truly self-made. We all grow and succeed through the support and involvement of those around us.

I am profoundly grateful to my family — my parents, S. Selva Kumar and S. Vijayalakshmi; my sister, Dr. Monisha Devi S. Their unwavering support and sacrifices have been the cornerstone of every opportunity I have had, including this transformative journey through my Master's program. Their encouragement has been my greatest source of motivation. I would also like to thank my uncle, R. Balachandar, for his support and the unique role he has played in my life and education.

I extend my deepest gratitude to Dr. Andrei Terechko, my supervisor at NXP Semiconductors. His comprehensive support and guidance throughout my thesis were invaluable, not just academically but also in terms of personal development. Dr. Terechko inspired me to view the larger picture, enhancing my professional skills and approach to meaningful research and engineering.

I am very thankful to Yuting Fu at NXP Semiconductors for her technical assistance and guidance, as well as for sharing crucial resources like research papers that have shaped my thesis. I also extend my thanks to Kees Schuerman for his insightful feedback, which has clarified the main focus of my work and deepened my understanding. Additionally, I appreciate the enriching discussions with Caspar Hanselaar, PhD candidate at TU Eindhoven, which extended beyond the thesis scope. The contributions from all three have not only influenced my thesis but also significantly enhanced my problem-solving skills and contributed to my growth as an engineer.

I would also like to thank and acknowledge Dr. R. R. Venkatesha Prasad, my supervisor at TU Delft, for his invaluable guidance, support and timely feedback. His readiness to engage with all my queries and to promote exploration across various topics has been motivating.

Finally, I wish to express my heartfelt gratitude to all my friends who have stood by me throughout this journey. Their support, whether through encouraging words or shared moments of relaxation, has been indispensable.

Each person mentioned here has played a significant role in this journey, and I am deeply grateful for their contributions.

Murali Manohar Selva Kumar
Netherlands,June 2024

Abstract

The rapid advancement of automated vehicles (AVs) can potentially improve transportation. However, ensuring the safety and reliability of Automated Driving Systems (ADS) remains a critical challenge, particularly when facing the expansion of Operational Design Domains (ODDs) and the continuous emergence of unknown hazardous scenarios. This thesis aims to address these challenges by developing a framework for monitoring the safety of multi-channel ADS and identifying hazardous scenarios using Safety Performance Indicators (SPIs) and Hazardous Scenario Identification (HSI) techniques.

The proposed SPI framework, based on the principles outlined in the UL 4600 standard, encompasses a comprehensive set of metrics for assessing the safety and performance of ADS. These metrics cover various critical functionalities, such as ego localization, object detection, trajectory planning, and overall ADS behaviour. By defining appropriate thresholds for each SPI, the framework enables the identification of potential safety issues and supports the continuous monitoring and improvement of ADS.

The HSI module, developed as part of this thesis, leverages the SPI framework and the NXP Daruma cross-channel analysis to detect hazardous scenarios. The HSI module's performance is evaluated using the CARLA simulator and advanced ADS software stacks (LAV and TFUSE) across diverse driving scenarios. The results demonstrate the HSI module's effectiveness in identifying hazardous scenarios such as ego vehicle tailgating, inconsistent ego localization, and ego vehicle being tailgated. However, our analysis also reveals challenges in terms of false positives and negatives, highlighting the need for further improvements in the ADS's perception and localization functionalities and in tuning the SPI thresholds appropriately based on testing as well as the characteristics of the ADS.

This thesis contributes to advancing ADS safety by developing a comprehensive SPI framework and implementing a proof of concept HSI module. We propose an architecture that integrates these components in a closed-loop process involving vehicle fleet data collection, cloud-based analysis, and targeted software updates. This framework enables the identification of areas for improvement and supports generating OpenSCENARIO files for reproducing and analyzing hazardous scenarios ad hoc. The findings from the experimental evaluation provide valuable insights into the performance and limitations of the SPI safety monitoring and HSI techniques, guiding the safe deployment and continuous improvement of ADS. This research ultimately paves the way for the widespread adoption of automated vehicles (AVs) in driving environments.

List of Figures

1.1 A few key requirements of Automated Vehicles [10] 2

2.1 SAE Levels of Automation [28] 6
2.2 Koopman's Levels of Automation [30] 7
2.3 Difference between ISO26262 and ISO21448 illustrated through an example [21] 10
2.4 Classification of safe and unsafe scenarios [34] 11

3.1 SPI Based Continuous Improvement Process For Automated Driving 21
3.2 Planned Functional Architecture for SPI and HSI 22

4.1 Illustration of the Fréchet problem's intuition [79] 32
4.2 Illustration of Daruma Cross Channel Analysis [21] 34

5.1 Example scenario where ego vehicle is tailgating another vehicle 40
5.2 Example scenario where ego vehicle is tailgated by another vehicle 41
5.3 Flowchart of SPIs that contribute to the identification of the respective hazardous scenarios 43
5.4 Flowchart depicting the generation of OpenScenario files from Daruma 44

6.1 Simplified architecture of the Daruma setup 45
6.2 Screenshot of the manual HSI identification process on the Daruma multi-channel AD test bed 47
6.3 Map of the CARLA town 1 [85] 48
6.4 The CARLA town 1 map (a) and the path followed by ego vehicle (b) 49
6.5 Example scenario of ADS misclassification 50
6.6 Map of the CARLA Town 2 [86] 52
6.7 The CARLA town 2 map (a) and the path followed by ego vehicle (b) 52
6.8 Map of the CARLA Town 3 [87] 55
6.9 The CARLA town 3 map (a) and the path followed by ego vehicle (b) used for the evaluation 55
6.10 Senario where an object (bicycle) was missed by both the ADS. 56
6.11 Example scenarios where trajectory similarity threshold (a) and object count scores threshold (b) were exceeded 56
6.12 Map of the CARLA Town 4 [88] 59
6.13 The CARLA town 4 map (a) and the path followed by ego vehicle (b) used for the evaluation 59
6.14 Scenario where object count score exceeded the threshold as LAV identified objects on the sidewalk. 60
6.15 Two scenarios where SPI algorithms identified significantly different ADS-guided paths for the vehicle 61
6.16 Scenario where the vehicle collided in Town 4 62
6.17 Map of the CARLA Town 5 [89] 65
6.18 The CARLA town 5 map (a) and the path followed by ego vehicle (b) used for the evaluation 65
6.19 Map of the CARLA Town 6 [90] 68
6.20 The CARLA town 6 map (a) and the path followed by ego vehicle (b) used for the evaluation 68
6.21 Scenario where ego vehicle veered off the road and was stuck for a short period 69
6.22 The map of the sixth town (a) and the path followed by ego vehicle (b) used for the evaluation 69

6.23 Time Taken by the SPI and HSI module in Town 1 73
6.24 Time Taken by the Similarity module in Town 1 73
6.25 Time Taken by the comfort module in Town 1 74
6.26 Time Taken by the motion prediction module in Town 1 74
6.27 Time Taken by the SPI and HSI module in Town 2 75
6.28 Time Taken by the Similarity module in Town 2 75
6.29 Time Taken by the comfort module in Town 2 76
6.30 Time Taken by the motion prediction module in Town 2 76
6.31 SPI Time Analysis for Town 3 77
6.32 Similarity Time Analysis for Town 3 77
6.33 Comfort Time Analysis for Town 3 78
6.34 Motion Prediction Time Analysis for Town 3 78
6.35 SPI Time Analysis for Town 4 79
6.36 Similarity Time Analysis for Town 4 79
6.37 Comfort Time Analysis for Town 4 80
6.38 Motion Prediction Time Analysis for Town 4 80
6.39 SPI Time Analysis for Town 5 81
6.40 Similarity Time Analysis for Town 5 81
6.41 Comfort Time Analysis for Town 5 82
6.42 Motion Prediction Time Analysis for Town 5 82
6.43 SPI Time Analysis for Town 6 83
6.44 Similarity Time Analysis for Town 6 83
6.45 Comfort Time Analysis for Town 6 84
6.46 Motion Prediction Time Analysis for Town 6 84

List of Tables

2.1 Comparison between ISO 26262 and ISO 21448 (SOTIF) 9
2.2 Summary of Safety Frameworks in the AV Industry 17

4.1 Overview of safety case arguments defined for the thesis 24
4.2 Safety Performance Indicators corresponding to each of the arguments of the safety case 25
4.3 Correlation between Safety Performance Indicators and Metrics 26
4.4 Categorised SPI Metrics 27
4.5 Reference SPI Thresholds 36
4.6 Summary of SPI Metrics and Targeted Channels 38

6.1 Summary of routes selected for the experimental evaluation 46
6.2 Results of HSI for CARLA town 1 49
6.3 Results of SPI monitoring for CARLA town 1 51
6.4 Results of HSI for CARLA town 2 53
6.5 Results of SPI monitoring for CARLA town 2 54
6.6 Results of HSI for CARLA town 3 57
6.7 Results of SPI monitoring for CARLA town 3 58
6.8 Results of HSI for CARLA town 4 63
6.9 Results of SPI monitoring for CARLA town 4 64
6.10 Results of HSI for CARLA town 5 66
6.11 Results of SPIs monitoring for CARLA town 5 67
6.12 Results of HSI for CARLA town 6 70
6.13 Results of SPI monitoring for CARLA town 6 71
6.14 SPI Time Statistics 85
6.15 Comfort Time Statistics 85
6.16 Similarity Time Statistics 85
6.17 Motion Prediction Time Statistics 86
6.18 Summary of Hazardous Scenarios, Precision and Success Rates for All CARLA Towns 86
6.19 Summary of SPI Threshold Violations Across All CARLA Towns 87

Contents

Acknowledgements **ii**

Abstract **iii**

List of Figures **v**

List of Tables **vi**

Nomenclature **ix**

Definitions **x**

1 Introduction **1**
1.1 Safety of Driving Automation 1
1.2 Problem Statement 2
1.3 Research Methodology 3
1.4 Thesis Structure 4

2 Related Work **6**
2.1 Levels of Automated Driving 6
2.1.1 SAE Levels of Automation 6
2.1.2 Philip Koopman's Levels of Automation 7
2.2 Safety Standards 8
2.2.1 Automotive Functional Safety Standard (ISO 26262) 8
2.2.2 Safety of Intended Functionality (ISO 21448) 8
2.2.3 Standard for Safety for the Evaluation of Autonomous Products (UL 4600) 9
2.3 Challenges in AV Safety 10
2.4 Operational Design Domain 11
2.4.1 Hazardous Scenarios Identification (HSI) 12
2.5 Redundant Automated Driving Systems 12
2.5.1 Daruma Design Pattern 13
2.5.2 Daruma Cross-Channel Analysis 13
2.6 AV Safety Monitoring 13
2.6.1 Safety Performance Indicators (SPIs) and Safety Case 14
2.7 Fleet Management and Safety Monitoring 15
2.8 State of the art AV safety frameworks 15
2.8.1 Aurora Safety Case Framework 15
2.8.2 Waymo Safety Framework 16
2.8.3 NVIDIA Self Driving Safety Report 16
2.8.4 IVEX Safety Framework 16
2.9 Software Ecosystem 17
2.9.1 CARLA Simulator 17
2.9.2 OpenSCENARIO 17
2.9.3 Open source ADS Software 18
2.9.4 Robot Operating System (ROS) 18
2.10 Related Work Conclusion 19

3 System Architecture **20**
3.1 Architecture for Continuous Improvement of ADS 20
3.2 Functional Architecture of SPI and HSI 21

4 Safety Performance Indicators **24**
4.1 Safety Case 24

4.2 Safety Performance Indicators in a multi-channel ADS . . . 25
4.3 SPI Algorithms . . . 27
4.3.1 Comfort Metrics . . . 27
4.3.2 Similarity Metrics . . . 28
4.3.3 Motion Prediction Metrics . . . 30
4.4 Safety Scores . . . 33
4.5 Thresholds For Defined SPIs . . . 35
4.6 Summary and Overview . . . 38

5 Hazardous Scenario Identification 39
5.1 Hazardous Scenarios . . . 39
5.1.1 Inconsistent Ego Localization . . . 39
5.1.2 Ego vehicle tailgating another vehicle (Ego Tailgating) . . . 40
5.1.3 Ego vehicle being tailgated by another vehicle (Ego Tailgated) . . . 40
5.2 SPIs Contributing to HSI . . . 41
5.2.1 Inconsistent Ego Localization . . . 42
5.2.2 Ego Tailgating . . . 42
5.2.3 Ego Tailgated . . . 42
5.3 OpenScenario export of hazardous scenarios . . . 43

6 Experimental Evaluation 45
6.1 The Experimental Setup . . . 45
6.1.1 Evaluation testbed . . . 45
6.1.2 Evaluation Methodology . . . 46
6.2 Evaluation results and discussion . . . 47
6.2.1 Experimental Evaluation of the Route in Town 1 . . . 48
6.2.2 Experimental Evaluation of the Route in Town 2 . . . 52
6.2.3 Experimental Evaluation of the Route in Town 3 . . . 55
6.2.4 Experimental Evaluation of the Route in Town 4 . . . 58
6.2.5 Experimental Evaluation of the Route in Town 5 . . . 64
6.2.6 Experimental Evaluation of the Route in Town 6 . . . 67
6.3 Profiling Results . . . 71
6.3.1 Profiling Results for the Route in Town 1 . . . 72
6.3.2 Profiling Results for the Route in Town 2 . . . 74
6.3.3 Profiling Results for the Route in Town 3 . . . 76
6.3.4 Profiling Results for the Route in Town 4 . . . 78
6.3.5 Profiling Results for the Route in Town 5 . . . 80
6.3.6 Profiling Results for the Route in Town 6 . . . 82
6.3.7 Statistical Summary . . . 84
6.4 Analysis and Discussion . . . 86
6.4.1 Summary of Results . . . 86
6.4.2 Analysis of False Positives and Negatives . . . 87
6.4.3 SPI Threshold Violations . . . 87
6.4.4 More Hazardous Scenario Identification . . . 87
6.4.5 Insights into the ADS Performance . . . 88
6.4.6 Profiling and Real-Time Performance . . . 88

7 Conclusions 90

8 Future Work 93

References 94

Nomenclature

Abbreviations

Abbreviation	Expansion
AV	Automated Vehicle
ISO	International Organisation for Standardisation
ADS	Automated Driving Systems
AD channel	Automated Driving Channel
ODD	Operational Design Domain
SDV	Software Defined Vehicles
OTA	Over The Air
GSN	Goal Structuring Notation
SPI	Safety Performance Indicators
HSI	Hazardous Scenario Identification

Definitions

Definitions

Automated Driving System (ADS)	The hardware and software that are collectively capable of performing the entire dynamic driving task continuously, regardless of whether it is limited to a specific operational design domain (ODD).
Ego Vehicle	The vehicle equipped with and controlled by the Automated Driving System (ADS), that is the primary focus of the study or analysis.
Hazardous Scenario	A situation or condition that poses a potential risk to the safety of the ego vehicle, its occupants, or other road users, requiring the ADS to take appropriate action to mitigate the risk.
Object	Any entity detected by the ADS in the surrounding environment, including other vehicles, pedestrians, cyclists, and stationary obstacles.
Operational Design Domain (ODD)	The specific conditions under which a given ADS or feature is designed to function, including but not limited to environmental, geographical, and time-of-day restrictions.
Planned Path	The intended trajectory generated by the ADS for the ego vehicle to follow, considering the current environment, traffic conditions, and the vehicle's destination.
Predicted Trajectory	The anticipated future path of an object detected by the ADS, based on the object's current state and motion.
Safety Performance Indicator (SPI)	A quantitative measure used to assess the safety performance of an ADS, providing evidence that the system is operating within acceptable risk levels and supporting the overall safety case.

1 Introduction

1.1. Safety of Driving Automation

The rapid development of automated vehicle (AV) technology has the potential to revolutionize transportation and change the nature of road safety. While AVs and human drivers make different types of mistakes, they also have different strengths and weaknesses. Human drivers can rely on experience, intuition, and contextual understanding to navigate complex situations, but they are also prone to errors caused by factors such as distraction, fatigue, and impairment [1]. In contrast, AVs can maintain constant vigilance and react quickly to detected hazards, but they may struggle in the presence of ambiguous or unexpected events [2] [3].

To illustrate the potential safety benefits of AVs, consider two common hazardous scenarios. In the first scenario, a distracted driver fails to notice a pedestrian crossing the street and collides with them. An AV equipped with advanced sensors and perception algorithms could detect pedestrians and quickly respond by braking or steering to avoid a collision. In the second scenario, a driver under the influence of alcohol fails to maintain their lane and crashes into an oncoming vehicle. An AV, unaffected by alcohol or fatigue, would maintain its lane and avoid the crash altogether. These examples demonstrate how AVs can mitigate the risks associated with human error and reduce the occurrence of common hazardous situations. By minimizing the role of human drivers, AVs have the potential to save countless lives and prevent injuries on the road [4].

However, the current state of AV technology still faces significant challenges despite the potential of increased safety, [5]. It is essential to recognize that AVs are not infallible and still face limitations in many situations, such as adverse weather conditions or complex urban environments. One of the primary concerns surrounding AVs is their ability to handle the wide range of complex and unpredictable situations encountered in real-world driving conditions [4]. AVs must depend on their sensors, algorithms, and decision-making systems to ensure safe operation [6]. This becomes particularly challenging in situations that fall outside the AV's operational design domain (ODD), which refers to the specific conditions and scenarios under which the AV is designed to function safely [7].

As AV technology advances, it is crucial to consider the different levels of autonomy and their implications for road safety. Philip Koopman's classification provides a practical insight into AV safety and operational reliability [8]. His system not only categorizes levels of autonomy by capability but also emphasizes how they ensure safety across different conditions, i.e. the system's ability to handle various driving scenarios and the level of human intervention required. By introducing categories like Driver Assistance, Supervised Automation, Autonomous Operation, and Vehicle Testing, Koopman's classification highlights the fail-operational requirements necessary for AVs. AVs must maintain safety functions even when failures occur in the ADS, closely aligning with real-world demands. Koopman's framework is vital for understanding AV safety in dynamic environments and for developing expanded ODDs that can accommodate both expected and unexpected scenarios [5] [9].

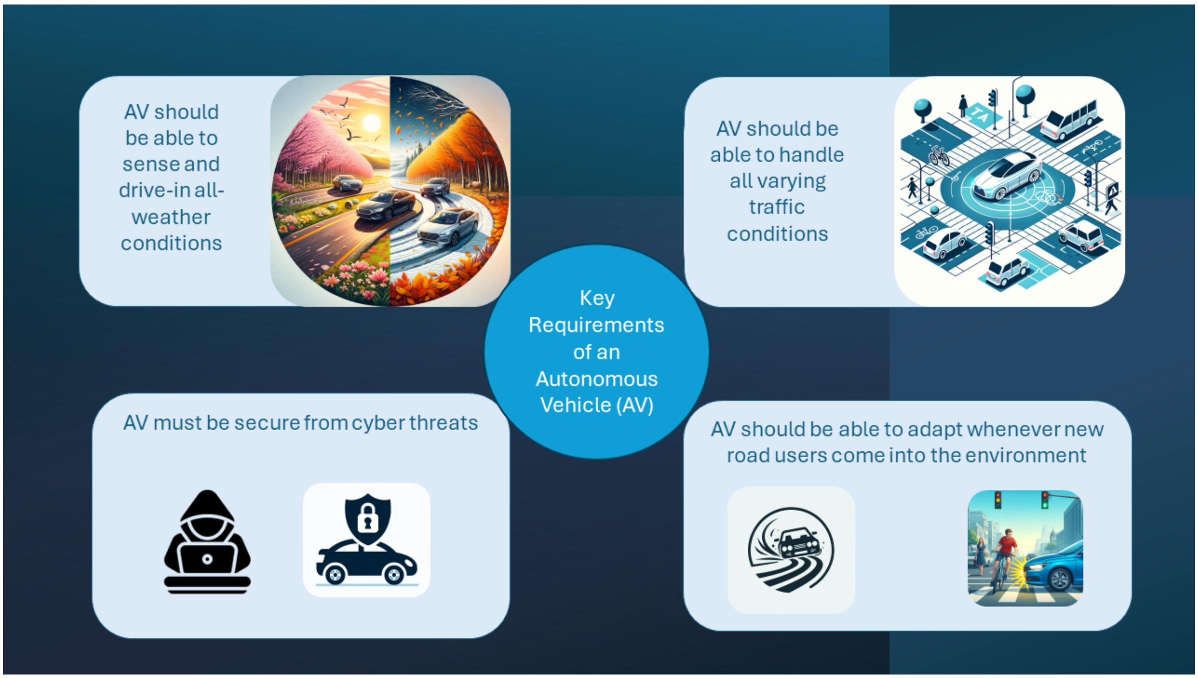


Figure 1.1: A few key requirements of Automated Vehicles [10]

Figure 1.1 shows a few of the key requirements of an AV vehicle. While these may seem trivial, having an AV that can handle all conditions is difficult due to the inherent complexities in the real world.

Currently, ODDs of existing ADS are relatively small, often restricted to certain scenarios, highways, or cities, and are directly correlated with the safety and confidence in the performance of AVs. As the world continues to evolve, with new actors and scenarios emerging in the traffic environment [11], continuous improvement of AV systems is necessary also to ensure that vehicles can correctly react to new scenarios and retain the level of safety achieved at release [12]. Software-defined vehicles with over-the-air updates provide an implementation platform for this continuous improvement [13]. Additionally, redundancy through multiple ADS channels operating in parallel is crucial for fail-safe operation, offering advantages over single-channel systems [14]. Most AVs currently operate on a single nominal channel with a ”cold backup” second channel is activated only when the first one fails [15]. Having two active channels parallely can provide additional benefits and improve overall safety [16].

This thesis explores a method in multi-channel AVs for monitoring and assessing vehicle safety while also contributing to the expansion of ODDs. By leveraging the advantages of redundant ADS channels and continuous improvement, this research aims to address the current limitations of AV safety and pave the way for safer and more reliable AVs.

1.2. Problem Statement

The expansion of ODDs for ADS presents significant challenges. As the world continuously changes, ADS must adapt to new and unforeseen scenarios to avoid misclassifications and potential safety hazards [17]. These challenges are compounded by the fact that analyzing diverse driving scenarios to enhance vehicle safety and performance is a costly and time-consuming process. This process requires extensive data review and monitoring, making it labour-intensive and expensive [18].

Ensuring the safety of AVs is crucial for their widespread adoption and public trust. Real-time safety monitoring is essential for promptly detecting and mitigating hazardous situations. Additionally, there is a need for methods to assess the safety of AVs and validate the claims made by manufacturers at runtime, particularly in the context of multi-channel architectures where multiple perception, planning, and control systems operate simultaneously [19] [16].

The problem statement can be formulated as follows -

The widespread deployment of Automated Driving Systems (ADS) is currently hindered by their inability to adequately handle unknown hazardous scenarios and the continuous emergence of new situations. Additionally, there is a need for effective methods to monitor the safety of AVs in real-time and assess the safety claims made by manufacturers, particularly in multi-channel ADS architectures.

The thesis primarily tries to explore and answer the following questions.

1. How can we monitor the safety of an AV in real-time through a cost-effective method, particularly in systems with redundant AD channels?
   This question investigates the development of efficient and economical strategies for real-time safety monitoring in AVs equipped with redundant channels, focusing on maximizing reliability without significant cost increases.
2. How can redundant channels be utilized to identify and mitigate hazardous scenarios in Automated Driving Systems (ADS)?
   This question explores the use of redundant channels in ADS to enhance the detection and management of hazardous scenarios, thus improving overall system safety and robustness.

Since the methods used and algorithms developed are to be run on an embedded device where computation is time-sensitive and the hardware resource is limited, the design constraints are set as -

1. The algorithms developed should be computationally efficient to avoid delaying the arbitration process in the ADS.
2. The maximum computational budget allocated for the complete module is +25 ms.

In summary, this thesis aims to address the challenges faced by ADS in expanding their ODDs, handling unknown hazardous scenarios, and ensuring real-time safety monitoring. By exploring effective methods for monitoring the safety of multi-channel AVs in real-time and attempting hazardous scenario identification, this research seeks to contribute to the continuous improvement and safe deployment of ADS. The design constraints ensure the proposed solutions are practical and can be integrated into existing ADS without significant performance impacts.

1.3. Research Methodology

This research adopts a comprehensive mixed-methods approach, leveraging both qualitative and quantitative analyses to investigate the efficacy of the adopted strategy for assessing AV safety and identifying hazardous scenarios [20]. The methodology is structured as follows:

- **Literature Review and State-of-the-Art Analysis:** A comprehensive review of current literature and state-of-the-art technologies will focus on AV safety. This literature review will cover various aspects, including SAE and Philip Koopman's levels of automated driving, safety standards such as ISO 26262, ISO 21448 (SOTIF), and UL 4600, and the challenges in AV safety. The concept of operational design domains (ODDs) and the importance of expanding them for the widespread deployment of AVs will also be explored. Additionally, the review will investigate the potential of the NXP Daruma Design Pattern to enhance the AV safety of redundant ADS architectures [21]. Existing approaches to monitoring AV safety employed by various companies will be examined to identify gaps in current knowledge and areas for improvement.
- **Definition of Safety Measures to Address the Problem Statement:** Based on the insights gained from the literature review, specific safety measures tailored to multi-channel ADS architectures will be defined. These measures will include a comprehensive set of SPIs, such as comfort metrics, similarity metrics, and motion prediction metrics, which will be used to assess the safety performance of the AV system. Additionally, techniques for identifying hazardous scenarios, such as inconsistent ego localization, ego tailgating, and ego tailgated, will be explained. These safety measures will be designed to address the challenges of expanding the ODD and ensuring the safe operation of AVs in a wide range of driving conditions.
- **Algorithm Design and Implementation:** Algorithms for monitoring the vehicle's safety will be designed and implemented based on the defined SPIs and hazardous scenario identification tech-

niques. These algorithms will leverage the NXP Daruma Design Pattern [22] for cross-channel analysis and arbitration, enabling the AV system to detect and mitigate potential safety risks. The algorithms will be implemented using C++ and Robot Operating System (ROS2), ensuring compatibility with the NXP Daruma setup and real-world ADS software stacks. The implementation will focus on computational efficiency and real-time performance to meet the design constraints specified in the problem statement.

- **Experimental Testing and Verification:** The developed algorithms will be tested and verified through a series of experiments utilizing the CARLA simulator [23] and real-world ADS software stacks – LAV [24] and TFuse [25]. The CARLA Leaderboard [26], a standardized set of urban driving scenarios for testing ADS performance, will be used to evaluate the effectiveness of the algorithms in identifying hazardous scenarios and assessing the safety performance of the AV system. The simulated routes will cover a range of driving scenarios and weather conditions. The algorithms' performance will be evaluated using both manual analysis and automated processing, allowing for a complete assessment of their accuracy, efficiency, and robustness.
- **Qualitative and Quantitative Analysis:** The results obtained from the experiments will be analyzed using both qualitative and quantitative methods. Qualitative analysis will involve interpreting the identified hazardous scenarios and their implications for AV safety. This analysis will provide insights into the strengths and limitations of the developed algorithms in detecting and mitigating potential safety risks. Quantitative analysis will involve the evaluation of the algorithms' performance by measuring the accuracy and consistency of identified hazardous scenarios. The analysis will validate the effectiveness of the proposed methodology in enhancing AV safety and expanding the ODD.

By following this structured methodology, the research aims to develop a comprehensive and validated approach for monitoring AV safety and identifying hazardous scenarios in multi-channel ADS architectures. The use of state-of-the-art simulation environments, real-world ADS software stacks, and standardized testing scenarios ensure the relevance and applicability of the proposed methodology to the development of safer and more reliable AVs. The combination of qualitative and quantitative analysis methods allows for a thorough evaluation of the developed algorithms and their potential impact on the advancement of AV technology.

1.4. Thesis Structure

Chapter 1 introduces the research topic, providing an overview of the safety challenges that AVs face and the need for continuous improvement in AV systems. The chapter also presents the problem statement, research questions, design constraints, and the research methodology adopted in this thesis.

Chapter 2 comprehensively reviews the related work and state-of-the-art techniques in AV safety, hazardous scenario identification, and safety performance indicators. The chapter also discusses existing approaches to monitoring AV safety and identifies the gaps in current knowledge that this thesis aims to address.

Chapter 3 describes the system architecture developed in this thesis for the continuous improvement of ADS using safety performance indicators (SPIs) and hazardous scenario identification (HSI). The chapter presents the overall architecture for continuous improvement, the functional architecture of the SPI and HSI modules, and the Daruma C++ implementation.

Chapter 4 focuses on developing safety performance indicators (SPIs) for monitoring AV safety. It presents the safety case framework used to define the SPIs, the algorithms and implementation details for calculating them, and a summary of the developed SPIs and their perceived value for cross-channel analysis.

Chapter 5 discusses identifying hazardous scenarios using the developed SPI framework and the Daruma cross-channel analysis. The chapter presents the identified hazardous scenarios, their characteristics, and the automated open scenario generation process used to create test cases for validating the ADS performance.

Chapter 6 presents the experimental evaluation of the developed SPI and HSI frameworks using a test bench setup and in a simulated environment that mimics real-world scenarios. The chapter discusses the results obtained from both manual and automated evaluations and the profiling results for the developed algorithms in terms of latency and CPU load.

Chapter 7 concludes the thesis by summarizing the research's main contributions, findings, and implications. The chapter also discusses the limitations of the current work and provides directions for future research.

Chapter 8 presents the potential future work that can be carried out to extend and improve the developed SPI and HSI frameworks.

2

Related Work

2.1. Levels of Automated Driving

To understand the capabilities and limitations of AVs, it is essential to consider the different levels of automation. These levels provide a standardized framework for describing how an AV system can perform driving tasks without human intervention. The most widely recognized framework for defining levels of automated driving is the SAE International's Levels of Driving Automation. A new Vehicle Safety level has recently been defined by Philip Koopman - a professor at Carnegie Mellon University and a leading expert in AV safety. While both frameworks share some similarities, they differ in their focus and the specific criteria used to define each level.

2.1.1. SAE Levels of Automation

The Society of Automotive Engineers (SAE) International has defined six levels of driving automation, ranging from no automation (Level 0) to full automation (Level 5) [27].

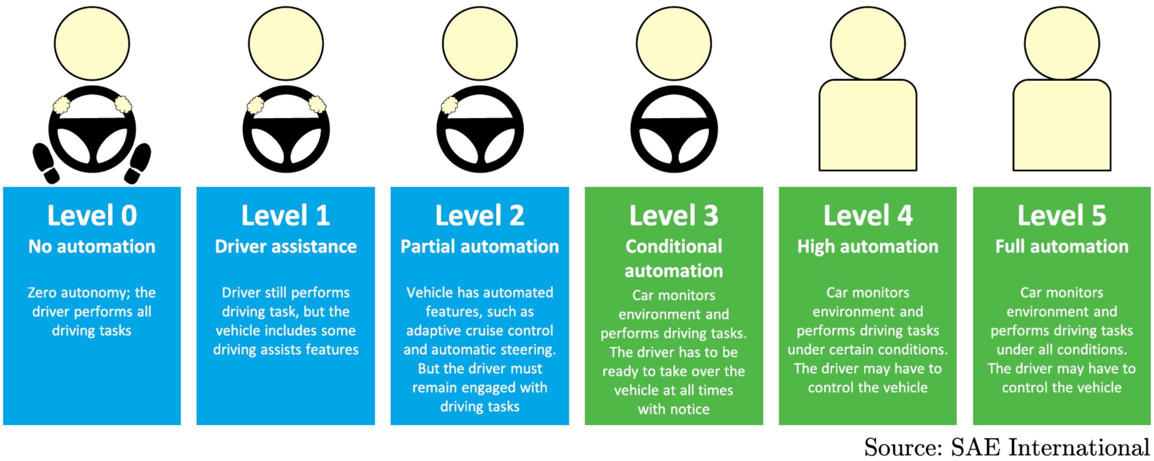


Figure 2.1: SAE Levels of Automation [28]

Figure 2.1 shows the overview of the levels of safety. The levels are:

- Level 0 (No Automation): The human driver is responsible for all driving tasks.
- Level 1 (Driver Assistance): The vehicle can assist with either steering or acceleration/deceleration, but the human driver is responsible for all other driving tasks.

- Level 2 (Partial Automation): The vehicle can control both steering and acceleration/deceleration under specific conditions, but the human driver must monitor the environment and be ready to take control at any time.
- Level 3 (Conditional Automation): The vehicle can perform all driving tasks under specific conditions, but the human driver must be ready to take control when requested by the system.
- Level 4 (High Automation): The vehicle can perform all driving tasks under most conditions without requiring human intervention, but the human driver may be able to take control.
- Level 5 (Full Automation): The vehicle can perform all driving tasks under all conditions without human intervention.

The SAE levels provide a useful framework for understanding the progression of AV capabilities and the roles and responsibilities of the human driver at each level.

2.1.2. Philip Koopman's Levels of Automation

Philip Koopman's classification of autonomy levels provides a clear framework for understanding the operational reliability and safety of AVs. Koopman's framework emphasizes on the ODDs and the safety requirements for each level rather than focusing solely on the division of driving tasks between the human and the AV system [29] [8]. This classification helps assess how AVs perform under various driving conditions and is integral to their safe deployment and public acceptance.

Koopman's model introduces specific categories of vehicle automation (as seen in Figure 2.2), each emphasizing different levels of automation and human interaction requirements:

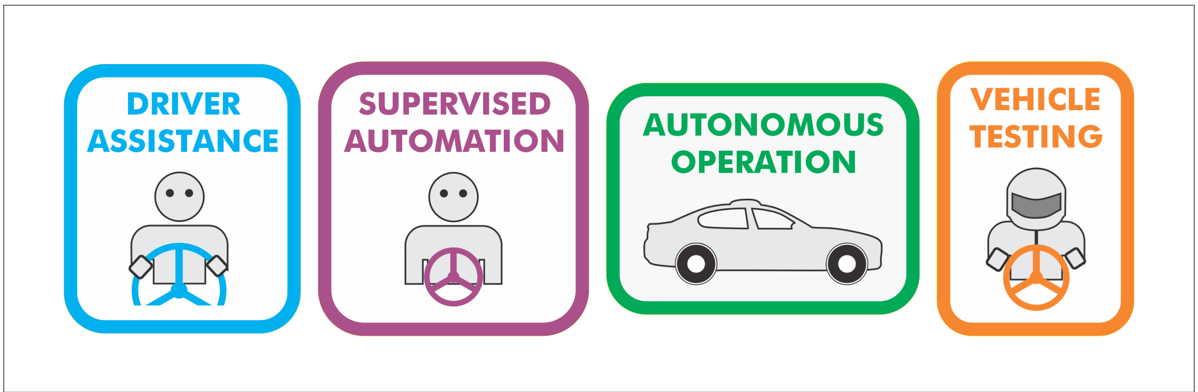


Figure 2.2: Koopman's Levels of Automation [30]

- **Driver Assistance**: In this mode, the vehicle provides support through active safety and convenience features like anti-lock brakes and adaptive cruise control. The human driver remains in control, with the technology designed to enhance the driver's ability to operate the vehicle safely.
- **Supervised Automation**: Here, the vehicle handles tasks like speed and lane keeping, while the human driver monitors the driving environment and intervenes when the vehicle encounters scenarios outside its capabilities. This mode requires effective driver monitoring systems to ensure that the human driver remains engaged and ready to take over as needed.
- **Autonomous Operation:** This mode represents a significant shift as the vehicle operates completely independently of a human driver. It includes comprehensive responsibility for all driving tasks and safety-related functions, capable of handling entire driving missions within its operational design domain (ODD).
- **Vehicle Testing:** Aimed at ensuring safe development and deployment, this mode involves a trained safety driver who oversees the operation of AVs, particularly when testing new and immature technologies. This mode highlights the importance of rigorous safety standards and testing protocols to manage the potential risks associated with automation technologies.

Koopman's classification emphasizes a practical and user-centric approach to vehicle automation. By redefining these roles, the classification aims to mitigate the confusion prevalent in public discussions and media portrayals of AV capabilities. It also addresses legal aspects such as driver liability in different modes, providing clear guidelines on responsibilities and expectations. This framework is relevant for consumers and manufacturers and essential for policymakers and regulators in crafting informed and effective legislation for AV technologies [31].

2.2. Safety Standards

Safety standards are crucial in ensuring the safe development, testing, and deployment of AVs. These standards provide guidelines and requirements for various aspects of AV safety, including functional safety, safety of intended functionality, and the evaluation of autonomous products [32].

2.2.1. Automotive Functional Safety Standard (ISO 26262)

The International Standard for Standardisation (ISO) 26262, titled " Road vehicles – Functional safety", - focuses on the functional safety of electrical and electronic systems in vehicles, including AVs [33]. This standard covers the entire lifecycle of these systems, from development to decommissioning, and provides guidelines for risk assessment, hazard analysis, and safety measures. Automotive companies extensively cover and track ISO 26262 compliance to ensure the functional safety of their vehicles. However, this standard does not address the safety challenges arising from the intended functionality of AVs, which are covered by ISO 21448 (SOTIF).

The main focus of the thesis is on Safety of the Intended Functionality (SOTIF), which is explained in subsequent sections.

2.2.2. Safety of Intended Functionality (ISO 21448)

The International Standard for Standardisation (ISO) 21448 titled "Road vehicles — Safety of the intended functionality", also known as Safety of the Intended Functionality (SOTIF), addresses the safety challenges arising from the intended functionality of AVs, particularly in situations where the system operates correctly but still poses risks due to functional insufficiencies [34]. For example, an AV might fail to detect a pedestrian in low-light conditions, even though its sensors function as intended. ISO 21448 provides guidance on identifying and mitigating these risks not covered by ISO 26262 [35].

Table 2.1: Comparison between ISO 26262 and ISO 21448 (SOTIF)

Feature	**ISO 26262**	**ISO 21448 (SOTIF)**
Focus	Focuses on ensuring that automotive systems are free from unreasonable risk due to hazards caused by malfunctioning behaviour of E/E systems.	Addresses the safety of intended functionality and the correct performance of E/E systems where performance is not caused by system malfunctions.
Scope	Covers all lifecycle phases of safety-related systems comprising electrical, electronic, and software components.	Complements ISO 26262 by addressing hazards resulting from functional insufficiencies or foreseeable misuse that are not considered to be malfunctions.
Objective	Prevention of failures and protection against system malfunctions.	Ensuring safety under conditions where the system behaves as intended, but these behaviours still lead to unsafe situations.
Methodology	Risk assessment through Hazard Analysis and Risk Assessment (HARA), leading to the definition of Safety Integrity Levels (ASILs).	Includes methodologies like scenario analysis to assess situations where intended functions might lead to hazardous events.
Application	Primarily applicable to the automotive industry for vehicles and their subsystems.	Particularly significant for advanced driver-assistance systems (ADAS) and AVs that rely heavily on complex sensors and algorithms.

From Table 2.1, it can be seen that while both standards are crucial for the automotive industry, ISO 21448 holds particular relevance for systems like ADAS and AV technologies that rely on complex interactions of software and hardware to interpret and respond to real-world conditions.

This thesis focuses primarily on the functional insufficiencies defined in SOTIF, as these are more relevant to the challenges faced by multi-channel AV architectures. Functional insufficiencies can arise from limitations in the AV's perception, decision-making, or control systems, as well as from the complexity of the operating environment. By addressing these insufficiencies through effective safety monitoring and hazardous scenario identification, this thesis aims to contribute to the development of safer and more reliable AVs.

2.2.3. Standard for Safety for the Evaluation of Autonomous Products (UL 4600)

The UL 4600 standard provides a comprehensive framework for the safety evaluation of autonomous products, including AVs [36] [37]. This standard emphasizes the importance of safety cases, which provide a structured argument for the safety of an AV system supported by evidence. UL 4600 also introduces the concept of Safety Performance Indicators (SPIs), which are quantitative measures used to assess the safety performance of an AV system.

Although UL 4600 is highly relevant to the development and evaluation of SPIs in the context of AV safety, it is important to note that this thesis was not directly influenced by the standard. The principles and guidelines outlined in UL 4600 serve as a useful reference point for understanding the role of SPIs in assessing AV safety, but the specific approaches and methods employed in this thesis were developed independently.

Nevertheless, the concepts presented in UL 4600, such as the importance of considering the operational design domain (ODD) when evaluating AV safety, align well with the objectives of this thesis. The focus on expanding the ODD of AVs through effective safety monitoring and hazardous scenario identification is consistent with the principles outlined in the standard. By addressing the challenges posed by functional insufficiencies and the complexity of real-world operating environments, this thesis aims to contribute to the safe deployment of AVs across a wider range of conditions, which is a key goal of UL 4600.

Figure 2.3 shows the differences between ISO 26262 and ISO 21448. The simplified hazard models in ISO 26262 (blue) and SOTIF (yellow) illustrate internal and external vehicle events, separated by a grey box. Examples are provided in brackets [21].

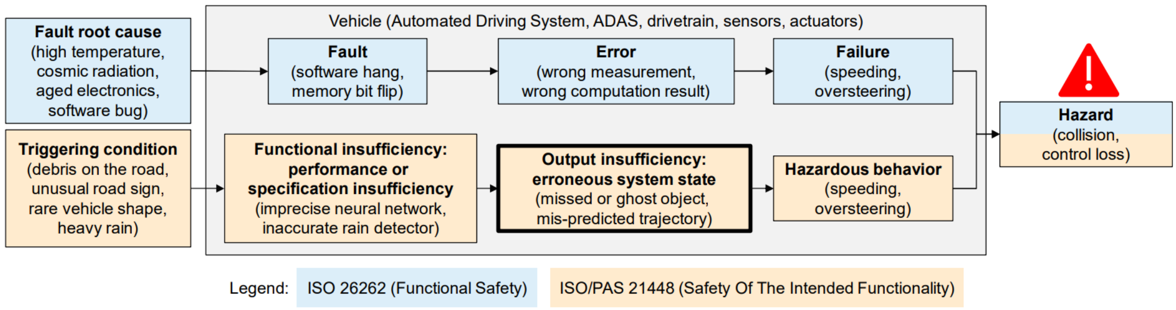


Figure 2.3: Difference between ISO26262 and ISO21448 illustrated through an example [21]

In summary, while ISO 26262 focuses on functional safety, this thesis primarily addresses the safety challenges covered by ISO 21448 (SOTIF) and the principles outlined in UL 4600. By focusing on functional insufficiencies, developing effective SPIs, and leveraging the safety case framework, this thesis aims to contribute to the development of safer and more reliable multi-channel AV architectures capable of operating in expanded ODDs.

2.3. Challenges in AV Safety

AVs face unique safety challenges that go beyond the scope of traditional automotive safety standards. These challenges arise from the complex interaction between the AV's sensors, algorithms, and decision-making systems and the unpredictable nature of real-world driving conditions [29] [21]. One of the key challenges is sensor reliability and perception accuracy. AVs rely on a combination of sensors, such as cameras, lidars, and radars, to perceive their environment. However, various factors can affect these sensors, such as weather conditions, lighting, and road surface quality, which can lead to inaccurate or incomplete perception of the surroundings [6]. This, in turn, can result in incorrect decision-making by the AV's algorithms.

Another challenge is the complexity of the decision-making algorithms themselves. AVs must be able to handle a wide range of scenarios, including those that are not encountered during simulation or training. This requires sophisticated algorithms that can adapt to novel situations and make safe decisions in real time [12]. Ensuring the robustness and reliability of these algorithms is a significant challenge, as they must be extensively tested and validated under various conditions.

In addition to these challenges, AVs must also contend with the potential for functional insufficiencies, as addressed by the ISO 21448 (SOTIF) standard. Functional insufficiencies occur when an AV system operates as intended but still poses risks due to limitations in its perception, decision-making, or control capabilities [34]. For example, an AV might struggle to detect pedestrians in low-light conditions or fail to anticipate the behaviour of other road users in complex traffic scenarios [17].

Other challenges in AV safety include the need for effective human-machine interaction, particularly in the context of handovers between the AV and the human driver [38], the management of ethical dilemmas in emergencies [1], and the establishment of legal and regulatory frameworks that can keep pace with the rapid development of AV technologies [39].

Cybersecurity is also a critical concern for AV safety, as the increasing connectivity and complexity of these systems make them potential targets for cyber attacks [40]. Hackers could potentially manipulate an AV's sensors, control systems, or communication channels, leading to dangerous situations on the road. Addressing cybersecurity risks requires implementing robust security measures, such as encryption, authentication, and intrusion detection systems, as well as regular security updates and patches [41].

This thesis focuses primarily on addressing the challenges posed by functional insufficiencies, as these are critical to ensuring the safety and reliability of AVs in real-world operating conditions. By developing effective methods for monitoring AV safety, identifying hazardous scenarios, and expanding the operational design domain (ODD), this research aims to contribute to the development of safer and more capable AVs that can handle the complexities of real-world driving.

2.4. Operational Design Domain

ODD is a critical concept in the development and deployment of AVs. An ODD refers to the specific conditions and scenarios under which an AV is designed to operate safely [27]. These conditions can include factors such as geography, road type, weather, and traffic density.

Currently, ODDs for AVs are relatively limited, often restricted to specific scenarios, highways, or cities [9]. This is due to the challenges associated with ensuring the safety and reliability of AVs in more complex and unpredictable environments. The size and scope of an AV's ODD are directly correlated with the safety and confidence in the vehicle's performance [5].

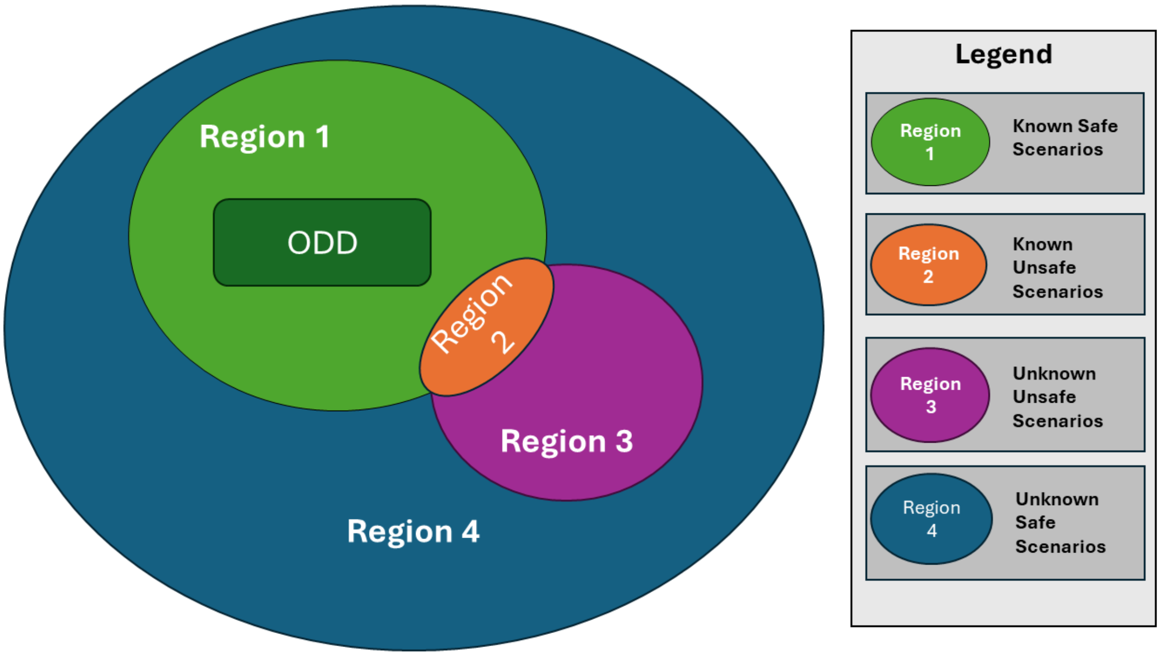


Figure 2.4: Classification of safe and unsafe scenarios [34]

The ODD can be visualized using a four-quadrant model as shown in Figure 2.4, categorising scenarios based on their potential hazards and the AV's ability to handle them [42]. The four quadrants are:

1. Known Unsafe scenarios or Known hazards: Scenarios that are known to be hazardous and can be handled by the AV.
2. Unknown Unsafe scenarios or Unknown hazards: Scenarios that are hazardous but not known or predictable by the AV.
3. Known Safe scenarios or Known non-hazards: Scenarios that are known to be safe and can be handled by the AV.
4. Unknown Safe scenarios or Unknown non-hazards: Scenarios that are safe but not known or predictable by the AV.

Expanding an AV's ODD primarily involves increasing the "known non-hazards" quadrant by identifying and addressing potential hazards through testing, validation, and continuous improvement of the AV's perception, decision-making, and control systems [12]. However, expanding the ODD alone is insufficient; improvements in the AV's overall capabilities are also necessary to move hazardous scenarios into the non-hazardous category.

Continuous improvement of AVs is essential for both expanding ODDs and ensuring that the vehicles can adapt to the ever-changing road environment. As new scenarios and edge cases are encountered, AV developers must update and refine their systems to maintain and improve safety performance. This process can be facilitated by software-defined vehicles (SDVs) with over-the-air (OTA) update capabilities, enabling the continuous improvement of AV functionalities without requiring physical hardware changes.

2.4.1. Hazardous Scenarios Identification (HSI)

In this thesis, the process of identifying hazardous scenarios is called HSI (Hazardous Scenario Identification). Hazardous scenarios are situations that pose a potential risk to the safety of the AV, its occupants, or other road users. These scenarios can arise due to various factors, such as adverse weather conditions, complex road layouts, or unexpected behaviour of other traffic participants [43]. Identifying and characterizing hazardous scenarios is crucial for assessing the safety performance of AVs and defining and expanding their ODDs. However, the definition of a hazardous scenario can vary depending on the context and the specific AV system. Generally, a scenario is considered hazardous if it challenges the AV's perception, decision-making, or control capabilities and requires the AV to take action to avoid or mitigate potential harm or take care of consideration before decision-making [44].

Some examples of hazardous scenarios include:

- Sudden braking of a lead vehicle
- Pedestrian unexpectedly crossing the road
- Adverse weather conditions (e.g., heavy rain, fog, or snow) reducing visibility and traction
- Complex or ambiguous road layouts (e.g., roundabouts, intersections with obstructed views)
- Aggressive or erratic behaviour of other traffic participants

By characterizing and identifying these hazardous scenarios, AV developers can focus their efforts on improving the system's capabilities to handle these situations safely and efficiently, ultimately expanding the ODD and enhancing overall safety performance.

2.5. Redundant Automated Driving Systems

Redundancy in Automated Driving Systems (ADS) is crucial for ensuring fail-operational behaviour and enhancing overall safety [34]. Our project experiments with vehicles with multiple ADS running in parallel, leveraging the concept of redundancy to improve the robustness and reliability of ADS [45]. There are several key advantages to having multiple ADS channels in a vehicle, some of which are:

1. **Enhanced safety:** Redundant AD channels can enhance safety, as the ground truth is unknown. By comparing the outputs of multiple channels, the system can identify potential errors or inconsistencies, increasing the overall confidence in the vehicle's decision-making [46] .
2. **Failover capability:** The cost of activating a redundant ADS channel when a current one fails is too high. Multiple parallel channels ensure seamless failover in case of a single channel failure, minimizing the impact on the vehicle's performance and safety [34].
3. **Improved perception:** While adding more sensors to a particular ADS may not always improve perception and can even work against it, having multiple ADS channels with different sensor configurations can provide a more comprehensive understanding of the environment [47].
4. **Specialized training:** Each ADS channel can be trained for specific use cases or operating conditions, allowing the vehicle to switch between channels based on the current scenario. This approach can help improve the vehicle's overall performance and confidence in handling diverse driving situations [48].

5. **Continuous validation:** Some ADS channels may require validation before fully deploying. Running these channels in parallel with existing, proven ADS allows for real-world testing and validation without compromising the vehicle's safety [49].

However, one challenge with having multiple ADS channels is determining which driving channel should control the vehicle under different conditions. This is where Daruma Design Pattern comes into play.

2.5.1. Daruma Design Pattern

The Daruma Design Pattern, developed by NXP Semiconductors, is a novel approach to improve both the safety and availability of Automated Driving Systems (ADS) [45]. The design pattern leverages the concept of redundancy by running multiple Automated Driving (AD) channels in parallel, enabling the system to select the optimal channel for each driving scenario. The core of the Daruma Design Pattern is the cross-channel analysis of the outputs of the different AD channels to ensure safe and consistent vehicle behaviour. The AD channels provide high-level channel state information to the Daruma module, including the world model, motion predictions, ego trajectory, and detected traffic rules [21].

2.5.2. Daruma Cross-Channel Analysis

The cross-channel analysis in Daruma evaluates the similarity and consistency of the decisions made by each AD channel. By analyzing the agreement between channels, the Daruma Design Pattern can determine the confidence level of the overall system and make informed decisions on which channel should be in control of the vehicle [45].

The Daruma cross-channel analysis employs two major types of algorithms: risk analysis and preferences. The analysis produces various safety and availability metrics that are fused with the output of classical fault monitors, ODD monitors, and other safety mechanisms. The aggregated score per channel influences the high-level arbiter's decision on which channel to select for driving at runtime [45]. Furthermore, Daruma is uniquely positioned in the ADS to monitor safety. By correlating the channels' world models and motion plans, Daruma can identify low safety performance and spot functional insufficiencies, such as repeated disagreements in object classifications between the main and backup channels. The data associated with driving scenarios can be uploaded to the cloud for fleet-wide analysis, enabling the creation of Over-The-Air updates to improve the ADS performance [45].

The first MATLAB [50] implementation of the Daruma Design Pattern, called Safety Shell, has been validated using the CARLA simulator and multiple heterogeneous AD channels with TU/e [16]. In our work, however, the NXP C++ implementation of the Daruma Design Pattern will be used, which has a different set of algorithms.

2.6. AV Safety Monitoring

Monitoring the safety of AVs is crucial for ensuring their reliable operation and building public trust in the technology. There are several approaches to monitoring AV safety, each with its own focus and methodology [5].

Real-Time Diagnostics and Monitoring: This approach focuses mainly on the functional safety aspects covered by the ISO 26262 standard. It involves continuously monitoring the AV's systems and components to detect and diagnose potential failures or malfunctions in real-time [51]. While these extensively cover ISO 26262, they do not solve the hazards faced by ISO21448.

Data-Driven Analysis: This approach relies on collecting and analysing data from AV operations, including incident reports, telemetry, and operational data. By analyzing this data, engineers can identify patterns, trends, and potential safety issues, which can then be addressed through updates and improvements to the AV system [52]. While this may cover real-time scenarios, it is a time-consuming and cumbersome process. All, if not most, of the analysis is done after the AV navigation is completed, and hence, no real-time monitoring and assistance to AV vehicles is possible.

Scenario-based testing (SBT): This is a method used to validate the safety and efficiency of AVs and ADAS. This approach involves a database of predefined scenarios for controlled simulation or real-world testing of AVs or ADAS, offering a systematic way to assess these systems under various driving conditions, including edge cases [53]. SBT advantages include reproducibility, systematic coverage of potential conditions, and safety due to controlled environment testing [54]. The main challenge or disadvantage of SBT is that a comprehensive scenario database must be maintained and updated as driving conditions and technology evolve, necessitating frequent updates [55]. Despite its thoroughness, SBT cannot fully replicate the complexity of real-world environments, making it necessary to integrate other testing methods for a comprehensive evaluation.

Key Performance Indicators (KPIs): KPIs are quantitative metrics used to evaluate the performance of AVs, including aspects of safety, perception accuracy, decision-making quality, and control stability [56]. Although these indicators address safety, their coverage is not extensive enough to ensure safe operation in all scenarios [57].

Safety Performance Indicators (SPIs): SPIs are a specific type of KPI that focuses on measuring the safety performance of an AV system. Unlike traditional KPIs, SPIs are directly linked to safety goals and are used to demonstrate that an AV system is operating within acceptable risk levels [37]. More about SPIs in Section 2.6.1.

While the above approaches address the functional safety aspects covered by ISO 26262, they do not adequately cover the safety of intended functionality (SOTIF) aspects addressed by ISO 21448. This is where SPIs, as mandated by the UL 4600 standard, are gaining traction [58]. AVSC best practice documents highlight the importance of monitoring AV safety [59] [60] [61] [62], and SPIs provide a more comprehensive and systematic approach to monitoring AV safety, particularly in the context of SOTIF. By defining and monitoring SPIs that are directly linked to safety goals and argumentation, AV developers can demonstrate that their systems are operating safely and identify areas for improvement.

In this thesis, we focus on the use of SPIs for monitoring AV safety. They provide a promising approach for addressing the challenges associated with SOTIF and enabling the safe deployment of AVs.

2.6.1. Safety Performance Indicators (SPIs) and Safety Case

Safety Performance Indicators (SPIs) are quantitative measures used to assess the safety performance of an AV system. They are closely tied to the concept of a safety case, which is a structured argument, supported by evidence, that a system is acceptably safe for a given application in a given operating environment. In a safety case, SPIs serve as the evidence that demonstrates the achievement of safety goals and the effective management of safety risks. Each SPI is linked to one or more safety goals and is used to monitor the system's performance against those goals[63] [64].

A safety case typically consists of several levels of argumentation, starting with high-level safety goals and breaking them down into more specific sub-goals and evidence. SPIs are usually associated with the lower levels of the safety case, providing tangible evidence that the system is operating within acceptable risk levels [37].

Some examples of SPIs in the context of AVs include:

- **Perception accuracy:** The percentage of correctly detected and classified objects in the AV's environment.
- **Decision-making quality:** The percentage of decisions that comply with traffic rules and prioritize safety.
- **Control stability:** The deviation from the AV's planned trajectory to the AV's actual (travel) trajectory.
- **Incident frequency:** The number of safety-related incidents (e.g., near-misses, disengagements) per mile travelled.

By defining and monitoring SPIs, AV developers can ensure that their systems meet safety objectives and identify areas for improvement. SPIs also provide a way to communicate the safety performance of an AV system to stakeholders, such as regulators, insurance companies, and the public [63].

The benefits of using SPIs in a safety case include:

- Improved transparency and accountability in the safety management process.
- Objective and quantifiable evidence of safety performance.

- Continuous monitoring and improvement of safety performance.
- Facilitation of safety certification and regulatory compliance.

In this thesis, we explore the application of SPIs within the context of multi-AD channel architectures, leveraging the Daruma framework's cross-channel analysis capabilities to enhance the safety monitoring and argumentation process.

2.7. Fleet Management and Safety Monitoring

Fleet management is a crucial aspect of deploying AVs on a large scale. It involves coordinating, operating, and maintaining a fleet of AVs to ensure their safe, efficient, and reliable performance. As AV technology advances and more companies begin to deploy fleets of AVs, the importance of effective fleet management and safety monitoring strategies becomes increasingly evident. One of the primary challenges in AV fleet management is ensuring the safety of the vehicles across a wide range of operating conditions and scenarios. This requires continuous monitoring of the vehicles' performance, identification of potential safety issues, and timely interventions to mitigate risks.

Several companies (such as Waymo [65], Aurora [66]) have highlighted the potential benefits of fleet management for AVs [67]. One key advantage is the ability to share data and insights across multiple vehicles, accelerating the development of safer and more reliable systems through collaborative learning. By pooling data from different AVs and analyzing common issues, fleet operators can prioritize targeted safety improvements and disseminate learnings throughout the fleet [65]. Additionally, frameworks have been proposed for continuous fleet-wide risk assessment and management. These involve monitoring AV performance data, identifying high-risk scenarios, and adapting control strategies based on real-world observations [68]. Such approaches can help fleet operators proactively address safety concerns and reduce overall risk levels.

However, a universal framework that analyses fleet-level information is still lacking, and hence, this thesis proposes a framework that enables the continuous improvements of ADS.

2.8. State of the art AV safety frameworks

This section discusses real-life examples of Safety monitors, safety case frameworks and safety performance indicators (SPIs) used by companies in the AV industry. The examples include Aurora, Waymo, NVIDIA, and IVEX. It is to be noted that all the companies stated in the subsequent sections focus on single channels AVs that do not leverage the benefits obtained from multichannel architectures as discussed in Section 2.5.

2.8.1. Aurora Safety Case Framework

Aurora, a leading self-driving vehicle company, has developed a comprehensive safety case framework for AVs. The framework, called the Safety Case Framework, is designed to provide a structured and transparent approach to demonstrating the safety of Aurora's AV system [66].

The Aurora Safety Case Framework is based on the Goal Structuring Notation (GSN) and consists of several layers of argumentation, from high-level safety goals to specific evidence and SPIs. The framework covers various aspects of AV safety, including perception, decision-making, control, and overall system performance.

One of the key features of the Aurora Safety Case Framework is its emphasis on continuous monitoring and updating. As the AV system evolves and new data becomes available, the safety case is updated to reflect the system's current state and ensure safety goals are met.

While the Aurora Safety Case Framework represents a significant step forward in systematically demonstrating AV safety, it is important to note that it primarily focuses on single-channel ADS architectures. The framework does not explicitly address the challenges and opportunities associated with multi-channel ADS architectures. Furthermore, it is not known publicly if the health and internal capabilities of the ADS are also monitored through its safety case. The publicly known information shows that only the external factors (such as safety distance, vehicle diagnostics, and object classification accuracy) are monitored.

2.8.2. Waymo Safety Framework

Waymo, a subsidiary of Alphabet Inc., has developed a comprehensive safety framework for its AVs. The framework, called the Waymo Safety Framework, outlines the company's approach to ensuring the safety of its AV throughout their development and deployment [69].

The Waymo Safety Framework consists of several key components, including:

1. Extensive testing and validation of AV in both simulated and real-world environments.
2. Continuous monitoring and analysis of AV performance data to identify potential safety issues.
3. Collaboration with stakeholders, such as regulators, industry partners, and the public, to promote transparency and build trust in AV technology.
4. Investment in research and development to advance AV safety technologies and methodologies.

One of the unique aspects of the Waymo Safety Framework is its focus on what the company calls "behavioural safety." This refers to the AV system's ability to make safe decisions and navigate complex traffic scenarios in a manner that is consistent with human expectations and social norms [69].

While the Waymo Safety Framework represents a comprehensive approach to AV safety, it primarily focuses on the operational aspects of AV deployment rather than the detailed technical implementation of safety monitoring and argumentation.

2.8.3. NVIDIA Self Driving Safety Report

NVIDIA has developed a safety framework for AVs that leverages the company's expertise in artificial intelligence and deep learning [70].

One of the key components of NVIDIA's approach to AV safety is the use of camera-based perception systems. By using deep learning algorithms to process and interpret camera data, NVIDIA aims to achieve high levels of perception accuracy and robustness. However, the reliance on camera-based perception also presents challenges in terms of computational complexity and data management. Processing and analyzing the vast amount of data generated by camera sensors requires significant computational resources, which can be expensive and energy-intensive [71].

Another aspect of NVIDIA's safety framework is using simulation environments to train and validate AV. By using simulation, NVIDIA can expose its AV to a wide range of scenarios and edge cases, including hazardous situations that would be difficult or dangerous to test in the real world. However, the process of identifying and annotating hazardous scenarios in simulation data can be time-consuming and labour-intensive. Engineers must manually review and label the data to ensure the AV system is trained and validated on relevant and representative scenarios.

While NVIDIA's safety framework represents a powerful approach to AV safety, it is primarily focused on single-channel ADS architectures with labour-intensive post-processing.

2.8.4. IVEX Safety Framework

IVEX, a provider of AV testing and validation solutions, has developed an approach to AV safety that focuses on scenario-based testing and validation using camera data [72].

One key component of IVEX's approach is using open scenarios scenes. Other than being simulated by AI, the IVEX framework uses scenes created from realistic camera feeds and includes both hazardous and non-hazardous situations. These scenes test and validate AV perception and decision-making systems, ensuring that they can accurately detect and respond to a wide range of driving scenarios.

However, similar to NVIDIA's AI based approach, the process of creating and validating these open scenario scenes can be time-consuming and labour-intensive. Engineers must manually review hours of camera footage to identify and annotate relevant scenarios, which can be a significant bottleneck in the AV development and testing process.

Table 2.2: Summary of Safety Frameworks in the AV Industry

Company	Focus	Key Components	Comments
Aurora	Safety case framework based on GSN	Continuous monitoring and updating	Primarily focused on single-channel ADS architectures
Waymo	Comprehensive safety framework	Extensive testing, continuous monitoring, collaboration with stakeholders	Focused on operational aspects, not detailed technical implementation
NVIDIA	AI-based perception using camera data	Deep learning for perception, simulation environments for training and validation	Computationally expensive, time-consuming data annotation and focus on single ADS AVs
IVEX	Scenario-based testing using camera data	Open scenario scenes including hazardous and non-hazardous situations	Time-consuming and labour-intensive scenario annotation, focused on single AD systems

Table 2.2 shows a quick overview of the existing technologies implemented by companies understood from publicly available information. It is important to note that while each of these companies (Aurora, Waymo, NVIDIA, and IVEX) has developed its own approach to AV safety, they all face similar challenges in terms of data management, scenario coverage, and validation efficiency. These challenges are particularly acute in the context of multi-channel ADS architectures, where the complexity and diversity of driving scenarios are compounded by the need to ensure consistent and safe behaviour across multiple perception and decision-making systems.

2.9. Software Ecosystem

A comprehensive software ecosystem is required to develop and test the Safety Performance Indicators (SPIs) and Hazardous Scenario Identification (HSI) algorithms. Our ecosystem includes open-source Automated Driving System (ADS) software, the use of Robot Operating System (ROS), the CARLA simulator, and the Open SCENARIO standard.

2.9.1. CARLA Simulator

The CARLA simulator is an open-source urban driving simulator that provides a realistic environment for developing, training, and validating ADS. It includes a flexible API for controlling vehicles, pedestrians, and traffic lights and a suite of sensors such as cameras, LiDARs, and GPS [23].

The CARLA Leaderboard is a benchmark platform for evaluating the performance of ADS in the CARLA simulator [26]. It provides a standardized set of scenarios and metrics to assess the perception, planning, and control capabilities of ADS. The CARLA Leaderboard has become a popular platform for researchers and developers to compare their ADS implementations' performance and identify areas for improvement.

This thesis uses the CARLA simulator to create realistic driving scenarios for testing the SPI and HSI algorithms. By leveraging the CARLA simulator, a wide range of hazardous and non-hazardous scenarios can be created and tested in a safe and controlled environment without the need for expensive real-world testing.

2.9.2. OpenSCENARIO

OpenSCENARIO is an open, standardized file format describing dynamic driving scenarios for simulation environments. It facilitates the specification of complex, synchronized manoeuvres among multiple entities within a simulation, such as vehicles, pedestrians, and traffic infrastructure [73]. This thesis utilizes OpenSCENARIO to capture and encode hazardous driving scenarios identified from runs conducted in the CARLA simulator.

OpenSCENARIO typically comprises two types of files: '.xodr' and '.xosc'. The '.xodr' file describes the road network used in the simulation, detailing the geometry, topology, and additional attributes of road elements. The '.xosc' file specifies the dynamic content of the scenario, including vehicle behaviours, pedestrian movements, and interactions with the traffic infrastructure.

These files are platform-independent and can be viewed on any compatible scenario player. For the purposes of this thesis, the Esmini player [74] is used to view and validate the scenario reproductions. It is important to note that OpenSCENARIO is an evolving standard that is continuously developed and refined by the OpenSCENARIO community.

OpenSCENARIO files can be generated using multiple tools, including the Python scenario-generation library and MATLAB, and by manually writing XML files according to the standard. To streamline the process and enable the automatic generation of scenarios, this thesis primarily utilizes the Python scenario-generation library.

While MATLAB [50] also supports creating OpenSCENARIO files, it is limited to version 1.1.0. In contrast, the Python scenario-generation library supports the latest version, 1.2.0. This capability to use the most current version is a key reason for selecting the Python library over MATLAB.

2.9.3. Open source ADS Software

We used two open-source ADS software stacks integrated into CARLA to simulate a multi-channel AV:

1. **LAV (Learning from All Vehicles)**: LAV introduces a novel neural network architecture that enhances ADS by leveraging data from all observed vehicles in the vicinity, not just the ego-vehicle. This method uses a multi-agent learning framework where the neural network models the behaviours and trajectories of multiple vehicles simultaneously. The approach enables the system to learn complex interactions and dependencies between different vehicles, enhancing prediction accuracy and decision-making under varied traffic conditions. By integrating multi-vehicle observational data, LAV's neural network architecture benefits from a richer dataset, resulting in more robust generalization capabilities across diverse driving scenarios. It is to be noted that the multi-vehicle training in LAV was conducted offline using pre-recorded data. [24].
2. **TFuse (TransFuser)**: TransFuser employs a hierarchical neural network that performs a multi-modal fusion of high-dimensional sensor data to control AV in urban settings. This network architecture innovatively combines features from different sensor modalities—such as images, LiDAR, and radar—through fusion layers that enhance the model's ability to interpret complex environmental data. The hierarchical design allows for efficient data processing, with lower layers handling raw sensory inputs and higher layers focusing on decision-making based on the fused information. This structure ensures that TransFuser can effectively deal with the challenges of sensor integration, providing a seamless sensorimotor control that is crucial for navigating through dynamic urban environments [25].

Both the ADS have been integrated into CARLA to simulate a multichannel AV.

2.9.4. Robot Operating System (ROS)

The Robot Operating System (ROS) is a flexible framework for writing robot software. It provides a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behaviour across a wide variety of robotic platforms [75].

ROS provides standardized communication methods between components (called nodes) using topics, services, and actions. This thesis uses ROS as the underlying framework for integrating the various components of the software ecosystem, including the open-source ADS software, the CARLA simulator, and the Daruma C++ implementation.

One of the key features of ROS used in this thesis is the rosbag functionality, which allows for the recording and playback of ROS message data. This is particularly useful as the simulations can be recorded and played faster for faster analysis. ROSBag also has features that allow it to play recorded scenarios slower, which enables better manual analysis.

2.10. Related Work Conclusion

In conclusion, the related work and state-of-the-art analysis have revealed several key insights that influencing the approach taken in this thesis:

1. Safety Performance Indicators (SPIs) have emerged as a promising approach for monitoring the safety of AVs and identifying hazardous scenarios for testing and validation. By adopting SPIs as a central component of our methodology, we aim to leverage their potential for comprehensive and quantitative safety assessment.
2. Existing safety frameworks and approaches primarily focus on single-channel Automated Driving System (ADS) architectures. It is also not clearly mentioned whether the safety aspects, such as the perception and navigation systems, also monitor the internal working of the ADS. This highlights the need for further research and exploration into the safety monitoring and scenario generation challenges specific to multi-channel ADS architectures, which is the focus of this thesis.
3. The Daruma Design Pattern, with its cross-channel analysis and arbitration capabilities, provides a unique opportunity to address the safety challenges associated with multi-channel ADS architectures.
4. Moreover, Daruma was developed considering the requirement in the ADS to derive Safety Performance Indicators (SPIs). By correlating the channels' world models and motion plans, Daruma can identify low safety performance and spot functional insufficiencies, such as repeated disagreements in object classifications between the main and backup channels. The SPIs and associated driving scenarios can be uploaded to the cloud for fleet-wide analysis, enabling the creation of Over-The-Air updates to improve the ADS performance [45].
5. The software ecosystem required for developing and testing the planned SPI and HSI algorithms involves a combination of open-source ADS software, such as LAV and TFuse, the Robot Operating System (ROS), the CARLA simulator, and the OpenSCENARIO standard. By leveraging these tools and frameworks, a realistic and flexible testbed for testing algorithms is aimed at while also ensuring compatibility with existing AV development workflows.
6. Fleet management and safety monitoring is an important aspect of AV deployment that has not been extensively addressed in the prior art. Most of the existing literature does not provide detailed explanations of the underlying architectures and algorithms used, making it challenging to evaluate the effectiveness and scalability of the proposed solutions. The thesis aims to propose SPI and HSI frameworks that have the potential to be applied in a fleet management context, enabling the continuous monitoring and improvement of AV safety across many vehicles.

In summary, this thesis aims to address the limitations of existing AV safety approaches by focusing on multi-channel ADS architectures and leveraging the capabilities of the Daruma Design Pattern. By integrating SPIs and HSI algorithms into a comprehensive software ecosystem, we seek to advance the state-of-the-art AV safety monitoring and validation, ultimately contributing to developing safer and more reliable AVs.

3

System Architecture

3.1. Architecture for Continuous Improvement of ADS

We propose the architecture, as shown in Figure 3.1, for continuous improvement of ADS is designed to enhance the performance, safety, and reliability of AVs through a closed-loop system involving fleet data collection, cloud-based analysis, and over-the-air (OTA) updates.

At the core of this architecture are AVs, each equipped with sensors, safety monitors, and onboard computing systems. These safety monitors continuously track and record various parameters related to the individual vehicle's performance, diagnostics, and environment perception in real-time as the vehicle navigates through its operational domain.

Each AV periodically transmits key data points and events (such as sensor data, SPIs, and identified hazardous scenarios) to a centralized, cloud-based system to enable fleet-wide learning and improvement. This data is collected from an entire fleet of AVs operating in diverse environments and under various conditions. The cloud-based system, managed by the ADS developer or a third-party provider, securely receives, stores, and processes the vast amounts of data the fleet generates.

The aggregated fleet data can undergo extensive analysis using advanced data processing techniques, machine learning algorithms, and statistical models. This analysis is performed either periodically or on-demand, depending on the specific needs and objectives of the stakeholders involved. The insights gained from this analysis serve multiple purposes and benefit various stakeholders in the ADS ecosystem.

One key beneficiary of the processed fleet data is the road authorities responsible for maintaining and upgrading transportation infrastructure. Based on the fleet data analysis, the ADS developer can generate detailed reports and recommendations, highlighting areas where infrastructure improvements are needed to support the safe and efficient operation of AVs. By addressing these issues, road authorities can create a more conducive environment for AV deployment, ultimately enhancing the safety and reliability of these systems.

In addition to infrastructure improvements, the fleet data analysis can also trigger real-time support for individual AVs in emergencies. If an AV encounters a critical event or an anomaly that requires immediate attention, it can send an alert to the cloud-based system. This alert can be promptly relayed to a fleet operator or a remote teleoperator who can assess the situation and provide guidance or take control of the vehicle if necessary, ensuring that AVs can handle unexpected scenarios gracefully and maintain safe operation.

Another critical aspect of the continuous improvement architecture is the ability to push software updates and enhancements to the AVs over-the-air. As the ADS developer identifies areas for improvement based on the fleet data analysis, they can develop software updates that address specific issues, optimize performance, or add new functionalities to the AVs. By delivering these updates wirelessly to

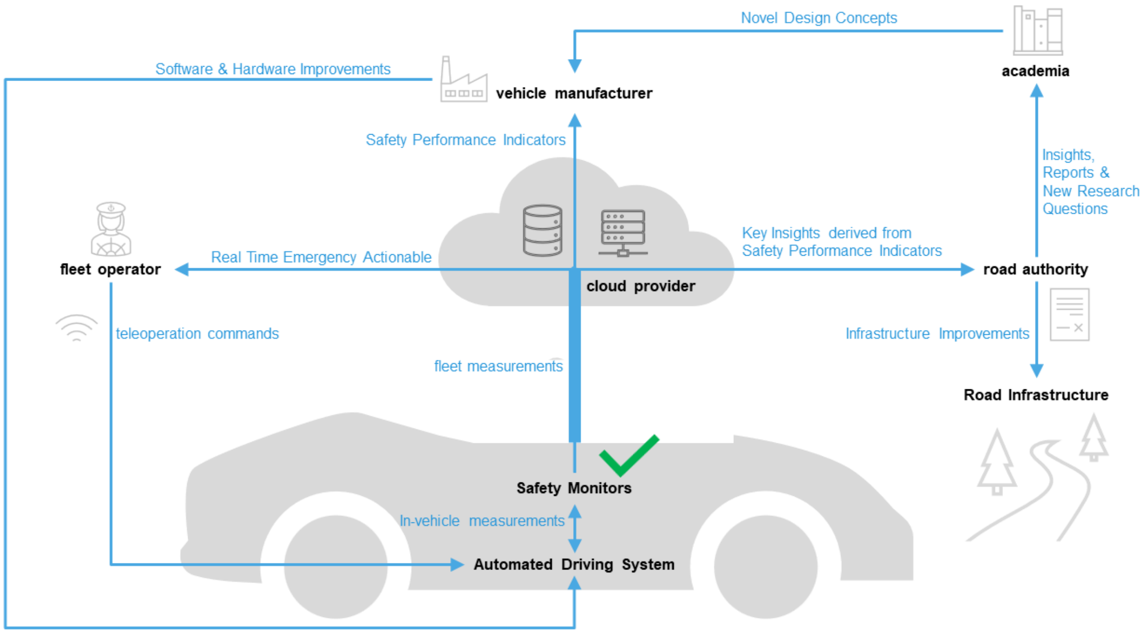


Figure 3.1: SPI Based Continuous Improvement Process For Automated Driving

the entire fleet, the ADS developer can ensure that all vehicles benefit from the latest advancements and maintain high performance and safety.

The continuous improvement process enabled by this architecture extends beyond the ADS developer and the road authorities. The insights derived from the fleet data analysis can also benefit academic institutions and research organizations focused on advancing AV technologies. By collaborating with ADS developers and accessing anonymized fleet data, researchers can validate their models, test new algorithms, and explore innovative solutions to the challenges faced by AVs, fostering a cycle of innovation and practical improvements.

In the context of this thesis, the architecture shown in Figure 3.1 for continuous improvement of ADS serves as a foundation for the functional architecture of Safety Performance Indicators (SPI) and Hazardous Scenario Identification (HSI) described in Section 3.2.

By leveraging the comprehensive data collection, cloud-based analysis, and OTA update capabilities of the continuous improvement architecture, the SPI and HSI functional architecture can effectively monitor, analyze, and enhance the safety and performance of the AV fleet. This seamless integration between the two architectures ensures that the insights derived from the SPI and HSI processes are effectively utilized to drive the continuous improvement of the ADS, ultimately leading to the development of safer, more reliable, and more capable AVs.

3.2. Functional Architecture of SPI and HSI

The functional architecture of SPI and HSI, as depicted in Figure 3.2, provides a detailed view of how these critical components are integrated into a multi-channel ADS to enable continuous improvement, enhanced safety, and proactive hazard mitigation. This thesis aims to implement and validate the proposed functional architecture, demonstrating its effectiveness in real-world scenarios.

At the foundation of this architecture are the multiple vehicles (V1, V2, ..., Vn) within the AV fleet, each equipped with a sophisticated multi-channel ADS. The multi-channel architecture is a key enabler for the SPI and HSI functionality because it allows for redundancy, diversity, and cross-validation of the ADS's perception, decision-making, and control functions.

Within each vehicle, every ADS processes the raw sensor data from the vehicle's onboard sensors, such as cameras, LiDAR, radar, and ultrasonic sensors, to generate a comprehensive understanding

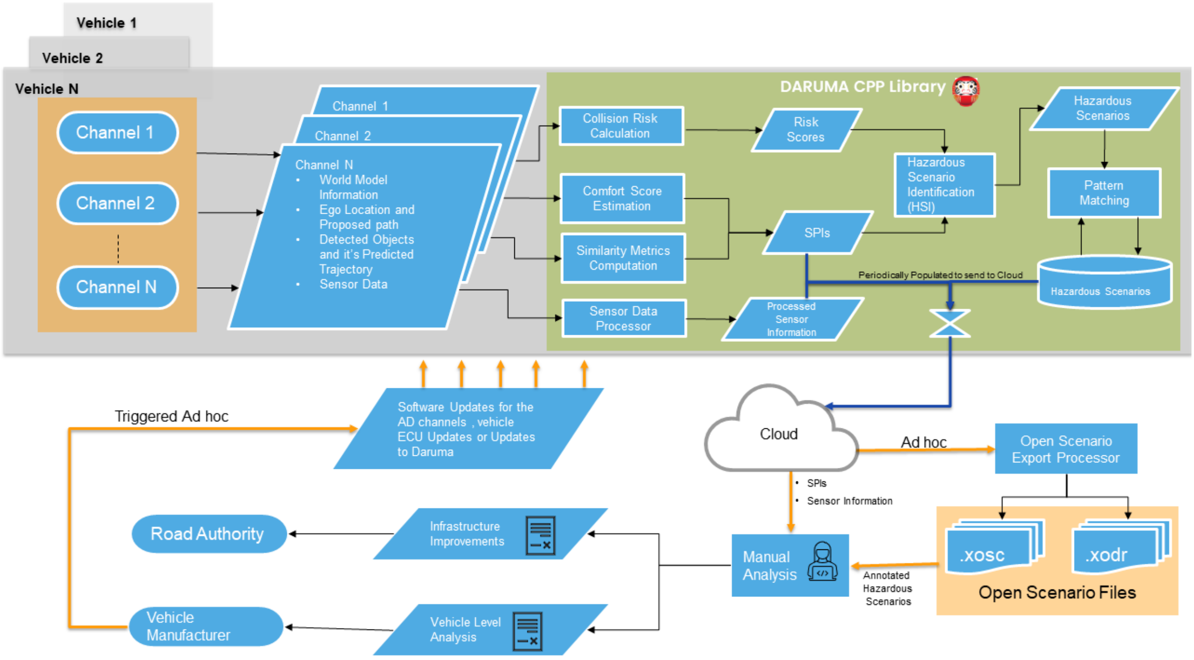


Figure 3.2: Planned Functional Architecture for SPI and HSI

of the surrounding environment. This understanding is encapsulated in channel-specific information, including the World Model, Ego Localization and Proposed Path, Detected Objects and their Predicted Trajectories, and Sensor Data.

Multiple modules then process the channel-specific information from each ADS to calculate various metrics, including the Collision Risk, Comfort Score, Similarity Metrics, and Sensor Data Processor. These metrics provide insights into the ADS's performance, safety, and potential hazards.

The calculated metrics are aggregated and synthesized to generate comprehensive risk scores and SPI metrics for each vehicle. One key output from the SPI metrics and Risk scores (based on preset thresholds) is the identification of hazardous scenarios encountered by the AV fleet. These hazardous scenarios are stored in a dedicated database, along with relevant contextual information, and pattern-matching techniques are applied to identify and consolidate similar or recurring hazardous scenarios.

The SPI metrics, risk scores, identified hazardous scenarios, and associated sensor information are continuously transmitted to a cloud-based system for storage, analysis, and dissemination. This centralized repository allows for fleet-wide monitoring, trend analysis, and data-driven decision-making.

Stakeholders, such as vehicle manufacturers or fleet operators, can access and analyze the SPI and HSI data ad hoc or at predefined intervals. This enables them to gain insights into the AV fleet's performance and safety, identify trends and patterns, and pinpoint areas for improvement.

One of the key processes in this functional architecture is the generation of OpenSCENARIO files based on the identified hazardous scenarios. This process occurs offline, typically triggered by manufacturers or other relevant stakeholders accessing the information from the cloud. The OpenSCENARIO files encapsulate the hazardous scenarios in a standardized format, allowing for their reproduction, simulation, and analysis in various virtual testing environments. The OpenSCENARIO generation process is carried out within the vehicle manufacturer's or fleet operator's domain, leveraging the data and insights the cloud-based system provides. The functional architecture ensures data security, confidentiality, and compliance with relevant regulations and policies by containerising this process within the stakeholder's environment.

The generated OpenSCENARIO files serve as valuable assets for the continuous improvement of the ADS. The vehicle manufacturer can use them to refine and validate the ADS algorithms, test the system's performance in challenging situations, and develop mitigation strategies for identified hazards.

Moreover, the OpenSCENARIO files can be shared with simulation platforms, such as the CARLA simulator, to perform virtual testing and validation of the ADS under various conditions.

In summary, as implemented in this thesis, the functional architecture of SPI and HSI provides a comprehensive framework for monitoring, analyzing, and improving the safety and performance of multi-channel ADS. By leveraging the power of fleet data, cloud computing, and standardized scenario representation (OpenSCENARIO), this architecture enables proactive safety management, data-driven decision-making, and continuous improvement of the AV fleet.

The successful implementation and validation of this functional architecture in real-world scenarios will contribute to developing safer, more reliable, and more trustworthy AVs, paving the way for their widespread adoption in the future.

4

Safety Performance Indicators

This chapter explains the safety case defined in this thesis and the associated SPIs defined to validate the arguments made to the safety case. The Daruma C++ implementation, based on the Daruma Design Pattern (as discussed in Section 2.5.1), is used as the framework for implementing and validating the SPIs in this thesis.

It is to be noted that the information available from each of the channels to the Daruma C++ implementation is as follows:

- Location, shape, and size of the Ego Vehicle, along with its velocity, orientation, and the planned trajectory by the respective AD channel.
- Location, shape, and size of the Objects detected around the ego vehicle by the respective AD Channel, along with their velocity, orientation, and predicted trajectory.

The algorithms have to be designed based on the information available only; hence, they can be considered a constraint on the system. More information about the setup can be found in Section 6.1.

4.1. Safety Case

As presented in Section 2.6.1, the safety case is a structured argument, supported by evidence, that a system is acceptably safe for a given application in a given operating environment [57]. Since the thesis aims to explore SPIs in the context of AV safety, a universal safety case that can be applied to most manufacturers is defined as ***EGO Vehicle goes from Point A to Point B safely***. Four main arguments support this top-level claim, each focusing on a specific aspect of the ADS's performance and safety. Table 4.1 shows the arguments that are linked to the safety case.

Table 4.1: Overview of safety case arguments defined for the thesis

Safety Case	**Arguments**
EGO vehicle goes from Point A to Point B safely	Channels can detect the location of the ego vehicle accurately
	Channels can detect Objects around the ego vehicle accurately
	Channels can correctly predict the motion of the objects detected
	Channel generates a safe trajectory for the vehicle to follow

The first argument, "Channels can detect the location of the ego vehicle accurately," focuses on the ADS's ability to determine its own position and orientation within the environment. Accurate ego localization is essential for safe navigation and decision-making.

The second argument, ”Channels can detect Objects around the ego vehicle accurately,” addresses the ADS’s capability to perceive and classify objects in its surroundings. Reliable object detection and localization are crucial for identifying potential hazards and planning appropriate actions.

The third argument, ”Channels can correctly predict the motion of the objects detected,” emphasizes the importance of predicting the future states and trajectories of detected objects. Accurate motion prediction enables the ADS to anticipate potential conflicts and make proactive decisions.

Finally, the fourth argument, ”Channel generates a safe trajectory for the vehicle to follow,” focuses on the ADS’s ability to plan and execute safe and comfortable trajectories. This argument encompasses various aspects of trajectory generation, such as smooth acceleration and braking, consistency with previous plans, and adherence to safety constraints.

These arguments collectively contribute to the overall safety case, providing a structured framework for assessing the ADS’s performance and identifying potential areas for improvement.

4.2. Safety Performance Indicators in a multi-channel ADS

As presented in Section 2.6.1, SPIs are quantitative measures that provide evidence for the arguments in the safety case [37]. Each SPI is designed to evaluate a specific aspect of the ADS’s performance, with scores ranging from 0 to 1, indicating the degree to which the corresponding argument is supported.

Table 4.2 presents the SPIs associated with each argument in the safety case. These SPIs are carefully defined to provide meaningful and measurable evidence for the arguments, enabling a comprehensive assessment of the ADS’s safety performance.

Table 4.2: Safety Performance Indicators corresponding to each of the arguments of the safety case

Arguments	**Safety Performance Indicators (SPI)**
Channels can detect the location of the ego vehicle accurately	The location of the ego vehicle detected is consistent across the channels
	The orientation of the vehicle detected is consistent across the channels
Channels can detect Objects around the ego vehicle accurately	The number of objects detected is consistent across channels
	The location of the objects detected are consistent across channels
	The size and orientation of the objects detected are similar across channels
Channels can correctly predict the motion of the objects detected	The predicted trajectory of the objects identified are similar across channels
	The objects follow the predicted trajectories proposed by the respective channels
Channel generates a safe trajectory for the vehicle to follow	The planned path generated by the channels contains smooth acceleration and breaking
	The planned path generated at every instant is consistent with the previously generated planned path by the respective channels
	The planned path is similar to the paths generated across channels

For the argument that ”Channels can detect the location of the ego vehicle accurately,” two SPIs are defined: consistency of ego location and consistency of ego orientation across channels. These SPIs evaluate the agreement between different ADS channels regarding the estimated position and orientation of the ego vehicle.

The argument ”Channels can detect Objects around the ego vehicle accurately” is supported by three SPIs: consistency of object count, location, and size/orientation across channels. These SPIs assess the ADS’s ability to perceive objects in its environment consistently.

The SPIs for the argument ”Channels can correctly predict the motion of the objects detected” focus on the similarity of predicted object trajectories across channels and the conformance of actual object motion to the predicted trajectories. These SPIs provide evidence for the ADS’s ability to anticipate the future states of detected objects accurately.

Finally, the argument ”Channel generates a safe trajectory for the vehicle to follow” is supported by three SPIs: smoothness of the planned path (acceleration and braking), consistency of the planned path over time, and similarity of the planned path across channels. These SPIs evaluate the generated trajectories’ safety, comfort, stability, and consistency.

By systematically evaluating the ADS’s performance using these SPIs, developers and safety assessors can comprehensively understand the system’s strengths and weaknesses. The SPI scores provide quantitative evidence for the arguments in the safety case, enabling data-driven decision-making and guiding the continuous improvement of the ADS.

To facilitate the implementation and analysis of the SPIs, we formally defined the metrics associated with each SPI, as shown in Table 4.3. This mapping allows for a clear understanding of the relationship between the SPIs and their corresponding metrics, which will be discussed in detail in the following sections.

Table 4.3: Correlation between Safety Performance Indicators and Metrics

Safety Performance Indicators	**Metrics**
The location of the ego vehicle detected is consistent across the channels	Ego Location Similarity Scores
The orientation of the vehicle detected is consistent across the channels	Ego Orientation Similarity Scores
The number of objects detected is consistent across channels	Object Count Similarity Scores
The location of the objects detected is consistent across channels	Object Location Similarity Scores
The size and orientation of the objects detected are similar across channels	Object Orientation Similarity Scores Object Area Similarity Scores
The predicted trajectory of the objects identified are similar across channels	Object Trajectory Similarity Scores Ego Trajectory Similarity Scores
The objects follow the predicted trajectories proposed by the respective channels	Objected Motion Prediction Validation Score
The planned path generated by the channels contains smooth acceleration and breaking	History Comfort Scores Planned Path Comfort Scores
The planned path generated at every instant is consistent with the previously generated planned path by the respective channels	Channel Specific Ego Planned Path Tracker Scores Cross Channel Ego Planned Path Tracker Score

It is important to note that each SPI metric provides a scalar score ranging from 0 to 1, which represents the safety performance of the system with respect to the corresponding argument

in the safety case. A score of 0 indicates that the argument is not satisfied, suggesting a potential safety concern, while a score of 1 indicates that the argument is fully satisfied, providing evidence that the system is operating safely in that particular aspect.

4.3. SPI Algorithms

To simplify the explanation of the metrics in the subsequent sections, the SPI metrics are categorized into three main categories: Similarity Metrics, Motion Prediction Metrics, and Comfort Metrics, as shown in Table 4.4.

Table 4.4: Categorised SPI Metrics

Similarity Metrics	**Motion Prediction Metrics**	**Comfort Metrics**
Ego Location Similarity Scores	Objected Motion Prediction Validation Score	History Comfort Scores
Ego Orientation Similarity Scores	Ego Trajectory On-track Scores	Planned Path Comfort Scores
Object Count Similarity Scores	Object Trajectory On-track Scores	
Object Location Similarity Scores	Channel Specific Ego Planned Path Tracker Scores	
Object Orientation Similarity Scores	Cross Channel Ego Planned Path Tracker Score	
Object Area Similarity Scores		

In the following subsections, we will discuss the algorithms and implementation details for each category of SPI metrics. We will explore how these metrics are calculated, their relevance to the safety case arguments, and their significance in evaluating the overall safety performance of the ADS. By providing a comprehensive understanding of these metrics, we aim to demonstrate how they contribute to the continuous improvement and validation of the ADS's safety. It is to be noted that the algorithms were designed with the constraints defined in Section 1.2 in mind.

4.3.1. Comfort Metrics

Comfort metrics such as acceleration and jerk are closely monitored, as excessive values can indicate risky or aggressive driving behaviours that may compromise vehicle stability and safety. Smooth trajectories minimize sudden movements, thereby reducing the potential for motion sickness in passengers and enhancing the predictability of vehicle behaviour for other road users. Moreover, trajectories prioritising comfort will likely align with safer driving practices, as they tend to avoid sharp manoeuvres and maintain a steady velocity, enhancing the AV's overall control and response time in dynamic traffic conditions [76].

The following subsections explain the comfort metrics defined to validate the safety case.

History Comfort Metric

The History Comfort Metric is channel-specific and evaluates the comfort of the ego vehicle's motion based on its historical location data. The algorithm stores the location of the ego vehicle perceived by each channel, along with the time interval. After collecting a specified number of data points, the algorithm calculates the velocity, lateral and longitudinal acceleration, and jerk for each channel. These values are then normalized between 0 and 1 using a sigmoid normalization function:

$$NV = \frac{1}{1 + e^{-s \cdot w}}$$

Where s is the raw value of the metric (acceleration or jerk), w is a predefined constant that determines the steepness of the normalization curve, and NV is the normalized value. The sigmoid normalization function maps raw score values into a 0 to 1 range, providing a smooth transition that reflects slight variations in comfort levels. This bounded scaling ensures that extreme values are handled gracefully, maintaining score consistency across different conditions and making it suitable for decision-making in safety assessments of automated vehicles.

Since, at every time step, the location of the ego vehicle in each of the channels is stored, the acceleration and jerk are calculated as

$$\begin{aligned}
V_x &= \frac{dx}{dt} \quad \text{(Longitudinal Velocity)}\\
V_y &= \frac{dy}{dt} \quad \text{(Lateral Velocity)}\\
LA &= \frac{dV_x}{dt} \quad \text{(Lateral Acceleration)}\\
LL &= \frac{dV_y}{dt} \quad \text{(Longitudinal Acceleration)}\\
\dot{J} &= \frac{dLL}{dt} \quad \text{(Jerk)}
\end{aligned}$$

Where t is the time interval obtained from the channels, x and y represent the difference in x and y coordinates between consecutive stored locations of the ego vehicle in their respective channels.

The overall comfort score is calculated by:

$$\text{Comfort Score} = 1 - \frac{(NV_{LL} + NV_{LA})/2 + NV_J}{2}$$

Planned Path Comfort Metric

The Planned Path Comfort Metric assesses the comfort of the ego vehicle's planned path. The algorithm takes the location of the ego vehicle and the planned path provided by the AD channel. It then calculates the planned path's velocity, acceleration, and jerk using the same methods as the History Comfort Metric. The raw values are normalized using the sigmoid normalization function, and the overall comfort score is calculated using the same formula as the History Comfort Metric.

4.3.2. Similarity Metrics

Similarity metrics evaluate the consistency and agreement between the trajectory decisions and world model outputs of different AV subsystems or channels. These metrics are particularly relevant in the context of multi-channel ADS architectures, where multiple perception, planning, and control pipelines operate in parallel.

Ego Location Similarity Scores

The ego location similarity (ELS) metric evaluates the proximity of ego vehicle locations across different channels. It is calculated by comparing the translation components (i.e., positions) of the ego vehicles in each channel and scoring their similarity. For each pair of channels, the similarity score is computed as:

$$\text{(ELS)} = \frac{1}{1 + \Delta D}$$

Where ΔD is the Euclidean distance between the two locations of the ego vehicle between the channels.

Additionally, a moving variance of the ELS scores was calculated using a window size of 5 to further analyse the stability and variability of location similarity over time. This analysis helps identify hazardous scenarios, which will be discussed in detail in subsequent chapters.

To calculate the moving variance of the ELS scores, the following formula is used:

$$s_m^2 = \frac{1}{n-1} \sum_{i=1}^{n} (ELS_i - \overline{ELS})^2$$

where:

- s_m^2 is the moving variance of the ELS scores.
- ELS_i represents the Ego Location Similarity score at the i-th position in the moving window.
- $\overline{ELS}$ is the mean of the ELS scores within the window.
- n is the number of scores in the window, which is 5 in this case.

Ego Orientation Similarity

The ego orientation similarity (EOS) metric quantifies the similarity in the orientation (rotation) of ego vehicles across channels. This is calculated by extracting the heading of the vehicle as angle θ from the rotation matrix of each ego vehicle's position and using the normalized angle difference formula:

$$\text{EOS} = 1.0 - \frac{diff}{\pi}$$

Where $diff$ is the absolute difference between any two angles, adjusted to the range $[0, \pi]$.

Object Location Similarity Scores

The object location similarity (OLS) quantifies how closely objects are located to each other across different channels. Similar to the ego location similarity, this score uses the translation components of object positions and calculates their proximity. Objects between channels are paired by proximity within a specified threshold. If objects are paired, their similarity score is calculated as follows:

$$\text{OLS} = \frac{1}{1.0 + \Delta D}$$

Where ΔD is the Euclidean distance locations of the paired objects. Scores are averaged over all pairs to yield the similarity for each channel. The similarity score is set to 1.0 if both channels have no objects.

Object Count Similarity Scores

The object count similarity (OCS) metric measures how similar the numbers of detected objects are across channels. It is computed by determining the absolute difference in object counts between channels and scoring them as follows:

$$\text{OCS} = \frac{1}{1.0 + \Delta N}$$

Where ΔN is the difference in the number of objects each channel identifies. The similarity score is set to 1.0 if both channels have no objects.

Object Orientation Similarity

The Object Orientation Similarity (OOS) metric assesses how similarly objects are oriented relative to one another across channels. For each pair of objects that have been paired by proximity, the similarity in their orientation (rotation) is calculated using:

$$\text{OOS} = 1.0 - \frac{diff}{\pi}$$

Where $diff$ is the normalized angle difference between their orientations, calculated and adjusted to the range $[0, \pi]$. If both channels have no objects, the similarity score is set to 1.0.

Object Area Similarity

The object Area Similarity (OAS) compares the areas of paired objects across channels. Objects are paired by proximity, and the area is computed using the Gauss area formulae, as explained in the section. For each pair, the area similarity is computed as:

$$\text{OAS} = \frac{small\_area}{big\_area}$$

where $small\_area$ and big_area are the smaller and larger of the two areas, respectively. This ratio provides a direct comparison of size irrespective of shape differences. The area of each object is calculated using the Gauss area formula (shoelace formula) [77] applied to the object's polygon. The similarity score is set to 1.0 if both channels have no objects.

Gauss Area Formula for Polygons with Ordered Vertices

The Gauss Area Formula, also known as the Shoelace Formula, is a method used to calculate the area of a simple polygon whose vertices are defined in a sequential order in the plane. This technique is particularly useful because it provides a straightforward way to determine the area of a polygon when the coordinates of its vertices are known. The formula is named the "Shoelace" because of how the summations in the formula crisscross, much like the lacing of a shoe.

To compute the area of a polygon with n vertices labeled $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, the formula is:

$$A = \frac{1}{2}\left|\sum_{i=1}^{n}(x_i y_{i+1} - y_i x_{i+1})\right|$$

where (x_{n+1}, y_{n+1}) are taken to be (x_1, y_1), effectively closing the polygon by connecting the last vertex back to the first. This method efficiently calculates the area by traversing the perimeter of the polygon just once, summing the areas of the trapezoids defined by the line segments of the polygon edges and the x-axis.

The elegance of the Gauss Area Formula lies in its efficiency and simplicity, making it a popular choice for computer graphics and geometric calculations where vertices of polygons are routinely used [77].

4.3.3. Motion Prediction Metrics

Motion prediction metrics assess the accuracy and reliability of the ADS's ability to predict the future states of detected objects and the ego vehicle itself. These metrics are crucial for evaluating the system's situational awareness and decision-making capabilities.

Object Trajectory Similarity Scores

The object trajectory similarity scores reflect the alignment of object movements over time across channels. Similar to ego trajectory similarity, it involves comparing the trajectories of paired objects point-by-point and computing similarity scores with decreasing weights to prioritize initial alignment. Objects are paired by proximity with a set threshold. For each pair of objects, their trajectories are extended to the maximum duration of the two, and the similarity for each trajectory point is computed as:

$$S_n = \frac{1}{1 + \Delta D} \times W_n$$

$$W_n = 1 \times (k)^{n-1}$$

Here, ΔD represents the Euclidean distance between corresponding waypoints in the trajectories of the objects between the different channels. It is then multiplied by a W_n, which is a weight. The weight for each waypoint starts at 1 (k) and decreases progressively by a factor of 5% for each subsequent waypoint. The decision to use a 5% factor was based on empirical observations and is configurable to accommodate different thresholds as needed, further detailed in subsequent sections. This decreasing weight factor emphasizes the importance of closer alignment at the beginning of the trajectory, which is often more critical for immediate decision-making in automated driving.

Finally, the Object Trajectory Similarity (OTS) score is obtained using:

$$OTS = \frac{\sum S_n}{\sum W_n}$$

and then averaged over the trajectory length. The similarity score is set to 1.0 if both channels have no objects.

Ego Trajectory Similarity Scores

The ego trajectory similarity (ETS) metric assesses the alignment of trajectories of the ego vehicles across channels over time. The similarity is quantitatively evaluated by a point-by-point comparison of the translation components of the trajectories for each channel over a minimum overlapping duration. The equations used for the calculation of Ego Trajectory Similarity scores are -

$$S_n = \frac{1}{1 + \Delta D} \times W_n$$

$$W_n = 1 \times (k)^{n-1}$$

Here, ΔD represents the Euclidean distance between corresponding waypoints in the trajectories between the different channels. It is then multiplied by a W_n, which is a weight. The weight for each waypoint starts at 1 and decreases progressively by a factor of 5% for each subsequent waypoint. This decreasing weight factor emphasizes the importance of closer alignment at the beginning of the trajectory, which is often more critical for immediate decision-making in automated driving.

The weighted similarities for all overlapping waypoints are summed to generate a cumulative score representing the overall trajectory alignment between channels. This final score reflects the integrated assessment of trajectory similarity, prioritizing early points in the trajectory and gradually considering less influence from later points.

The Ego Trajectory Similarity (ETS) is then obtained with the equation:

$$ETS = \frac{\sum S_n}{\sum W_n}$$

Object Motion Prediction Validation

The Object Motion Prediction Validation (OMPV) module validates the accuracy of trajectory predictions for objects detected by each AD channel.

During each operational cycle, the AV's channels provide real-time data about detected objects, including their current positions and predicted trajectories. These trajectories forecast where each object is expected to move in the near future based on its current motion patterns and dynamics. This information is stored each time it is received.

In the subsequent cycle, when updated information is received from the respective channels regarding the environment and objects within it, the Object Motion Prediction Validation process then involves comparing these objects' new, actual positions against the predicted positions stored from the previous cycle. Specifically, the algorithm evaluates how many of the objects have moved according to their previously predicted trajectories.

The validation score is calculated by assessing the proportion of objects whose actual locations match their predicted locations. If the actual positions significantly deviate from the predictions or new objects not previously detected appear in current cycle, it is recorded as a miss. OMPV is calculated using:

$$OMPV = \frac{N1}{N2}$$

Where $N1$ is the total number of objects that correctly followed the predicted part and $N2$ is the total number of objects detected in the previous cycle.

Ego Planned Path Tracker

The Ego Planned Path Tracker is specifically designed to evaluate and ensure the consistency and reliability of the ego vehicle's trajectory across different channels, which is crucial for safe navigation and efficient manoeuvring in various driving conditions. This module has two primary components: Channel-Specific Path Similarity and Cross-Channel Path Similarity. The Path Similarity is calculated using Fréchet as explained in the following section.

Discrete Fréchet Distance

The Discrete Fréchet Distance offers a measure to assess the similarity between two curves, which are typically represented as sequences of points. Often illustrated by the analogy of a person walking their dog on leashes of varying lengths, the Fréchet distance is the minimum length of a leash necessary for both the dog and the owner to walk their respective paths from start to finish without backtracking, but they are allowed to control their speed independently [78].

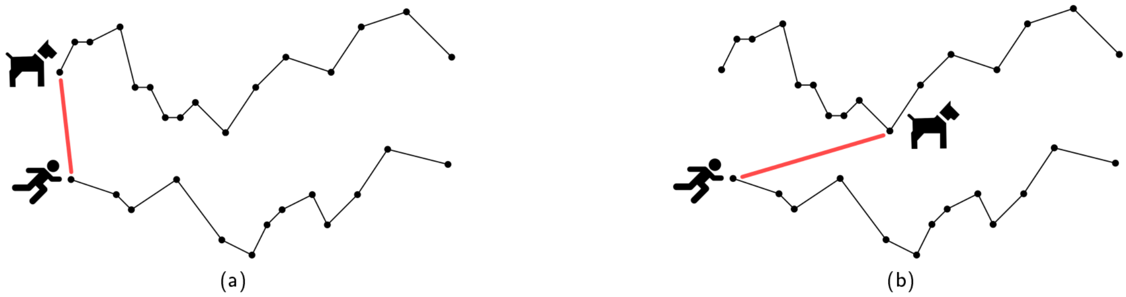


Figure 4.1: Illustration of the Fréchet problem's intuition [79]

Figure 4.1 shows an illustration of the Fréchet Distance. A man and his dog on a trail with a red leash between them. This explanation uses the Fréchet distance as the minimum leash length. Man and dog can only go forward or halt, never backwards, and must both meet their end. In Figure 4.1 (a), the man and dog are in the starting position. In Figure 4.1 (b), the dog has moved, increasing the leash length.

The paths are visualized as two sequences of points, and at each step, the distance between corresponding points (the person and the dog at each step) is considered. The Fréchet distance is the maximum of these point-to-point distances minimized over all possible points along the two paths. In simpler terms, it captures the idea of a "best match" between two paths where the paths are similar if they stay close to each other at every step along their lengths.

For a discrete calculation, this involves creating a matrix where each element (i, j) represents the distance between the i-th point on one curve and the j-th point on the other. The Discrete Fréchet Distance is then computed by evaluating the least-cost path through this matrix, which connects the beginning of both curves to their ends while minimizing the maximum distance travelled at any step. This process translates the intuitive concept of the Fréchet distance into a practical algorithm for quantitatively comparing the geometric similarity of curves [80].

These concepts are foundational in fields like geographic information systems (GIS), computer graphics, and pattern recognition, where measuring similarities between spatial curves is crucial for various applications, from map matching and trajectory analysis to animation and movement synthesis, and in this thesis, Fréchet distance is used to compute the trajectory similarity scores.

Channel-Specific Path Similarity

Each time the AD channel provides data, it includes a planned trajectory for the ego vehicle. The system checks this new trajectory against the previously received ones when new data is received to assess consistency. The expectation is that there should be a significant overlap (around 90%) between the new trajectory and the previous ones, with the new trajectory extending further as it is based on more recent data. This consistency check uses the Fréchet distance to measure how closely the trajectories align over their overlapping segments.

The trajectory data for each channel is stored in a fixed-size buffer, maintaining a history of up to ten (configurable) previous trajectories. The system calculates the Fréchet distance between the most recent trajectory and each stored trajectory, applying a decreasing weight to older trajectories. This weighted score quantitatively measures path consistency over time for each channel. The formula used to compute the similarity score is:

$$\text{Channel Specific Planned Path Similarity Scores} = \frac{1}{1 + fd}$$

where fd is the fréchet distance computed between the trajectories.

Cross-Channel Path Similarity

This component evaluates the alignment and synchronization of the most recent trajectories received from different channels. It is essential to ensure that all sensor systems and processing paths within the AV agree regarding the planned path, particularly as driving conditions change. The system adjusts for any differences in the starting positions of the ego vehicle across channels by applying a translation adjustment. It then calculates the Fréchet distance between the adjusted latest trajectories from each channel to assess their similarity.

The Fréchet distances are averaged to compute a final score for cross-channel consistency using the same normalization formula as in the channel-specific analysis:

$$\text{Cross Channel Planned Path Similarity Scores} = \frac{1}{1 + fd}$$

where fd is the fréchet distance computed between the trajectories.

By continuously monitoring both intra-channel and inter-channel trajectory consistency, the Ego Planned Path Tracker ensures that the vehicle's path planning is robust, reliable, and harmonized across all operational channels, enhancing the automated driving system's overall safety and reliability.

4.4. Safety Scores

Safety scores were already used in the automotive industry before the development of the Safety Performance Indicators (SPIs) discussed in this thesis. Although safety scores are not considered SPIs, they are crucial in determining which AD channel should be the driving channel in a cross-channel architecture. Furthermore, safety scores are vital for identifying hazardous scenarios.

Figure 4.2 illustrates the concept of Daruma cross-channel analysis [22] by comparing the high-level states of three AD channels. In this example, a driving scenario (shown at the top of the figure) is processed by three distinct AD channels, represented by the colours red, green, and blue. Each channel independently computes its high-level states, including the perceived world model, ego vehicle trajectory, and object motion predictions. By applying the Daruma design pattern, the ego vehicle trajectories proposed by each AD channel are cross-analyzed against the world models of all other channels. This cross-analysis aims to identify potential hazards or inconsistencies in the proposed actions of each channel [21].

For a 3-channel AV, the cross-channel analysis results in a 3x3 matrix of possible driving scenarios, where each cell represents the evaluation of an ego vehicle trajectory from one channel against the world model of another channel. This matrix provides a comprehensive assessment of the safety and compatibility of the proposed actions across all AD channels, enabling the system to identify the safest and most appropriate course of action for the given driving scenario [21].

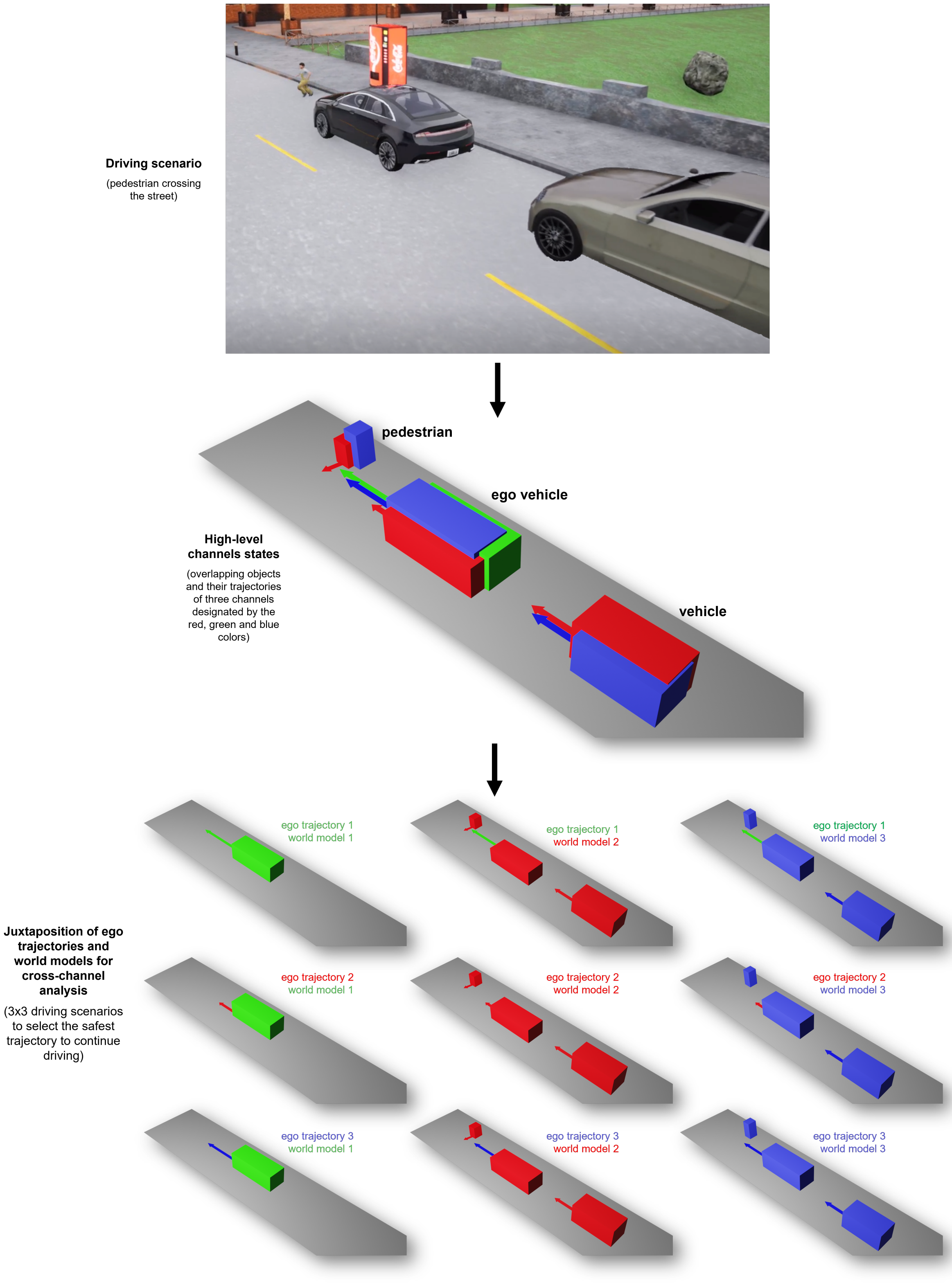


Figure 4.2: Illustration of Daruma Cross Channel Analysis [21]

The Daruma C++ implementation included the cross-channel risk analysis algorithm where the algorithm leverages the concepts of Post-Encroachment Time (PET) [81] and Time-to-Collision (TTC) [82] to identify potential collisions between the ego vehicle and other objects in the environment. PET quantifies the time difference between the moments when an ego vehicle and another object, such as a pedestrian or a vehicle, would occupy the same space, highlighting the potential for a collision if their paths were to intersect. TTC, on the other hand, calculates the time remaining until a potential collision could occur between the ego vehicle and another object, assuming both continue on their current paths and speeds.

The Daruma cross-channel risk analysis algorithm generates a risk matrix that quantifies the safety of each channel's proposed actions by evaluating the ego vehicle's trajectory from each AD channel against the world models of all other channels. This risk matrix serves as a key input to the safety fusion module proposed in this thesis, which combines the risk scores with other safety metrics to determine the most appropriate AD channel to control the vehicle at any given time.

4.5. Thresholds For Defined SPIs

Defining appropriate thresholds for each SPI is essential when using SPIs to assess an AV's safety. These thresholds act as decision boundaries, determining whether the ADS operates within acceptable safety limits or if intervention/analysis is later required to mitigate potential risks.

Setting SPI thresholds is a delicate balance between detecting true safety hazards and minimizing false positives. If the thresholds are set too loosely, the system may generate excessive false positives, incorrectly identifying safe situations as hazardous. This can lead to unnecessary interventions and extra effort needed by the manufacturer when later analysing the performance of the AV. On the other hand, if the thresholds are set too strictly, the system may fail to detect some true safety hazards.

Determining appropriate SPI thresholds involves carefully considering various factors, such as the specific ADS architecture, the operating environment, and the acceptable level of risk. It often requires a combination of theoretical analysis, simulation studies, and empirical observations to strike the right balance between detecting true safety hazards and minimizing false positives.

In the context of this thesis, the SPI thresholds have been specifically tuned for the multi-channel ADS architecture and the test bench setup described in Section 6.1. These thresholds were derived through an iterative process of manual observation and adjustment based on the performance of the AV in the simulated environment.

It is important to note that the thresholds discussed in this section are tailored to the setup used in our test bench. If a different ADS architecture or operating environment is used, these thresholds may need to be re-tuned accordingly to ensure optimal performance and safety.

Table 4.5 presents the permissible thresholds for each SPI used in our setup. All the SPIs are designed to produce a normalized scalar score between 0 and 1, where a higher score generally indicates better performance or safety (except for those SPIs that use moving variance in their algorithm). The thresholds define the minimum acceptable scores for each SPI, below which the ADS is considered to be operating in an unsafe manner.

Table 4.5: Reference SPI Thresholds

SPI Metric	Threshold	Threshold Type
Ego Location Similarity Scores (ELSS)	0.6	Lower bound (detection if below)
Moving Variance of ELSS	0.05	Upper bound (detection if above)
Planned Path Comfort Metric	0.8	Lower bound (detection if below)
Ego Trajectory Similarity Scores	0.6	Lower bound (detection if below)
Ego Orientation Similarity	0.8	Lower bound (detection if below)
Object Location Similarity Scores	0.5	Lower bound (detection if below)
Object Trajectory Similarity Scores	0.6	Lower bound (detection if below)
Object Count Similarity Scores	NA	NA
Object Orientation Similarity	0.6	Lower bound (detection if below)
Object Area Similarity	0.6	Lower bound (detection if below)
Object Motion Prediction Validation	0.5	Lower bound (detection if below)
Channel Planned Path Similarity	0.7	Lower bound (detection if below)
Cross-Channel Planned Path Similarity	0.5	Lower bound (detection if below)

The rationale for setting the above thresholds is as follows-

Safety Scores (SS): The primary goal of Safety Scores is to facilitate the arbitration process, i.e., to determine which channel should be the driving channel. As such, a definitive threshold is not set for SS, as the system will always choose the channel with the highest safety score. However, thresholds are defined for hazardous scenario identification, which will be explained in Chapter 5. The safety scores are not part of Table 4.5 as SS are part of the arbitration decision and not specifically an SPI tied to a safety case.

Ego Location Similarity Scores (ELSS): The threshold for ELSS is set to 0.6to ensure that the ego vehicle's location is consistent across channels. A lower score indicates a significant discrepancy in the perceived location, which could lead to low confidence in decision-making and potential safety hazards. Consistently low ELSS scores across multiple trips may suggest a need for the manufacturer to investigate and improve the localization algorithms or sensor calibration.

Moving Variance of ELSS: The moving variance of ELSS is set to be less than 0.05 to ensure that the ego vehicle's location remains stable over time. A higher moving variance indicates that the perceived location fluctuates significantly, which could be a sign of sensor inconsistencies or environmental factors affecting localization accuracy. This is further explained in Chapter 5. If the moving variance of ELSS consistently exceeds the threshold, the manufacturer may need to assess the robustness of the localization system and consider improvements to handle challenging environmental conditions.

Planned Path Comfort Scores (PPCS): The threshold for the Planned Path Comfort Metric is set to be greater than 0.8 to ensure that the planned trajectory is comfortable for passengers and does not involve sudden or aggressive manoeuvres. A lower score suggests that the planned path may compromise passenger comfort and potentially lead to unsafe situations. Consistently low scores may indicate a need for the manufacturer to fine-tune the motion planning algorithms to prioritize passenger comfort without compromising safety.

Ego Trajectory Similarity Scores (ETSS): The threshold for ETSS is set to 0.6 to ensure that the ego vehicle's planned trajectory is consistent across channels. A lower score indicates that the channels have different plans for the ego vehicle's future path, which could lead to conflicting decisions and potential safety risks.

Ego Orientation Similarity Scores (EOSS): A threshold of 0.8 is set because if the channels show conflicting headings beyond a difference of 20%, it could mean that the vehicles are facing completely different directions, and the planned trajectory may be unsafe for the ego vehicle. Even a momentary dip in this score is considered hazardous for the system.

Object Location Similarity Scores (OLSS): The score for OLSS is leniently set to 0.5 to account for minor discrepancies in object localization across channels. The rationale for leniency is that the number of objects could be substantially high, and a small discrepancy between them could cascade and cause a huge fall in the final score. A significantly lower score indicates that the channels perceive the location of objects differently, which could affect the accuracy of the world model and the safety of decision-making.

Object Trajectory Similarity Scores (OTSS): The threshold for OTSS is set to 0.6 to ensure that the predicted trajectories of objects are consistent across channels. A lower score suggests that the channels have different expectations for the future motion of objects, which could lead to incorrect predictions and potential collisions due to incorrectly planned trajectories.

Object Count Similarity Scores (OCSS): No threshold is set for OCSS, as this metric indicates how differently the channels perceive the world. A score of 1 indicates that both channels detect the same number of objects, while a score closer to 0 indicates that the channels see the world differently. The absence or presence of objects alone does not necessarily indicate a safety-critical situation.

Object Orientation Similarity Scores (OOSS): The threshold for OOSS is set to 0.6 to ensure that the perceived orientation of objects is consistent across channels. The leniency of this score is also for the same reason as that of OLSS, as there may be a high number of objects, and a small difference between them could cascade onto the final score. Hence, the acceptable limit is given a lot of leniency.

Object Area Similarity Scores (OASS): The threshold for OASS is set to 0.6 to ensure that the perceived size of objects is consistent across channels. A lower score suggests that the channels have a different understanding of object dimensions, which could impact the accuracy of the world model and the safety of decision-making. Consistently low OASS scores may indicate a need for the manufacturer to improve the object size estimation algorithms and ensure a more consistent understanding of the surrounding environment.

Object Motion Prediction Validation Score (OMPVC): The threshold for OMPVC is set to 0.5 to ensure that the predicted motion of objects aligns with their actual motion. A lower score indicates that the predicted trajectories deviate significantly from the observed motion, which could lead to incorrect expectations. If this score is consistently low, it indicates that the ADS's motion prediction algorithm needs to be revisited or analyzed.

Channel Planned Path Similarity Scores (ChPPS): The threshold for ChPPS is set to 0.7 to ensure that the planned path within each channel is consistent over time. A lower score suggests that the channels' planned path varies significantly between consecutive time steps, which could indicate instability or inconsistency in the planning process.

Cross-Channel Planned Path Similarity Scores (CrPPS): The threshold for CrPPS is set to 0.5 to ensure that the planned paths across channels are consistent with each other. A lower score indicates that the channels have significantly different plans for the ego vehicle's trajectory, which could indicate a safety critical scenario or one or more channels are faulty.

The thresholds for the SPIs are set in the context of safety performance indicators to balance allowing minor discrepancies and ensuring the overall safety of the system. By setting these thresholds, we aim to identify situations where the inconsistencies or deviations in the SPIs exceed acceptable levels, potentially indicating safety-critical scenarios that require attention or intervention.

4.6. Summary and Overview

Table 4.6 summarizes the SPI metrics and their targeted channels (cross-channel or single-channel)

Table 4.6: Summary of SPI Metrics and Targeted Channels

SPI Metric	Targeted Channel(s)
History Comfort Metric	Single-channel
Planned Path Comfort Metric	Single-channel
Channel Planned Path Similarity	Single-channel
Ego Location Similarity Scores	Cross-channel
Ego Trajectory Similarity Scores	Cross-channel
Ego Orientation Similarity	Cross-channel
Object Location Similarity Scores	Cross-channel
Object Trajectory Similarity Scores	Cross-channel
Object Count Similarity Scores	Cross-channel
Object Orientation Similarity	Cross-channel
Object Area Similarity	Cross-channel
Object Motion Prediction Validation	Cross-channel
Cross-Channel Planned Path Similarity	Cross-channel

The implemented Safety Performance Indicators provide a comprehensive framework for assessing the safety and performance of Automated Driving Systems. By leveraging cross-channel analysis and incorporating SPI thresholds, the SPI framework enables the continuous monitoring and improvement of ADS safety, paving the way for developing more robust and reliable automated vehicles.

Long-term monitoring of SPIs can provide valuable insights for manufacturers to identify areas for improvement in their ADS. Scores consistently over the SPI thresholds may indicate systemic issues that require attention, such as improving localization algorithms, refining object detection and tracking methods, or enhancing the consistency and coordination of path planning across channels. By addressing these issues, manufacturers can continuously improve their ADS's safety, reliability, and overall performance.

In conclusion, this chapter has presented a detailed description of the Safety Performance Indicators defined in this thesis, their relevance to the defined safety case, and the algorithms and methods employed to calculate them. The SPI framework encompasses a wide range of metrics, categorized as comfort metrics, similarity metrics, and motion prediction metrics, which collectively provide a holistic assessment of the ADS's performance and safety:

- The comfort metrics, such as acceleration and jerk, evaluate the smoothness and stability of the AV's motion
- The similarity metrics assess the consistency and agreement between the decisions and outputs of different AV subsystems or channels
- The motion prediction metrics, including object motion prediction validation and ego planned path tracking, help to evaluate the ADS's situational awareness and decision-making capabilities

5 Hazardous Scenario Identification

Hazardous scenario identification is a critical aspect of ensuring the safety and reliability of ADS. By automatically detecting and classifying potentially dangerous situations, ADS manufacturers can easily and quickly run analyses that enable them to take appropriate actions, including releasing software updates for the ADS. This chapter focuses on the hazardous scenarios that the proposed Safety Performance Indicator (SPI) framework can identify, along with an explanation of why each scenario is considered hazardous and which SPIs contribute to their identification.

5.1. Hazardous Scenarios

The SPI framework presented in this thesis is designed to detect and classify the following hazardous scenarios:

- Inconsistent Ego Localization
- Ego Tailgating
- Ego Tailgated

Each of these scenarios poses unique challenges and risks to the safe operation of an ADS, and their identification is crucial for ensuring the system's ability to navigate complex and dynamic environments.

5.1.1. Inconsistent Ego Localization

Inconsistent Ego Localization refers to a scenario where the ego vehicle's perceived location varies significantly across different sensor channels. In this situation, the ADS is unable to determine its precise location within the environment, leading to potential conflicts and uncertainties in decision-making.

This scenario becomes particularly problematic when the sensors of different channels do not agree on the ego vehicle's location or exhibit inconsistent errors between them. In such cases, the ego vehicle's position in each channel may appear to jump or drift, making it difficult for the ADS to establish a reliable estimate of its true location. Furthermore, since the ground truth of the ego vehicle's position is unknown, the AV faces the challenge of determining which channel to trust and follow. Inconsistent Ego Localization can arise due to various factors, such as sensor malfunctions, calibration errors, or environmental conditions that affect sensor performance (e.g. poor weather, reflective surfaces, or occlusions).

Inconsistent Ego Localization is considered hazardous because accurate localization is essential for an ADS to plan and execute safe manoeuvres. When the ego vehicle's location is inconsistent across channels, the system may struggle to make informed decisions about its trajectory, speed, and interactions with other road users. This can result in erratic behaviour, sudden changes in direction or speed, and an increased risk of collisions. For example, the ADS may incorrectly estimate its distance from other vehicles or obstacles, leading to inappropriate or delayed reactions. It may also struggle to maintain a stable and predictable trajectory, as the inconsistencies between sensors of different channels can cause the system to readjust its path planning and control outputs constantly.

5.1.2. Ego vehicle tailgating another vehicle (Ego Tailgating)

Ego Tailgating describes a scenario in which the ego vehicle is following another vehicle at an unsafe short distance or when one of the ADS has not detected the vehicle in front and is proposing a trajectory that can potentially cause a collision. This situation is considered hazardous because it significantly reduces the ego vehicle's ability to react to sudden changes in the lead vehicle's behaviour, increasing the risk of rear-end collisions. Figure 5.1 shows an example (taken from a route on the CARLA simulator) where the ego vehicle is tailgating another vehicle. The vehicle marked by the blue arrow is the ego vehicle, and the vehicle marked by the red arrow is that vehicle which the ego vehicle is tailgating.

A tailgating scenario is considered hazardous when at least one of the ADS proposes a planned path that could potentially lead to a collision with the vehicle in front. In Figure 5.1, the LAV channel's planned path, represented by the red line extending from the ego vehicle, overlaps with the vehicle directly ahead. The LAV channel's detections are enclosed in red bounding boxes, while the TFuse channel's detections are shown in green bounding boxes. It is important to note that the mere presence of a vehicle in close proximity to the ego vehicle does not necessarily constitute a hazardous tailgating scenario. The scenario becomes hazardous when the planned path generated by one or more ADS channels indicates a risk of collision with the leading vehicle. Section 6.1.2 provides further information on the fused world model, and the details on the CARLA routes.

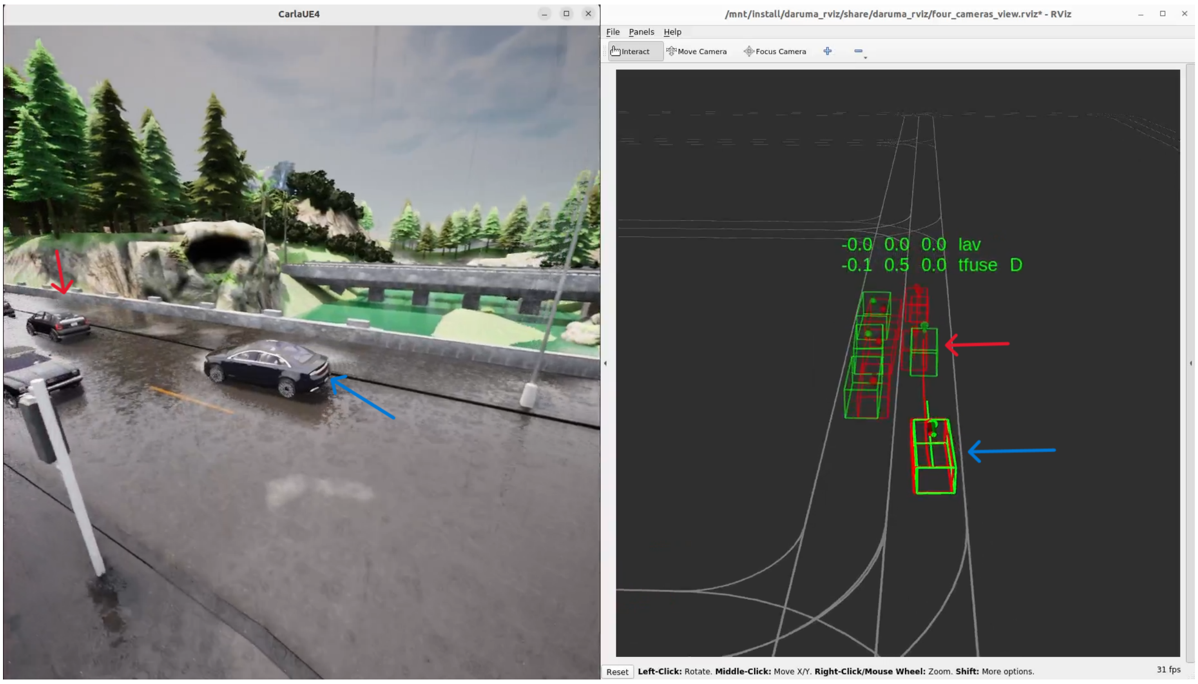


Figure 5.1: Example scenario where ego vehicle is tailgating another vehicle

When the ego vehicle is tailgating, it has less time and space to respond to events such as the lead vehicle's sudden braking, deceleration, or lane changes. This can result in the ego vehicle having to perform abrupt manoeuvres, which may compromise its stability and controllability and the safety of its occupants and surrounding road users.

5.1.3. Ego vehicle being tailgated by another vehicle (Ego Tailgated)

Ego Tailgated refers to a scenario where another vehicle follows the ego vehicle at an unsafe distance. While the ego vehicle may not have direct control over the following vehicle's behaviour, this situation still poses significant risks and challenges for the ADS. It's important to note that the ego vehicle can mitigate this risk by accelerating or changing lanes. Similarly, risk can be introduced by decelerating or moving into a lane occupied by another object. Hence, this scenario is crucial for analyzing the ADS's decision-making capabilities. Figure 5.2 shows an example (taken from a route on the CARLA simulator) where the ego vehicle is being tailgated by a truck. The vehicle marked by the blue arrow is the ego vehicle and the vehicle marked by the red arrow is the vehicle that the ego is being tailgated by.

An ego tailgated scenario is considered hazardous when at least one of the ADS predicts a proposed trajectory for the object behind the ego vehicle, where the trajectory could potentially lead to a collision with the ego vehicle. In Figure 5.2, the LAV channel predicts that the vehicle behind the ego vehicle (whose trajectory is represented by the red line extending from the object marked by the red arrow) will overlap with the ego vehicle based on its predicted trajectory. The LAV channel's detections are enclosed in red bounding boxes, while the TFuse channel's detections are shown in green bounding boxes. It is important to note that the mere presence of a vehicle in close proximity to the ego vehicle does not necessarily constitute a hazardous ego tailgated scenario. The scenario becomes hazardous when the proposed trajectory for objects generated by one or more ADS channels indicates a risk of collision with the ego vehicle. Section 6.1.2 provides further information on the fused world model. It also includes details on the CARLA routes.

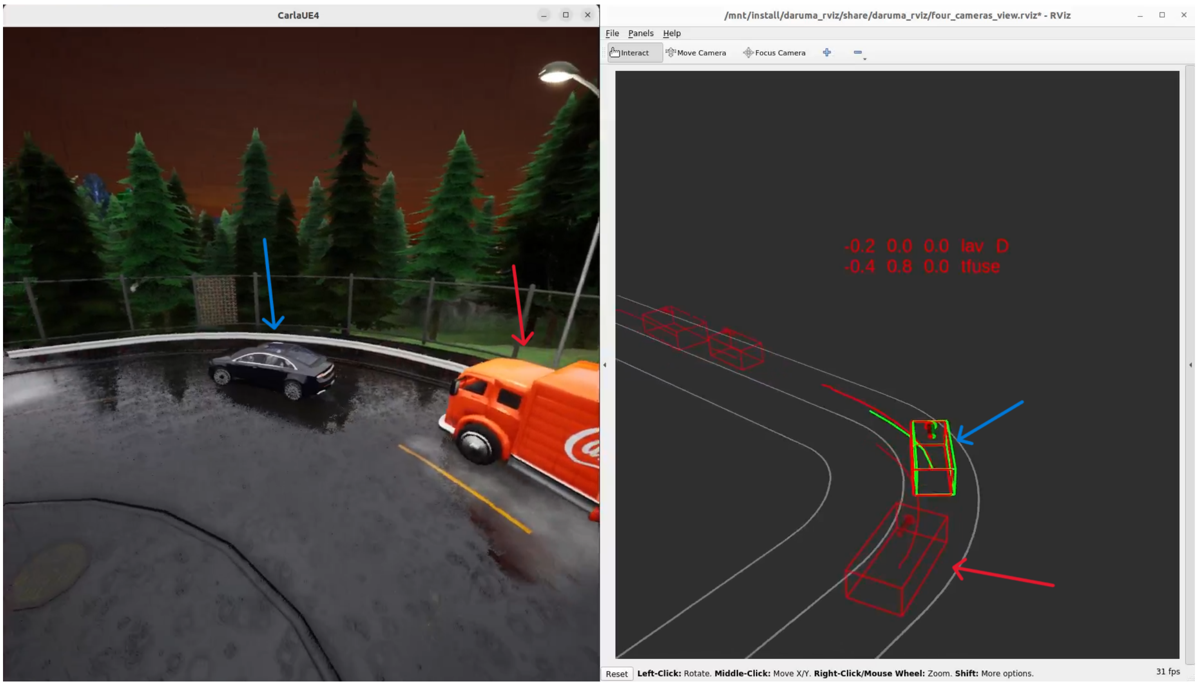


Figure 5.2: Example scenario where ego vehicle is tailgated by another vehicle

When the ego vehicle is being tailgated, it has limited room for manoeuvring and may face increased pressure to maintain or increase its speed to avoid a potential rear-end collision. This can lead to the ego vehicle making suboptimal decisions, such as accelerating unnecessarily or failing to yield to other road users when required. Furthermore, the presence of a tailgating vehicle can limit the ego vehicle's ability to perform safe and smooth decelerations, as abrupt braking may result in a collision with the following vehicle. This can be particularly problematic in situations that require the ego vehicle to slow down quickly, such as when approaching a traffic light, a stop sign, or a slower-moving vehicle.

5.2. SPIs Contributing to HSI

SPIs are crucial in identifying the hazardous scenarios described in the previous section. By monitoring and analyzing various SPIs, the ADS can detect potentially dangerous situations and take appropriate actions to mitigate risks. This section discusses the key SPIs that contribute to identifying each hazardous scenario listed in Section 5.1 and provides an intuitive explanation of how they work and why they make sense.

To understand why the SPIs contribute to the identification of hazardous scenarios, it is worthwhile to understand the following characteristics of the ADS (LAV and TFuse) described in Section 2.9.3 and Section 6.1.

- LAV can detect objects in all directions of the ego vehicle (front, back, left and right), while TFuse cannot detect objects that appear behind the vehicle. Due to this, whenever an object appears behind the vehicle, the Object Count Similarity Scores between LAV and TFuse dip.
- By default, the planned path proposed by LAV is longer (or covers a larger distance) than the path proposed by the TFuse channel. This is also an important factor in deciding the thresholds of trajectory similarity scores, as explained in Section 4.5.

5.2.1. Inconsistent Ego Localization

The primary SPI that contributes to identifying Inconsistent Ego Localization is the Ego Location Similarity Score (ELSS). ELSS measures the consistency of the ego vehicle's location across different sensor channels. When the ELSS fluctuates, it indicates that the ego vehicle's perceived location varies significantly between channels, suggesting an inconsistency in localization.

Hence, the SPI for detecting Inconsistent Ego Localization is the Moving Variance of ELSS. The Moving Variance of ELSS captures the stability and variability of location similarity over time. A high Moving Variance of ELSS indicates that the ego vehicle's perceived location fluctuates significantly across channels over a short period, which is a strong indicator of localization inconsistency. In the current setup, including the ADS configured, if the moving variance crosses the threshold of 0.05 , it indicates inconsistent Ego Localization.

Intuitively, monitoring ELSS and its Moving Variance makes sense because they directly measure the agreement between different sensor channels regarding the ego vehicle's location. If the channels provide consistent location information, the ELSS will be high, and the Moving Variance will be low. However, when there are discrepancies in the perceived location, the ELSS will decrease, and the Moving Variance will increase, signalling a potentially Inconsistent Ego Localization scenario.

5.2.2. Ego Tailgating

The SPIs that contribute to the identification of Ego Tailgating are the Safety Scores (SS) and Planned Path Comfort Scores (PPCS). When the difference between Object Count Similarity Scores (OCSS) and Object Location Similarity Scores (OLSS) is significant (e.g., greater than 0.5), and the SS is high (e.g., greater than or equal to 0.8) while the PPCS is low (e.g., less than 0.9 and greater than or equal to 0.7), it indicates that the ego vehicle is following another vehicle too closely, suggesting an Ego Tailgating scenario.

Intuitively, a high SS suggests that there is no potential collision or a collision will occur at the end of the planned trajectory, combined with a low PPCS further supports the presence of the Ego Tailgating scenario, as it suggests that the ego vehicle is maintaining an unsafe distance from the lead vehicle (and has to decelerate, causing a low PPCS), compromising both safety and comfort.

Intuitively, a SS indicates that there is no imminent risk of collision or that any potential collision is likely to occur only at the end of the planned trajectory. Furthermore, a low PPCS complements this scenario by suggesting that the ego vehicle actively manages its distance from the object ahead. This management involves proposed adjustments in acceleration and deceleration to maintain a safe following distance, as reflected by the low PPCS.

5.2.3. Ego Tailgated

The Object Count Similarity Scores (OCSS), Object Location Similarity Scores (OLSS), Safety Scores (SS), and Planned Path Comfort Scores (PPCS) contribute to the identification of the Ego Tailgating scenario. When the difference between OCSS and OSS is significant (e.g., greater than 0.5), and the SS is low (e.g., less than 0.8) while the PPCS is high (e.g., greater than or equal to 0.8), it indicates that the ego vehicle is being followed by another vehicle too closely, suggesting the ego vehicle is being tailgated.

Intuitively, a large difference between OCSS and OLSS implies that the number and location of objects detected by different channels are inconsistent, which may occur since in the setup we used, which will be described in Section 6.1, LAV cannot detect objects behind the vehicle, and hence will be reflected with a low OCSS. A low SS indicates that the future state of another object (from the trajectory) will collide with the ego vehicle. The high PPCS indicates that the vehicle continues to go at its current velocity and does not indicate any acceleration or breaking in its trajectory.

Figure 5.3 summarizes the key SPIs that contribute to identifying each hazardous scenario. By continuously monitoring and analyzing these SPIs, the framework enables the ADS to identify hazardous scenarios that the AV might have overcome in its journey.

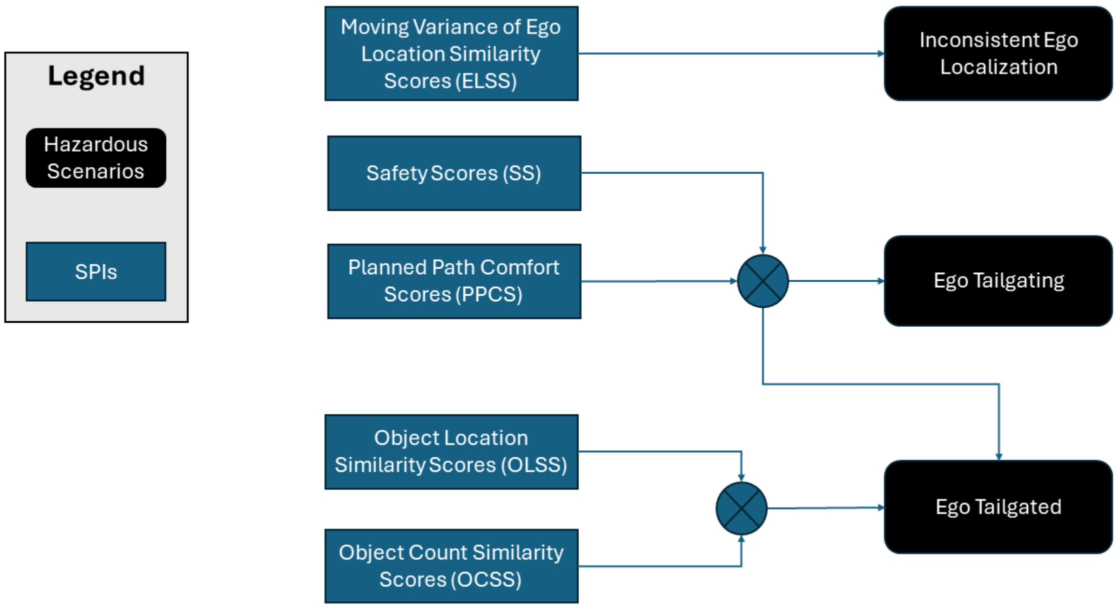


Figure 5.3: Flowchart of SPIs that contribute to the identification of the respective hazardous scenarios

5.3. OpenScenario export of hazardous scenarios

Figure 5.4 illustrates the process flow for generating .xodr and .xosc files in accordance with the OpenScenario format. This process facilitates the efficient export and analysis of hazardous scenarios identified by the Daruma C++ implementation.

As Daruma is intended to operate on an embedded processor, the generation of OpenScenario files is performed during post-processing to optimize computational resources. Creating these files for each detected scenario can take up to 12 seconds, which may introduce significant latency if executed in real time.

To facilitate the offline OpenScenario generation, the Daruma C++ implementation is designed to output a lightweight text file containing the identified scenario and encoded information about the world observed by the ADS. This approach minimizes the computational overhead during runtime while still capturing the essential data required for scenario analysis.

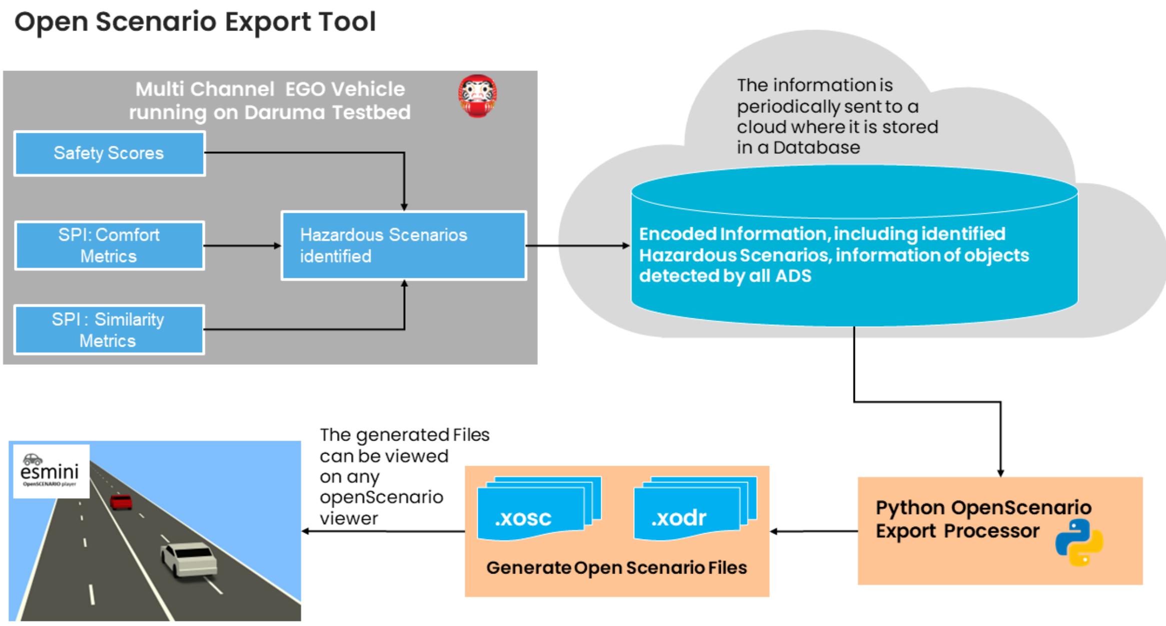


Figure 5.4: Flowchart depicting the generation of OpenScenario files from Daruma

The generated text file can be periodically transmitted to a server for centralized storage and processing. When manual analysis is required, a dedicated Python tool we developed (as explained in Section 2.9.2) is employed to decode the text file produced by Daruma. This tool extracts the encoded data and generates the corresponding OpenScenario files (.xodr and .xosc) based on the identified hazardous scenarios. By decoupling the scenario identification process from the file generation, this architecture ensures that the embedded processor running Daruma can operate efficiently without being burdened by the computationally intensive task of creating OpenScenario files in real time.

The generated OpenScenario files can be easily viewed and analyzed by stakeholders using any compatible OpenScenario reader. This enables a streamlined workflow for examining hazardous scenarios, facilitating the continuous improvement of the ADS and enhancing overall system safety.

In summary, the OpenScenario export functionality we developed on top of the existing Daruma testbed is designed to optimize performance, minimize latency, and provide a convenient means for stakeholders to access and analyze hazardous scenarios identified by the system. By leveraging a lightweight text file format and a dedicated Python tool for file generation, our architecture ensures the efficient transfer and processing of critical safety data while maintaining the real-time performance of the embedded Daruma implementation.

6

Experimental Evaluation

6.1. The Experimental Setup

6.1.1. Evaluation testbed

Figure 6.1 illustrates the architecture of the Daruma multi-channel ADS experimental setup [35]. The setup utilizes the CARLA simulator and the CARLA Leaderboard framework for scenario-based testing and evaluation. The system consists of two main AD channels: the LAV channel and the TFuse channel.

In this setup, the CARLA AD simulator generates the ego vehicle's sensor data, actuators, vehicle dynamics, and other road users' information, which is then fed into the CARLA Python RPC API. Acting as a scenario manager, the CARLA leaderboard framework communicates with the CARLA Python RPC API to control the simulation and manage the testing scenarios.

The CARLA simulator sensor data is sent to the LAV and TFuse containers via a lock-step synchronization and secondary vehicle channel control command using a barrier. The LAV and TFuse channels process the sensor data independently, generating their respective control commands. These commands are then sent to the LAV-Daruma and Transfuser-Daruma bridges, respectively.

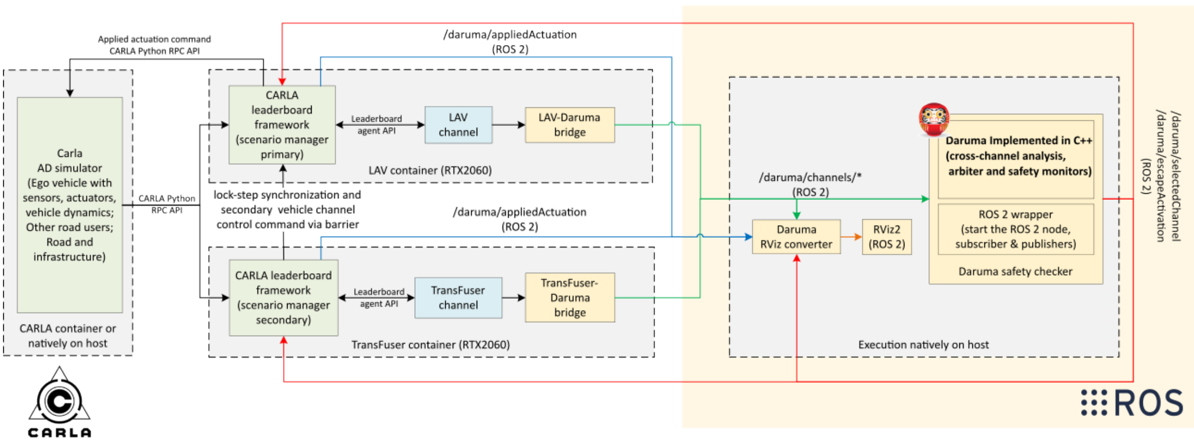


Figure 6.1: Simplified architecture of the Daruma setup

The C++ Daruma with a ROS 2 wrapper performs cross-channel analysis and arbitration to determine the safest control command. The selected command is then returned to the CARLA simulator via the respective ROS2 topic.

The Daruma safety checker also monitors the system's performance and safety using the specific ROS2 topics. This is where the algorithms of SPIs are designed (as described in Chapter 4) and implemented.

The entire setup is executed natively on the host machine, allowing for efficient testing and evaluation of the ADS's performance in various scenarios provided by the CARLA leaderboard framework.

The CARLA Leaderboard framework was utilized to evaluate the effectiveness of the developed SPIs and HSI algorithms. This framework provides a standardized set of routes and scenarios for testing ADS. The CARLA Leaderboard offers a variety of towns and routes that cover diverse driving conditions, including urban areas, highways, and residential districts, as well as different weather and lighting conditions.

Six representative routes from the CARLA Leaderboard were selected for the experimental setup, each in a different simulated town, as shown in Table 6.1. The total duration of all the individual routes adds up to 8 hours, and the total distance travelled by the ego vehicle is approximately 8.24 Km. The routes are explained in subsequent subsections.

These routes were chosen to comprehensively assess the SPI and HSI algorithms' performance across various driving scenarios. The driving scenarios of each selected route were recorded in the format of ROSbags while a two-channel AV navigated through the CARLA simulator to facilitate the evaluation process. Rosbags are a convenient tool for storing and replaying sensor data, allowing for offline analysis of the vehicle's performance in a reproducible manner.

Table 6.1: Summary of routes selected for the experimental evaluation

Town	Weather	Time of Day	Total Duration (hr)	Distance of the Route (meters)
1	Hard rain	Noon	0.60	732
2	Hard rain	Sunset	1.23	973
3	Mid rain	Twilight	1.91	1748
4	Soft rain	Night	1.51	1862
5	Mid rain	Morning	1.11	1071
6	Wet	Dawn	1.59	1859

6.1.2. Evaluation Methodology

The SPI and HSI algorithms were evaluated through a combination of manual analysis and automated processing of the recorded ROSbags. For each route, the following steps were performed:

1. **Manual Analysis:** The recorded videos of each route were visually inspected to identify potentially hazardous scenarios and assess the overall safety performance of the AV. This manual analysis served as a ground truth for evaluating the effectiveness of the SPI and HSI algorithms. During the manual analysis, all the hazardous scenarios encountered and the instances where the safety of the vehicle was compromised were noted.

   Figure 6.2 shows the screens used for the manual evaluation. The screen on the left shows the route generated by the CARLA leaderboard. The vehicle pointed by the blue arrow is the Ego Vehicle. The screen on the right displays the combined/fused world model from both the ADS-LAV and TFuse systems. It shows objects detected by each system's channels. The Ego Vehicle is indicated by the blue arrow. Objects detected by the LAV AD channel are enclosed in red bounding boxes, while those detected by the TFuse AD channel appear within green bounding boxes. Additionally, the planned path proposed by the TFuse channel is depicted as a green line extending forward from the Ego Vehicle, and the LAV channel's proposed path is marked by a red line.

2. **Automated Hazardous Scenarios Logging Process:** The corresponding Rosbag for each of the routes mentioned in Table 6.1 were played, and the results from Daruma were obtained in the text file format. Note that the Daruma cross-channel analysis is executed every 0.1s with the world information, which is the frequency of the ADS. Hence, basic filtering algorithms are executed based on the results from Daruma to remove repeated detections. The filtering algorithms are

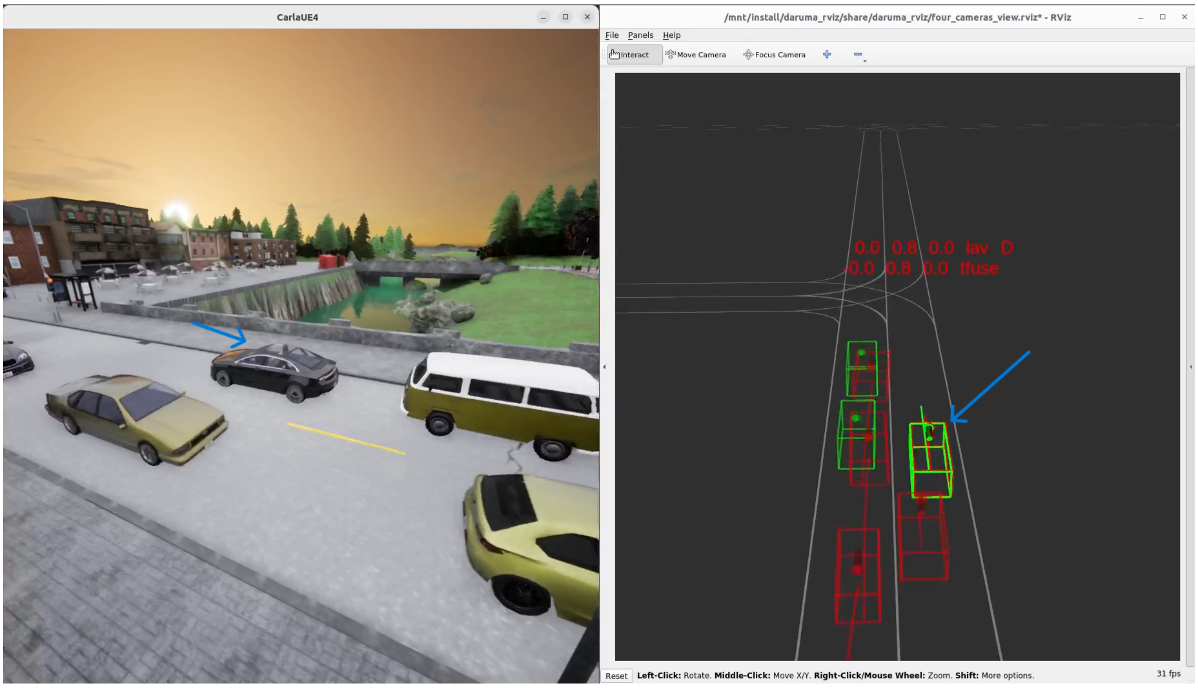


Figure 6.2: Screenshot of the manual HSI identification process on the Daruma multi-channel AD test bed

time-based, where if the same scenario is detected for a prolonged period, it is counted as one entity. This is applicable to hazardous scenario identification as well as the SPI threshold violation

3. **Comparison and Evaluation:** The results obtained from the automated processing were compared against the manual analysis to determine the accuracy and effectiveness of the SPI and HSI algorithms. For each route, the number of correctly identified hazardous scenarios missed detections (false negatives), and incorrect detections (false positives) were assessed. This evaluation helped understand the algorithms' strengths and limitations in identifying hazardous scenarios and assessing the vehicle's safety performance. Furthermore, when an SPI falls below its threshold, the ground truth and the ADS information are also analysed to check if any insights can be derived.

The evaluation results for each route were analyzed individually to gain insights into the algorithms' performance under different driving conditions and scenarios. This route-wise analysis identified the strengths and limitations of the SPI and HSI algorithms and their applicability to real-world ADS.

6.2. Evaluation results and discussion

This section presents and discusses the evaluation results for each of the six selected routes. For each route, an overview of the driving scenario, the manual analysis findings, and the performance of the SPI and HSI algorithms in identifying hazardous scenarios and assessing the vehicle's safety performance are provided.

In the experimental evaluation of the HSI module, we use the metrics of Precision and Recall to quantitatively assess its performance. These metrics are particularly valuable in scenarios where it is critical to distinguish between accurately detected hazards and false identifications [83].

- **Precision** quantifies the accuracy of the positive identifications made by the HSI module. It measures the proportion of true positive identifications among all the instances that the module classified as hazardous, and it is defined as the ratio of true positives (i.e., hazards correctly identified as hazardous) to the total number of items identified as hazardous (both correctly and incorrectly). A high Precision indicates that when the HSI module identifies a scenario as hazardous, it is highly likely to be an actual hazardous scenario. Precision is calculated using:

$$Precision = \frac{TP}{TP + FP}$$

Where TP is the number of True Positives, and FP is the number of False Positives [84]. However, Precision alone does not tell us whether the module is identifying all the hazardous scenarios present.

- **Recall** measures the completeness of the HSI module in identifying all actual hazardous scenarios. It measures the proportion of true positive identifications among all the actual hazardous scenarios present in the data. It is calculated as the ratio of true positives to the total actual hazards present in the scenarios, which includes those not identified by the module. A high Recall indicates that the HSI module is capable of identifying a large portion of the hazardous scenarios, minimizing the number of false negatives (i.e., hazardous scenarios that are missed by the module). This is crucial for ensuring that no hazardous scenarios are overlooked . Recall is calculated using:

$$Recall = \frac{TP}{TP + FN}$$

Where TP is the number of True Positives, and FN is the number of False Negatives [84].

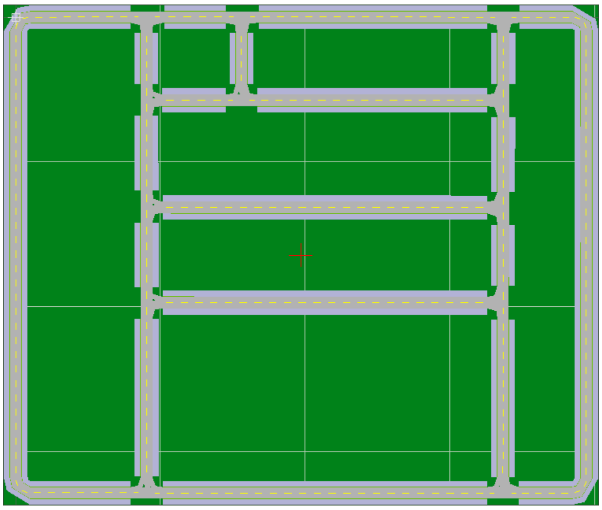
Together, Precision and Recall provide a comprehensive view of the HSI module's performance, indicating how trustworthy its identifications are (Precision) and how effective it is at detecting all relevant hazards (Recall).

6.2.1. Experimental Evaluation of the Route in Town 1

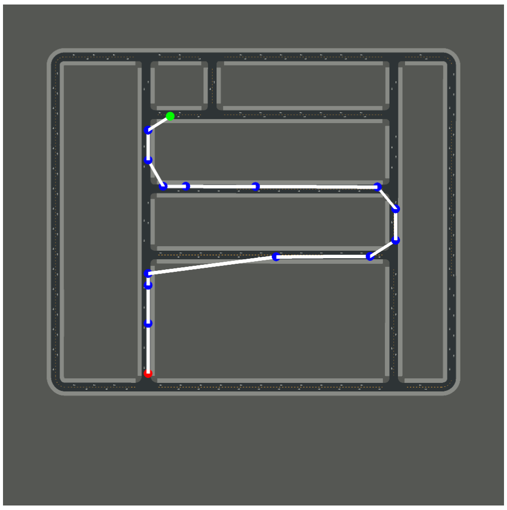
Scenario Overview

Route 1 (as seen in Figure 6.4b) is situated in CARLA Town 1 (as seen in Figures 6.3, and 6.4a), a small town with numerous T-junctions and a variety of buildings. The route takes place during noon and features heavy rainfall [85]. The ego vehicle navigates through the town for approximately 0.60 hours.

Figure 6.3: Map of the CARLA town 1 [85]



(a) Layout of the town.



(b) The route followed by the ego vehicle.

Figure 6.4: The CARLA town 1 map (a) and the path followed by ego vehicle (b)

Results

Table 6.2 shows the number of hazardous scenarios classified by manually observing the ground truth (CARLA simulator) and the automated HSI modules. Table 6.3 shows the results of the SPI module when the set threshold (discussed in Section 4.5) was crossed.

Table 6.2: Results of HSI for CARLA town 1

Results/Hazardous Scenario	**Ego Vehicle Tailgated**	**Inconsistent Ego Localization**	**Ego Vehicle Tailgating**
Manually observed number of scenarios considered hazardous on CARLA Simulator	6	NA	4
Manually observed number of scenarios considered hazardous in the combined world model of ADS	10	2	4
Number of scenarios correctly identified by the HSI module (automated)	10	1	3
Number of false negatives by the HSI module (automated)	0	1	1
Number of false positives by the HSI module (automated)	0	0	2
Precision of the HSI module	100.00 %	100.00 %	60.00 %
Recall of the HSI module	100.00 %	50.00 %	75.00 %

The HSI module successfully detected 100% of the ”Ego Tailgated” scenarios manually observed in the combined world model of the Autonomous Driving System (ADS). This indicates high accuracy in identifying situations where the ego vehicle was being tailgated. The difference in the number of scenarios manually identified on the CARLA simulator versus those manually identified in the fused/-combined world model of the ADS is due to errors in the ADS's perception system. Specifically, the ADS incorrectly misclassified stationary objects on the sidewalks, such as dustbins, as moving objects with predicted trajectories that seemingly followed the ego vehicle. This misclassification by the ADS represents an inconsistency within its perception system. Figure 6.5 shows an example where a pot to the right of the Ego vehicle is misclassified as a dynamic object with a predicted trajectory. Despite these misclassifications by the ADS, the HSI module accurately identified the scenario as presented by the ADS. This accuracy is due to the HSI module's design, which does not possess knowledge of the ground truth but relies solely on the data and interpretations provided by the ADS. Thus, even if the ADS delivers flawed or inaccurate data, the HSI module evaluates and responds based on that information alone.

However, in the case of the ”Inconsistent Localization” hazardous scenario, the ADS identified two scenarios. Out of these, only one was correctly identified, leading to a precision of 100% and recall of 50% for this type of scenario. The missed scenario (resulting in a false negative) involved a situation where the Ego Localization Stability Score (ELSS) showed fluctuations. Although the ELSS came close to the threshold necessary to trigger a detection, it did not cross this threshold. Consequently, the HSI module did not detect this hazardous scenario, marking it as a false negative.

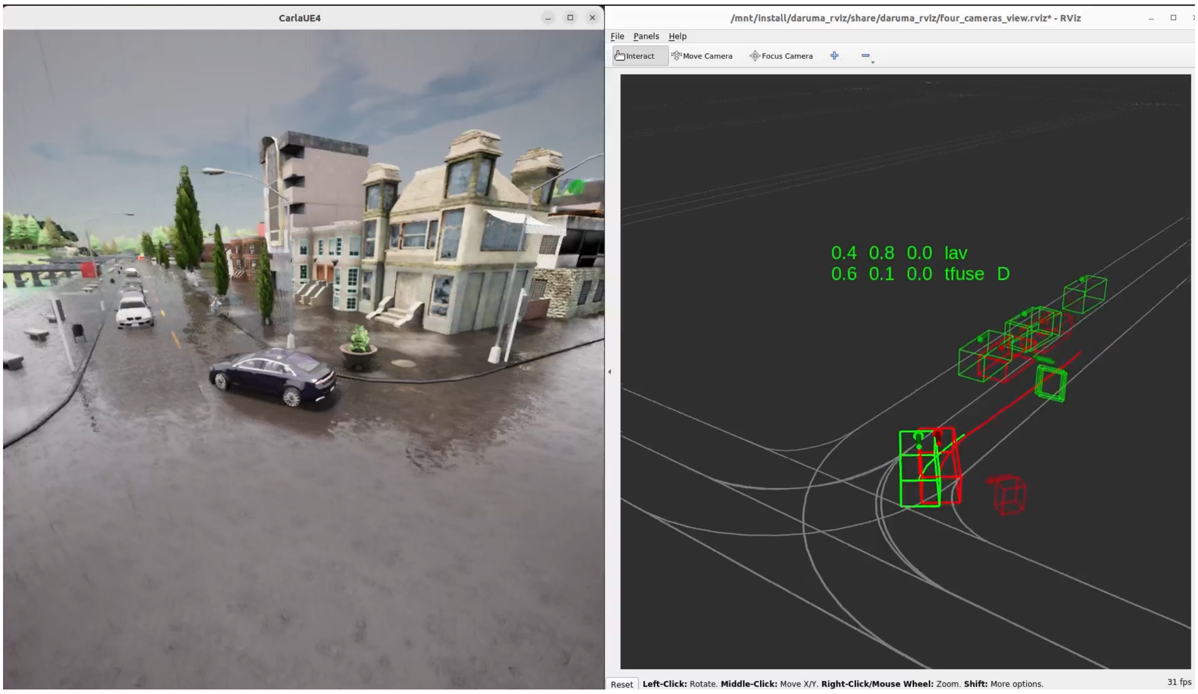


Figure 6.5: Example scenario of ADS misclassification

For the scenario of Ego tailgating, 4 scenarios were manually observed on CARLA Simulator, and the same scenarios were detected by the ADS in the combined world model. 3 of these scenarios were correctly identified. The one false negative scenario occurred because, while there was a vehicle in front of the ego vehicle, the distance to the ego vehicle was large, and hence, the safety scores did not go below the threshold as the TTC values were high. In addition to the false negative, there were two false positives obtained in the automated process:

- During a Turn: One false positive was detected when an oncoming vehicle's predicted path aligned with the ego vehicle's planned path while the ego vehicle was turning. This alignment caused a drop in the safety score because the system predicted a potential collision.

- At a Red Traffic Light: When the light turned red, and another vehicle was behind the ego vehicle, the PPCS decreased as the ego vehicle decelerated. Concurrently, the safety scores dropped due to the algorithm predicting a collision trajectory with the vehicle behind, which was incorrectly perceived as a potential hazard.

This leads to a precision of 60% and recall of 75% for ”Ego Tailgating” detection.

Table 6.3: Results of SPI monitoring for CARLA town 1

SPI	# Threshold Violations	Cause
Ego Location Similarity Scores	6	Violations occurred due to inconsistent localization, either prolonged or short-term.
Ego Orientation Similarity Scores	0	
Object Count Similarity Scores	4	Violations occurred when TFuse or LAV misidentified stationary objects on the sidewalk as obstacles, though these were irrelevant to the driving scenarios.
Object Location Similarity Scores	2	Violations occurred when both the ADS detected a high number of objects near the ego vehicle.
Object Orientation Similarity Scores	0	
Object Area Similarity Scores	0	
Objected Motion Prediction Validation Score	8	Violations occurred when many objects exited the ADS perception field.
Ego Trajectory Similarity Scores	4	Violations occurred during stoplight periods.
Object Trajectory Similarity Scores	4	Violations occurred when numerous detected objects led to small errors that compounded, causing trajectory scores to drop below the threshold.
Channel Specific Ego Planned Path Tracker Scores	3	Violations occurred at red lights for LAV and during turns for TFuse.
Cross Channel Ego Planned Path Tracker Score	4	Violations occurred during red lights and turns.
Planned Path Comfort Scores	6	Violations were inherently linked to the ADS’s handling of scenarios involving tailgating, red lights, and turns.
History Comfort Scores	8	Violations were inherently linked to the ADS’s handling of scenarios involving tailgating, red lights, and turns.

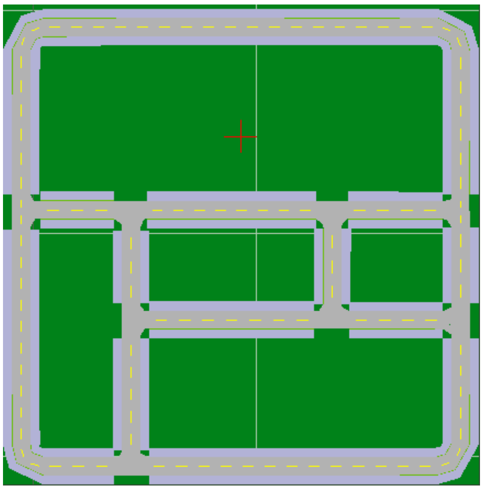
Table 6.3 summarizes the number of threshold violations for each SPI defined and the corresponding causes during the route. These results provide valuable insights into the ADS's performance in detecting hazardous scenarios and the causes of SPI threshold violations. The high accuracy in detecting hazardous scenarios demonstrates the ADS's effectiveness in these situations.

6.2.2. Experimental Evaluation of the Route in Town 2

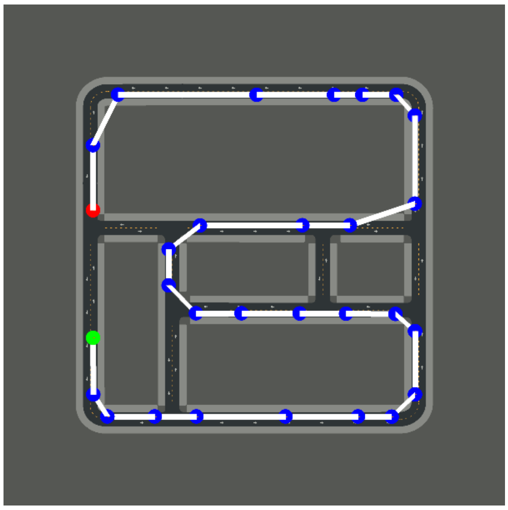
Scenario Overview

Route 2 (as seen in Figure 6.7b) takes place in Town 2 (as seen in Figures 6.6, and 6.7a), a small town with numerous T-junctions and the route is set during sunset and features heavy rainfall [86]. The ego vehicle spends around 1.23 hours traversing this route.

Figure 6.6: Map of the CARLA Town 2 [86]



(a) Layout of the town.



(b) The route followed by the ego vehicle.

Figure 6.7: The CARLA town 2 map (a) and the path followed by ego vehicle (b)

Results

Table 6.4 shows the number of hazardous scenarios classified by manual observation of CARLA simulation and the automated HSI modules. Table 6.5 shows the results of the SPI module when the set threshold (discussed in Section 4.5) was crossed.

Table 6.4: Results of HSI for CARLA town 2

Results/Hazardous Scenario	Ego Vehicle Tailgated	Inconsistent Ego Localization	Ego Vehicle Tailgating
Manually observed number of scenarios considered hazardous on CARLA Simulator	14	NA	5
Manually observed number of scenarios considered hazardous in the combined world model of ADS	17	3	5
Number of scenarios correctly identified by the HSI module (automated)	17	2	3
Number of false negatives by the HSI module (automated)	0	1	2
Number of false positives by the HSI module (automated)	2	0	2
Precision of the HSI module	89.47 %	100.00 %	60.00 %
Recall of the HSI module	100.00 %	66.67 %	60.00 %

The HSI module achieved a detection recall of 100% for ”Ego Vehicle Tailgated” scenarios, indicating perfect alignment with the observations from the CARLA Simulator. This scenario highlights the ADS's robust perception capabilities in specific contexts. The difference in the number of scenarios identified on the CARLA simulator versus those identified by the ADS is due to the errors in the ADS's perception system.

The two false negative scenarios for ”Ego Vehicle Tailgating” occurred because, while there was a vehicle in front of the ego vehicle, the distance to the ego vehicle was large, and hence, the safety scores did not go below the threshold as the TTC values were high. There were 2 false positives detected, one of which was due to a pedestrian crossing the street. While momentarily this could be classified as Ego tailgating, since the pedestrian is ahead of the EGO vehicle, it strictly does not fall in its definition and hence is categorized as false positive. The other scenario occurred at a red traffic light. As soon as the traffic light turned red, and there was another vehicle behind the ego vehicle, the PPCS score dropped due to the de-acceleration of the planned path, and the safety scores fell due to the trajectory of the vehicle being that was predicted to collide with the ego vehicle. This leads to a precision of 60% and recall of 60% for ”Ego Vehicle Tailgating” detection.

For the ”Inconsistent Ego Localization” scenario, the HSI module achieved a precision of 100% and recall of 66.67%, correctly identifying two out of three scenarios observed by the ADS.

Table 6.5: Results of SPI monitoring for CARLA town 2

SPI	# Threshold Violations	Cause
Ego Location Similarity Scores	9	Violations occurred due to inconsistent localization, either prolonged or short-term.
Ego Orientation Similarity Scores	0	
Object Count Similarity Scores	16	Violations occurred when TFuse or LAV misidentified stationary objects on the sidewalk as obstacles, though these were irrelevant to the driving scenarios.
Object Location Similarity Scores	5	Violations occurred when both the ADS detected a high number of objects near the ego vehicle.
Object Orientation Similarity Scores	0	
Object Area Similarity Scores	0	
Objected Motion Prediction Validation Score	6	Violations occurred when many objects exited the ADS perception field.
Ego Trajectory Similarity Scores	7	Violations occurred during stoplight periods.
Object Trajectory Similarity Scores	9	Violations occurred when numerous detected objects led to small errors that compounded, causing trajectory scores to drop below the threshold.
Channel Specific Ego Planned Path Tracker Scores	7	Violations occurred at red lights for LAV and during turns for TFuse.
Cross Channel Ego Planned Path Tracker Score	11	Violations occurred during red lights and turns.
Planned Path Comfort Scores	7	Violations were inherently linked to the ADS's handling of scenarios involving tailgating, red lights, and turns.
History Comfort Scores	12	Violations were inherently linked to the ADS's handling of scenarios involving tailgating, red lights, and turns.

Table 6.5 summarizes the number of threshold violations for each SPI defined and the corresponding causes during the route.

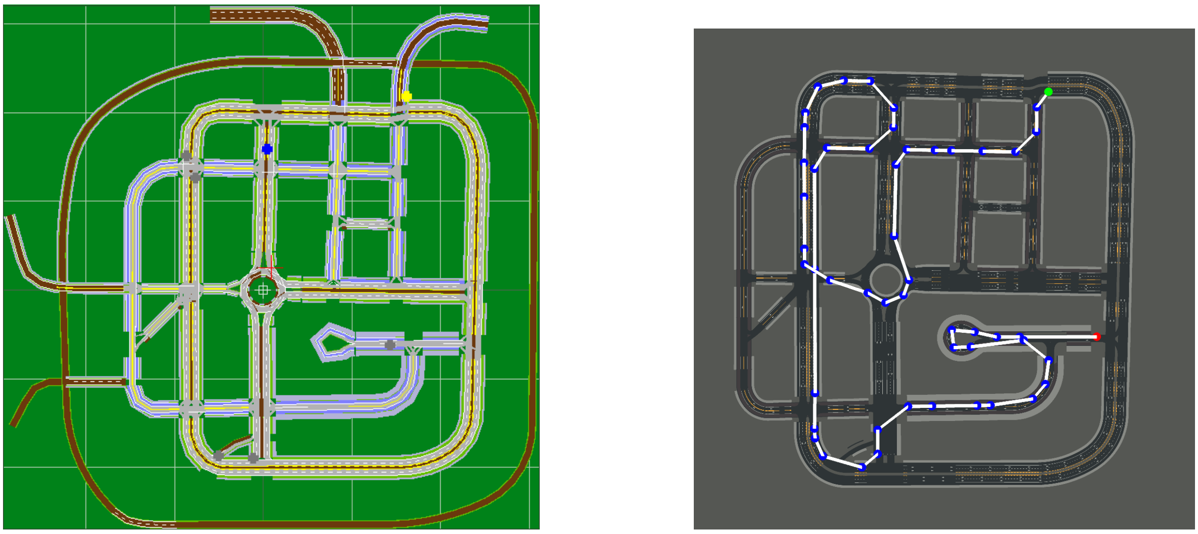
The ”Ego Trajectory Similarity Scores” and ”Object Trajectory Similarity Scores” had seven and nine violations during stop lights and when there were too many objects, respectively, causing small errors to cascade and lower the scores below the threshold.

6.2.3. Experimental Evaluation of the Route in Town 3

Scenario Overview

Route 3 (as seen in Figure 6.9b) is located in Town 3 (as seen in Figures 6.8 and 6.9a), a larger town with features of a downtown urban area. The town includes interesting road network features such as a roundabout, underpasses, and overpasses. The route occurs during twilight and features moderate rainfall [87]. The ego vehicle navigates through this town for approximately 1.91 hours.

Figure 6.8: Map of the CARLA Town 3 [87]



(a) Layout of the town. **(b)** The route followed by the ego vehicle.

Figure 6.9: The CARLA town 3 map (a) and the path followed by ego vehicle (b) used for the evaluation

Results

Table 6.6 shows the number of hazardous scenarios classified by manually observing the ground truth (CARLA simulator) and the automated HSI modules. The HSI module demonstrated mixed performance in detecting hazardous scenarios in Town 3. For the ”Ego Vehicle Tailgated” scenario, the module achieved a precision of 100% and recall of 84.61% concerning the combined ADS world model. Two tailgated events were missed due to threshold settings, and one tailgated event observed in the CARLA simulator was not detected by the ADS.

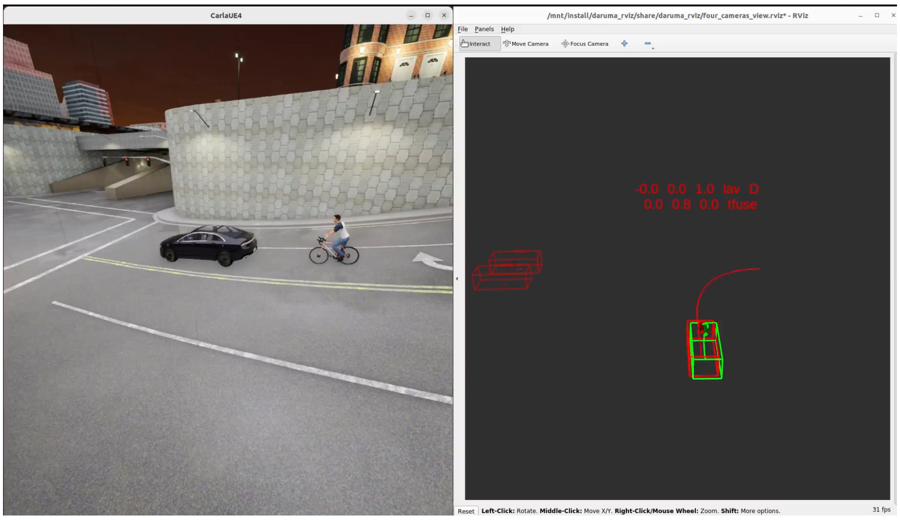
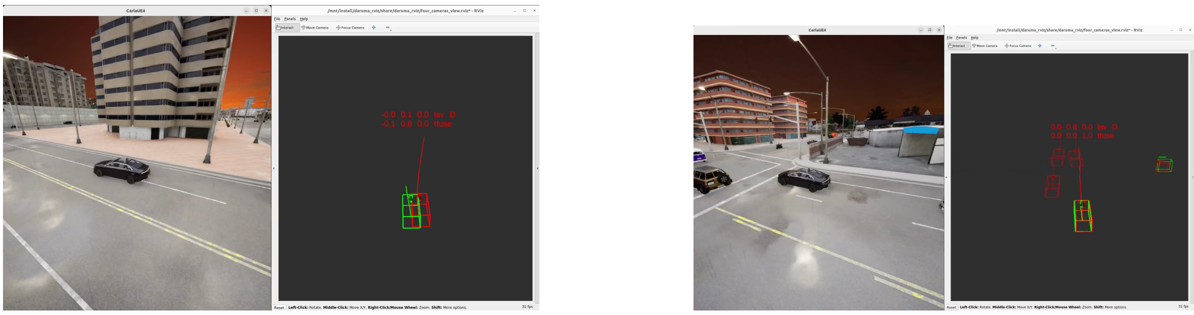


Figure 6.10: Senario where an object (bicycle) was missed by both the ADS.

In the "Ego Vehicle Tailgating" scenario, the HSI module achieved a precision of 80% and recall of 66.67%, correctly identifying four out of six scenarios observed by the ADS. There were two false negatives due to threshold settings and one false positive where an object crossing was misclassified as tailgating. Additionally, there was one instance of ghost tailgating, where the ADS incorrectly detected an object not present in the CARLA simulator. For the "Inconsistent Ego Localization" scenario, the HSI module demonstrated a precision and recall of 100%, correctly identifying both scenarios observed by the ADS.

Table 6.7 summarizes the threshold violations for each SPI defined and the corresponding causes during the route.



(a) (b)

Figure 6.11: Example scenarios where trajectory similarity threshold (a) and object count scores threshold (b) were exceeded

Figure 6.11a illustrates a scenario where each ADS suggests an alternate trajectory for the ego vehicle, which may pose a hazard (captured when there was a cross-channel ego planned path score threshold violation). Meanwhile, Figure 6.11b (captured when there was an object count score threshold violation) depicts a case where only LAV detects three vehicles to the left and ahead of the ego vehicle, while TFuse fails to identify them. This discrepancy provides manufacturers with valuable insights, prompting them to examine additional data, such as sensor specifics, to determine why the ADS did not detect these vehicles.

These results highlight the challenges the HSI module faces in accurately detecting hazardous scenarios in complex urban environments like Town 3. The presence of ghost objects and misclassifications underscores the need for further improvements in the ADS's perception and classification capabilities. The SPI threshold violations provide valuable insights into the ADS's performance and the impact of specific driving conditions on the system's safety and reliability.

Table 6.6: Results of HSI for CARLA town 3

Results/Hazardous Scenario	**Ego Vehicle Tailgated**	**Inconsistent Ego Localization**	**Ego Vehicle Tailgating**
Manually observed number of scenarios considered hazardous on CARLA Simulator	12	NA	4
Manually observed number of scenarios considered hazardous in the combined world model of ADS	13	2	6
Number of scenarios correctly identified by the HSI module (automated)	11	2	4
Number of false negatives by the HSI module (automated)	2	0	2
Number of false positives by the HSI module (automated)	0	0	1
Precision of the HSI module	100.00 %	100.00 %	80.00 %
Recall of the HSI module	84.61 %	100.00 %	66.67 %

Table 6.7: Results of SPI monitoring for CARLA town 3

SPI	# Threshold Violations	Cause
Ego Location Similarity Scores	4	Violations occurred due to inconsistent localization, either prolonged or short-term.
Ego Orientation Similarity Scores	0	
Object Count Similarity Scores	15	Violations occurred when TFuse or LAV misidentified stationary objects on the sidewalk as obstacles, though these were irrelevant to the driving scenarios.
Object Location Similarity Scores	6	Violations occurred when both the ADS detected a high number of objects near the ego vehicle.
Object Orientation Similarity Scores	0	
Object Area Similarity Scores	0	
Objected Motion Prediction Validation Score	11	Violations occurred when many objects exited the ADS perception field.
Ego Trajectory Similarity Scores	3	Violations occurred during stoplight periods.
Object Trajectory Similarity Scores	17	Violations occurred when numerous detected objects led to small errors that compounded, causing trajectory scores to drop below the threshold.
Channel Specific Ego Planned Path Tracker Scores	4	Violations occurred at red lights for LAV and during turns for TFuse.
Cross Channel Ego Planned Path Tracker Score	11	Violations occurred during red lights and turns.
Planned Path Comfort Scores	8	Violations were inherently linked to the ADS's handling of scenarios involving tailgating, red lights, and turns.
History Comfort Scores	14	Violations were inherently linked to the ADS's handling of scenarios involving tailgating, red lights, and turns.

6.2.4. Experimental Evaluation of the Route in Town 4

Scenario Overview

Route 4 (as seen in Fig 6.13b) is situated in Town 4 as seen in Fig 6.12 and 6.13a), a small town with a backdrop of snow-capped mountains and conifers. A multi-lane road circumnavigates the town in a "figure of 8" style. The route takes place during nighttime and features light rainfall. The ego vehicle spends around 1.51 hours navigating the road network, which consists of a small network of short streets and junctions, with a "figure of 8" style ring road circumnavigating the buildings and a nearby mountain. The cross of the figure 8 presents an underpass/overpass and circular slip roads, testing the ego vehicle's ability to operate in low-light conditions while maintaining safe speeds and distances.

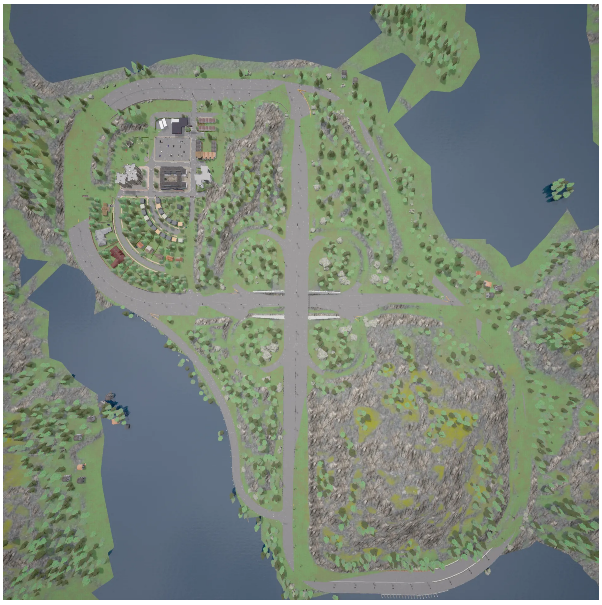
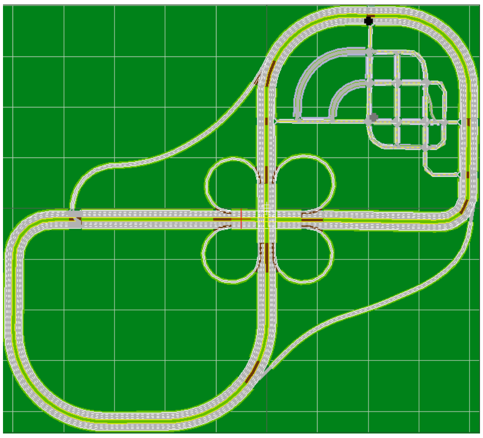
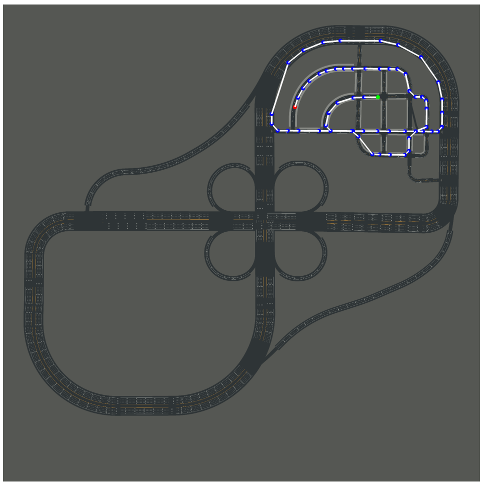


Figure 6.12: Map of the CARLA Town 4 [88]



(a) Layout of the town.



(b) The route followed by the ego vehicle.

Figure 6.13: The CARLA town 4 map (a) and the path followed by ego vehicle (b) used for the evaluation

Results

Table 6.8 shows the number of hazardous scenarios classified by manually observing the ground truth (CARLA simulator) and the automated HSI modules.

The HSI module's performance in Town 4 was influenced by the presence of red lights and the ADS's perception limitations in low-light conditions. For the "Ego Vehicle Tailgated" scenario, the module achieved a precision and recall of 92.30%, correctly identifying 12 out of 13 scenarios observed by the ADS. There was one false negative due to a red light and one false positive where the CARLA simulator didn't see the car, but the ADS detected a ghost object.

In the "Ego Vehicle Tailgating" scenario, the HSI module achieved a precision of 57.14% and recall of 80%, correctly identifying four out of five scenarios observed by the ADS. There was one false negative

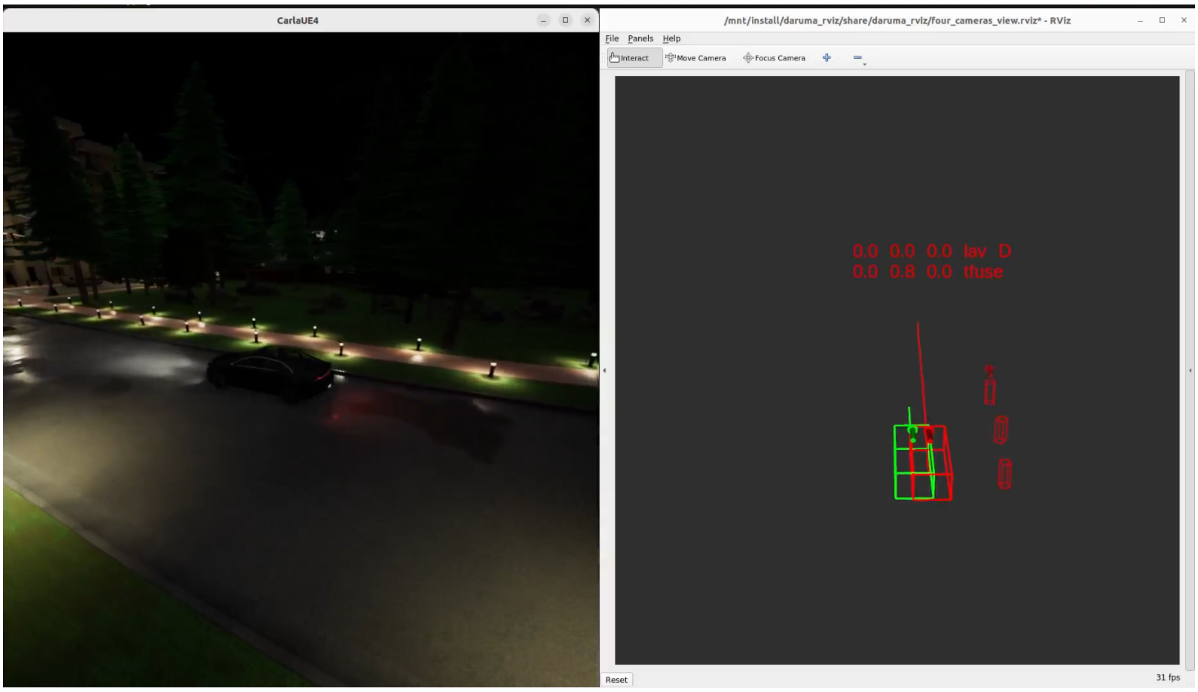


Figure 6.14: Scenario where object count score exceeded the threshold as LAV identified objects on the sidewalk.

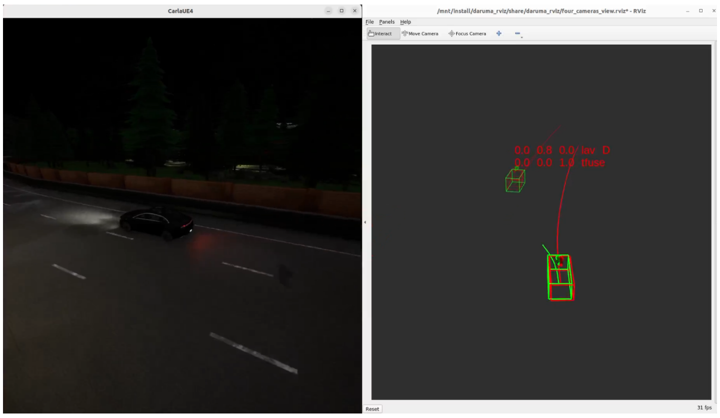
due to a red light and three false positives, primarily caused by the ego vehicle's behaviour at red lights.

For the "Inconsistent Ego Localization" scenario, the HSI module demonstrated a precision of 100% and recall of 88.89%, correctly identifying 8 out of 9 scenarios observed by the ADS. There was one false negative due to a threshold setting.

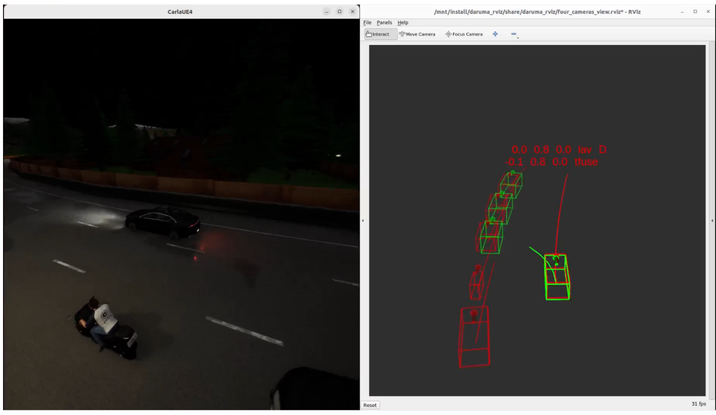
The TFuse channel exhibited significant ego localization issues in the nighttime conditions of Town 4, indicating its sensitivity to low-light environments. This highlights the need for robust localization techniques that consistently perform across different lighting conditions.

Table 6.9 summarizes the threshold violations for each SPI defined and the corresponding causes during the route. Figure 6.14 shows a sample scenario where the object count score crossed the threshold mainly because TFuse identified objects on the side, but LAV did not. This pattern was observed throughout the journey

The "Channel Specific Ego Planned Path Tracker Scores" and "Cross Channel Ego Planned Path Tracker Score" encountered fourteen and twenty-two violations. Upon reviewing the exported Open-SCENARIO files, the object locations initially appeared identical. However, closer examination of the ADS world model revealed conflicting paths proposed by the ADS, as depicted in Figures 6.15a and 6.15b. This pattern of conflicting trajectory decisions was consistently observed along this route, particularly under low visibility conditions, where TFuse's object detection performance diminished.



(a)



(b)

Figure 6.15: Two scenarios where SPI algorithms identified significantly different ADS-guided paths for the vehicle

In this route, all the SPI parameters were almost stable after a certain period, with only fluctuations in safety scores and object count scores. On closer examination manually, it was found that the ADS had collided with an object in front of it (as seen in Figure 6.16), and then it detected it as a vehicle with no trajectory being generated. The Object Count scores fluctuated due to the vehicles passing it. It is due to this reason that the number of detections for HSI and SPI is low in this town.

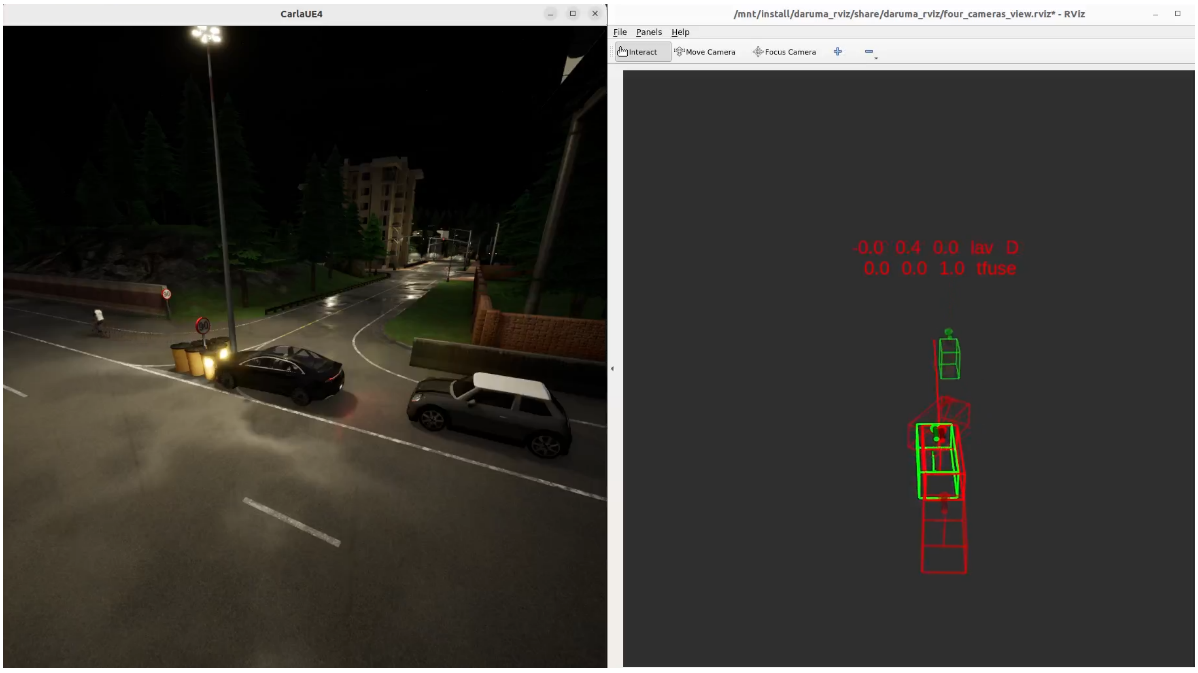


Figure 6.16: Scenario where the vehicle collided in Town 4

This presents a clear example of how monitoring SPI thresholds effectively identified issues with the ADS world model detection, prompting a detailed analysis and highlighting opportunities for enhancing the ADS system.

Table 6.8: Results of HSI for CARLA town 4

Results/Hazardous Scenario	**Ego Vehicle Tailgated**	**Inconsistent Ego Localization**	**Ego Vehicle Tailgating**
Manually observed number of scenarios considered hazardous on CARLA Simulator	9	NA	5
Manually observed number of scenarios considered hazardous in the combined world model of ADS	13	9	5
Number of scenarios correctly identified by the HSI module (automated)	12	8	4
Number of false negatives by the HSI module (automated)	1	1	1
Number of false positives by the HSI module (automated)	1	0	3
Precision of the HSI module	92.30 %	100.00 %	57.14 %
Recall of the HSI module	92.30 %	88.89 %	80.00 %

Table 6.9: Results of SPI monitoring for CARLA town 4

SPI	# Threshold Violations	Cause
Ego Location Similarity Scores	20	Violations occurred due to inconsistent localization, either prolonged or short-term.
Ego Orientation Similarity Scores	0	
Object Count Similarity Scores	4	Violations occurred when TFuse or LAV misidentified stationary objects on the sidewalk as obstacles, though these were irrelevant to the driving scenarios.
Object Location Similarity Scores	12	Violations occurred when both the ADS detected a high number of objects near the ego vehicle.
Object Orientation Similarity Scores	4	
Object Area Similarity Scores	0	
Objected Motion Prediction Validation Score	8	Violations occurred when many objects exited the ADS perception field.
Ego Trajectory Similarity Scores	4	Violations occurred during stoplight periods.
Object Trajectory Similarity Scores	18	Violations occurred when numerous detected objects led to small errors that compounded, causing trajectory scores to drop below the threshold.
Channel Specific Ego Planned Path Tracker Scores	14	Violations occurred at red lights for LAV and during turns for TFuse.
Cross Channel Ego Planned Path Tracker Score	22	Violations occurred during red lights and turns.
Planned Path Comfort Scores	30	Violations were inherently linked to the ADS's handling of scenarios involving tailgating, red lights, and turns.
History Comfort Scores	21	Violations were inherently linked to the ADS's handling of scenarios involving tailgating, red lights, and turns.

6.2.5. Experimental Evaluation of the Route in Town 5

Scenario Overview

Route 5 (as seen in Figure 6.18b) takes place in Town 5 (as seen in Figures 6.17 and 6.18a), an urban environment featuring a raised highway and large multi-lane roads and junctions. The route is set during morning hours and features moderate rainfall. The ego vehicle navigates through this town for approximately 1.11 hours, demonstrating its adaptability to different road types and traffic conditions while traversing the road network, which consists of numerous dual-lane urban roads intersecting at numerous large junctions.

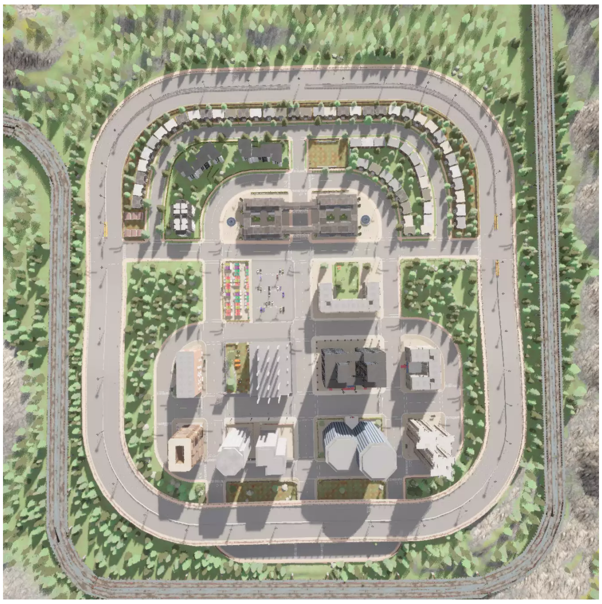
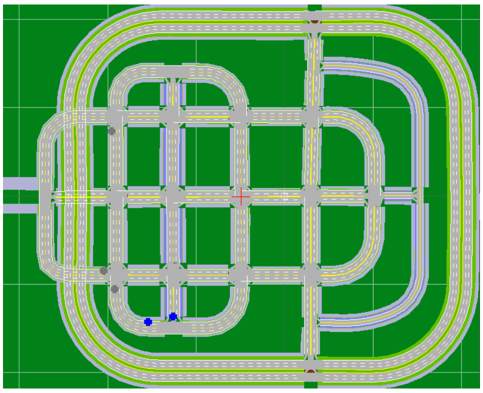
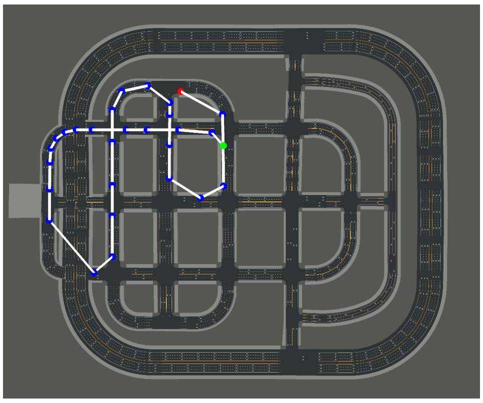


Figure 6.17: Map of the CARLA Town 5 [89]



(a) Layout of the town.



(b) The route followed by the ego vehicle.

Figure 6.18: The CARLA town 5 map (a) and the path followed by ego vehicle (b) used for the evaluation

Results

Table 6.10 shows the number of hazardous scenarios classified by manually observing the ground truth (CARLA simulator) and the automated HSI modules.

The HSI module's performance in Town 5 was relatively stable, with high accuracies in detecting hazardous scenarios. For the "Ego Vehicle Tailgated" scenario, the module achieved a precision of 83.34% and recall of 90.90%, correctly identifying 10 out of 11 scenarios observed by the ADS. There was one false negative and two false positives, one of which occurred when the ADS detected an object behind the ego vehicle at a red light and incorrectly classified it as a tailgating event.

In the "Ego Vehicle Tailgating" scenario, the HSI module demonstrated a precision of 75% and recall of 100%, correctly identifying all 3 scenarios observed by the ADS. There was one false positive due to an ego tailgating event at a red light.

For the "Inconsistent Ego Localization" scenario, the HSI module achieved a precision of 100% and recall of 66.67%, correctly identifying 2 out of 3 scenarios observed by the ADS.

Table 6.11 summarizes the number of threshold violations for each SPI defined and the corresponding causes during the route.

These results suggest that the ADS performs relatively well in urban environments with good lighting like Town 5, with high accuracies in detecting hazardous scenarios. However, the SPI threshold violations indicate that the ADS still faces challenges in handling complex traffic situations, such as red lights and turns, which can impact the system's overall safety and comfort scores.

Table 6.10: Results of HSI for CARLA town 5

Results/Hazardous Scenario	Ego Vehicle Tailgated	Inconsistent Ego Localization	Ego Vehicle Tailgating
Manually observed number of scenarios considered hazardous on CARLA Simulator	8	NA	3
Manually observed number of scenarios considered hazardous in the combined world model of ADS	11	3	3
Number of scenarios correctly identified by the HSI module (automated)	10	2	3
Number of false negatives by the HSI module (automated)	1	1	0
Number of false positives by the HSI module (automated)	2	0	1
Precision of the HSI module	83.34 %	100.00 %	75.00 %
Recall of the HSI module	90.90 %	66.67 %	100.00 %

Table 6.11: Results of SPIs monitoring for CARLA town 5

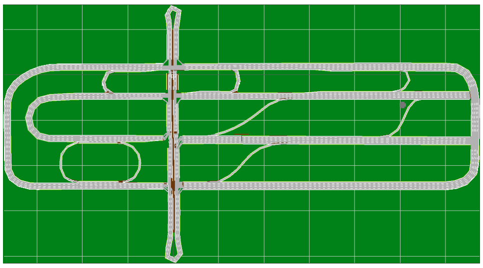
SPI	# Threshold Violations	Cause
Ego Location Similarity Scores	7	Violations occurred due to inconsistent localization, either prolonged or short-term.
Ego Orientation Similarity Scores	0	
Object Count Similarity Scores	12	Violations occurred when TFuse or LAV misidentified stationary objects on the sidewalk as obstacles, though these were irrelevant to the driving scenarios.
Object Location Similarity Scores	8	Violations occurred when both the ADS detected a high number of objects near the ego vehicle.
Object Orientation Similarity Scores	0	
Object Area Similarity Scores	0	
Objected Motion Prediction Validation Score	15	Violations occurred when many objects exited the ADS perception field.
Ego Trajectory Similarity Scores	17	Violations occurred during stoplight periods.
Object Trajectory Similarity Scores	22	Violations occurred when numerous detected objects led to small errors that compounded, causing trajectory scores to drop below the threshold.
Channel Specific Ego Planned Path Tracker Scores	14	Violations occurred at red lights for LAV and during turns for TFuse.
Cross Channel Ego Planned Path Tracker Score	9	Violations occurred during red lights and turns.
Planned Path Comfort Scores	14	Violations were inherently linked to the ADS's handling of scenarios involving tailgating, red lights, and turns.
History Comfort Scores	9	Violations were inherently linked to the ADS's handling of scenarios involving tailgating, red lights, and turns.

6.2.6. Experimental Evaluation of the Route in Town 6

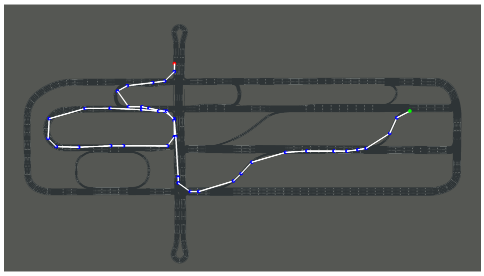
Scenario Overview

Route 6 (as seen in Figure 6.20b) is located in Town 6 (as seen in Figures 6.19 and 6.20a), a low-density town exhibiting a multitude of large, 4-6 lane roads and special junctions like the Michigan Left. The route takes place during dawn and features wet road conditions. The ego vehicle spends around 1.59 hours navigating through this town, showcasing its robustness in handling diverse terrain and wet weather conditions while traversing the road network, which exhibits 4 large parallel roads with 4 to 6 lanes, interconnected by several slip roads and junctions [90].

Figure 6.19: Map of the CARLA Town 6 [90]



(a) Layout of the town.



(b) The route followed by the ego vehicle.

Figure 6.20: The CARLA town 6 map (a) and the path followed by ego vehicle (b) used for the evaluation

Results

Table 6.12 shows the number of hazardous scenarios classified by manually observing the ground truth (CARLA simulator) and the automated HSI modules.

The HSI module's performance in Town 6 was characterized by high accuracies in detecting "Ego Vehicle Tailgated" and "Inconsistent Ego Localization" scenarios, achieving 100% recall in both cases. For the "Ego Vehicle Tailgated" scenario, the module achieved a precision of 90%, correctly identifying all nine scenarios observed by the ADS, with one false positive. In the "Ego Vehicle Tailgating" scenario, the HSI module achieved a precision and recall of 50%, correctly identifying one out of two scenarios observed by the ADS.

The HSI module's performance in Town 6 was also notable for a prolonged period of stable SPI scores, with no significant fluctuations observed across most metrics. However, the Cross Channel Planned Path Scores exhibited repeated deviations over a short period of time, indicating potential issues with the ADS's path planning and execution. Upon closer examination, it was observed that the ADS veered off the road and repeatedly attempted to generate a path (as seen in Figure 6.21) to navigate around the deviation but was unsuccessful in doing so. This behaviour suggests that the SPI framework can effectively identify situations where the ADS becomes stuck or trapped in a particular location, providing valuable insights for system improvements and failure mode analysis.

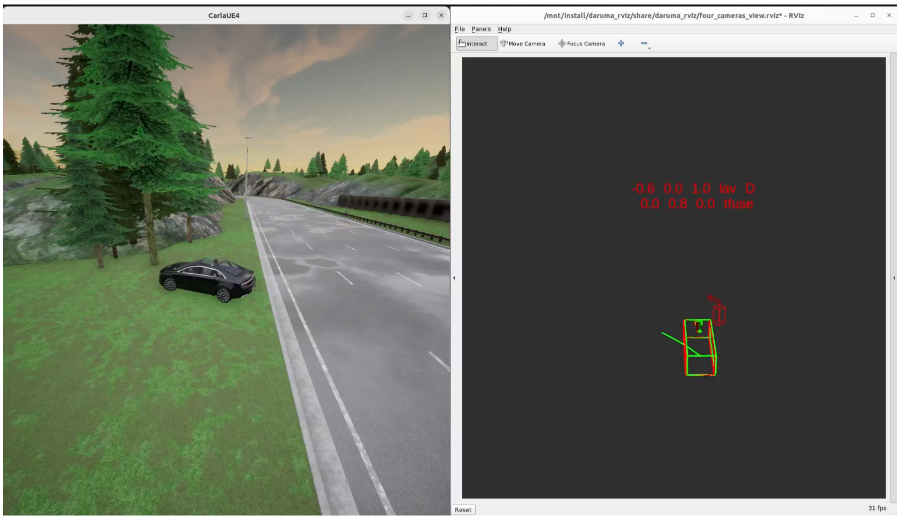
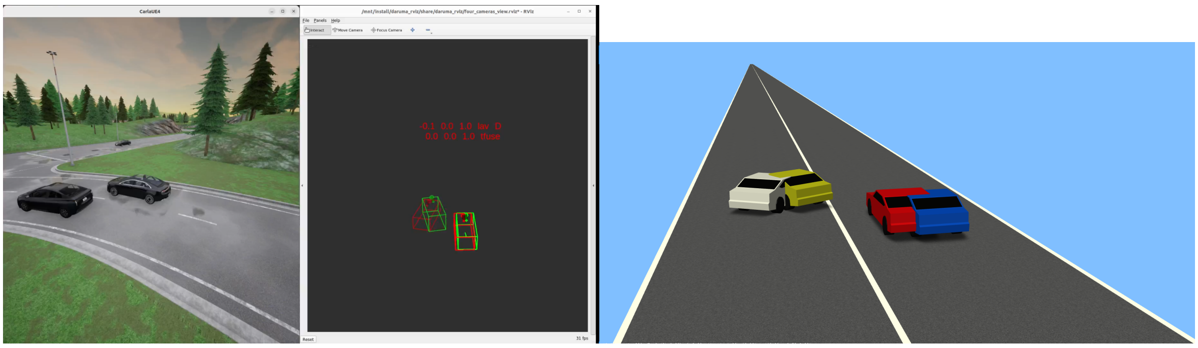


Figure 6.21: Scenario where ego vehicle veered off the road and was stuck for a short period

Table 6.13 summarizes the number of threshold violations for each SPI defined and the corresponding causes during the route. The ”Ego Location Similarity Scores” violated the threshold eighteen times due to inconsistent localization. The ”Object Count Similarity Scores” and ”Object Location Similarity Scores” experienced twelve and eight violations, respectively, primarily when either of the channels detected irrelevant objects on the sidewalk or when there were too many objects in the scene.

In this particular route, an interesting situation arose where the ego orientation score exceeded the predefined threshold. Figure 6.22a illustrates the scenario captured by the CARLA simulator alongside the ADS's perception, while Figure 6.22b presents the OpenScenario export observed when the threshold was breached. This incident highlights a discrepancy in the detection capabilities between the two ADS channels: LAV accurately identified the object's orientation, whereas TFuse did not.

In Figure 6.22b, the ego vehicles identified by each channel are marked in red and blue for LAV and TFuse, respectively, showing an overlap. The object detected by LAV is depicted in white, and the object perceived by TFuse is in yellow. It is important to note that Daruma did not have access to lane information, prompting the export to include a generic depiction of the road.



(a) Snapshot of the CARLA simulator and ADS world model

(b) OpenScenario files viewed on Esmini

Figure 6.22: The map of the sixth town (a) and the path followed by ego vehicle (b) used for the evaluation

These results highlight the importance of monitoring SPI scores and their fluctuations over time, as they can provide valuable insights into the ADS's performance and potential failure modes. The stable SPI scores observed in Town 6, combined with the deviations in the Cross Channel Planned Path Scores, demonstrate the effectiveness of the SPI framework in identifying situations where the ADS may become stuck or exhibit suboptimal behavior. This information can be used to guide further development and refinement of the ADS's path planning and decision-making capabilities.

Table 6.12: Results of HSI for CARLA town 6

Results/Hazardous Scenario	Ego Vehicle Tailgated	Inconsistent Ego Localization	Ego Vehicle Tailgating
Manually observed number of scenarios considered hazardous on CARLA Simulator	8	NA	2
Manually observed number of scenarios considered hazardous in the combined world model of ADS	9	2	2
Number of scenarios correctly identified by the HSI module (automated)	9	2	1
Number of false negatives by the HSI module (automated)	0	0	1
Number of false positives by the HSI module (automated)	1	0	1
Precision of the HSI module	90.00 %	100.00 %	50.00 %
Recall of the HSI module	100.00 %	100.00 %	50.00 %

Table 6.13: Results of SPI monitoring for CARLA town 6

SPI	# Threshold Violations	Cause
Ego Location Similarity Scores	18	Violations occurred due to inconsistent localization, either prolonged or short-term.
Ego Orientation Similarity Scores	1	
Object Count Similarity Scores	9	Violations occurred when TFuse or LAV misidentified stationary objects on the sidewalk as obstacles, though these were irrelevant to the driving scenarios.
Object Location Similarity Scores	8	Violations occurred when both the ADS detected a high number of objects near the ego vehicle.
Object Orientation Similarity Scores	0	
Object Area Similarity Scores	0	
Objected Motion Prediction Validation Score	8	Violations occurred when many objects exited the ADS perception field.
Ego Trajectory Similarity Scores	6	Violations occurred during stoplight periods.
Object Trajectory Similarity Scores	17	Violations occurred when numerous detected objects led to small errors that compounded, causing trajectory scores to drop below the threshold.
Channel Specific Ego Planned Path Tracker Scores	9	Violations occurred at red lights for LAV and during turns for TFuse.
Cross Channel Ego Planned Path Tracker Score	11	Violations occurred during red lights and turns.
Planned Path Comfort Scores	14	Violations were inherently linked to the ADS's handling of scenarios involving tailgating, red lights, and turns.
History Comfort Scores	8	Violations were inherently linked to the ADS's handling of scenarios involving tailgating, red lights, and turns.

6.3. Profiling Results

A profiling analysis was conducted on the NXP S32G274A vehicle network processor to evaluate the computational efficiency and real-time performance of the developed SPI and HSI algorithms. The S32G274A processor combines ASIL D safety, hardware security, high-performance real-time and application processing, and network acceleration, making it a suitable platform for automotive applications.

The S32G processor features a quad Arm Cortex-A53 core cluster with optional lockstep for applications and services, a triple Arm Cortex-M7 lockstep core for real-time applications, and various hardware accelerators such as the Low Latency Communication Engine (LLCE) for automotive networks, the Packet Forwarding Engine (PFE) for Ethernet networks, and the Hardware Security Engine (HSE) for secure boot and security services. The processor operates at a speed of 1.1 GHz [91].

The routes in each town were subject to profiling analysis. To ensure accurate and reliable profiling results, several steps were taken to minimize interruptions and optimize the performance of the NXP Daruma application during the profiling process:

- **CPU Affinity**: The CPU affinity of the NXP Daruma application was set to 0 using the `taskset` linux command. This command allows you to specify which CPU core(s) a process should run on. By setting the affinity to 0, the application was constrained to run on a specific processor, preventing interruptions from other processes. All other processes' CPU affinity was set to 1, ensuring they would not interfere with the profiling of the NXP Daruma application.
- **Process Priority**: The NXP Daruma application was assigned the highest priority using the `nice` linux command. The `nice` command allows you to adjust the scheduling priority of a process. Giving the application the highest priority minimised the risk of interruptions or pauses during the profiling process. This ensured that the application received the maximum available CPU time and resources.
- **CPU Governor Mode**: The device was set to performance governor mode using the `cpupower frequency-set` linux command. The CPU governor is responsible for managing the CPU's frequency and power consumption. By setting the governor to performance mode, the CPU was configured to operate at its maximum frequency, providing consistent and optimal performance throughout the profiling process. This eliminated any potential fluctuations in CPU performance that could impact the profiling results.

These optimizations were crucial for obtaining accurate and reliable profiling results. By isolating the NXP Daruma application on a dedicated CPU core, granting it the highest priority, and ensuring consistent CPU performance, the impact of external factors and interruptions on the profiling process was minimized. This allowed for a more precise measurement of the application's execution times and resource utilization.

6.3.1. Profiling Results for the Route in Town 1

Figure 6.23 shows the time taken by the SPI and HSI module for the route in Town 1. The graph illustrates that the execution time remains well below the threshold of 25ms for most instances, with only a few sporadic spikes approaching or slightly exceeding the limit. This indicates that the developed algorithms can operate efficiently and meet the real-time requirements in this scenario. The statistical summary is present in Tables 6.14, 6.15, 6.16, 6.17 and Figures 6.24, 6.25, and 6.26 present the execution times for the Similarity, Comfort, and Motion Prediction modules, respectively.

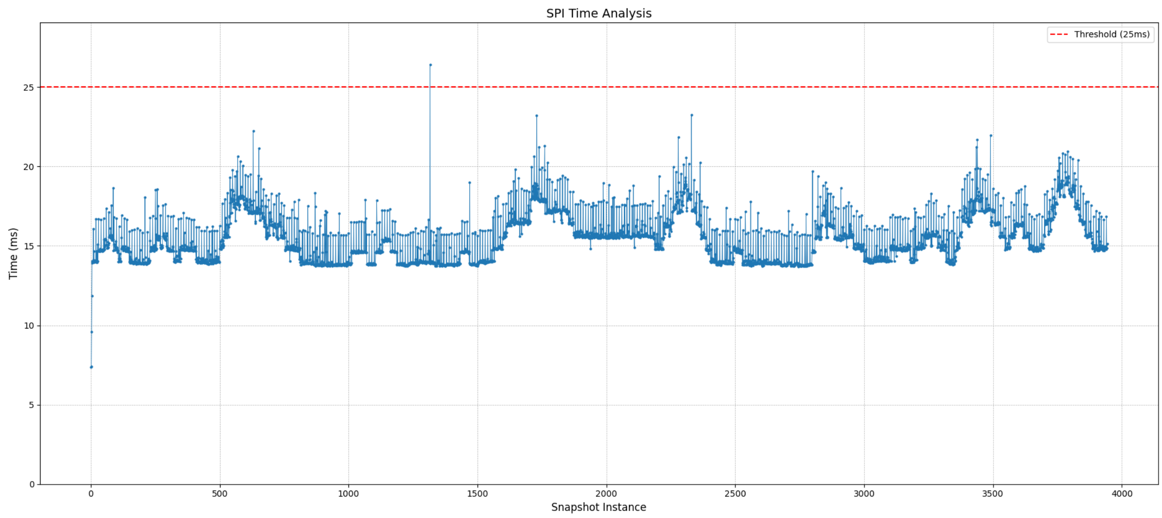


Figure 6.23: Time Taken by the SPI and HSI module in Town 1

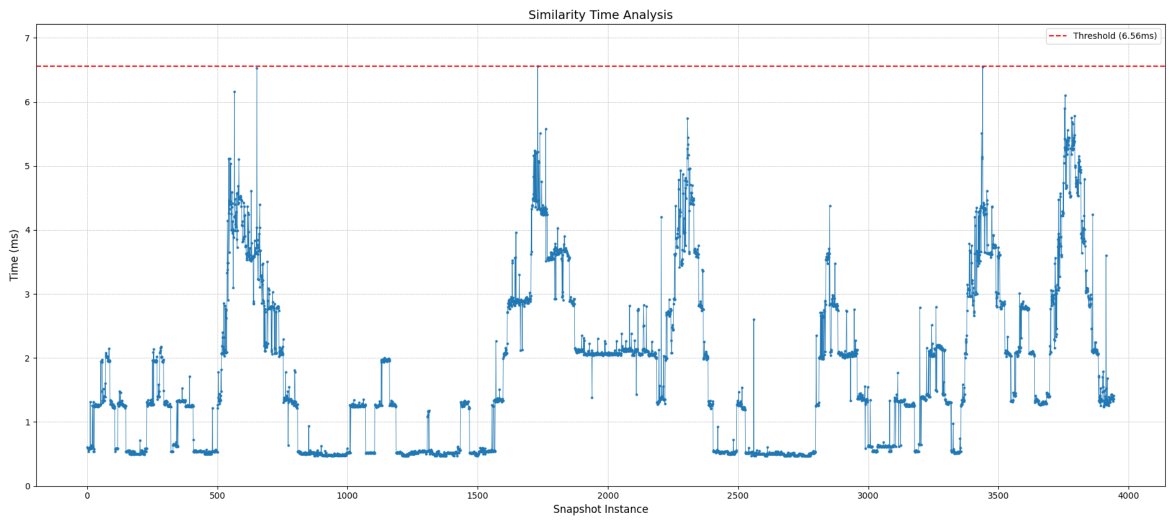


Figure 6.24: Time Taken by the Similarity module in Town 1

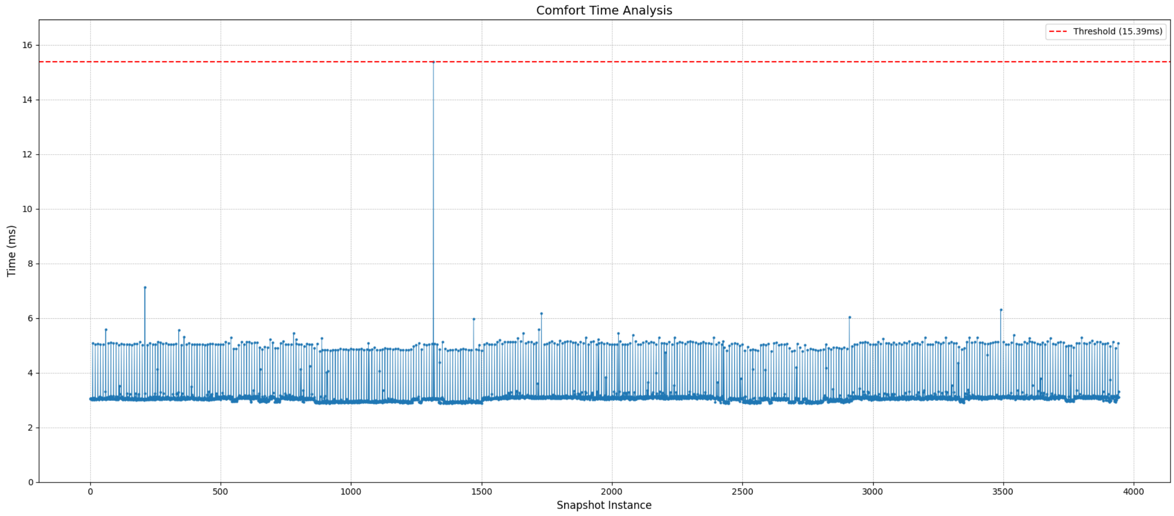


Figure 6.25: Time Taken by the comfort module in Town 1

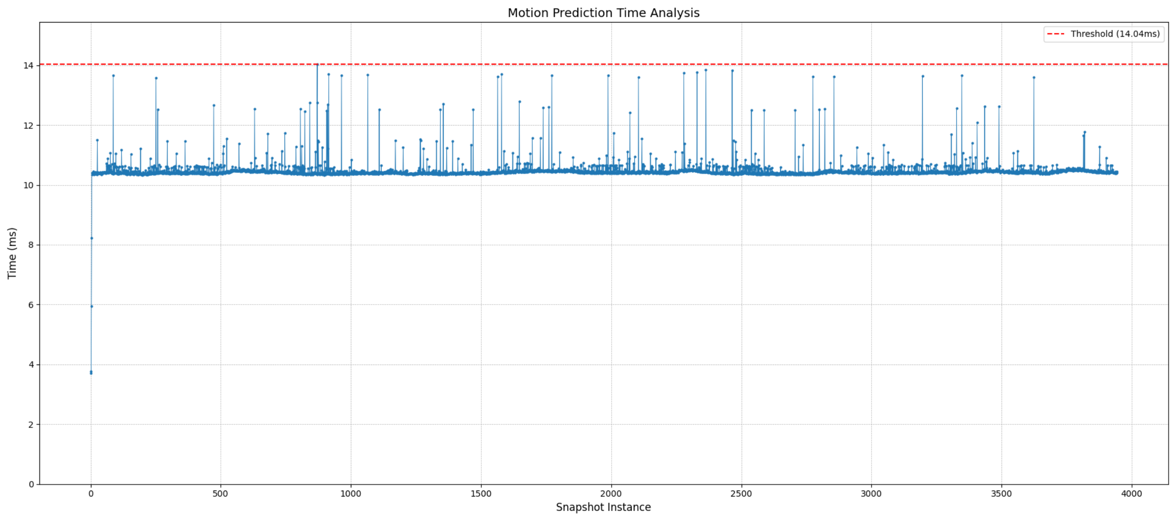


Figure 6.26: Time Taken by the motion prediction module in Town 1

6.3.2. Profiling Results for the Route in Town 2

Figure 6.27 shows the time taken by the SPI and HSI module for the route in Town 2. Similar to Town 1, the execution time remains well below the 25ms threshold for most instances, demonstrating the algorithms' efficiency in this scenario. The statistical summary is present in Tables 6.14, 6.15, 6.16, 6.17, and Figures 6.28, 6.29, and 6.30 present the execution times for the Similarity, Comfort, and Motion Prediction modules, respectively.

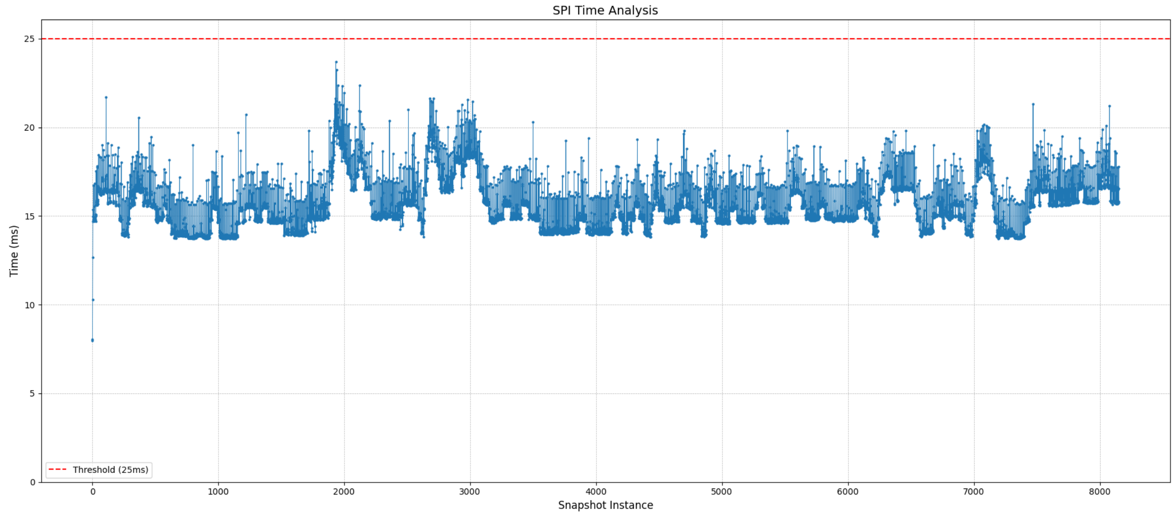


Figure 6.27: Time Taken by the SPI and HSI module in Town 2

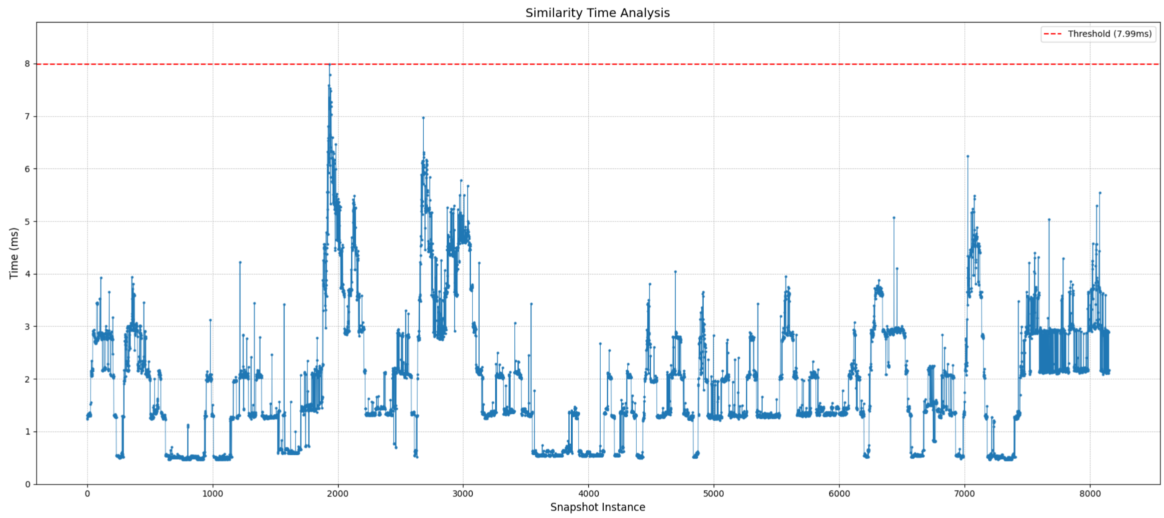


Figure 6.28: Time Taken by the Similarity module in Town 2

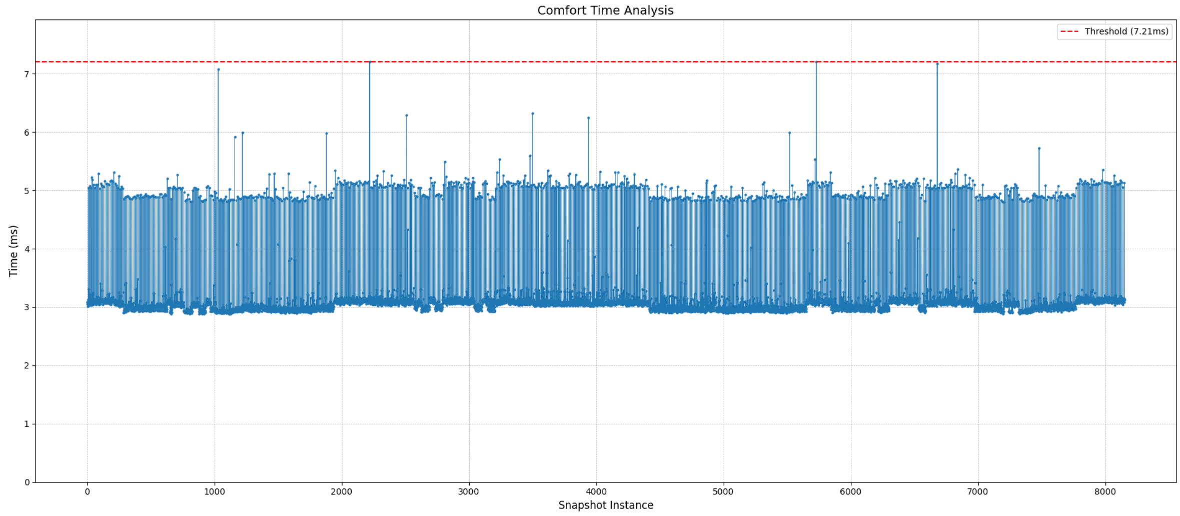


Figure 6.29: Time Taken by the comfort module in Town 2

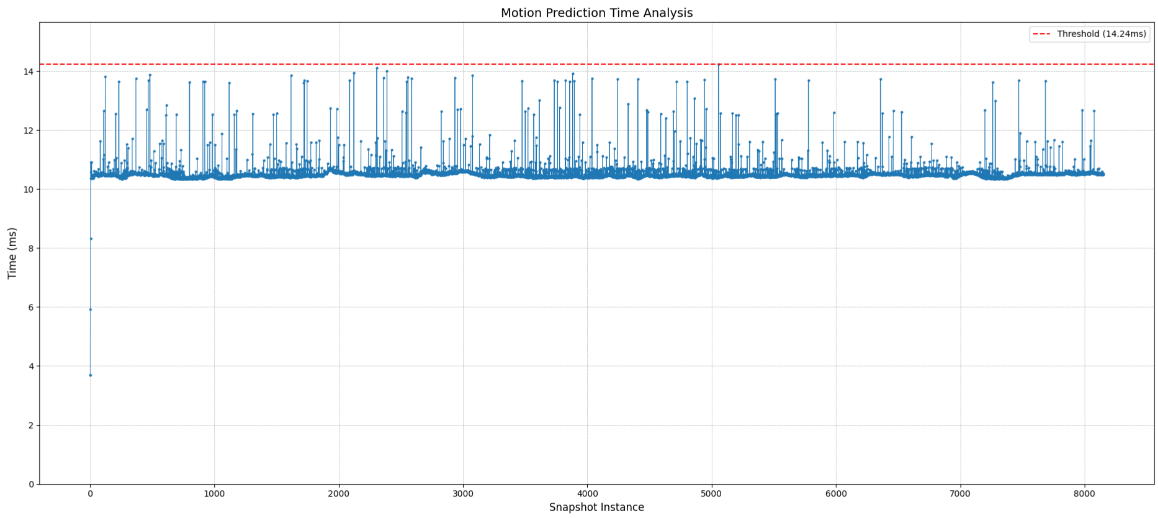


Figure 6.30: Time Taken by the motion prediction module in Town 2

6.3.3. Profiling Results for the Route in Town 3

Figure 6.31 shows the time taken by the SPI and HSI module for the route in Town 3. The statistical summary is present in Tables 6.14, 6.15, 6.16, 6.17 and Figures 6.32, 6.33, and 6.34 present the execution times for the Similarity, Comfort, and Motion Prediction modules, respectively.

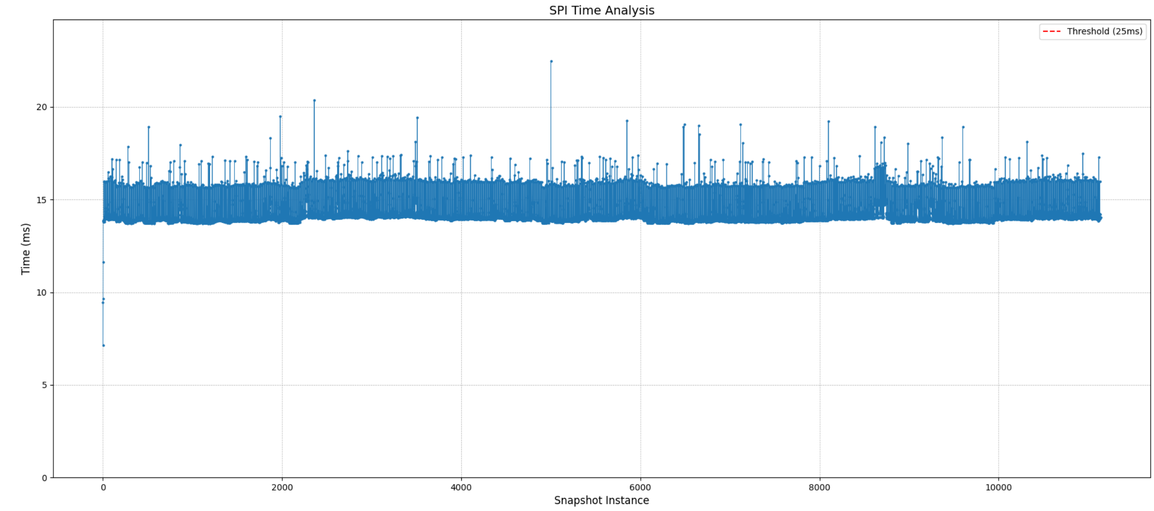


Figure 6.31: SPI Time Analysis for Town 3

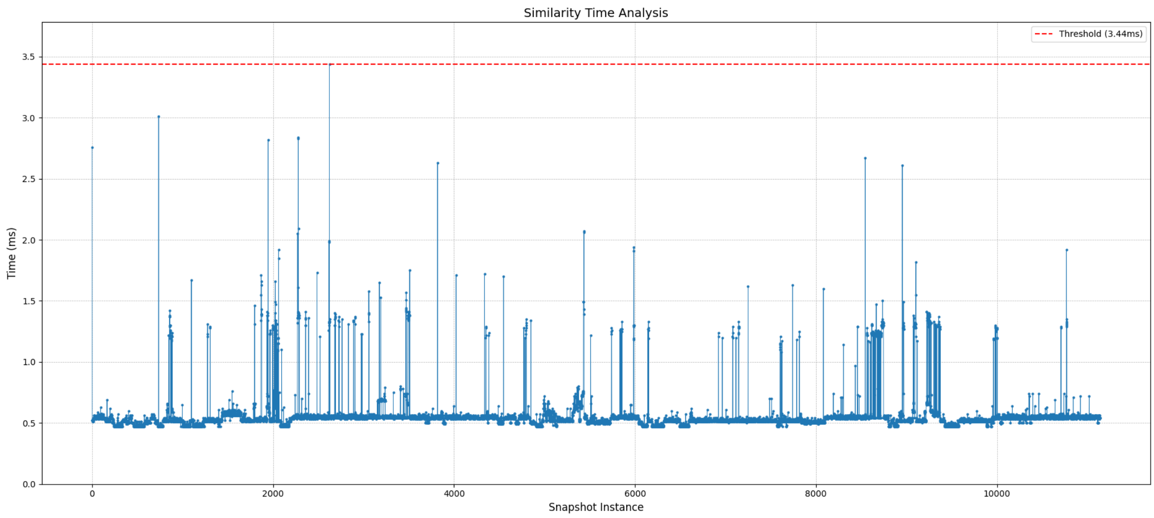


Figure 6.32: Similarity Time Analysis for Town 3

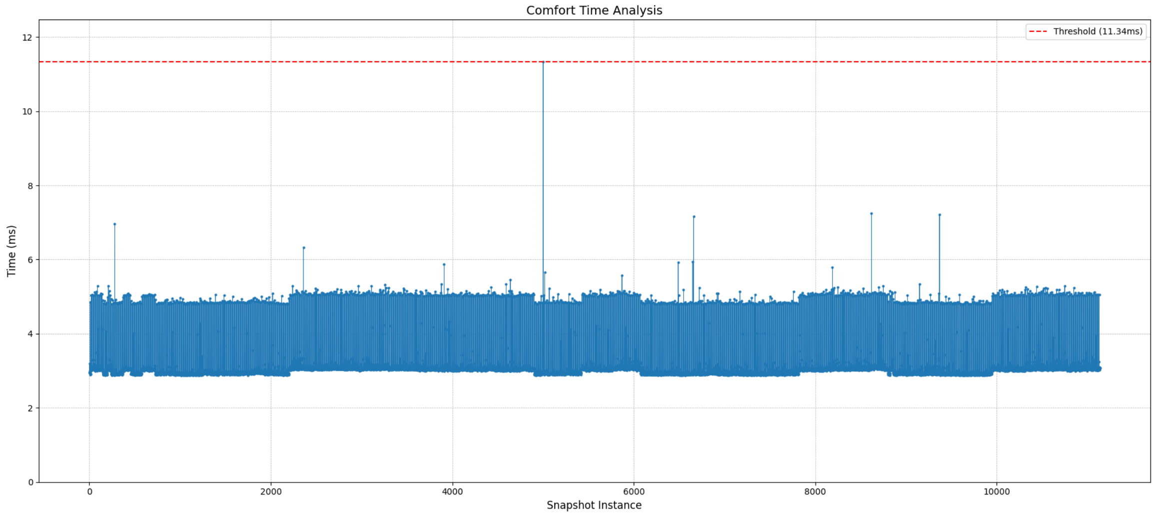


Figure 6.33: Comfort Time Analysis for Town 3

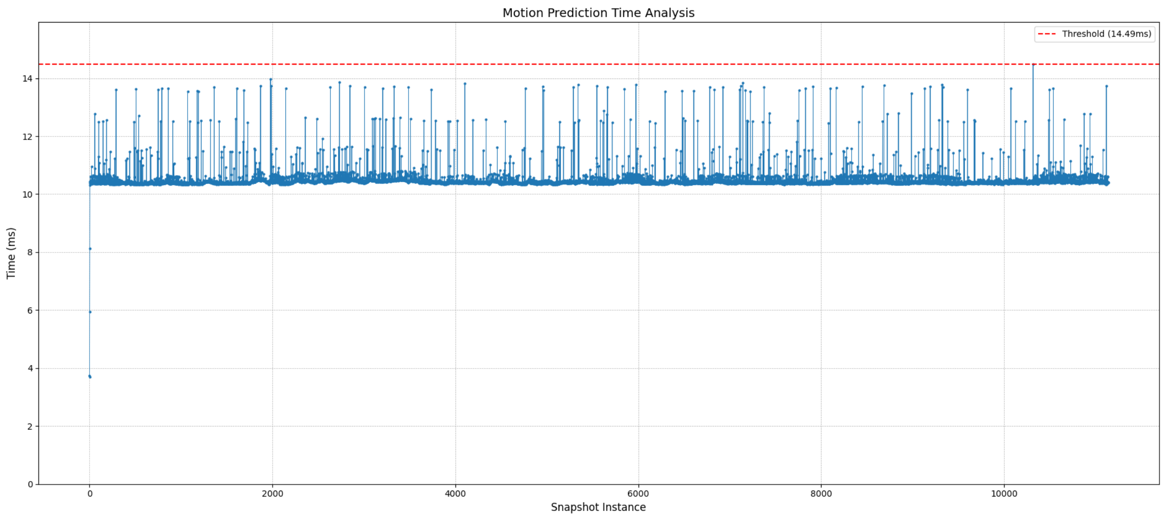


Figure 6.34: Motion Prediction Time Analysis for Town 3

6.3.4. Profiling Results for the Route in Town 4

Figure 6.35 shows the time taken by the SPI and HSI module for the route in Town 4. The statistical summary is present in Tables 6.14, 6.15, 6.16, 6.17, and Figures 6.36 , 6.37, 6.38 present the execution times for the Similarity, Comfort, and Motion Prediction modules, respectively, for the route in Town 4.

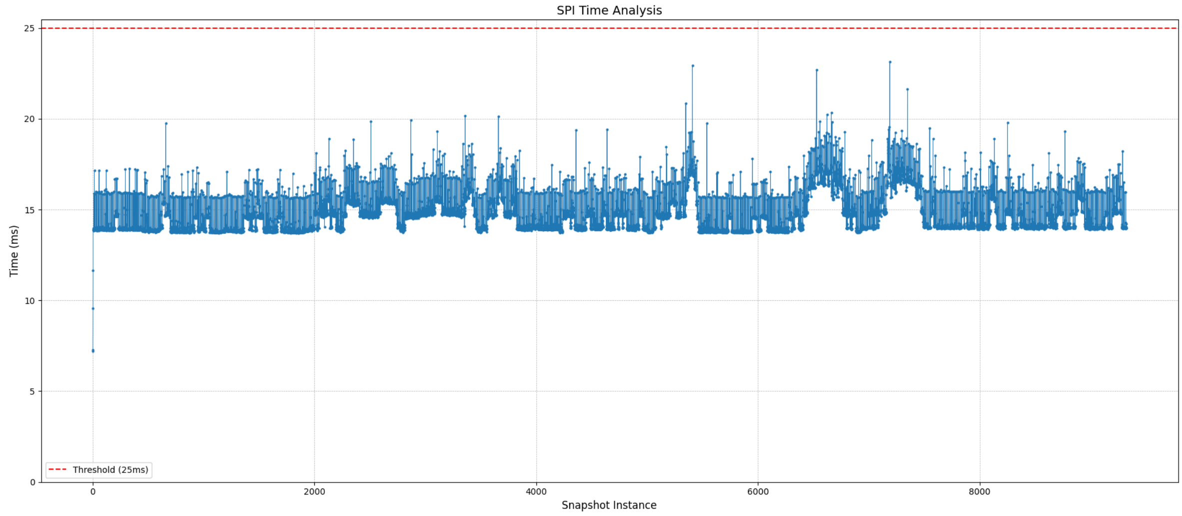


Figure 6.35: SPI Time Analysis for Town 4

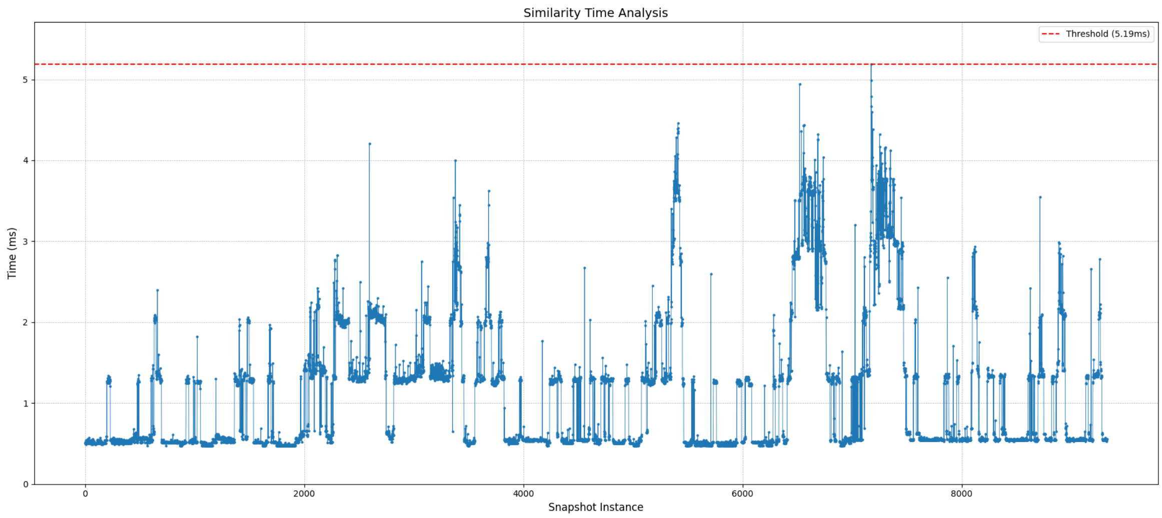


Figure 6.36: Similarity Time Analysis for Town 4

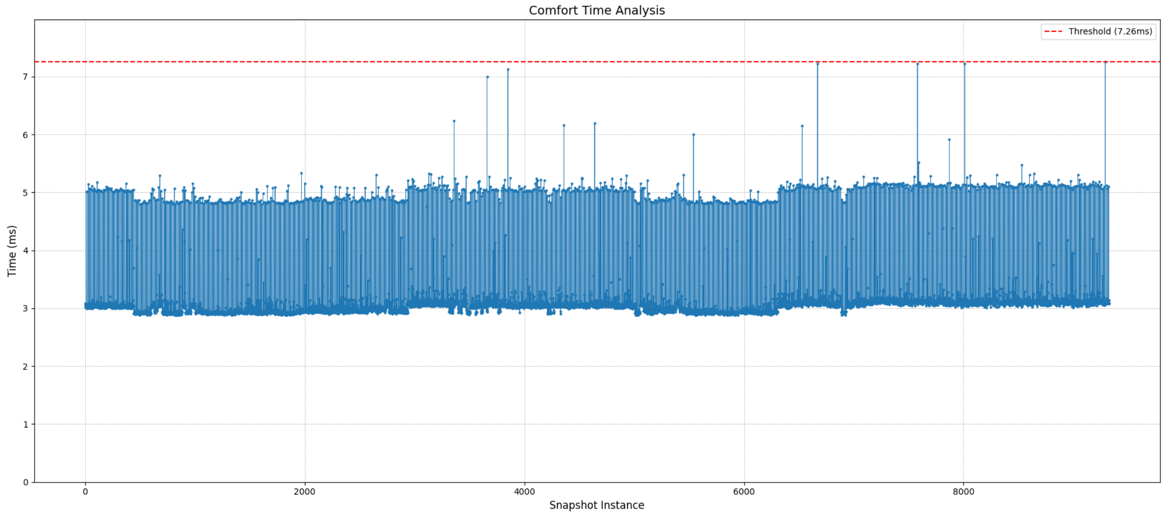


Figure 6.37: Comfort Time Analysis for Town 4

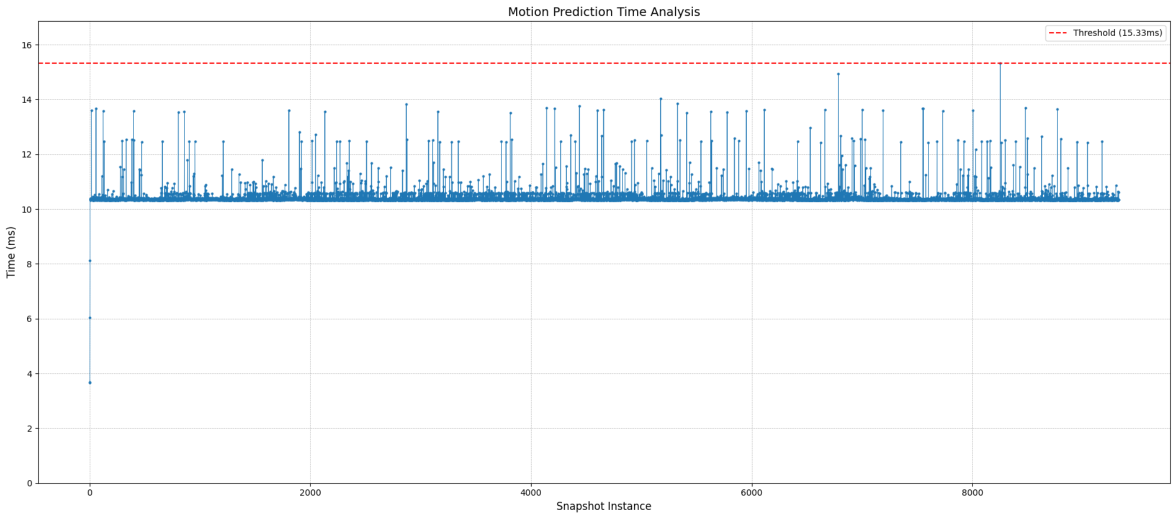


Figure 6.38: Motion Prediction Time Analysis for Town 4

6.3.5. Profiling Results for the Route in Town 5

Figure 6.39 shows the time taken by the SPI and HSI module for the route in Town 5. The statistical summary is present in Tables 6.14, 6.15, 6.16, 6.17, and Figures 6.40, 6.41, and 6.42 present the execution times for the Similarity, Comfort, and Motion Prediction modules, respectively.

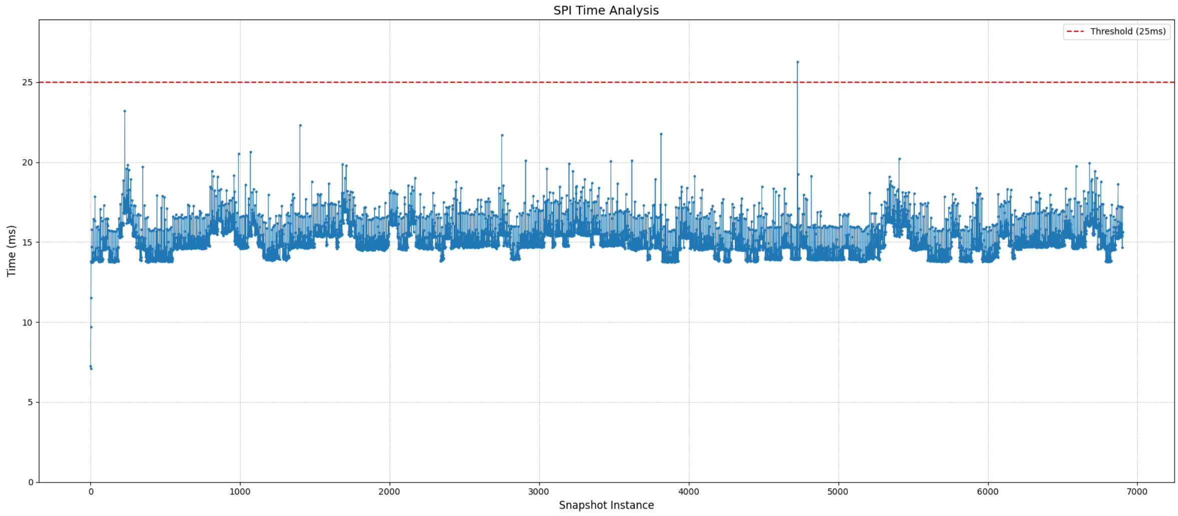


Figure 6.39: SPI Time Analysis for Town 5

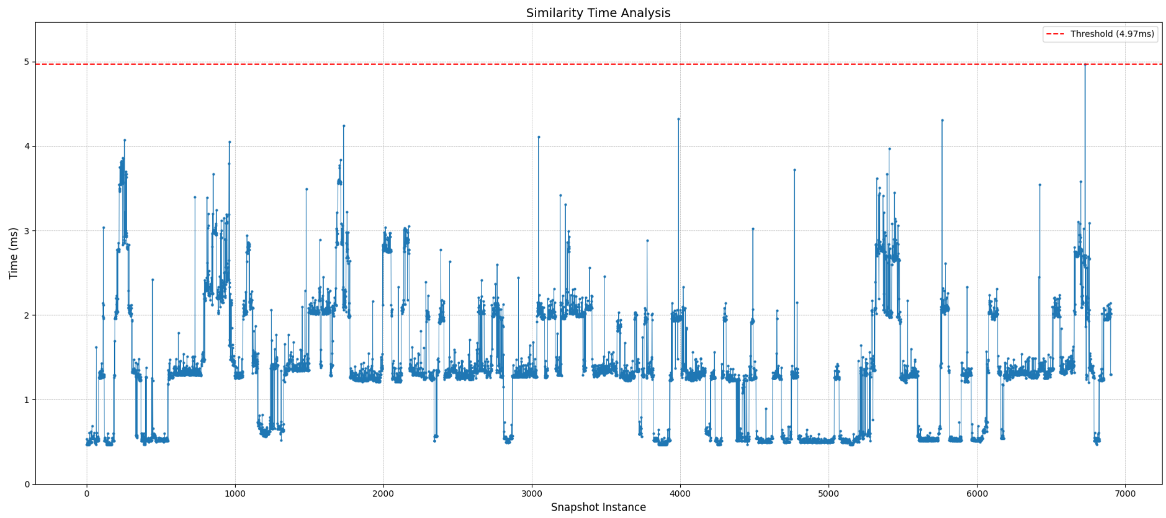


Figure 6.40: Similarity Time Analysis for Town 5

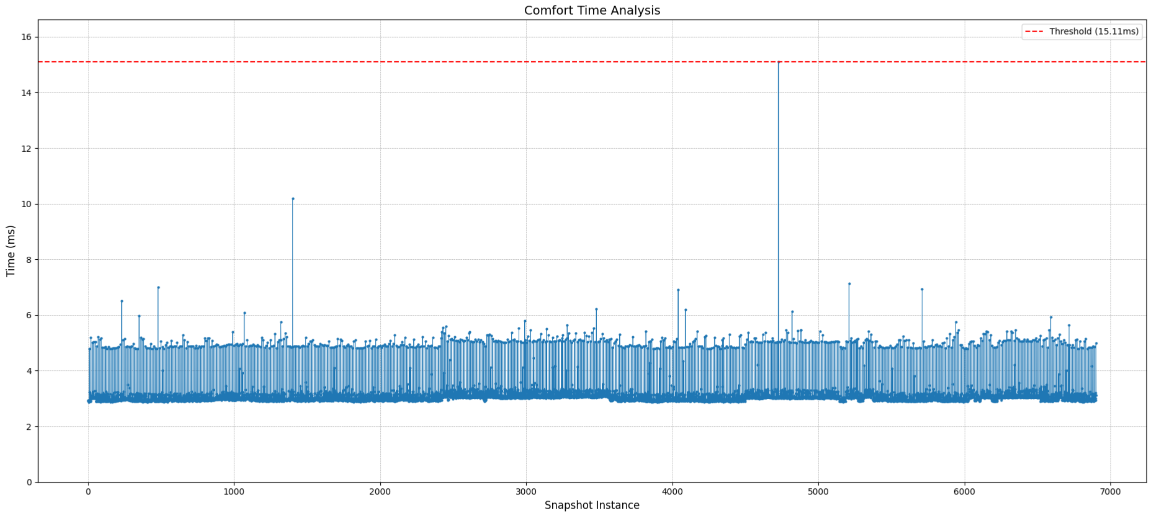


Figure 6.41: Comfort Time Analysis for Town 5

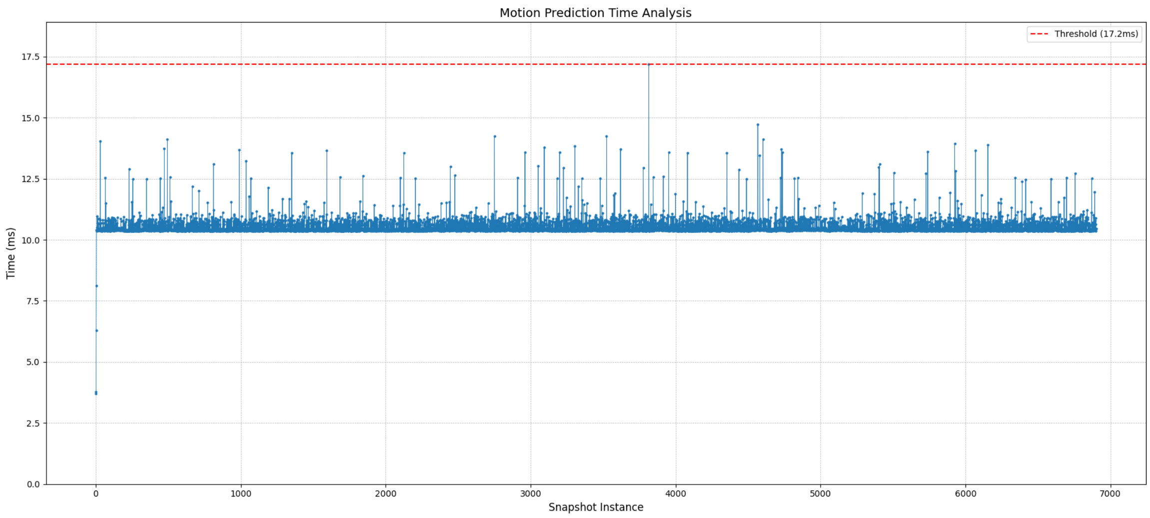


Figure 6.42: Motion Prediction Time Analysis for Town 5

6.3.6. Profiling Results for the Route in Town 6

Figure 6.43 shows the time taken by the SPI and HSI module for the route in Town 6. The statistical summary is present in Tables 6.14, 6.15, 6.16, 6.17, and Figures 6.44, 6.45, and 6.46 present the execution times for the Similarity, Comfort, and Motion Prediction modules, respectively.

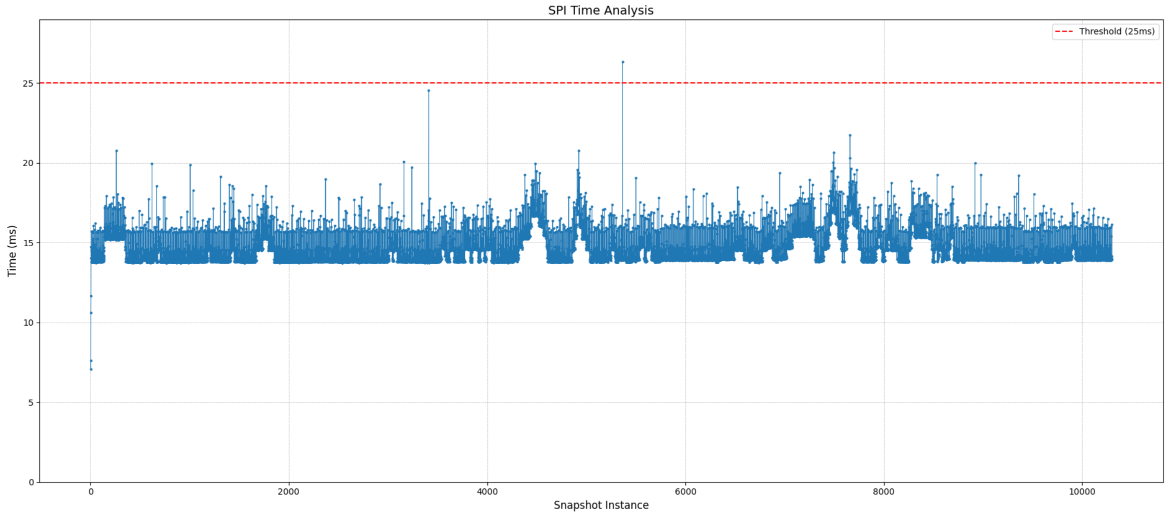


Figure 6.43: SPI Time Analysis for Town 6

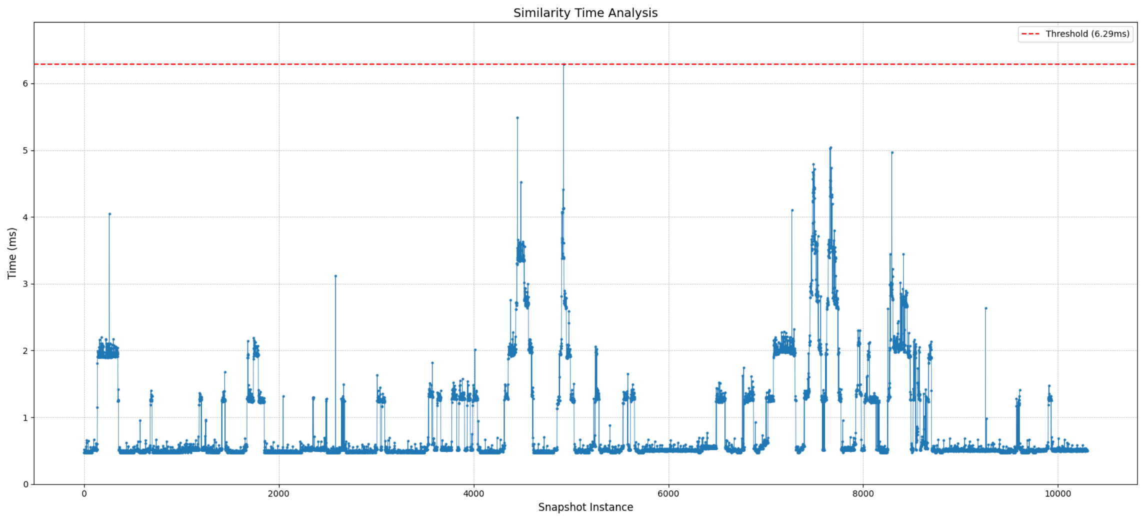


Figure 6.44: Similarity Time Analysis for Town 6

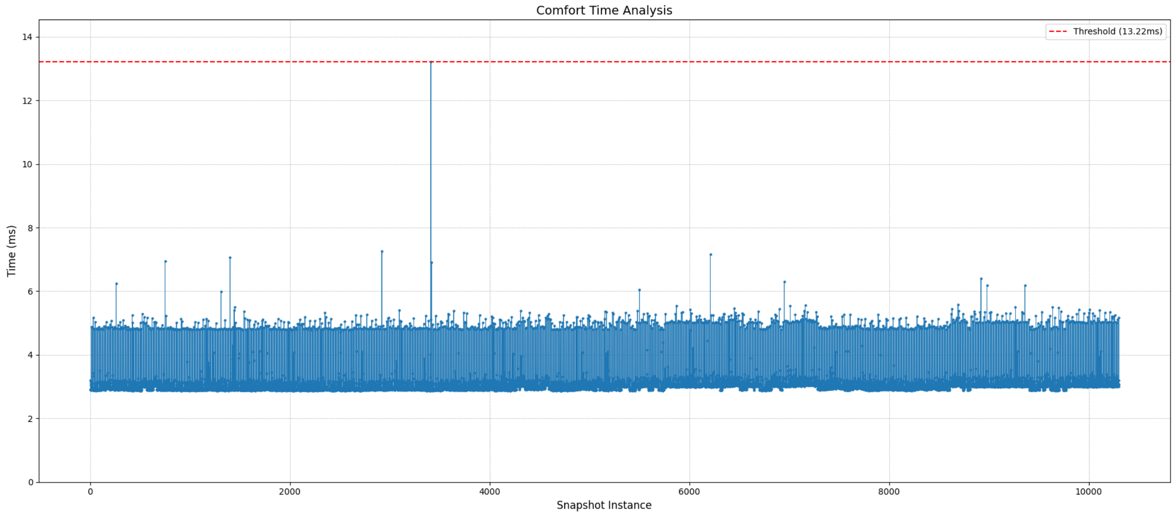


Figure 6.45: Comfort Time Analysis for Town 6

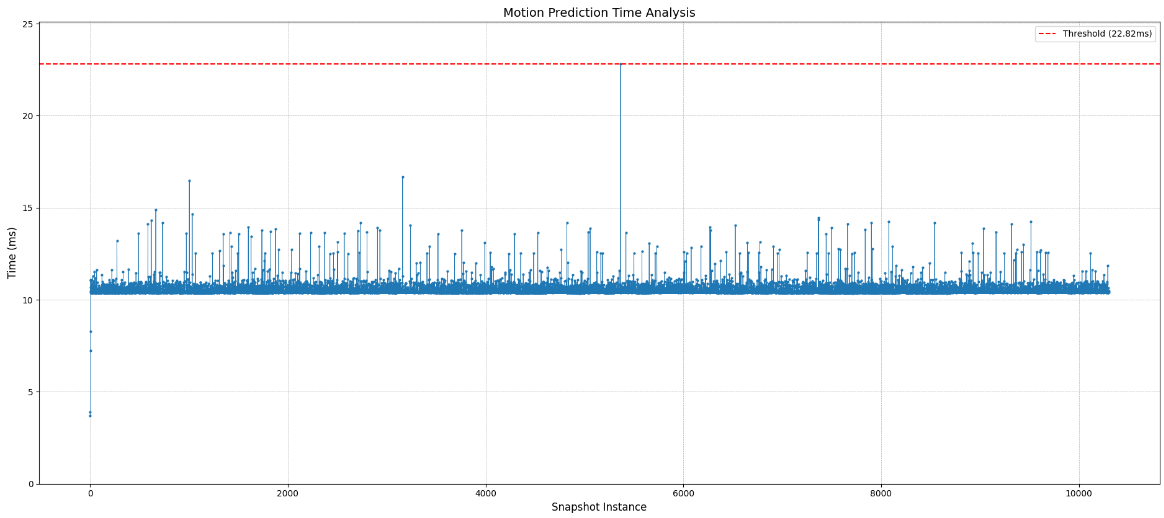


Figure 6.46: Motion Prediction Time Analysis for Town 6

6.3.7. Statistical Summary

The statistical summaries for the SPI and HSI module execution times, as well as the execution times of the Similarity, Comfort, and Motion Prediction modules, are presented in Tables 6.14, 6.15, 6.16, and 6.17, respectively.

For the SPI Time statistics, the threshold is 25 ms, and for the remaining statistics (comfort time, similarity time and motion prediction time statistics), the threshold is the Maximum Time (ms) in that particular route. This, when compared to the Low Values (<50% of threshold), helps understand if the time taken is in the 50 percentile or if the maximum time is just an outlier.

Table 6.14: SPI Time Statistics

Statistic	Town1	Town2	Town3	Town4	Town5	Town6
Average Time (ms)	15.52	15.69	14.26	14.78	15.17	14.66 %
Maximum Time (ms)	26.43	23.71	22.47	23.15	26.29	26.35%
Minimum Time (ms)	7.38	7.98	7.15	7.20	7.08	7.07%
Standard Deviation (ms)	1.52	1.44	0.75	1.10	1.06	1.05 %
Breaches of 25 ms	0.03%	0.00%	0.00%	0.00%	0.01%	0.01%
Within 90% of threshold	0.08%	0.02%	0.00%	0.03%	0.03%	0.02%
Low Values (<50% of threshold)	0.10%	0.04%	0.04%	0.03%	0.06%	0.04%

Table 6.15: Comfort Time Statistics

Statistic	Town1	Town2	Town3	Town4	Town5	Town6
Average Time (ms)	3.27	3.25	3.21	3.24	3.25	3.22
Maximum Time (ms)	15.39	7.21	11.34	7.26	15.11	13.22
Minimum Time (ms)	2.88	2.88	2.88	2.88	2.87	2.86
Standard Deviation (ms)	0.65	0.61	0.62	0.62	0.65	0.63
Within 90% of threshold	0.03%	0.05%	0.01%	0.06%	0.01%	0.01%
Low Values (<50% of threshold)	99.97%	89.46%	99.91%	89.17%	99.97%	99.94%

Table 6.16: Similarity Time Statistics

Statistic	Town1	Town2	Town3	Town4	Town5	Town6
Average Time (ms)	1.77	1.91	0.57	1.12	1.38	0.88
Maximum Time (ms)	6.56	7.99	3.44	5.19	4.97	6.29
Minimum Time (ms)	0.47	0.47	0.47	0.47	0.47	0.47
Standard Deviation (ms)	1.26	1.20	0.18	0.81	0.69	0.71
Within 90% of threshold	0.13%	0.10%	0.01%	0.04%	0.01%	0.01%
Low Values (<50% of threshold)	85.11%	93.25%	99.78%	91.97%	92.51%	97.89%

Table 6.17: Motion Prediction Time Statistics

Statistic	Town1	Town2	Town3	Town4	Town5	Town6
Average Time (ms)	10.48	10.54	10.48	10.42	10.54	10.56
Maximum Time (ms)	14.04	14.24	14.49	15.33	17.20	22.82
Minimum Time (ms)	3.70	3.69	3.69	3.67	3.69	3.69
Standard Deviation (ms)	0.36	0.34	0.34	0.32	0.37	0.40
Within 90% of threshold	0.66%	0.59%	0.57%	0.05%	0.01%	0.01%
Low Values (<50% of threshold)	0.08%	0.04%	0.03%	0.03%	0.06%	97.84%

6.4. Analysis and Discussion

This experimental evaluation of the SPI and HSI modules utilized with the CARLA simulator and ADS software stacks (LAV and TFuse) has yielded significant insights into the strengths and developmental needs for the continuous improvement of AV safety. The assessment spanned a variety of urban and highway environments under different weather conditions, providing a platform for analyzing the effectiveness of the proposed techniques for assessing AV safety.

6.4.1. Summary of Results

The results summarized in Table 6.18 underscore the effectiveness and areas for improvement in the SPI and HSI modules across different driving conditions and scenarios. It includes the total number of scenarios observed for each hazardous situation, the number of correctly identified scenarios, false positives, false negatives, and the overall success rate for all the routes taken through the different towns.

Table 6.18: Summary of Hazardous Scenarios, Precision and Success Rates for All CARLA Towns

Results/Hazardous Scenario	**Ego Vehicle Tailgated**	**Inconsistent Ego Localization**	**Ego Vehicle Tailgating**
Manually observed number of scenarios considered hazardous in the combined world model of ADS	73	21	25
Number of scenarios correctly identified by the HSI module (automated)	69	17	18
Number of false negatives by the HSI module (automated)	4	4	7
Number of false positives by the HSI module (automated)	6	0	10
Overall Precision of the HSI module	92.00 %	100.00 %	64.28 %
Overall Recall of the HSI module	94.52 %	80.95 %	72.00 %

Table 6.19: Summary of SPI Threshold Violations Across All CARLA Towns

SPI	# Threshold Violations
Ego Location Similarity Scores	64
Ego Orientation Similarity Scores	1
Object Count Similarity Scores	60
Object Location Similarity Scores	41
Object Orientation Similarity Scores	4
Object Area Similarity Scores	0
Objected Motion Prediction Validation Score	56
Ego Trajectory Similarity Scores	43
Object Trajectory Similarity Scores	91
Channel Specific Ego Planned Path Tracker Scores	51
Cross Channel Ego Planned Path Tracker Score	68
Planned Path Comfort Scores	79
History Comfort Scores	92

Overall, the SPI and HSI modules achieved an average recall of 82.39 % in identifying hazardous scenarios across all towns, correctly identifying 104 out of 119 total observed scenarios. During the evaluation, 16 false positives and 15 false negatives were observed, resulting in an average precision of 85.43 %.

6.4.2. Analysis of False Positives and Negatives

The occurrence of false positives and negatives in the ”Ego Vehicle Tailgating” scenarios underscores some inherent challenges in complex driving environments. False positives were frequently triggered by misinterpreting other vehicles’ movements, especially during turns or at traffic lights, suggesting that the current models may overly generalize from insufficient data or misinterpret the intent of surrounding traffic. Conversely, false negatives often resulted from situations where the proximity thresholds set for triggering alerts were not breached despite potentially risky scenarios. This indicates a need for refining the threshold settings or employing more SPIs that can aid the detection of the existing known hazardous scenarios and, if possible, more.

6.4.3. SPI Threshold Violations

The SPI threshold violations provide a window into operational discrepancies within the ADS. Notably, deviations in planned paths between different ADS channels during manoeuvres, such as red lights and turns, were observed. These findings suggest inconsistencies in the ADS’s decision-making processes and emphasize the importance of enhanced cross-channel analysis for safe and robust arbitration mechanisms. Ensuring consistent performance across different subsystems is crucial for the reliability and safety of AV operations.

6.4.4. More Hazardous Scenario Identification

From analysing the HSI and SPI modules and observing the ADS, more hazardous scenarios could be identified by the system by properly tuning and defining SPIs. For example, the scenario of an Object crossing in front of the ego vehicle can be identified by looking at the moving variance of the cross channel Planned path similarity scores and Channel Planned Path Similarity scores along with the safety scores. This is because when an object is crossing, the safety scores go down due to a potential collision, and from the inherent behaviours of LAV and TFuse, both the channels give out diverging trajectories to follow (causing a dip in cross-channel planned path scores) and they within themselves change their own trajectory as the object crosses the ego vehicle. By perhaps running more statistics on this score or training an ML model to understand the behaviour of these scores along with others, this object-crossing scenario can be identified.

Another scenario that is already being identified by the current system is the conflicting paths generated. Since the ground truth is unknown whenever there is a significant dip in cross-channel planned path similarity scores and ego trajectory similarity scores, it indicates that the ADS does not agree with each other, possibly both of them seeing the world differently and hence need post-analysis.

By defining more SPIs and/or performing more such analyses, more complex hazardous scenarios that are of interest can be identified, thus contributing to the continuous improvement of AV vehicle safety.

6.4.5. Insights into the ADS Performance

Running the SPI monitoring module on all the routes provided valuable insights into the performance of the Automated Driving Systems (ADS):

1. In CARLA Town 4, it was observed that the Object Count Scores, Similarity Scores, Channel Specific Ego Planned Path Tracker Scores, and Cross Channel Ego Planned Path Tracker Scores crossed their respective thresholds multiple times. Analyzing the scenarios revealed that the perception of LAV is limited in low-light conditions, often detecting numerous ghost objects, as illustrated in Figure 6.14. Furthermore, the ADS were unable to suggest consistent planned paths for the ego vehicle in these conditions.
2. In the same CARLA Town 4, it was observed that the SPIs under the Motion Prediction Metrics (see table 4.4) continued to take a lot of compute time even after a collision, as shown in Figure 6.38. Upon analysis, it was found that both the ADS (LAV and TFuse) continued to suggest planned paths and attempted to drive even after the collision.
3. During turns, the planned paths generated by TFuse and LAV often differed slightly, leading to threshold violations of the respective SPIs. This discrepancy highlights the challenges in achieving consistent path planning across multiple ADS.
4. The LAV channel demonstrated the capability to detect stop signs, whereas the TFuse channel did not possess this functionality. As a result, when the vehicle stopped at a red light, the TFuse channel would continue to plan a path for the ego vehicle, causing violations of the relevant SPI thresholds.
5. The Object Area Similarity Scores consistently remained within the set thresholds throughout all test scenarios, demonstrating a high level of agreement between LAV and TFuse in estimating object areas. This consistency suggests that both ADS channels employ reliable algorithms for object size estimation, which is crucial for accurate perception and decision-making in various driving scenarios.

These insights, derived from the SPI monitoring module, underscore its effectiveness in understanding AV vehicle behaviour. By identifying potential issues related to perception, trajectory planning, and overall safety, the module can prompt ADS manufacturers to investigate and provide updates to enhance AV vehicles' world perception, trajectory planning, and safety. The analysis highlights the value of the SPI monitoring approach in facilitating continuous improvement and ensuring the reliable operation of autonomous vehicles in diverse driving scenarios.

6.4.6. Profiling and Real-Time Performance

The profiling results on the NXP S32G274 processor show the real-time capability of the developed SPI and HSI algorithms. These algorithms consistently operated within set time constrains of 25ms, affirming their feasibility for deployment in real-world AV systems. However, occasional spikes in execution times, particularly in motion prediction modules, highlight areas where computational efficiency could be further optimized.

A notable pattern observed in the similarity analysis execution time is the presence of regular peaks, as shown in Figures 6.24, 6.28, 6.32, 6.36, 6.40, and 6.44. These peaks occur when the number of objects detected by the ADS are high (about 5 and above). The increased execution time is attributed to the trajectory comparison, area calculation, pairing, and matching operations performed for each detected object. The peaks are more prominent in highways and traffic signals scenarios, where long trajectories are predicted for each object. Interestingly, the individual peaks, followed by a dip in the subsequent values, are indicative of the presence of ghost objects. These ghost objects are typically static objects on the sides of the road that are momentarily misclassified by the TFuse channel as vehicles with trajectories. However, these ghost objects quickly disappear, leading to the observed dip in execution time.

The analysis of SPI threshold violations offers valuable insights into the ADS's performance, identifying specific aspects that require further optimization, such as passenger comfort, object trajectory prediction, and cross-channel consistency. These findings can guide targeted improvements in the ADS algorithms and decision-making processes.

In conclusion, the experimental evaluation of the SPI and HSI modules using the CARLA simulator and advance ADS software stacks demonstrates the effectiveness of the proposed techniques in assessing AV safety and identifying hazardous scenarios.

7 Conclusions

This thesis has explored the development and application of SPIs (Safety Performance Indicators) and HSI (Hazardous Scenario Identification) techniques for the continuous improvement of driving automation. The research aimed to address the challenges faced by ADS in expanding their ODDs (Operational Design Domains), handling today's and future unknown hazardous scenarios, and ensuring real-time safety monitoring, particularly in the context of multi-channel architectures.

The proposed SPI framework provides a comprehensive set of metrics for assessing the safety and performance of redundant ADS. The SPIs cover various critical aspects, such as ego localization, object detection, trajectory planning, and overall system behaviour. By defining appropriate thresholds for each SPI, the framework enables the identification of potential safety issues and supports the continuous monitoring and improvement of ADS. The SPI monitoring module has proven to be an effective tool for gaining valuable insights into ADS performance, highlighting issues related to perception, trajectory planning, and overall safety. These insights can prompt ADS manufacturers to investigate and provide updates to enhance the world perception, trajectory planning, and safety of AV vehicles.

The HSI module, developed as part of this thesis along with the Python OpenSCENARIO file generation tool, demonstrates the effectiveness of leveraging SPIs and the Daruma cross-channel analysis for detecting hazardous scenarios. The experimental evaluation, conducted using the CARLA simulator and advanced ADS software stacks (LAV and TFuse), highlights the module's ability to accurately identify hazardous scenarios.

However, the evaluation also reveals the challenges the HSI module faces, including false positives and false negatives. False positives were frequently triggered by misinterpreting other vehicles' movements, especially during turns or at traffic lights, suggesting that the current models may overly generalize from insufficient data or misinterpret the intent of surrounding traffic. False negatives often resulted from situations where the proximity thresholds set for triggering alerts were not breached despite potentially risky scenarios. These findings highlight the need for further improvements in the ADS's perception and localization capabilities and the importance of refining threshold settings or associated HSI algorithms.

The SPI threshold violations observed during the experiments provide valuable insights into the ADS's performance and the causes of these violations. Divergences in the planned paths between channels were observed during red lights and turns, indicating potential issues with the coordination and consistency of the ADS's decision-making processes. These inconsistencies emphasize the importance of cross-channel analysis and the need for robust arbitration mechanisms to ensure safe and reliable ADS operation.

The experimental evaluation also revealed the potential for identifying additional hazardous scenarios by leveraging the SPI framework and cross-channel analysis. For example, the scenario of an object crossing in front of the ego vehicle can be identified by analyzing the moving variance of the cross-channel planned path similarity scores and channel planned path similarity scores, along with the safety scores. Similarly, conflicting paths generated by different ADS channels can be detected by monitoring significant dips in cross-channel planned path similarity scores and ego trajectory similarity scores. These findings suggest that by defining more SPIs and performing in-depth analysis, more complex and critical hazardous scenarios can be identified, contributing to the continuous improvement of AV safety.

The proposed architecture for the continuous improvement of ADS, incorporating SPIs and HSI techniques and the OpenScenario export tool in Python, offers a structured approach to enhance the safety and performance of AVs. By enabling the collection and analysis of real-world driving data, the architecture facilitates the identification of areas for improvement and supports the creation of targeted software updates. The generation of OpenSCENARIO files based on identified hazardous scenarios allows for the reproduction and analysis of these scenarios in simulation environments, enabling the development of more robust and reliable ADS.

The profiling results demonstrate that the developed SPI and HSI algorithms can operate efficiently and meet the time requirements when executed on the NXP S32G274 automotive processor. The execution times remain within the specified limits, with only rare breaches or instances approaching the threshold. The sub-category analysis reveals that the motion prediction time dominates the execution time, while the comfort and similarity time sub-categories exhibit lower execution times and more variability. These findings validate the suitability of the algorithms for real-time automotive applications and provide valuable insights for further optimization and resource allocation.

In conclusion, this thesis has made contributions to the development of safer and more reliable AV by proposing a comprehensive SPI framework, implementing an effective HSI module, and designing an architecture for the continuous improvement of ADS. The experimental evaluation, conducted using a diverse set of routes from the CARLA simulator, provides valuable insights into the performance of the SPI and HSI modules, highlighting their strengths and areas for improvement. The results underscore the potential of the proposed techniques in enhancing the safety and reliability of ADS, while also revealing the challenges and limitations that need to be addressed through further research and development efforts. The insights derived from the SPI monitoring module highlight its effectiveness in understanding AV vehicle behaviour and identifying potential issues related to perception, trajectory planning, and overall safety. These findings can aid ADS manufacturers in their efforts to continuously improve and ensure the reliable operation of autonomous vehicles in diverse driving scenarios.

By continuously refining SPI thresholds, expanding the range of monitored scenarios, and leveraging advanced data analysis techniques, ADS can achieve higher safety and performance standards, effectively addressing current and emerging challenges in AV operations. The insights and methodologies presented in this thesis pave the way for the development of more robust, reliable, and adaptable AVs, bringing us closer to the widespread deployment of safe and trustworthy self-driving cars.

Limitations of the Thesis

The following are the limitations of the thesis:

- One of the main limitations of this work is that the HSI heavily depends on the perception and localization capabilities of the ADS. If all the ADS present do not, for example, detect an object near a vehicle, then the system cannot identify the hazardous scenario. If the ADS present are also very similar, while SPIs can still effectively monitor the safety of the AVs, the HSI may not be able to identify a wide range of hazardous scenarios.
- In terms of the method, the thesis relies on simulation environments for the verification. While the CARLA simulator provides a realistic and diverse range of driving scenarios, it does not fully capture the complexity and unpredictability of real-world driving conditions. The performance of the SPI framework and HSI module in real-world settings may differ from the results obtained in the simulation environment.
- Although the experimental evaluation covered several routes with varying weather conditions and road layouts, the number of driving scenarios tested is still limited compared to the vast array of possible situations an ADS may encounter in the real world. The effectiveness of the SPI framework and HSI module in identifying hazardous scenarios across a broader range of driving conditions remains to be validated.

- The SPI framework and HSI module are designed to complement existing safety mechanisms in ADS, such as functional safety (ISO 26262) and safety of the intended functionality (SOTIF). However, the seamless integration of these techniques with other safety measures may require further research efforts.
- The algorithms developed for this thesis rely on the 2D location and shape information of objects and the ego vehicle provided by the open-source ADS (LAV and TFuse) to Daruma. While this 2D data has been sufficient for developing the SPIs and HSI techniques presented, it is important to acknowledge that production ADS will have access to three-dimensional (3D) information. The availability of 3D data would enable the development of more sophisticated and comprehensive SPIs, allowing for a more accurate assessment of potential hazards and a more precise characterization of the driving scenario. With 3D information, SPIs could consider factors such as object height, the vertical distance from the ego vehicle, road surface slope, and complex environmental geometry, leading to a more comprehensive assessment of the ADS's safety performance and, in return, possibly more hazardous scenarios being identified.

8 Future Work

The future directions based on the conclusion and limitations of the thesis are listed below:

- Incorporating 3D data into the SPI framework and HSI techniques could significantly enhance the safety assessment of AV. Future research should explore the integration of 3D information along with other information about the perceived world by the ADS (such as lanes, static obstacles and traffic signs) to develop more sophisticated and comprehensive SPIs. By leveraging all the information the ADS detects, researchers can create SPIs that consider factors like object height, vertical distance, road surface slope, and complex environment, leading to better safety monitoring and potential hazards.
- As more data is collected from ADS deployments, machine learning models can be trained on the SPIs to enhance the accuracy and efficiency of hazardous scenario identification. By learning patterns and relationships within the SPI data, these models can help detect potential safety compromises and predict the likelihood of hazardous situations. This data-driven approach can complement the rule-based methods used in the HSI module, enabling more adaptive safety monitoring. Additionally, machine learning techniques can be employed to optimize SPI thresholds and improve the overall performance of the safety assurance framework.
- Creation of a sophisticated pattern matching algorithm. In this thesis, the output of the HSI module was run through a basic filter that removed repetitive identifications in a very short period of time. Future research can look into more advanced pattern-matching algorithms such that different hazardous scenarios are identified during one journey, enabling quicker manufacturer analysis.
- Incorporating vehicle dynamics information from the ADS can provide valuable insights into the stability and controllability of the AV. The safety framework can detect potential loss of control situations by defining SPIs that monitor parameters such as lateral acceleration, yaw rate, and slip angles. These SPIs can help identify instances where the vehicle's behaviour deviates from expected patterns or exceeds predetermined thresholds, indicating a possible compromise in the ADS's ability to maintain stable control.

References

[1] D. J. Fagnant and K. Kockelman. “Preparing a Nation for Autonomous Vehicles: Opportunities, Barriers and Policy Recommendations”. In: *Transp. Res. Part A Policy Pract.* 77 (2015).

[2] National Highway Traffic Safety Administration. *Automated Vehicles for Safety*. https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety. (Accessed: June 19, 2024). 2020.

[3] N. Kalra and D. G. Groves. *The Enemy of Good: Estimating the Cost of Waiting for Nearly Perfect Automated Vehicles*. Research Report RR-2150. (Accessed: June 19, 2024). RAND Corporation, 2017. URL: https://www.rand.org/pubs/research_reports/RR2150.html.

[4] A. Shetty et al. “Automated Vehicle Safety and Deployment: Lessons from Human Crashes”. In: *2022 International Conference on Connected Vehicle and Expo (ICCVE)*. 2022, pp. 1–6.

[5] P. Koopman and M. Wagner. “Challenges in Autonomous Vehicle Testing and Validation”. In: *SAE Int. J. Trans. Safety* 4.1 (2016).

[6] J. Z. Varghese and R. G. Boone. “Overview of Autonomous Vehicle Sensors and Systems”. In: *International Conference on Operations Excellence and Service Engineering*. 2015.

[7] SAE International. “SAE J3016 - Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles”. In: *SAE Int.* 4970.724 (2021).

[8] Autoevolution. *AV Safety Expert Proposes a New Classification for Autonomy Levels*. https://www.autoevolution.com/news/av-safety-expert-proposes-a-new-classification-for-autonomy-levels-181024.html. (Accessed: December 15, 2023).

[9] M. Gyllenhammar et al. “Towards an Operational Design Domain That Supports the Safety Argumentation of an Automated Driving System”. In: Jan. 2020.

[10] All icons in this image were created using OpenAI’s DALL-E.

[11] L. Mendiboure et al. “Operational Design Domain for Automated Driving Systems: Taxonomy Definition and Application”. In: *IEEE Intelligent Vehicles Symposium*. 2023.

[12] H. J. Vishnukumar et al. “Machine Learning and Deep Neural Network - Artificial Intelligence Core for Lab and Real-World Test and Validation for ADAS and Autonomous Vehicles: AI for Efficient and Quality Test and Validation”. In: *2017 Intelligent Systems Conference, IntelliSys 2017*. Vol. 2018-January. 2017.

[13] S. Halder, A. Ghosal, and M. Conti. “Secure Over-the-Air Software Updates in Connected Vehicles: A Survey”. In: *Comput. Netw.* 178 (2020).

[14] Daimler. “Safety First for Automated Driving”. In: *White paper of different car manufacturers and suppliers* (2019). (Accessed: June 19, 2024).

[15] T. Stolte et al. “Hazard Analysis and Risk Assessment for an Automated Unmanned Protective Vehicle”. In: *IEEE Intelligent Vehicles Symposium*. 2017.

[16] C. A. J. Hanselaar et al. “The Safety Shell: An Architecture to Handle Functional Insufficiencies in Automated Driving”. In: *IEEE Trans. Intell. Transp. Syst.* (2024), pp. 1–19.

[17] K. Czarnecki. *Operational Design Domain for Automated Driving Systems - Taxonomy of Basic Terms*. July 2018. DOI: 10.13140/RG.2.2.18037.88803.

[18] J. E. Stellet et al. “Testing of Advanced Driver Assistance Towards Automated Driving: A Survey and Taxonomy on Existing Approaches and Open Questions”. In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. 2015, pp. 1455–1462.

[19] A. Wardziński. “Safety Assurance Strategies for Autonomous Vehicles”. In: *Computer Safety, Reliability, and Security*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 277–290.

[20] J. W. Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. 3rd ed. Sage Publications, Inc., 2009.

[21] Y. Fu et al. *Characterization and Mitigation of Insufficiencies in Automated Driving Systems*. 2024. arXiv: 2404.09557 [cs.RO].

[22] A. Terechko et al. *Daruma Design Pattern for Safe and Continuous Automated Driving*. (Accessed: November 20, 2023). 2023. URL: https://community.nxp.com/t5/Blog/Daruma-Design-Pattern-for-Safe-and-Continuous-Automated-Driving/ba-p/1609514.

[23] A. Dosovitskiy et al. "CARLA: An Open Urban Driving Simulator". In: *Proceedings of the 1st Annual Conference on Robot Learning*. 2017, pp. 1–16.

[24] D. Chen and P. Krähenbühl. "Learning from All Vehicles". In: *CVPR*. (Accessed: January 21, 2024). 2022.

[25] H. Chitta et al. "TransFuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving". In: *arXiv e-prints*, arXiv:2205.15997 (May 2022), arXiv:2205.15997. DOI: 10.48550/arXiv.2205.15997.

[26] Carla. *Carla Leader Board*. https://leaderboard.carla.org/. (Accessed: May 21, 2024).

[27] SAE International Standards and ISO. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles - SAE International*. 2019.

[28] P. J. van der Perk. *A Distributed Safety Mechanism for Autonomous Vehicle Software Using Hypervisors*. Oct. 2019.

[29] P. Koopman and M. Wagner. "Autonomous Vehicle Safety: An Interdisciplinary Challenge". In: *IEEE Intell. Transp. Syst. Mag.* 9.1 (2017), pp. 90–96.

[30] P. Koopman. https://archive.org/details/vehicle-automation-modes-sae. (Accessed: December 1, 2023).

[31] P. Koopman. https://safeautonomy.blogspot.com/2022/01/simplified-proposal-for-vehicle.html. (Accessed: April 10, 2024).

[32] P. Koopman and M. Wagner. "Autonomous Vehicle Safety: An Interdisciplinary Challenge". In: *IEEE Intell. Transp. Syst. Mag.* 9.1 (2017).

[33] R. Debouk. "Overview of the Second Edition of ISO 26262: Functional Safety— Road Vehicles". In: *J. Syst. Saf.* 55.1 (2019).

[34] ISO. "ISO / PAS 21448 : 2019 Road Vehicles — Safety of the Intended Functionality". In: *Sustainability* 11.1 (2019).

[35] C. A. J. Hanselaar et al. "Detection and Mitigation of Functional Insufficiencies in Autonomous Vehicles: The Safety Shell". In: *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. 2022, pp. 2021–2027.

[36] P. Koopman. "Credible Autonomy Safety Argumentation". In: *Proceedings of 27th Safety-Critical Systems Symposium* 27 (2019). (Accessed: November 20, 2023). URL: https://users.ece.cmu.edu/~koopman/pubs/Koopman19_SSS_CredibleSafetyArgumentation.pdf.

[37] P. Koopman and P. Laplante. "UL 4600: What to Include in an Autonomous Vehicle Safety Case". In: *Computer* 56.5 (May 2023), pp. 101–104.

[38] C. Gold et al. "Taking Over Control From Highly Automated Vehicles in Complex Traffic Situations: The Role of Traffic Density". In: *Proceedings of the 5th International Conference on Applied Human Factors and Ergonomics AHFE*. July. 2014.

[39] B. W. Smith. "Automated Driving and Product Liability". In: *Detroit College of Law at Michigan State University Law Review* 2017.1 (2016). (Accessed: November 20, 2023). URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2923240.

[40] S. Parkinson et al. "Cyber Threats Facing Autonomous and Connected Vehicles: Future Challenges". In: *IEEE Trans. Intell. Transp. Syst.* 18.11 (2017).

[41] A. Ferdowsi et al. "Robust Deep Reinforcement Learning for Security and Safety in Autonomous Vehicle Systems". In: *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*. Vol. 2018-November. 2018.

[42] T. Colwell et al. “An Automated Vehicle Safety Concept Based on Runtime Restriction of the Operational Design Domain”. In: *IEEE Intelligent Vehicles Symposium, Proceedings*. Vol. 2018-June. 2018.

[43] T. Ponn et al. “Identification of Challenging Highway-Scenarios for the Safety Validation of Automated Vehicles Based on Real Driving Data”. In: *2020 Fifteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*. 2020, pp. 1–10. DOI: 10.1109/EVER48776.2020.9242539.

[44] S. Ulbrich et al. “Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving”. In: *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*. Vol. 2015-October. 2015.

[45] A. Terechko et al. *Daruma Design Pattern for Safe and Continuous Automated Driving*. (Accessed: November 20, 2023). NXP Blog. Mar. 2023. URL: https://community.nxp.com/t5/New-to-NXP-Blog/Daruma-Design-Pattern-for-Safe-and-Continuous-Automated-Driving/ba-p/1368047.

[46] A. Boubakri and S. M. Gammar. “A New Architecture of Autonomous Vehicles: Redundant Architecture to Improve Operational Safety”. In: vol. 1. Sept. 2021, pp. 355–368. DOI: 10.31763/ijrcs.v1i3.437.

[47] T. Ishigooka and S. Honda H. Takada. “Cost-Effective Redundancy Approach for Fail-Operational Autonomous Driving System”. In: *2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC)*. 2018, pp. 107–115. DOI: 10.1109/ISORC.2018.00023.

[48] C. Katrakazas, M. Quddus, and W. H. Chen. “A New Integrated Collision Risk Assessment Methodology for Autonomous Vehicles”. In: *Accid. Anal. Prev.* 127 (2019).

[49] L. Oliveira et al. “Driving Style: How Should an Automated Vehicle Behave?” In: *Information* 10.6 (2019).

[50] Matlab. https://en.wikipedia.org/wiki/MATLAB. (Accessed: June 1, 2024).

[51] A. Sinha, S. Chhaya, and R. Rai. “Real-time Diagnostics and Prognostics for Autonomous Vehicles: A Survey”. In: *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE. 2019, pp. 1–8.

[52] D. Zhao et al. “Accelerated Evaluation of Automated Vehicles Safety in Lane-Change Scenarios Based on Importance Sampling Techniques”. In: *IEEE Trans. Intell. Transp. Syst.* 18.3 (2017). (Accessed: June 19, 2024).

[53] *Safety Pool - Powered by Deepen AI and WMG University of Warwick*. https://www.safetypool.ai/. (Accessed: January 14, 2024).

[54] H. Elrofai et al. *StreetWise: Scenario-Based Safety Validation of Connected and Automated Driving*. Tech. Rep. Helmond: TNO, July 2018.

[55] S. Geronimi and E. Arnoux. *ADScene - A Path to a European Scenarios Database for ADS and ADAS Specification, Validation and Homologation*. https://www.vvm-projekt.de/fileadmin/user_upload/Mid-Term/Presentations/VVM_HZE_EmmanuelArnoux.pdf. (Accessed: January 12, 2024). Mar. 2022.

[56] X. Xiong et al. “A Forward Collision Avoidance Algorithm Based on Driver Braking Behavior”. In: *Accid. Anal. Prev.* 129 (2019), pp. 30–43.

[57] P. Koopman. *Safety Performance Indicators (SPIs) for Autonomous Vehicles*. (Accessed: November 22, 2023). 2020. URL: https://users.ece.cmu.edu/~koopman/lectures/L124_SPI_vs_KPI.pdf.

[58] UL4600. *UL 4600: Standard for Evaluation of Autonomous Products*. Standard for safety. (Accessed: November 12, 2023). Underwriters Laboratories, 2020. URL: https://books.google.nl/books?id=7fH3zQEACAAJ.

[59] AVSC. *AVSC Best Practice for Safety Assessment of Automated Driving Systems*. (Accessed: December 11, 2023). 2023.

[60] AVSC. *AVSC Best Practice for Metrics and Methods for Assessing Safety Performance of Automated Driving Systems*. (Accessed: December 11, 2023). 2023.

[61] AVSC. *AVSC Best Practice for Developing ADS Safety Performance Thresholds Based on Human Driving Behavior*. (Accessed: December 11, 2023). 2023.

[62] AVSC. *AVSC Best Practice for Metrics and Methods for Assessing Safety Performance of Automated Driving Systems*. (Accessed: December 11, 2023). 2021.

[63] P. Koopman et al. "A Safety Standard Approach for Fully Autonomous Vehicles". In: *Computer Safety, Reliability, and Security*. (Accessed: November 12, 2023). Aug. 2019, pp. 326–332. ISBN: 978-3-030-26249-5. DOI: 10.1007/978-3-030-26250-1_26. URL: https://users.ece.cmu.edu/~koopman/pubs/Koopman19_WAISE_UL4600.pdf.

[64] M. Wagner and C. Carlan. *The Open Autonomy Safety Case Framework*. 2024. arXiv: 2404.05444 [cs.SE].

[65] Waymo. *Sharing Our Safety Framework for Fully Autonomous Operations*. https://waymo.com/blog/2020/10/sharing-our-safety-framework/. (Accessed: April 10, 2024). 2020.

[66] Aurora. *Aurora Safety Case Framework*. https://safetycaseframework.aurora.tech/gsn. (Accessed: December 20, 2023).

[67] V3 Smart Technologies. *How AI Driving Risk Management Enhances AV Safety in Fleet Management*. https://v3smarttech.com/blog/how-ai-driving-risk-management-enhances-av-safety-in-fleet-management/. (Accessed: March 12, 2024).

[68] Volpis. *AI and ML in Fleet Management (Guide for 2024)*. https://volpis.com/blog/ai-and-ml-in-fleet-management/. (Accessed: January 19, 2024). 2024.

[69] Waymo. *Waymo Safety Case Approach White Paper*. https://assets.ctfassets.net/e6t5diu0txbw/66j0jPtNIjzawaK0ZjpU3q/7f081b392cf29a3355c97d0d758fe6cf/Waymo_Safety_Case_Approach.pdf. (Accessed: December 25, 2023).

[70] Nvidia. *Self-driving Safety Report*. https://www.nvidia.com/content/dam/en-zz/Solutions/self-driving-cars/safety-report/auto-print-safety-report-final-web.pdf. (Accessed: December 5, 2023).

[71] M. Bojarski et al. "End to End Learning for Self-Driving Cars". In: *arXiv* abs/1604.07316 (2016). (Accessed: December 11, 2023). URL: https://api.semanticscholar.org/CorpusID:15780954.

[72] IVEX. *Scenario Based Testing within IVEX*. https://webinars.sw.siemens.com/en-US/scenario-based-testing-for-adas-and-autonomous-vehicles/. (Accessed: December 11, 2023).

[73] ASAM. *OpenScenario Guide*. https://www.asam.net/index.php?eID=dumpFile&t=f&f=4092&token=d3b6a55e911b22179e3c0895fe2caae8f5492467. (Accessed: March 12, 2024).

[74] Esmini Open Scenario Player. https://esmini.github.io/. (Accessed: March 1, 2024).

[75] M. Quigley et al. "ROS: an Open-Source Robot Operating System". In: *ICRA Workshop on Open Source Software*. Vol. 3. 3.2. 2009.

[76] J. Zhang et al. "Trajectory Planning with Comfort and Safety in Dynamic Traffic Scenarios for Autonomous Driving". In: *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. 2021, pp. 342–349.

[77] *Shoelace Formula*. https://en.wikipedia.org/wiki/Shoelace_formula. (Accessed: January 19, 2024).

[78] *How Long Should Your Dog Leash Be?* https://medium.com/tblx-insider/how-long-should-your-dog-leash-be-ba5a4e6891fc. (Accessed: January 19, 2024).

[79] T. W. T. van Diggelen. "Efficiently Answering Fréchet Queries". (Accessed: March 1, 2024). Master's thesis. Location of University: TU Eindhoven, 2023.

[80] K. Buchin et al. "Fréchet Distance for Uncertain Curves". In: *ACM Trans. Algorithms* 19.3 (July 2023), pp. 1–47. DOI: 10.1145/3597640.

[81] R. van der Horst et al. "Traffic Conflicts on Bicycle Paths: A Systematic Observation of Behaviour from Video". In: *Accid. Anal. Prev.* 62 (Apr. 2013). DOI: 10.1016/j.aap.2013.04.005.

[82] R. van der Horst and J. Hogema. “Time-To-Collision and Collision Avoidance Systems”. In: (Jan. 1994).

[83] *Precision and Recall*. https://en.wikipedia.org/wiki/Precision_and_recall. (Accessed: June 17, 2024).

[84] *Precision and Recall | Essential Metrics for Machine Learning*. https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/. (Accessed: June 17, 2024).

[85] Carla Leaderboard Town 1. https://carla.readthedocs.io/en/latest/map_town01/. (Accessed: April 10, 2024).

[86] Carla Leaderboard Town 2. https://carla.readthedocs.io/en/latest/map_town02/. (Accessed: April 10, 2024).

[87] Carla Leaderboard Town 3. https://carla.readthedocs.io/en/latest/map_town03/. (Accessed: April 10, 2024).

[88] Carla Leaderboard Town 4. https://carla.readthedocs.io/en/latest/map_town04/. (Accessed: April 10, 2024).

[89] Carla Leaderboard Town 5. https://carla.readthedocs.io/en/latest/map_town05/. (Accessed: April 10, 2024).

[90] Carla Leaderboard Town 6. https://carla.readthedocs.io/en/latest/map_town06/. (Accessed: April 10, 2024).

[91] NXP Semiconductors. https://www.nxp.com/products/processors-and-microcontrollers/s32-automotive-platform/s32g-vehicle-network-processors/s32g2-processors-for-vehicle-networking:S32G2. (Accessed: June 1, 2024).