SAFETY ASSURANCE FRAMEWORK FOR CONNECTED, AUTOMATED MOBILITY SYSTEMS

D5.3

Quality metrics for scenario database content

Project short name
SUNRISE

Project full name
Safety assUraNce fRamework for connected, automated mobIlity SystEms

Horizon Research and Innovation Actions | Project No. 101069573
Call HORIZON-CL5-2021-D6-01

Funded by
the European Union

ccam-sunrise-project.eu/

Dissemination level	Public (PU) - fully open
Work package	WP5: Content harmonisation of scenario data framework
Deliverable number	D5.3: Quality metrics for scenario database content
Deliverable responsible	Erwin de Gelder, TNO
Status - Version	Final – V1.0
Submission date	08/07/2024
Keywords	Scenario, Metrics, Coverage, Representativity

Authors/Contributors

Role	Name
Main author	Erwin de Gelder (TNO)
Contributing authors	Emre Kaynar, Eren Mungan, Hannes Schneider (AVL) Anastasia Bolovinou, Elena Daskalaki (ICCS) Fouad Hadj Selem (VEDECOM) Sergi Vidal (IDIADA) Sven Tarlowski (ika) Jason Xizhe Zhang, Jerein Jeyachandran (UoW) Tajinder Singh, Mohsen Alirezaei (Siemens) Marcos Nieto Doncel, Yavar Taghipour Azar (Vicomtech)

Quality Control

	Name	Organisation	Date
Peer review 1	Stefan de Vries	IDIADA	02/06/2025
Peer review 2	Anders Thorsén	RISE	09/06/2025
Peer review 3	Sjef van Montfort	TNO	13/06/2025

Version history

Version	Date	Author	Summary of changes
0.1	09/01/2024	All authors	Initial version
0.2	01/10/2024	All authors	Intermediate version as part of milestone MS5

0.3	23/05/2025	All authors	Updated deliverable ready for peer review
0.4	10/06/2025	Peer reviewers	Peer review completed
1.0	08/07/2025	All authors	Final version

Legal disclaimer

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

Abbreviation	Meaning
AD	Automated Driving
ADS	Automated Driving System
ALKS	Automated Lane Keeping System
API	Application Programming Interface
CCAM	Connected, Cooperative, and Automated Mobility
CIPV	Closest In-Path Vehicle
CP	Criticality Phenomenon
CPM	Collective Perception Message
DDT	Dynamic Driving Task
DSCQ	Dynamic Scenario Complexity Quantification
DTW	Dynamic Time Warping
FOT	Field Operational Test
FVAE	Factor Variational Autoencoder
GIDAS	German In-Depth Accident Study
GP	Gaussian Process
GPC	Gaussian Process Classification
GUI	Graphical User Interface
HiL	Hardware in the Loop
HLP	High-Level Parameter
HuD	Head-up Display
KDE	Kernel Density Estimation
LCL	Lane Change Left
LCR	Lane Change Right
NDD	Naturalistic Driving Data
NF	Normalizing Flows

OD	Operational Domain
ODD	Operational Design Domain
PDF	Probability Density Function
PSS	Parameter Similarity Score
SDL	Scenario Description Language
SAF	Safety Assurance Framework
SBERT	Sentence Bidirectional Encoder Representations from Transformers
SC	Scenario Category
SCDB	SCenario DataBase
SDL	Scenario Description Language
SiL	Software in the Loop
SOTIF	Safety of the Intended Functionality
SSM	Surrogate Safety Metric
STP	Stop
SUT	System Under Test
TP	Traffic Participant
TSS	Tag Similarity Score
TTC	Time-to-Collision
V&V	Verification and Validation
ViL	Vehicle in the Loop
UC	Use Case
URI	Uniform Resource Identifier
WP	Work Package

EXECUTIVE SUMMARY

Safety assurance of Cooperative, Connected, and Automated Mobility (CCAM) systems is a crucial factor for their successful adoption in society, yet it remains a significant challenge. It is generally acknowledged that for higher levels of automation, the validation of these systems by conventional test methods would be infeasible. Furthermore, certification initiatives worldwide struggle to define a harmonized safety assurance approach enabling massive deployment of CCAM systems.

The **SUNRISE** project develops and demonstrates a **CCAM Safety Assurance Framework** (SAF). The overall objective of the SUNRISE project is to accelerate the large-scale and safe deployment of CCAM systems. In alignment with international twin projects and initiatives, the project aims to achieve this objective by providing a SAF consisting of three main components: a Method, a Toolchain and a Data Framework. The **Method** is established to support the SAF safety argumentation, and includes procedures for scenario selection, sub-space creation, dynamic allocation to test instances and a variety of metrics and rating procedures. The **Toolchain** contains a set of tools for safety assessment of CCAM systems, including approaches for virtual, hybrid and physical testing. The **Data Framework** provides online access, connection and harmonization of external Scenario Databases (SCDBs), allowing its users to perform query-based extraction of safety relevant scenarios, allocation of selected scenarios to a variety of test environments, and reception of the test results.

This deliverable, D5.3 "Quality metrics for scenario database content", represents a contribution to Work Package 5 of the SUNRISE project. It addresses methods to evaluate the quality of SCDBs, which play a central role in the validation of Connected, Cooperative, and Automated Mobility (CCAM) systems. By providing a structured set of quality metrics, this deliverable directly supports the overarching project goal of establishing a SAF that is harmonized, scalable, and transparent.

The document provides contributions to five key types of quality metrics, each targeting a different aspect of scenario database assessment:

- **Testing Purpose Metrics**: Evaluate how well a scenario or scenario set aligns with its intended use, such as validating specific system functions or addressing particular safety concerns.
- **Scenario Description Metrics**: Assess the completeness, clarity, and consistency of scenario definitions, including parameter validity and documentation quality.
- **Scenario Exposure Metrics**: Quantify how frequently a scenario – or a type of scenario – occurs in the real world, often based on statistical traffic data.
- **Dissimilarity Metrics**: Measure the diversity among scenarios by evaluating how different they are from one another, helping to avoid redundancy.
- **Scenario Coverage Metrics**: Evaluate how comprehensively the scenario set spans relevant parameter spaces, operational design domains, or system functions.

These metrics are based on both an in-depth literature review and novel developments within the SUNRISE project. They allow stakeholders to assess individual scenarios as well as entire

scenario sets in terms of completeness, relevance, diversity, and representativeness. The deliverable also demonstrates the feasibility of applying these metrics through illustrative examples using real-world scenarios, validating their practical utility.

This work is relevant to SAF users and SAF auditors, such as SCDB developers, testers, and policymakers, as it offers implementable guidelines for assessing and improving scenario quality in a consistent manner. The introduction of a risk-based scenario relevance metric, along with metrics for coverage and diversity, contributes to the advancement in scenario-based testing. These outcomes are intended to be integrated into SCDB interfaces, enabling more effective scenario selection, database optimization, and test planning.

The work lays the foundation for upcoming tasks, including the integration of these metrics into the SAF for streamlined scenario generation and evaluation. Overall, this deliverable provides actionable insights that will support harmonization and innovation in CCAM safety assurance across Europe and worldwide.

1 INTRODUCTION

1.1 SUNRISE project

Safety assurance of Connected, Cooperative, and Automated Mobility (CCAM) systems is a crucial factor for their successful adoption in society, yet it remains a significant challenge. CCAM systems need to demonstrate reliability in all driving scenarios, requiring robust safety argumentation. It is acknowledged that for higher levels of automation, the validation of these systems by means of real test-drives would be infeasible. In consequence, a carefully designed mixture of physical and virtual testing has emerged as a promising approach, with the virtual part bearing more significant weight for cost efficiency reasons.

Worldwide, several initiatives have started to develop test and assessment methods for Automated Driving (AD) functions. These initiatives already transitioned from conventional validation to a scenario-based approach and combine different test instances (physical and virtual testing) to avoid the million-mile issue.

The initiatives mentioned above, provide new approaches to CCAM validation, and many expert groups formed by different stakeholders, are already working on CCAM systems' testing and quality assurance. Nevertheless, the lack of a common European validation framework and homogeneity regarding validation procedures to ensure safety of these complex systems, hampers the safe and large-scale deployment of CCAM solutions. In this landscape, the role of standards is paramount in establishing common ground and providing technical guidance. However, standardising the entire pipeline of CCAM validation and assurance is in its infancy, as many of the standards are under development or have been very recently published and still need time to be synchronised and established as common practice.

Scenario Databases (SCDBs) are another issue tackled by several initiatives and projects, that generally tends to silo solutions. A clear concrete approach should be used (at least at European level), dealing with scenarios of any possible variations, including the creation, editing, parameterisation, storing, exporting, importing, etc. in a universally agreed manner.

Furthermore, validation methods and testing procedures still lack appropriate safety assessment criteria to build a robust safety case. These must be set and be valid for the whole parameter space of scenarios. Another level of complexity is added, due to regional differences in traffic rules, signs, actors and situations.

Evolving from the achievements obtained in HEADSTART and taking other project initiatives as a baseline, it becomes necessary to move to the next level in the development and demonstration of a commonly accepted **Safety Assurance Framework** (**SAF**) for the safety validation of CCAM systems, including a broad portfolio of Use Cases (UCs) and comprehensive test and validation tools. This will be done in **SUNRISE**, which stands for **S**afety ass**U**ra**N**ce f**R**amework for connected, automated mob**I**lity **S**yst**E**ms.

The SAF is the main product of the SUNRISE project. As the following figure indicates, it takes a central role, fulfilling the needs of different automotive stakeholders that all have their own interests in using it, see Figure *1*.

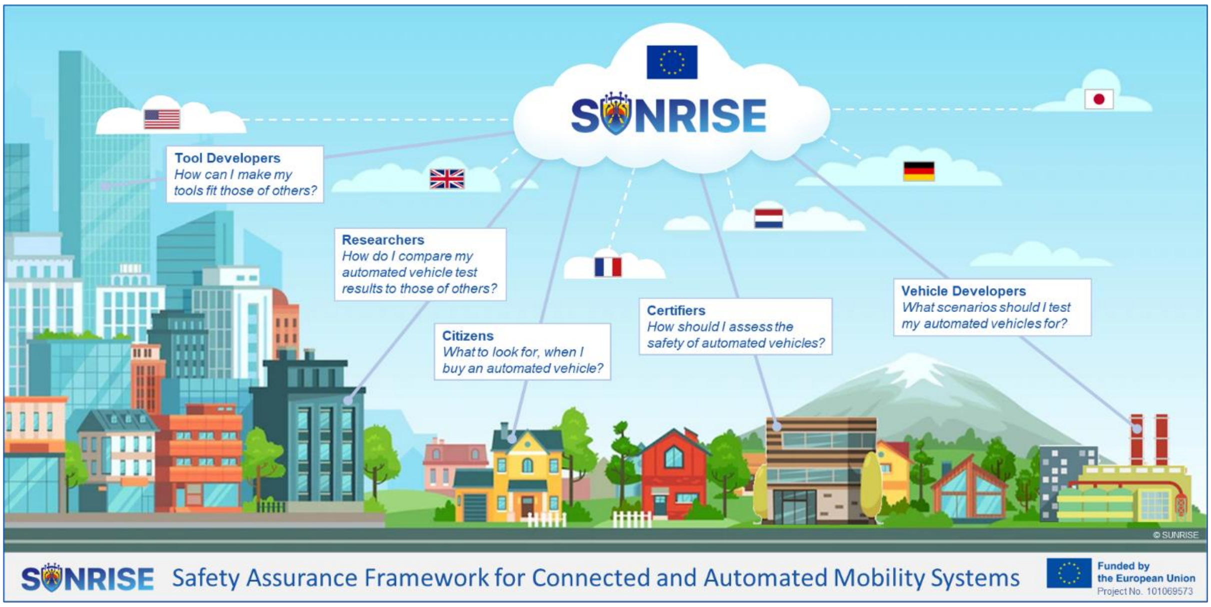

SUNRISE
Tool Developers
How can I make my tools fit those of others?
Researchers
How do I compare my automated vehicle test results to those of others?
Citizens
What to look for, when I buy an automated vehicle?
Certifiers
How should I assess the safety of automated vehicles?
Vehicle Developers
What scenarios should I test my automated vehicles for?
© SUNRISE
SUNRISE Safety Assurance Framework for Connected and Automated Mobility Systems
Funded by the European Union
Project No. 101069573

Figure 1: Safety Assurance Framework stakeholders

The **overall objective** of the SUNRISE project is to accelerate the safe deployment of innovative CCAM technologies and systems for passengers and goods by creating demonstrable and positive impact towards safety, specifically the EU's long-term goal of moving close to zero fatalities and serious injuries by 2050 (Vision Zero), and the resilience of (road) transport systems. The project aims to achieve this objective by providing a SAF consisting of three main components: a Method, a Toolchain and a Data Framework. The **Method** is established to support the SAF safety argumentation, and includes procedures for scenario selection, sub-space creation, dynamic allocation to test instances and a variety of metrics and rating procedures. The **Toolchain** contains a set of tools for safety assessment of CCAM systems, including approaches for virtual, hybrid and physical testing. The **Data Framework** provides online access, connection and harmonization of external Scenario Databases (SCDBs), allowing its users to perform query-based extraction of safety relevant scenarios, allocation of selected scenarios to a variety of test environments, and generation of the test results. The SAF will be put to the test by a series of **Use Cases demonstrations**, designed to identify and solve possible errors, gaps and improvements to the underlying methods, tools and data.

Following a common approach will be crucial for present and future activities regarding the testing and validation of CCAM systems, allowing to obtain results in a standardised way, to improve analysis and comparability, hence maximising the societal impact of the introduction of CCAM systems.

Figure 2 shows the general workplan of the SUNRISE project.

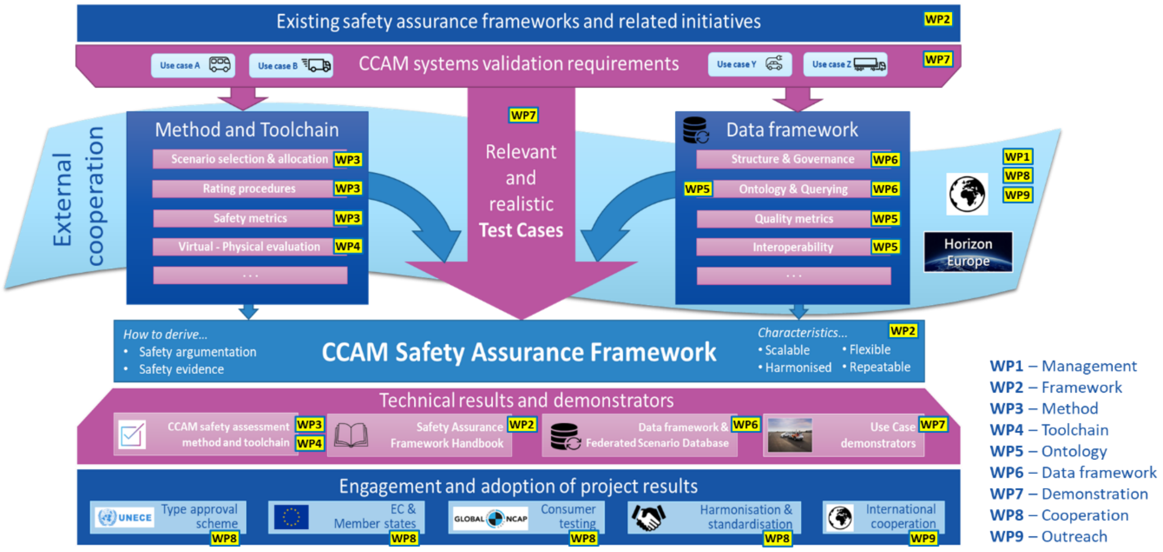


Figure 2: Workplan of the SUNRISE Project

1.2 Purpose of the deliverable

The main objective of Work Package (WP) 5 is to define a future-proof set of commonly accepted and harmonised descriptions for the definition of the SUNRISE Data Framework. This Data Framework is essential to ensure a robust scenario-based safety assessment of CCAM systems, where a scenario is understood as a description of the temporal development of a traffic constellation. An important component of the scenario-based testing is the source from which the designation of executed tests is selected, the so-called SCenario DataBase (SCDB). WP5 targets to describe the full required properties and features of such SCDBs.

Since the tests of CCAM systems relies on SCDBs, it is essential that the content of these SCDBs meets certain requirements. For example, the scenarios of SCDBs must sufficiently represent what the CCAM system will encounter on the road. Also, the scenarios need to span the Operational Design Domain (ODD) definition of the Automated Driving Systems (ADSs) and/or CCAM systems.

The purpose of this deliverable is to describe quality metrics of the content of the SCDBs. These quality metrics could pertain to individual scenarios, such as the likelihood of the scenario occurring in the real world or the relevance of a scenario given a certain testing purpose. The quality metrics could also pertain to sets of scenarios. For example, the similarity or diversity of a group of scenarios or the degree to which a set of scenarios cover an ODD. It is expected that the metrics could be valuable metadata for individual scenarios as well as sets of scenarios.

This deliverable will provide an overview of the different types of quality metrics related to SCDBs. An overview is given of relevant metrics that are available in the literature. In addition, this deliverable will present novel quality metrics that have been developed as part of the SUNRISE project.

The partner contributions to this deliverable are summarized in Table 1.

Table 1: Partner contribution to D5.3.

Partner	Role
TNO	Led the task, contributed to metrics for coverage and exposure.
AVL	Contributed to research over existing metrics on scenario coverage, developed Scenario Completeness Metric and demonstrated the metric with UC 2.1 Scenarios using AVL SCDB.
ICCS	Contributed to metrics for scenario criticality and complexity.
IDIADA	Contributed to metrics for testing purposes and scenario exposure.
ika	Contributed by undertaking research into the existing scenario completeness metrics and based on these findings, further developing the guidelines for the scenario completeness assessment which were then applied within the SUNRISE use case 2.1.
Siemens	Contributed to metrics for scenario dissimilarity
University of Warwick	Contributed to scenario description requirements, coverage metrics for SCDBs and to Chapter 2 of the deliverable
VEDECOM	Contributed to coverage metrics and gap analysis in a scenarios database using a generative AI technique.
Vicomtech	Contributed to similarity metrics for scenarios at various levels of abstraction.

1.3 Intended audience

This deliverable serves multiple stakeholders. The main stakeholder is the SUNRISE project itself, as this deliverable presents quality metrics that can be utilized to quantify several aspects of the SCDBs that connect to the Data Framework developed in WP6.

This deliverable is also useful for SCDB owners and SCDB hosts. The presented quality metrics might be implemented as part of the Application Programming Interface (API) or Graphical User Interface (GUI), if applicable, of their SCDB. Also, the resulting values of the quality metrics might be used to define what kind of data might need more attention during the data collection process.

Other stakeholders for which this deliverable is intended are the users and possible auditors of the SAF and, more specifically, the users of the SCDBs. To properly assess the safety of an ADS or CCAM system, it is important to know the quality of the scenarios that form the input of the safety assessment. The stakeholders could use the presented quality metrics for this purpose.

1.4 Structure of the deliverable and its relation with other work packages/deliverables

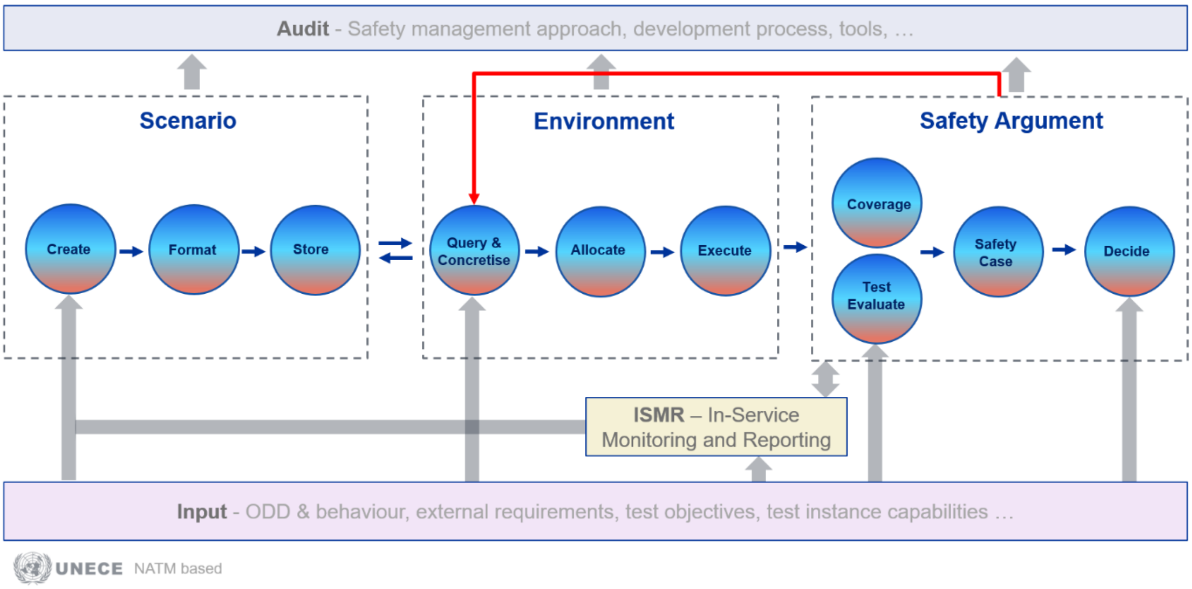
This deliverable is structured as follows. Chapter 2 further explains the use of metrics within the scope of SUNRISE, and more specifically the SAF that is developed within SUNRISE. Chapter 3 lists the different types of metrics that are considered in this deliverable. Metrics that are available in the literature are discussed in Chapter 4. Chapter 5 presents the metrics that are developed within the SUNRISE project. The results of applying the developed metrics of Chapter 5 are presented in Chapter 6. This deliverable ends with conclusions and directions for future work in Chapter 7.

The following deliverables are related to this deliverable (D5.3):

- D2.3: Final SUNRISE SAF. This is the deliverable from Task 2.2 and describes the SAF. The metrics presented in this deliverable are utilized within the SUNRISE SAF.
- D3.4 [1]: Report on Scenario Selection and Subspace Creation Methodology. This deliverable discusses methods to generate concrete scenarios. Part of the methods use an iterative approach where new scenarios are generated based on metrics related to an earlier set of generated scenarios, such as a scenario similarity and scenario coverage metrics. Thus, the (dis)similarity metrics and coverage metrics presented in this deliverable can be employed by methods discussed in D3.4.
- D5.1 [2]: Requirements for CCAM safety assessment data framework content. Part of the metrics presented in this deliverable should quantify whether the scenarios from the SCDBs comply with the requirements or the extent to which they comply.

2 SAFETY ASSURANCE FRAMEWORK

Based on the latest SAF draft, see Figure 3, this section will follow the same workflow in order to provide an overview of the various places the concept of metrics are discussed during the project.



Audit - Safety management approach, development process, tools, …
Scenario
Create
Format
Store
Environment
Query & Concretise
Allocate
Execute
Safety Argument
Coverage
Test Evaluate
Safety Case
Decide
ISMR – In-Service Monitoring and Reporting
Input - ODD & behaviour, external requirements, test objectives, test instance capabilities …
UNECE NATM based

Figure 3: Safety Assurance Framework (SAF) workflow.

Creating diverse and comprehensive scenarios is crucial to ensure a well-rounded safety assurance process. As an example, if a set of scenarios is collected under some specific conditions, there might be a lack of diversity, which is why it is useful to measure this diversity using a diversity metric. This metric can also be applied to optimize the creation of knowledge-based scenarios. In fact, various metrics could be utilized in this context, although it is important to note that the creation of scenarios goes beyond the scope of SUNRISE, but using such diversity metrics to evaluate the retrieved scenarios is within the scope.

When considering the scenario format, it is essential that the formats cater to multiple levels of abstraction, aligning with the testing purposes and ensuring they are interpretable by users who may not be as technically proficient. These formats should also be standardized to facilitate consistent understanding. Another key consideration is data completeness, which assesses how much of the original content is accurately covered by the format.

For the scenario storage and scenarios concretization, the focus will be on the scenario processability, scenario similarity, scenario coverage, scenario exposure, and data completeness. When allocating these scenarios to test instances, the testing purpose can guide the allocation process, while ensuring that the information covered in the scenario descriptions and that the data completeness is adequately addressed for the specific instance. During the execution of test scenarios, the testing purpose informs what needs to be measured, a process that might have already been partly addressed in the concretization phase.

Coverage analysis is another vital step, measuring the extent to which test scenarios cover the ODD and behaviour scope, and ensuring diversity within the scenarios. As part of the coverage analysis, it is important to consider how much of the logical scenario parameter ranges have been covered by individual concrete scenarios with their concrete parameter values through the parameter exploration methods. Such exploration is determined by the performance of the system under test (SUT) and evaluated through testing. Other coverage consideration points may include the diversity of scenario generation sources, and the diversity of (test) scenarios. An example of the diversity of scenario generation sources could be: a given SUT that is tested only using traffic-rule-derived scenarios. Even if they cover the ODD spaces of the system, from a scenario source perspective, the diversity can benefit from other sources such as accident data. An example of the diversity of test instances could be an SUT that is purely tested in virtual environment, although the test instances are validated, the diversity of the test environment should consider evidence from real-world test tracks or public road test. In addition to Coverage, the Test Evaluate block assesses individual test executions with the pass/fail criteria, such criteria are mostly part of the user input based on the individual use cases. Safety case will then consolidate the outcome from both the Coverage and Test Evaluate to form Safety Argument for the final safety assurance outcome.

3 TYPES OF METRICS

This chapter provides an overview of the different types of quality metrics for SCDBs. An overview of the different types of metrics is shown in Figure 4. Six different types of metrics are considered:

1. Metrics related to the testing purpose, which are further divided into metrics related to scenario relevance, scenario criticality, and scenario complexity;
2. Metrics related to the scenario description;
3. Metrics related to the scenario exposure;
4. Metrics related to the (dis)similarity of a set of scenarios;
5. Coverage of scenarios, for which we distinguish between ODD coverage by scenarios and the coverage of a parameter space; and
6. General SCDB metrics.

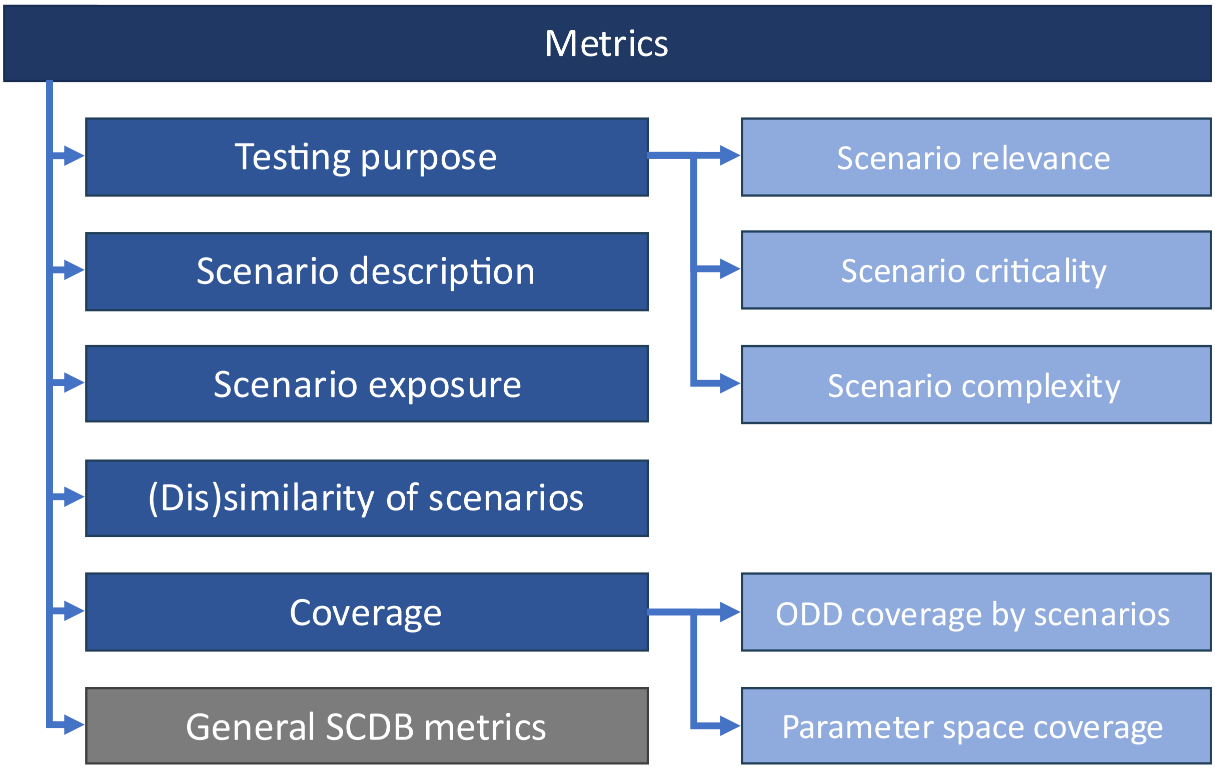


Figure 4: Overview of the types of metrics considered in this deliverable. For the metrics in blue, also new metrics are proposed in this deliverable (see Chapter 5).

Figure 5 provides an overview of the uses of the first five types of metrics within the SAF. Each metric type contributes to specific phases of the SAF workflow, supporting scenario generation, selection, execution, and the formulation of safety arguments:

- *Testing purpose metrics*: Metrics related to the testing purpose – including scenario relevance, criticality, and complexity – are instrumental across multiple stages of the SAF. During the concretisation phase, scenario relevance helps determine which logical scenarios should be instantiated into concrete test cases. In the allocation phase, both relevance and complexity influence how scenarios are assigned to test environments or configurations. Furthermore, in the execution phase, these metrics guide the level of testing detail and focus, as higher relevance or complexity may necessitate more intensive or targeted testing approaches.
- *Scenario description metrics*: Metrics assessing the quality and completeness of scenario descriptions are particularly relevant during format validation and scenario allocation. These metrics ensure that scenarios include all required information for execution in the intended simulation or test environment. A well-defined description may influence the selection of the test platform, as certain environments (e.g., SiL vs. HiL) may require different levels of detail or fidelity.
- *Scenario exposure metrics*: Scenario exposure metrics, which quantify the likelihood of encountering a given scenario in real-world driving, inform both the concretisation process and the safety case development. Scenarios with higher probability may be prioritised for testing, ensuring that the assessed system is validated against representative and regulatory-relevant cases. Additionally, exposure metrics can serve as weighting factors in the safety argumentation, enhancing the credibility of risk-based safety assessments.
- *(Dis)similarity metrics*: Metrics assessing scenario similarity or diversity are valuable in both the creation and evaluation of scenario sets. During scenario creation, especially when using knowledge-based methods, similarity metrics help ensure that generated scenarios are meaningfully distinct and cover a broad range of operating conditions. Within the coverage analysis phase, these metrics support the evaluation of scenario set diversity, ensuring that the testing scenarios are diverse.
- *Coverage metrics*: Coverage metrics are foundational to the SAF and are directly applicable in the concretisation, coverage analysis, and safety case development phases. During concretisation, these metrics ensure that the resulting scenarios sufficiently span the system's ODD. In the coverage phase, they provide quantitative assessments of the extent to which the scenario set addresses relevant operational conditions. Finally, in the safety case, coverage metrics underpin argumentation of completeness, helping to justify that the system has been validated across all relevant operational contexts.

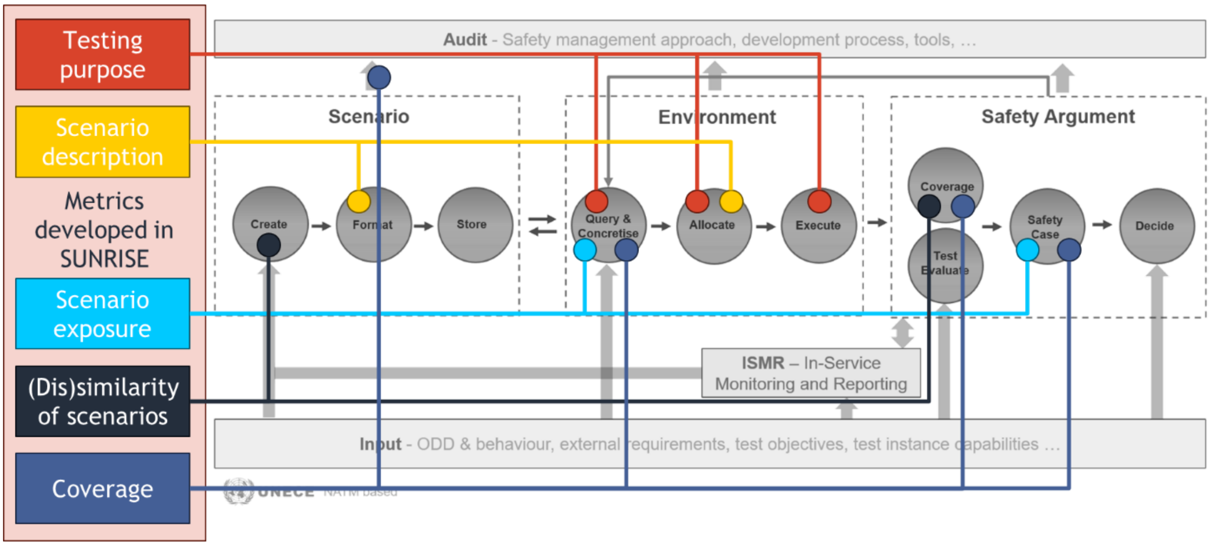

Testing purpose
Scenario description
Metrics developed in SUNRISE
Scenario exposure
(Dis)similarity of scenarios
Coverage
Audit - Safety management approach, development process, tools, …
Scenario
Environment
Safety Argument
Create
Format
Store
Query & Concretise
Allocate
Execute
Coverage
Test Evaluate
Safety Case
Decide
ISMR – In-Service Monitoring and Reporting
Input - ODD & behaviour, external requirements, test objectives, test instance capabilities …
UNECE NATM based

Figure 5: Overview of where the different types of metrics can contribute to the Safety Assurance Framework (SAF) that has been introduced in Chapter 2.

Each of the six different types of metrics are shortly introduced in this chapter. In Chapter 4, metrics related to these seven types of metrics are discussed. In this deliverable, new metrics related to the first five types of metrics, denoted the blue colour in Figure 4, are proposed. For the general SCDB metric (in grey in Figure 4), no new metrics are introduced in this deliverable as existing metrics are adopted in SUNRISE.

3.1 Testing purpose

Testing purpose metrics refer to the criteria used to evaluate and select scenarios within an SCDB to ensure comprehensive testing and validation of CCAM systems. These metrics encompass aspects such as scenario relevance (Section 5.1.1), which determines how closely a scenario aligns with the specific testing requirements and objectives; scenario criticality, which assesses the potential risk or danger presented by the scenario, ensuring the robustness of safety mechanisms; and scenario complexity, which assesses the degree of difficulty a scenario presents based on aspects like environmental conditions and surrounding traffic participants.

By systematically applying these metrics, testers can prioritise scenarios that are relevant to system requirements, focus on those that involve significant risks, and ensure broad system applicability, ultimately contributing to the development of safer and more reliable CCAM systems.

3.2 Scenario description

Once the scenario contents are created, the next stage is to format it into a specific format to represent the scenario. Considering the multiple stakeholders involved in a scenario-based testing workflow, such as regulators, research engineers, test engineers, system engineers, the public, there is a need to use multiple abstraction levels for the scenario format. During the benchmarking stage in T3.1 [3], four levels of scenario abstractions have been identified: functional-level scenarios, abstract-level scenarios, logical-level scenarios, and concrete-level scenarios. Each level focus on different properties, for example functional and abstract levels may focus on human readability, whereas logical and concrete levels may focus on machine readability.

The completeness of the scenario description can be divided into technical completeness and test case completeness. Technical completeness should answer the question whether a given scenario description can be read and interpreted by a simulation software or the user. A metric should also answer whether the software might have to make assumptions. A metric for the completeness of the description of a use case should answer whether the given description is complete in covering the test case.

There may be different levels of detail in the description of a scenario. Depending on the environment in which the scenario is used, the SUT, and the test objectives, different levels of detail are required. If a scenario is used in a Hardware in the Loop (HiL) environment with physics-based sensor models, the reflection of different surfaces is required. This may not be the case in a Software in the Loop (SiL) environment with non-physics-based sensor models. Therefore, the level of detail of a given scenario affects the (simulation) environment in which the scenario can be used. This also means that the completeness of the scenario description depends on the use case. In addition, an attribute, such as the trajectory of a vehicle, can be described in different resolutions. This also affects the level of detail of a scenario.

The completeness of the description statistics indicates that all actors and the environment are described. Even with a complete description of the scenario, the unambiguity must be checked. As an example, a car can be given a non-deterministic driver model. The scenario description may be complete, but the scenario description is ambiguous. The unambiguity of a scenario is that it should be reproducible. The unambiguity is in conjunction with the scenario detail level. For interpolation to be unnecessary, the resolution of the trajectory must be sufficiently high. The objective of scenario simplification is to determine which information in a scenario is necessary for the desired result. The specific information that is not needed in the scenario description depends on the use case and the environment in which the scenario is replayed. Additionally, it is possible to assess whether the detail level may be higher than necessary.

3.3 Scenario exposure

Scenario exposure metrics are related to frequency, time spent, or distance travelled in a specific driving scenario in real-world. These metrics encompass aspects such as the probability of specific scenarios, the uncertainty associated with this probability, and the ability to predict and prepare for future scenarios.

In practice, the exposure is typically estimated. It can be useful to also consider the uncertainty of the estimation. Hence, the scenario probability uncertainty is also considered.

The UN R157 [4] requires that the ADS avoids any collisions that are reasonably foreseeable and preventable. Thus, it is required to determine all scenarios that are reasonably foreseeable. Since this is related to the scenario exposure, metrics related to this are considered to be part of the scenario exposure metrics.

3.4 (Dis)similarity of scenarios

Scenario dissimilarity metric is used to compare individual scenarios and identify how similar or different the scenarios are from each other. The metrics may be applied at any scenario abstraction stage, whether functional, logical, or concrete. The dissimilarity of a scenario may be assessed with respect to scenario parameters or aspects such as trajectories of actors or actor types in scenarios. It may also be considered with respect to the behaviour of the SUT under different scenarios.

Applications for such metrics in the SAF include:

- Identifying redundant scenarios, such that they can be skipped to reduce test effort. By prioritizing unique and diverse scenarios, computational and validation resources can be allocated more effectively. In addition, by filtering out near-duplicate scenarios, test databases can be optimized, preventing unnecessary resource expenditure on redundant test cases (e.g., the same scenario stored across different SCDBs under different identifiers).
- Clustering and categorisation of concrete scenarios to obtain logical scenarios or scenario categories. Logical scenarios / scenario categories help with understanding, storage, and querying of scenarios.
- Promoting diversity in scenarios when using scenario generation methods such as optimization. Selecting representative scenarios ensures that the full spectrum of driving behaviours and environmental conditions is captured. This topic is further explored in SUNRISE D3.4 [1].

Given these objectives, it is imperative to establish quantitative measures of scenario (dis)similarity to support efficient, structured, and scalable testing strategies.

3.5 Scenario coverage

According to the ISO 34505 standard [5], “Scenarios, test scenarios and test cases are being used in order to increase coverage of various metrics and test conditions […] The coverage data shall be aggregated in order to support the overall argumentation for approval.” In this deliverable, coverage with respect to a set of (test) scenarios will be only handled since coverage with respect to the test cases is something broader (evaluated against test objectives coverage goals) and out of scope of SDDB quality metrics.

SCDB coverage refers to the adequacy of the database in terms of the representation of all possible situations that the vehicle will encounter in its ODD. The ODD, for all purposes, details the environment and conditions in which the vehicle will operate – in urban streets, highways, weather conditions, and traffic situations. These scenarios need to be detailed to include all variables and all conditions for safety and performance: from intersection types to weather conditions. Therefore, comprehensive sampling and the subsequent testing of those scenarios are necessary so that we can be certain the vehicle will perform safely in any given situation. In that way, a coverage metric would allow us to determine if we need yet more data or scenarios or if there are enough existing data/scenarios to validate the competency of the vehicle.

Two types of metrics can be considered by:

1. Coverage of an ODD by a set of scenarios;

2. Coverage of a parameter space by a set of parameter vectors.

Note: Methods to generate concrete scenarios could be based on an optimization toward a goal, such as coverage maximization. Thus, coverage metrics can also be used to steer the process of generating concrete scenarios. This is discussed in more detail in Deliverable 3.4 of the SUNRISE project and linked with this deliverable in Chapter 5.

3.6 General scenario database metrics

General SCDB metrics help to optimize SCDB content. As the project progresses, general scenario database metrics could be used to display improvements and updates of the SCDB. By regularly reviewing metrics like data accuracy, number of scenarios, and covered kilometres, the SCDB owner/host can continuously refine and update the scenario database to keep pace with new findings, real-world data, and emerging edge cases.

4 LITERATURE REVIEW ON METRICS

This chapter explores various existing metrics from the literature that can be used to assess the quality of the scenario database content. The aim is to provide a comprehensive understanding of the fundamental concepts underlying these metrics, rather than delving into the detailed (mathematical) formulations, which can be found in the referenced literature. It follows the same structure as Chapter 3, which can also be found in Figure 4.

4.1 Testing purpose

The primary purpose of scenarios within the scenario-based testing framework is to test specific CCAM functions in particular situations. The focus of the test is typically on analysing the CCAM function, such as a lane-keeping system, an automated parking system, or enabling technologies like decision-making, perception, or communications. Scenario metrics related to the testing purpose help in assessing the suitability of each scenario for testing purposes. These types of metrics are further divided into scenario relevance, scenario criticality, and scenario complexity.

4.1.1 Scenario relevance

Scenario relevance assesses how well a given scenario aligns with a specific CCAM function or enabling technology. For example, testing a pedestrian detection system with a scenario lacking pedestrians is ineffective, as it only measures false positives. Similarly, evaluating an automated parking system on a highway scenario is misaligned. Such considerations are often assumed to be common sense and thus are not explicitly addressed in related literature. When humans do not select scenarios, tags [6] can help automated systems choose relevant scenarios. Scenario relevance depends on the CCAM function under test. A scenario may be tagged for its relevance to specific CCAM functions, such as "cut-in" for frontal obstacle detection or more explicit tags like "relevant to Euro NCAP," "relevant to Automated Parking," and "relevant to Driver Monitoring”.

For safety assurance, criticality or risk indicators contribute to “scenario relevance”.

Criticality based: Huber et al. [7] developed a multidimensional criticality analysis framework for evaluating virtual traffic situations. This framework integrates various criticality metrics to assess scenario relevance, focusing on parameters such as the likelihood of collision and the complexity of traffic interactions. By combining these metrics, the framework aims to identify scenarios that are most critical for testing automated driving functions. Baumann et al. [8] proposed an automatic generation method for critical test cases. Their approach emphasizes the generation of scenarios that push the boundaries of the automated driving system's capabilities. The criticality metrics used in this method include parameters such as time-to-collision, braking force required, and evasive manoeuvre necessity, which are crucial for assessing the robustness of highly automated driving functions. Koné et al. [9] introduced an approach to guide the search for potentially hazardous scenarios. Their method leverages criticality metrics to validate the safety of autonomous vehicles. The metrics include factors like the presence of vulnerable road users, unexpected obstacles, and adverse weather conditions. These criticality metrics help in identifying scenarios that pose significant challenges to the vehicle's safety systems.

Risk based: De Gelder et al. [10] developed a method for risk quantification in real-world driving scenarios. This method uses exposure, severity, and controllability as key metrics to evaluate scenario risk. Exposure measures the frequency of a particular scenario occurring, severity assesses the potential consequences of an incident, and controllability evaluates the ability of the vehicle or driver to mitigate the situation. Together, these metrics provide a comprehensive assessment of the risk associated with different driving scenarios. In another study, de Gelder et al. [11] presented a detailed approach for scenario risk quantification. This approach emphasizes the importance of systematically quantifying the risk to enhance the safety of automated driving systems. The risk metrics include parameters such as accident probability, injury severity in the event of a collision, and the vehicle's response time. These metrics are crucial for identifying high-risk scenarios that require rigorous testing and validation.

4.1.2 Scenario criticality

Criticality metrics are fundamental in evaluating and ensuring the safety and reliability of CCAM systems. These metrics quantify the potential risks and challenges in various traffic scenarios, offering a framework for assessing and mitigating hazards.

In ISO 34502 standard [12] (Annex A-D), scenario criticality is approached by decomposing the scenario space into three sub-classes corresponding to the three main AD functions, namely perception, planning (traffic) and control, in accordance with the physics of the ADS. It is there argued that, if risk factors and their corresponding potentially critical scenarios (scenarios including one or more risk factors) are decomposed and logically structured in accordance with the physics of the ADS, then it is possible to provide a holistic coverage of all the reasonably foreseeable safety-relevant root causes for a given Dynamic Driving Task (DDT). This motivates the specific recommendations for perception, traffic, and vehicle control related risk factors, and the corresponding scenario structures elaborated in detail in Annex B, Annex C and Annex D respectively, of the standard. A similar approach is also followed by SUNRISE partners, see Section 5.1.2.

Cai et al. [13] categorize criticality metrics into five distinct classes, each serving a specific purpose:

Trajectory-based. These metrics calculate the spatial or temporal gaps between traffic participants based on their trajectories or positions within a scene. Examples include time head way [14], gap time, distance headway, Time-to-Collision (TTC) [15], worst TTC [16], time to closest encounter [17], time exposed TTC [18], time integrated TTC [18], time to zebra [19], and post encroachment time [20]. These metrics are crucial for scenarios where the precise movement and interaction of vehicles are central to assessing risk.

Manoeuvre-based. These metrics measure the difficulty of avoiding an accident through specific manoeuvres such as braking and steering. For braking, key metrics include time to brake, deceleration to safety time, brake threat number [21], required longitudinal acceleration, and longitudinal jerk. For steering, important metrics include time to steer, steer threat number

[21], required lateral acceleration, required longitudinal acceleration, and lateral jerk. These metrics are essential for evaluating the immediate actions required to prevent collisions.

Energy-based. These metrics assess the severity of a crash. For example, Yue et al. [22] used the kinematic energy of the ego vehicle to compute the scenario risk index. These metrics are critical for understanding the potential impact and damage severity in crash scenarios.

Uncertainty-based. These metrics capture the uncertainties inherent in traffic scenarios. The level of uncertainty in a scenario generally correlates with the number of challenges faced by the System Under Test (SUT). Examples include Cafiso et al.'s [23] pedestrian risk index, which quantifies the temporal variation of estimated collision speed between a vehicle and a pedestrian, and Cunto et al.'s [24] crash potential index, which estimates the average crash possibility if the required deceleration exceeds the maximal available deceleration. Schreier et al. [25] utilized Monte-Carlo simulations to estimate behavioural uncertainties of traffic participants with the time-to-critical-collision-probability. These metrics are pivotal for scenarios with high variability and unpredictability.

Combination-based. These metrics integrate several criticality metrics, addressing different aspects of a scenario to provide a more comprehensive assessment. Huber et al. [7] presented a multidimensional criticality analysis combining various metrics to evaluate overall scenario criticality. Baumann et al. [8] proposed a combination-based metric that includes longitudinal acceleration, time headway, and TTC. These metrics offer a holistic view but require careful consideration of the weights assigned to different components.

The diverse approaches to criticality metrics underscore the complexity and multifaceted nature of traffic scenarios. Each class of metrics addresses specific aspects of risk, yet no single metric can be universally applied to all scenarios. Researchers are encouraged to design or adopt appropriate criticality metrics tailored to the specific conditions of different scenarios, as a general and objective criticality metric for all scenarios does not yet exist.

4.1.3 Scenario complexity

In the literature, scenario complexity is typically understood as the degree of challenge a scenario presents to an ADS, often influenced by the number and behaviour of dynamic agents, the richness of the environment, and the temporal evolution of events. However, there is no universally accepted definition or metric, and multiple approaches have emerged depending on the application context.

An increasingly prominent approach to scenario complexity is based on information theory and machine learning. The COMP-AV-IT framework, for example, leverages entropy-based metrics to quantify the unpredictability and variability of surrounding agent behaviour, directly linking scenario complexity to the decision-making challenge for the AV [26].

One line of research explores environmental and infrastructural complexity, incorporating factors such as road topology (e.g., intersections, merging lanes), traffic control elements (e.g., signals, signs), and environmental conditions (e.g., lighting, weather). Recent efforts such as the Dynamic Scenario Complexity Quantification (DSCQ) method propose a layered view of complexity, integrating environmental, road, and traffic dynamics to estimate “Dynamic Effect Entropy” – a metric for quantifying uncertainty over time in evolving traffic scenes [27].

A structured approach to quantifying complexity has been explored in recent research. The DSCQ method proposed by Liu et al. [27] integrates static complexity, which assesses fixed scenario properties like road type and environment, and other traffic participants. Liu et al. [27] considers three key dimensions: natural environmental conditions, road conditions, and dynamic entities, denoted as C_1, C_2, and C_3 respectively. For continuous variables, the complexity indices are derived directly from their values or via functional relationships, while for discrete variables, values are normalized between 0 and 1. For example, weather complexity indices are ranked based on the severity of different conditions (as outlined in [28]), and indices for time and types of traffic participants are adopted from [29, 30]. The overall static scene complexity, C_{scene} is computed as the product of the sum of the environmental (C_1) and road complexities (C_2) with the dynamic entity complexity (C_3), as given by

$$C_{\text{scene}} = (C_1 + C_2) \cdot C_3$$

Here, the environmental complexity and road complexity are calculated as weighted sums:

$$C_1 = \omega_{11}x_{11} + \omega_{12}x_{12} + \omega_{13}x_{13}$$

$$C_2 = \omega_{21}x_{21} + \omega_{22}x_{22}$$

In these equations, ω_{ij} represents the weight assigned to the j-th factor within the i-th dimension and the values of x_{ij} are defined in Table 2.

The dynamic entity complexity C_3 is calculated to capture how the proximity and behavior of other traffic participants (TPs) influence the overall scenario complexity.

$$C_3 = \beta \cdot V_{\text{ego}} \cdot \sum_{j=1}^{m} f(x_j,\, z_j) \cdot \ln(1 + e^{x_{j,\text{type}}}) \cdot \ln(1 + e^{x_{j,\text{occ}}})$$

$$f(x, z) = 0.5e^{-|x|} + 0.5e^{-|z|}$$

where β denotes as a scale factor for adjusting the influence of C_3 in the overall static complexity equations, m is the number of TPs, V_{ego} is the speed of the ego vehicle, and x_j and z_j represent the longitudinal and lateral distances between the j-th TP and the ego vehicle, respectively. The values of $x_{j,\text{type}}$ and $x_{j,\text{occ}}$ are defined in Table 2.

Table 2 systematically maps each key factor affecting scenario complexity considering environmental conditions, road conditions, and dynamic entity attributes to a corresponding complexity index. The complexity indices are directly derived from parameters specified in OpenSCENARIO XML files, ensuring that the quantification framework is tightly aligned with a standardized scenario description file.

Table 2: Scenario complexity indices.

	Influence factor	Notation	Value	Complexity index
Environment	Weather	x_{11}	Clear	0
			Rainy	0.25
			Light fog	0.5
			Snow	0.75
			Dense fog	1
	Illumination	x_{12}	Best computer vision weather	0
			Low dynamic range	0.33
			High dynamic range	0.66
			Overall dark	1
	Time	x_{13}	Day	0
			Night	1
Road	Obstacles	x_{21}	0	0
			1	0.33
			2	0.67
			3 or more	1
	Road condition	x_{22}	Dry	0
			Wet	0.33
			Slushy	0.67
			Full snow coverage	1
Dynamic entities	Types of traffic participants	$x_{j,\mathrm{type}}$	Pedestrian	0.7
			Ridable vehicle	0.8
			Passenger car	0.9
			Large vehicle	1
	Occlusion level	$x_{j,\mathrm{occ}}$	Up to 10%	0
			10% to 40%	0.1
			40% to 80%	0.4
			80% to 100%	1

In conclusion, the resulting complexity score for each scenario can be stored as part of the scenario's metadata in the SCDB. The scenario complexity assessment can be applied for both logical and concrete scenarios. This enables efficient categorization and visualization of scenarios by complexity, facilitating targeted testing of CCAM systems under a diverse range of driving conditions.

4.2 Scenario description

Metrics regarding the scenario description can take in different aspects. Aspects can be the completeness, level of detail, or unambiguity of the scenario description. To the best knowledge of the authors, there are currently no metrics in the classical sense like TTC or THW, which are commonly used in the scenario-based approach. However, there are a lot of requirements regarding the description which also includes the goal of a complete scenario description.

In the context of scenario description, a variety of formats exist, such as ASAM OpenSCENARIO [31]. If a format is employed, it defines the way the constituent parts of the scenario are described. Consequently, it is possible to ascertain whether a given scenario file adheres to the prescribed format. However, this does not imply that the scenario description is either complete or that the level of detail is sufficient. This can only be determined when it is established that this is the case for the specific scenario format in question. In [32], an object-oriented description of driving scenarios is proposed. This Framework supports the goal of common understanding for scenario description. Here, attributes can be set which can support the required completeness of the scenario description by aligning with the format. [33] supports the aspect, that an alignment towards a scenario format is the first key aspect for assessing the scenario completeness. Thus, it argues that scenarios can be broken down into fundamental parts. However, we still lack the assessment of the completeness regarding the scenario use case or the detail level of the scenario. Other aspects like an unambiguity of the scenario description can be tackled by using a proposed scenario description format.

As described in [34], the scenario description must assess different requirements. It states that scenarios must be transferred from the linguistically formulation into a semi-formal representation. Therefore, according to [34], an abstraction of the scenario description must be possible. Scenario shall also be human and machine readable. Although this can lead to a contradiction, as not all formats are compatible with these requirements. In particular, to be human and machine readable is not always the case.

In [35], the scenario description level is described. It states that scenarios can either be described on a functional level, on a logical level, or on a concrete level. This goes in hand with the requirements given in [34], which require that a scenario has a representation in parameter ranges or be human readable. However, both sources only file for a need but do not provide a metric to verifying the fulfilment of the requirement.

In [34], the scenario description is required to be complete and have a sufficient detail level. This is needed, so that the test procedure can be executed. However, no way of checking the fulfilment is given.

A different aspect than the completeness can be the part of simplification. In [36], an approach is discussed how a scenario can be simplified by removing aspects of the scenario that are irrelevant to the goal of the scenario. However, this approach needs a simulation of the scenario. It is not capable of determining a simplification prior to the execution which would be helpful to reduce the number of overall simulations that are needed.

4.3 Scenario exposure

The exposure of a scenario refers to the likeliness that such a scenario is encountered. A common way to express this is using the scenario probability or scenario probability density. In Section 4.3.1, this will be discussed in more detail. Typically, the scenario probability (density) is estimated. In those cases, it is often desired to know how accurate this estimation is. This is further discussed in Section 4.3.2. The concept of “foreseeability” is also related to the scenario exposure, so this is discussed in Section 4.3.3.

4.3.1 Scenario probability

Several methodologies have been developed to estimate scenario exposure using Naturalistic Driving Data (NDD) and Field Operational Test (FOT) data.

For the scenario probability, typically a distinction is made between the probability of a “type of scenario”, such as an “abstract scenario” [37] and a “scenario category” [32], and the probability density of the parameter values within such a “type of scenario”.

Regarding the exposure of different types of scenarios, De Gelder et al. [10, 38] have expressed the exposure as the expected number of encounters per unit of time for scenarios within a specific scenario category. Their work relies on real-world driving data, such as the extensive dataset from Paardekooper et al. [39], which includes 6000 km of public-road driving. This data-driven approach provides a robust basis for estimating exposure frequencies and identifying critical scenarios.

Hakkert et al. [40] defined exposure within the context of road safety, focusing on various measures such as the number of kilometres travelled, time spent in traffic, and traffic volumes at intersections. These measures offer a practical way to quantify exposure but often require extensive and high-quality data, which can be challenging and expensive to collect.

Regarding the exposure at parameter value level, much literature is available. Already a few decades ago, the probability density of the scenario parameters have been estimated using Gaussian distributions [41]. With the increase of data, more sophisticated (but data-hungry) methods could be employed when estimating the probability densities, such as kernel density estimation [10, 38]. These methods, however, generally scale very badly with increasing number of parameters, which is why it is not uncommon to assume that the parameters are independent, see, e.g., [42].

4.3.2 Scenario probability uncertainty

Despite the importance of the uncertainty of estimated probabilities, this has not been discussed often in the literature in relation to scenario exposure. However, outside the field of automated driving, extensive literature is available on this topic. Here, two different approaches can be distinguished:

1. With the first approach, a parametric distribution is used to estimate the probability density, such as a normal or Gaussian distribution, or a gamma distribution. In those cases, the distribution parameters (not to be confused with the scenario parameters for which the density is estimated) are typically fitted to some data. When using a Bayesian approach to fit those distribution parameters, the posterior uncertainty of the distribution parameters can be used to estimate the uncertainty of the density [43].

2. With the second approach, a non-parametric distribution is used to estimate the probability density, such as kernel density estimation. In those cases, the uncertainty is either based on a theoretical model or bootstrapping is used [44].

Note that in [45, 38], bootstrapping is used to estimate the probability uncertainty of the scenario parameters' probability density.

4.3.3 Scenario foreseeability

Regulations for the type-approval of ADSs require that the activated system does not cause any collisions that are reasonably foreseeable [46]. To determine what scenarios are "reasonably foreseeable", one can look at the probability density of the parameters and consider the parameter values at the "edges" to be not reasonably foreseeable. Nakamura et al. [47] exploited this idea to determine the "reasonably foreseeable" range of parameter values. Their approach assumes scenario parameters are independently distributed according to the Beta distribution. From this, a parameter range capturing 99 % of the distribution is calculated and all these parameter values are reasonably foreseeable. Nakamura et al. [47] applied this analysis methodology to cut-in scenarios. Extending this work, Muslim et al. [48] performed a similar analysis for cut-out scenarios.

De Gelder and Op den Camp [49] expanded on this approach, proposing two alternative methods to estimate "reasonably foreseeable" parameter values. Their first method employs non-parametric kernel density estimator, allowing the probability density function to adapt to the data without assuming parameter independence. The second approach utilizes extreme value theory, applying the generalized Pareto distribution to model extreme parameter values. These methods are demonstrated through case studies involving scenarios from [47] and an additional scenario where the ego vehicle approaches a slower vehicle.

4.4 (Dis)similarity of scenarios

Existing dissimilarity metrics for scenarios can be broadly classified as follows:

- Dissimilarity using scenario parameters (Section 4.4.1)
- Dissimilarity using scenario trajectories (Section 4.4.2)
- Dissimilarity using scenario features (Section 4.4.3)
- Dissimilarity using scenario manoeuvres (Section 4.4.4)

4.4.1 Dissimilarity using scenario parameters

These metrics are applied particularly to multiple concrete scenarios of the same logical scenario. The logical scenario is parameterized, and parameter values of different concrete scenarios can be used for comparison. This comparison is fast and efficient, as the scenarios need not be simulated, nor complete trajectory information is required. Some works which proposed and/or applied this method in literature are as follows:

- Zhu et al. [50] – Used Euclidean distance in parameter space to evaluate scenarios, and whether they are sufficiently different to include in an archive.

- Zhong et al. [51] - Found unique traffic violations based on percentage difference between scenarios in parameter space. If dissimilarity was above a certain threshold, they were said to be unique.

4.4.2 Dissimilarity using scenario trajectories:

These metrics compute dissimilarity considering complete trajectories of all actors in each scenario. Such a method provides a more exhaustive comparison of the scenario but requires complete trajectories of the actors. Various metrics exist in literature:

- Ries et al. [52] - Used Dynamic Time Warping to estimate similarity between trajectories of traffic objects.

- Nguyen et al. [53] – Used Levenshtein edit distance to measure similarity between scenarios and filtered out scenarios using a threshold distance.

- Lin et al. [54] – Created matrix profiles, which consist of dissimilarity between the sub-sequences of one trajectory with the nearest neighbour sub-sequences from other trajectory. The dissimilarity is based on the number of elements lower than a certain threshold.

4.4.3 Dissimilarity using scenario features:

These metrics perform comparison on scenario features, which are defined and extracted from use case specifications, e.g., features of road infrastructure which is part of the ODD. Such metrics allow emphasis on relevant aspects (features) of the scenario and additionally compare beyond the dynamic behaviours of actors by also considering features related to the environment and road infrastructure. Some existing literature is highlighted below:

- Kerber et al. [55] – Defined a scene distance based on occupancy of an 8-cell grid around the ego vehicle, and extended it to a scenario distance by summing over the entire scenario.

- Kruber et al. [56] – Defined features of the road infrastructure and the object trajectories and use them for Unsupervised random forest clustering and learning a similarity measure.

- Wheeler et al. [57] – Clustered scenarios using features extracted from scenarios at criticality transition. Different features such as relative speeds, accelerations, relative speed change, acceleration change were investigated.

- Zohdinasab et al. [58] - Created feature maps using structural features such as such as road smoothness, complexity and orientation, and behavioural features like steering angle standard deviation and mean lateral position.

- Nyugen et al. [53] - Used feature maps similar to [58] using features such as direction coverage and number of turns, to select scenarios.

The diversity of a set of scenarios builds upon the dissimilarity measure; by providing a measure to quantify overall dissimilarity across a set of scenarios. Some existing works on this topic are as follows:

- Tian et al. [59] - Measure the average dissimilarity of a new scenario from an existing set as an indicator of diversity increase due to the scenario or novelty of the scenario.

- Zohdinasab et al. [58] - Measure diversity using a sparseness measure, defined as the average maximum Manhattan distance between cells in feature maps, which were created by mapping scenarios to certain cells based on feature values.

4.4.4 Dissimilarity using scenario manoeuvres

At the logical scenario level, it is possible to extract manoeuvre information from the scenario, then manoeuvre sequence can be used as metric related to scenario story. This metric, introduced by Braun et al. [60], employs manoeuvre-based similarity metrics to compare traffic scenarios based on driving manoeuvres rather than raw trajectories. The core similarity measure is sequence alignment, specifically using the Needleman-Wunsch algorithm, which aligns two manoeuvre sequences by assigning scores for matches, substitutions, and insertions/deletions. A higher alignment score indicates greater similarity in the manoeuvre sequences. Additionally, this work introduces a graph-based scenario similarity representation, where scenarios are mapped as nodes and similarity scores determine the edge weights. This enables clustering of functionally similar scenarios, helping detect redundant test cases and ensuring comprehensive scenario selection. The authors also incorporate event-type weighting, where different driving manoeuvres (e.g., lane change vs. braking) have different impacts on similarity scoring. The final similarity score is derived from a weighted sum of manoeuvre alignments, accounting for both event sequence ordering and manoeuvre type importance.

4.5 Scenario coverage

The term “coverage” is commonly understood as the extent to which something addresses or deals with something else. In software engineering, coverage is defined as “a measure of verification and completeness” [61]. As described by Pizali [61], there is no single best way to define coverage. Coverage metrics can be customized to assess verification progress from different perspectives, such as functional requirements (functional coverage), the executed portions of code (code coverage), or the evaluated assertions (assertion coverage). For instance, code coverage can be measured by the lines of code executed, the branches tested, or the paths traversed during verification.

In [62], the importance of coverage metrics for testing autonomous vehicles is emphasized. The authors argue that inadequate coverage of the situations an autonomous vehicle may encounter is tantamount to inadequate testing. To address this, Alexander et al. [62] proposed a “situation coverage metric”. They suggest that this metric should be tractable, which has two key implications:

1. It should be expressible as a percentage. For example, metrics based solely on the number of kilometres driven or the number of (simulated) scenarios are insufficient, as

both can be infinite. Similarly, the number of failures found is not suitable, as the total number of potential failures is unknown.

2. Achieving 100 % coverage should be possible under realistic, practical conditions. Therefore, measures that only reach 100 % exponentially are not applicable as these types of measures cannot reach 100 % with any practical means.

In the testing of ADSs, “coverage” is often used to evaluate the adequacy of a testing effort and to determine when testing can be stopped [63]. Riedmaier et al. [64] defined ”scenario coverage” as the extent to which test scenarios cover the entire scenario space, though they did not provide quantitative measures. In [65], this concept is expanded by proposing several metrics to measure the coverage of test scenarios relative to the ODD of an ADS. Given that the number of possible concrete scenarios is virtually infinite [66], and considering Alexander et al.'s reasoning [62], concrete scenarios alone are not sufficient for a reliable coverage metric. Instead, scenario types or types of scenes – defined as specific moments within a scenario – can be considered. Hauer et al. [67] proposed a metric to estimate the number of unaddressed scenario types during testing, without explicitly mentioning “coverage”. In [68], a coverage metric based on scenes is defined, although no practical results are provided.

In addition to measuring the coverage of testing, measures of the coverage of real-world driving data could be utilized. One reason to do so is because one may derive tests from the real-world data [69]. Another reason is because the driven kilometres may directly be used as a validation of the absence of unknown hazards. Compared with the amount of literature on coverage regarding the testing effort of ADSs, there is little literature available regarding the coverage of the real-world data. In [70], a criterion is proposed for the collection of naturalistic driving data. In [45], the asymptotic mean integrated squared error of an estimated probability density function is used as a metric to quantify the coverage of the collected data. A disadvantage of both these works is that a 100 % coverage can only be reached exponentially, i.e., not by any practical means. In [71], a metric is proposed based on the number of distinct sequences of manoeuvres of an observed object. A disadvantage of this metric is that the total number of distinct sequences is unknown, so a percentage cannot be calculated. Glasmacher et al. [72] proposed a coverage metric based on scenario parameter values. This approach requires selecting a parameterization and limiting the number of parameters, as achieving 100 % coverage could be impractical otherwise.

De Gelder et al. [73] discusses four different coverage metrics for evaluating whether collected scenarios adequately represent the ODD of an ADS:

- Tag-based coverage: This metric evaluates whether collected scenarios cover all relevant aspects of an ODD by checking the presence of specific tags associated with scenario characteristics (e.g., environmental conditions, vehicle positions). It provides a quantitative measure to ensure diversity and completeness in scenario generation.
- Time-based coverage: Time-based coverage checks whether all timestamps in the driving data are represented by one or more scenarios, ensuring that all moments within the data are adequately tested. It focuses on identifying gaps where certain periods lack scenario representation.

- Actor-based coverage: This metric assesses whether all relevant actors (e.g., vehicles, pedestrians) are included in at least one scenario, based on their proximity and interaction with the ego vehicle. It ensures that all important entities influencing the driving environment are considered in the test scenarios.
- Actor-over-time-based coverage: Actor-over-time-based coverage extends actor-based coverage by ensuring that relevant actors are included in scenarios throughout the period they are considered important. It helps identify cases where actors may be briefly included but not consistently represented over time.

Laurent et al. [74] introduces a novel approach for testing ADS by focusing on parameter coverage. The paper proposes parameter coverage as a criterion for ensuring that the parameters influencing the decision-making process of an ADS are adequately tested. A parameter is considered covered if changing its value leads to different simulation outcomes, which signifies that it affects the driving decisions in a scenario. The method involves running multiple simulations with different parameter values to assess whether the altered parameter leads to a statistically significant difference in outcomes, considering metrics like path deviation, safety (minimum distance to other objects), and comfort (maximum acceleration).

Tahir & Alexander [75] provides an overview of three different coverage-based testing techniques used for Verification and Validation (V&V) and safety assurance of CCAM systems:

- Scenario Coverage: Scenario coverage involves testing a set of predefined scenarios that represent different possible situations the vehicle might encounter. It aims to cover all combinations of temporal developments between scenes, such as lane changes or following another vehicle.
- Situation Coverage: Situation coverage considers the different internal and external situations that a vehicle can face, ensuring testing under both expected and unexpected conditions. This approach aims to cover a broad range of potential situations to verify the robustness of SAVs in diverse environments.
- Requirements Coverage: Requirements coverage tests whether the system under test meets all identified safety and functional requirements. It involves assessing the acceptability of scenarios based on predefined criteria.

4.6 General scenario database metrics

General SCDB metrics describe general characteristics, properties, or other relevant elements of the SCDB, typically without considering the actual content of the scenarios that are part of the SCDB. This could include:

- Data Completeness: Evaluates whether all necessary data fields for each scenario are populated. It checks for missing values and ensures that all relevant information is available.
- Data Accuracy: Assesses the correctness of the data entered into the SCDB. This involves cross-referencing with known benchmarks or ground truth data to ensure the scenarios reflect real-world conditions accurately.

- Data Consistency: Ensures that the data is consistent across all scenarios. This includes checking for uniformity in units of measurement, formats, and terminologies used.
- Data Freshness: Measures how up to date the scenarios are. This is crucial for reflecting current conditions and ensuring the SCDB remains relevant.
- Number of Scenarios: The total count of distinct scenarios available in the database. This provides an overview of the database's comprehensiveness.
- Covered kilometres: The cumulative distance covered by all scenarios. This metric helps in understanding the breadth and scale of the scenarios included.
- Scenario Distribution: Breakdown of scenarios by various categories such as geographic regions, road types, weather conditions, time of day, and traffic density.
- Scenario Complexity: Assesses the level of difficulty presented by the scenarios. This can include factors like the number of vehicles, presence of pedestrians, and road complexity.
- Detection Accuracy: The percentage of scenarios correctly identified and classified by the detection system. High accuracy indicates reliable scenario recognition.
- False Positives/Negatives: The rate at which the system incorrectly identifies scenarios (false positives) or fails to detect them (false negatives). Lower rates are better.

The following references are relevant for the general SCDB metrics:

- ISO 21448 Safety of the Intended Functionality (SOTIF) [76]: This ISO standard outlines safety requirements and testing protocols for autonomous driving systems, including how scenarios are designed, tested, and validated using metrics like detection accuracy and data quality.
- SAE J3016 Standard – Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles [77]: Provides the framework for scenario classification, detection performance, and the use of metrics in developing and validating driving automation systems.
- UNECE Regulation No. 157 on Automated Lane Keeping Systems (ALKSs) [46]: This regulation focuses on the safety, performance, and validation requirements for automated driving systems and provides guidance on metrics related to scenario coverage and system detection performance.

5 METRICS DEVELOPED WITHIN SUNRISE

This chapter presents metrics that have been developed in the SUNRISE project. The structure of this chapter follows the same structure as Chapters 3 and 4, meaning that metrics related to testing purpose are presented first. Next are metrics related to scenario description and scenario exposure in Sections 5.2 and 5.3, respectively. Section 5.4 presents metrics related to the (dis)similarity of a set of scenarios. Lastly, coverage metrics are presented in Section 5.5.

5.1 Testing purpose

Two different metrics related to the testing purpose have been developed. The first metric focuses on the relevance of the scenario by considering the risk to which the system is exposed in a scenario. The second metric is related to this and aims at quantifying on the criticality of a scenario. For measuring the complexity of a scenario, the method of Liu et al. [27] is applied in Chapter 6. Since this metric has already been discussed in Section 4.1.3, this chapter does not contain a contribution for the scenario complexity metric.

5.1.1 Scenario relevance

SUNRISE has developed a systematic scenario quality metric to prioritize high-risk scenarios from a SCDB for testing. This metric uses a hypothesis test that compares a subset of testing scenarios against the full set of ODD scenarios, using risk as the test variable. Risk is calculated based on three critical factors: severity, controllability, and exposure. By analysing the risk distribution across these scenario sets, the method ensures that the selected scenarios reflect a higher overall risk, increasing the relevance and effectiveness of testing.

The main objective is to provide a structured approach for selecting scenarios that accurately represent the most demanding real-world challenges faced by CCAM systems. This ensures that proving ground tests focus on the most critical situations, enhancing system safety and robustness.

Figure 6 presents an overview of the procedure for evaluating the relevance of scenarios, broken down into four steps (highlighted by coloured boxes).

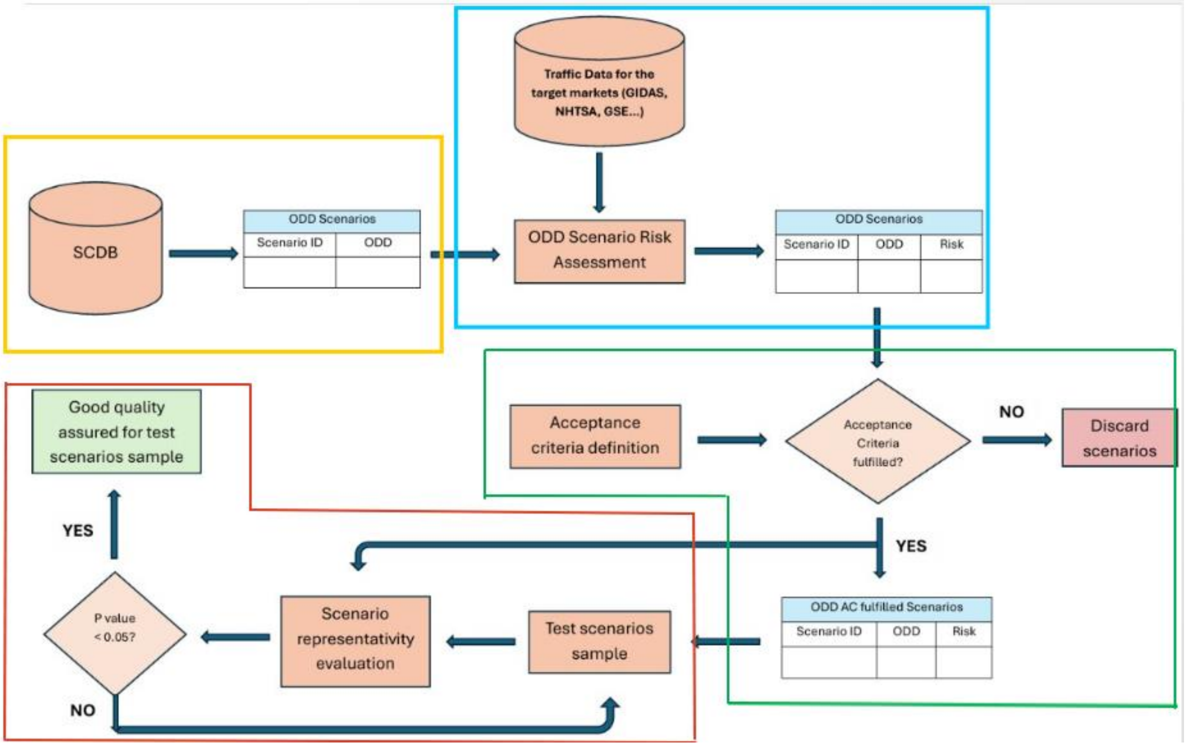

Traffic Data for the target markets (GIDAS, NHTSA, GSE...)
SCDB
ODD Scenarios
Scenario ID
ODD
ODD Scenario Risk Assessment
ODD Scenarios
Scenario ID
ODD
Risk
Good quality assured for test scenarios sample
Acceptance criteria definition
Acceptance Criteria fulfilled?
NO
Discard scenarios
YES
YES
ODD AC fulfilled Scenarios
Scenario ID
ODD
Risk
P value < 0.05?
Scenario representativity evaluation
Test scenarios sample
NO

Figure 6: Schema of the overall procedure for evaluating the relevance of scenarios.

Step 1 - Scenario Selection

A subset of scenarios is selected from existing SCDBs (such as Safety Pool, StreetWise, PEGASUS, ADScene) for a specific ODD. This selection process can be facilitated by SUNRISE DF, which provides a structured methodology for querying and retrieving relevant scenarios. Using the SUNRISE DF Dashboard, users can specify the SCDBs to be queried, apply advanced filtering criteria based on scenario metadata (e.g., road geometry, traffic conditions, and environmental factors), and retrieve scenario packages.

Step 2 - Risk Estimation

Each selected scenario's risk is estimated using a methodology adapted from [78] which evaluates accident risk using a predefined catalogue of influencing risk factors called Criticality Phenomena (CPs). These CPs represent specific factors linked to increased criticality in traffic scenarios, derived from a detailed analysis of the German In-Depth Accident Study (GIDAS) database.

The approach begins by decomposing the Operational Domain (OD) (e.g., urban scenarios involving passenger cars) into relevant CPs, guided by the criticality analysis framework proposed by Neurohr et al. [37]. CPs are categorized into two groups according to Damm and Galbas [79]:

- CPs relevant to human traffic – These are influencing factors independent from perception systems, such as occluded pedestrians at urban intersections or reduced road friction.

- CPs specific to ADSs – These are phenomena influenced by the reliance on sensor technology for perception. These are not covered in the current analysis due to the absence of ADS-specific accident data in GIDAS.

Following ISO 26262 [80] and ISO 21448 [76] guidelines, accident risk is factorized into three elements: exposure, controllability, and severity. Bayes' theorem is used to formulate the risk estimation, decomposing it into three key probabilities:

- $\mathbb{P}(\mathrm{CP}|\mathrm{Accident}, \mathrm{OD})$: Likelihood of a specific CP occurring in an accident, based on GIDAS data.

- $\mathbb{P}(\mathrm{Accident}|\mathrm{OD})$: Overall accident probability within a given OD, obtained from accident statistics.

- $\mathbb{P}(\mathrm{Severity}|\mathrm{Accident}, \mathrm{CP}, \mathrm{OD})$: Severity distribution of accidents, derived from GIDAS severity data.

Using these three probabilities, the risk can be calculated as follows:

$$\begin{aligned}&\mathrm{Risk}(\mathrm{CP}, \mathrm{Accident}, \mathrm{Severity}|\mathrm{OD})\\&= \underbrace{\mathbb{P}(\mathrm{CP}|\mathrm{OD})}_{\mathrm{Exposure}} \underbrace{\mathbb{P}(\mathrm{Accident}|\mathrm{CP}, \mathrm{OD})}_{\mathrm{Controllability}} \underbrace{\mathbb{P}(\mathrm{Severity}|\mathrm{Accident}, \mathrm{CP}, \mathrm{OD})}_{\mathrm{Severity}}\\&= \mathbb{P}(\mathrm{CP}|\mathrm{Accident}, \mathrm{OD})\mathbb{P}(\mathrm{Accident}|\mathrm{OD})\mathbb{P}(\mathrm{Severity}|\mathrm{Accident}, \mathrm{CP}, \mathrm{OD})\end{aligned}$$

This probabilistic decomposition enables CP-related risk estimation using accident databases and national statistics, ensuring data-driven and system-independent risk quantification. For scenarios with multiple CPs, risk values are aggregated, allowing for a comprehensive evaluation of complex scenarios:

$$\mathrm{Risk}(\mathrm{scenario}) = 1 - \prod_{i=1}^{n} (1 - \mathrm{Risk}(\mathrm{CP,Accident,Severity} \geq 1|\mathrm{OD})_i) \tag{1}$$

where:

- Risk(CP,Accident,Severity $\geq 1|$OD$)_i$ is the i-th CP risk in the scenario;
- $\prod_i^n (1 - \mathrm{Risk}(\mathrm{CP,Accident,Severity} \geq 1|\mathrm{OD})_i)$ is the probability that none of the CPs occur;
- $\mathrm{Risk}(\mathrm{scenario})$ is the risk that at least one CP occurs, which represents the risk of the scenario.

By using this structured approach, the scenario quality metric ensures a rigorous and representative risk assessment, leading to more relevant and effective scenario selection for testing. The framework is also adaptable for future criticality analyses involving ADSs by highlighting requirements for accident data collection, particularly for ADSs at SAE Levels 4. Although the current study focuses on human traffic-relevant CPs, the framework is adaptable to future ADS data sources.

Step 3 - Acceptance Criteria Definition

A threshold is defined to filter out low-risk scenarios, prioritizing the most critical scenarios for testing. For example, the threshold can be set as a risk higher than the mean plus one standard deviation of the ODD scenario risks. Scenarios that meet this criterion form the pool from which the final test sample is randomly selected.

Step 4 - Relevance Evaluation

The relevance of the test scenario sample is evaluated against the reference distribution (from the scenarios that meet the acceptance criteria) using statistical hypothesis testing. The null hypothesis states that the mean risk of the test sample is lower than that of the reference distribution.

The relevance metric is defined by the p-value obtained from this hypothesis test. Specifically, if the p-value is below 5%, the null hypothesis is rejected, and the alternative hypothesis is accepted. This confirms that the selected scenarios are representative of the reference distribution, accurately reflecting the most critical scenarios.

5.1.2 Scenario criticality

There exist two high-level approaches in assessing criticality of a scenario. The first approach (scenario based) is based on attributes and characteristics of the scenario itself as well as the agents involved and can be assessed before the scenario is executed (i.e., final trajectories are not known). The second approach (test case outcome based) is based on the assessment of the scenario outcome as a test scenario – including a given SUT – and is mainly considering the final trajectories of all agents involved upon execution. This means that a scenario characterized as critical upon execution is tightly connected to a specific SUT. The metrics discussed in Section 4.1.2, i.e., trajectory, manoeuvre and energy-based metrics (e.g., TTC, post encroachment time) are relevant here.

These metrics are used in the context of SUNRISE critical scenario generation from methods designed to efficiently explore the scenario space for critical scenarios based on a given set of criticality metrics (see SUNRISE D3.4 [1]). The existing metrics as reviewed in Section 4.1.2 are mostly looking at the criticality of the full AD stack and utilize the outcome trajectories or manoeuvres for criticality assessment. However, a similar approach to the ISO 34502 standard [12] could be employed by distinguishing the metrics into sub-classes according to the AD function of interest, that is perception, planning (traffic), and control or full AD stack. In this direction, we propose the use of:

1. Perception outcome metrics (e.g., false positives, accuracy, f-score): in the context of automatic scenario generation, these will help us generate scenarios critical for the Perception subsystem (as recommended by the ISO 34502):
    a. Minimum Precision/Recall of vehicle detection
    b. Minimum Precision/Recall of pedestrian detection
    c. Minimum Road/Lane recognition: mean intersection of union
2. Collective perception outcome metrics (e.g., V2X messages loss rate): in the context of automatic scenario generation, these will help us generate scenarios critical for the Collective Perception subsystem:

    a. Maximum V2X messages loss rate
    b. Minimum Precision/Recall of vehicle detection in the global Collective Perception Message (CPM)
    c. Minimum Precision/Recall of pedestrian detection in the global CPM

3. Planner/Control outcome metrics (e.g., TTC, trajectory prediction accuracy, RMSE): in the context of automatic scenario generation, these will help us generate scenarios critical for the Control subsystem (as recommended by the ISO 34502):
    a. Metrics that can also be computed online
        i. Minimum TTC
        ii. Minimum time to intersection violation
        iii. Collision incident
        iv. Collision type/severity (categorical)
    b. Metrics that are computed offline
        i. Minimum post encroachment time

5.2 Scenario description

First, some guidelines for the scenario description are presented. Even though these guidelines do not come with metrics, they are important enough to be presented here. Conformance to these guidelines requires a qualitative assessment. Next, conformance to standardized language using a taxonomy is discussed. Finally, a metric is presented to quantify the completeness of a scenario description.

5.2.1 Scenario description guidelines

For a given scenario description, the quality and completeness should be assessed. For this, a three-step methodology has been developed. Based on three guidelines, the completeness can be determined. Completeness requires requirements to assess the completeness, thus a use case for the scenario is needed. Therefore, completeness cannot be determined in general, therefore the guidelines will help to determine it for the use case.

The methodology applies for concrete scenarios [35]. A concrete scenario should be given in a known scenario format (e.g. OpenSCENARIO XML). This enables interoperability. In terms of completeness, a scenario format gives a baseline on how elements are defined. For the completeness on the technical level, the following guidelines is given:

Guideline 1: A given concrete scenario should align with a given scenario format.

This allows to implement rules that can be checked against. For example, an OpenSCENARIO XML file can be checked for conformity with the XML format. Generally, there can be multiple rules for each scenario format to assess the conformity. The conformity is the first step to assess the completeness, as conformity ensures that the scenario adheres to the required structure and syntax before evaluating its content for consistency and sufficiency.

In the second step, the content of the scenario itself should be checked for completeness. The check can be done independently of the use case and only the scenario with itself is checked. Thus, it is required that the content of the scenario is plausible within itself. This leads to the following guideline:

Guideline 2: The content of the scenario should be plausible within itself.

In the third step, requirements of the use case will be checked for the completeness of the scenario. As the scenario is given in a scenario format, this gives a boundary of what can be defined. It is assumed that the scenario format allows to describe the intended scenario. Thus, only a gap between the concrete scenario and the use case must be determined. Therefore, requirements on the description can be derived from the use case and checked for in the scenario.

Guideline 3: The scenario description must contain all required information from the use case.

5.2.2 Consistent use of taxonomy

As the number of scenarios within a SCDB continues to grow, along with increased collaboration among industrial and research stakeholders, there is a rising demand for a systematic structure that enables efficient storage and retrieval. The SCDB must be able to accommodate scenarios at varying levels of abstraction and ensure that they are stored in a structured manner to support scalability, consistency, and efficient querying.

With ODD and behaviour specifications serving as inputs to the SAF – whose primary goal is to ensure safe operation within defined boundaries, it is essential to align the terminologies used in scenario descriptions with that of the ODD and behaviour definitions. This alignment establishes a clear traceability between the scenario descriptions and the system's intended operational limits. Maintaining consistency in the terminologies used within the scenario description is crucial to the SAF, as it ensures a cohesive evaluation framework for assessing the safety claim. When scenario descriptions adhere to a common vocabulary and standardized structure, the SAF can reliably assess whether the input ODD and behaviour specifications are sufficiently represented within a set of scenarios.

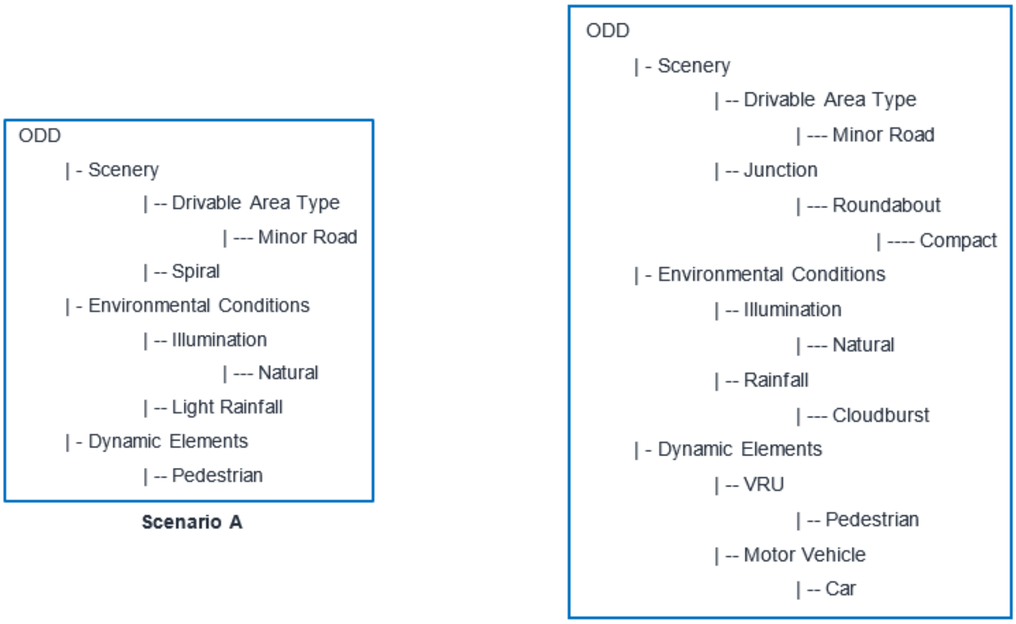
Additionally, the usage of common terminologies across the database facilitates efficient retrieval of the scenarios. To achieve this, a tagging system can be implemented, which relies on shared keywords across different levels of scenario abstraction. This system ensures that scenarios can be systematically categorized and retrieved based on specific attributes. However, for tagging to be effective, scenario descriptions within the database must follow a standardized vocabulary at one level of abstraction in the least. This allows relevant tags to be consistently assigned, ensuring interoperability across different stakeholders.

The common vocabulary used across the SCDB may be based on ISO 34503, which provides a taxonomy for ODD attributes. This taxonomy has been further incorporated into BSI Flex 1889, which expands on ODD attributes and includes behavioural specifications within formalized natural language descriptions of scenarios. By adhering to a well-established taxonomy, the SCDB can maintain consistency across scenarios, even as different stakeholders work with varying levels of abstraction.

One key advantage of using a standardized taxonomy is that it allows for a structured extension of attributes while maintaining uniformity. When stakeholders introduce new scenario attributes, they can extend existing categories systematically. For instance, consider a scenario that involves a pedestrian crossing. If a user wants to refine this attribute by specifying different types of pedestrian crossings, the parent attribute ("pedestrian crossing") can be expanded to include child attributes, such as pelican crossing, puffin crossing, toucan crossing, etc.

By explicitly defining the parent-child relationship, new attributes are seamlessly integrated into the SCDB while preserving a structured format. This structured extension benefits other users who may not have initially considered these attributes but can now relate new additions to existing taxonomy categories.

Consider Scenario A in Figure 7, which does not comply with the hierarchy of the database it belongs to. In this case, the attribute "Spiral" is incorrectly categorized as a direct child of "Scenery", making it difficult to assess its relationship with other attributes, especially in a large-scale SCDB containing numerous scenarios. Here, "Spiral" refers to a specific type of roundabout that is commonly be found only in a certain geographical region. However, this term may not be immediately recognizable to other SCDB users, particularly those unfamiliar with the road infrastructure of that specific region.



Scenario A

Scenario B

Figure 7: Examples of an ODD class hierarchical branch from a scenario description that are non-compliant (Scenario A) and compliant (Scenario B) with a standard taxonomy.

An ideal scenario description (such as Scenario B in Figure 7) follows the standardized taxonomy, ensuring that attributes are correctly associated with their relevant parent categories. If Scenario A were structured accordingly, "Spiral" would be explicitly linked to "Roundabouts", making its function and relevance immediately clear to all users. This structured approach not only enhances clarity but also enables deeper analytical assessments, such as evaluating scenario coverage to identify gaps or imbalances in scenarios within the database.

Ultimately, the consistent use of both a standardized language and taxonomy across scenario descriptions while not a metric, is foundational to enabling meaningful quality assessments of the SCDB. A shared vocabulary ensures that scenario attributes can be accurately tagged and retrieved, while the taxonomy provides the structural framework to extend branches. This consistency supports downstream analyses such as scenario coverage, where the presence or absence of specific attributes can be systematically identified and compared against a given ODD definition which would follow the standardised vocabulary.

5.2.3 Scenario description completeness

The scenario description completeness metric evaluates the overall completeness of a scenario description by assessing whether it contains all required and optional suggested elements. The completeness is categorized into two levels: core completeness and descriptive completeness.

Core completeness considers whether the individual scenario has the required elements that the scenario is meaningful and can be executed in a test execution environment. These elements consist of:

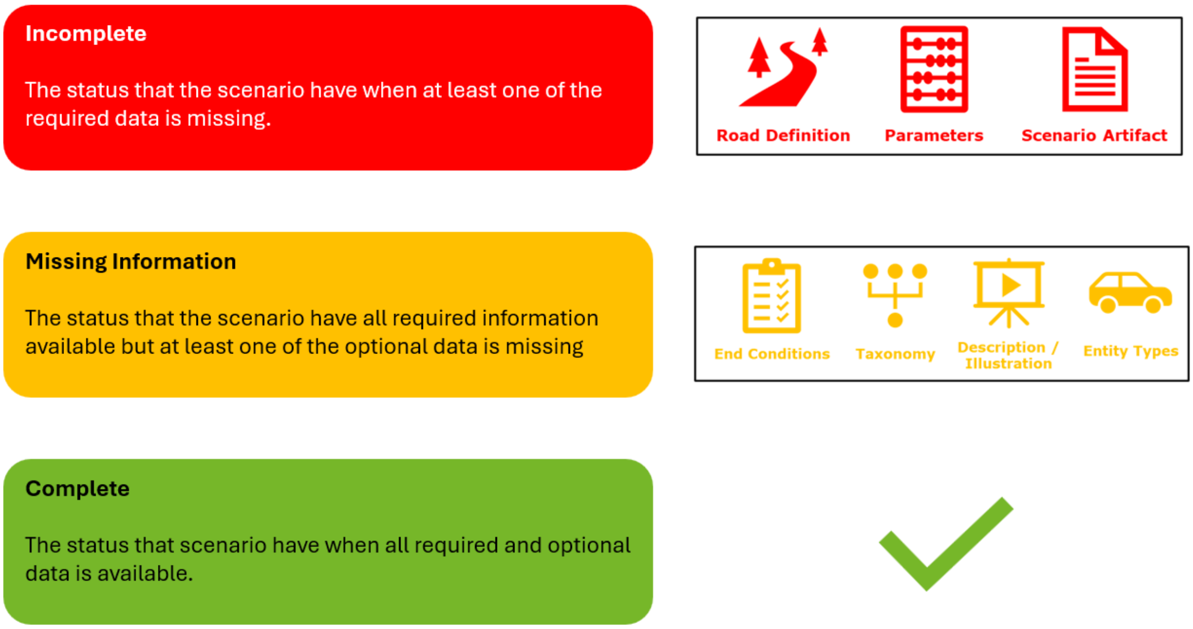
- *Scenario Artifact*: The scenario definition shall be included at least in one file that has been defined on a standard. E.g., a validated OpenSCENARIO XML file (.xosc).
- *Road Definition*: The defined road in a scenario file must be present in a SCDB. If the road file does not exist in the SCDB, the scenario could not be executed.
- *Scenario Parameters*: Parameters shall be defined in the scenario description file so that the parameter space can be configured in the SAF.

Descriptive completeness assesses whether a scenario includes optional metadata, such as end conditions, taxonomy classification, descriptions, illustrative media, and defined entity types, to enhance usability and clarity.

- *End Conditions*: Clearly defined success/failure conditions for terminating the test.
- *Taxonomy*: Proper categorization within the SCDB taxonomy.
- *Description & Illustrative*: Informative descriptions and media (e.g., images, videos).
- *Entity (Actor) Types*: Clear definition of scenario participants (vehicles, pedestrians, cyclists, etc.).

Finally, the overall completeness state of a SCDB is assessed as follows (see Figure 8):

- *Incomplete*: If any scenario lacks at least one core completeness criteria.
- *Missing Information*: If any scenario lacks descriptive completeness but meets core completeness.
- *Complete*: If all scenarios meet both core and descriptive completeness requirements.



Incomplete

The status that the scenario have when at least one of the required data is missing.

Missing Information

The status that the scenario have all required information available but at least one of the optional data is missing

Complete

The status that scenario have when all required and optional data is available.

Figure 8: Data completeness table.

5.3 Scenario exposure

One of the goals of safety validation is to prospectively evaluate the risk of an ADS dealing with real-world traffic. Scenario-based assessment is a widely used approach, where test cases are derived from real-world driving data. To allow for a quantitative analysis of the system performance, accurately estimating the scenario exposure is essential for reliable safety assessment. A Probability Density Function (PDF) quantifies the exposure of scenarios at the parameter level. However, assumptions about the PDF, such as parameter independence, can introduce errors, while avoiding assumptions often leads to oversimplified models with limited parameters to mitigate the curse of dimensionality.

One method that does not rely on assumptions on the shape of the PDF is Kernel Density Estimation (KDE), which is a non-parametric method for estimating the PDF of a dataset. Given a set of observations $\{x_i\}_{i=1}^{N}$, KDE estimates the density at a point x by averaging kernel functions K centered around each data point:

$$\hat{p}(x) = \frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right),$$

where h is the bandwidth parameter that controls the level of smoothing. The choice of h is crucial – too small a bandwidth leads to overfitting, while too large a bandwidth oversmooths the density estimate. A common method for determining h is leave-one-out cross-validation because this minimizes the difference between the real PDF and the estimated PDF according to the Kullback-Leibler divergence [81, 82]. Commonly used kernel functions include Gaussian, Epanechnikov, and uniform kernels [83]. KDE is widely used due to its simplicity and ability to approximate arbitrary distributions without assuming an underlying parametric model.

In the field of assessment for ADSs, KDE has been used to estimate the exposure of concrete traffic scenarios. For example, in [10], a method is proposed for quantifying the risk in alignment with the ISO 26262 standard, where risk is determined based on exposure, controllability, and severity. In their approach, exposure is estimated by constructing a PDF using KDE. However, the KDE-based PDF estimation becomes inefficient in high-dimensional spaces due to the curse of dimensionality [84].

To address the curse of dimensionality that KDE is suffering from, Normalizing Flows (NF) offer a more flexible, deep-learning-based alternative capable of modelling complex, high-dimensional distributions. NF are a class of generative models that estimate complex probability distributions by transforming a simple base distribution (e.g., Gaussian) through a sequence of invertible and differentiable mappings [85]. If f_θ is a transformation parameterized by θ, we can compute the density of x using the change of variables formula:

$$p(x) = p_z(z)\left|\det\frac{\mathrm{d}f_\theta^{-1}(x)}{\mathrm{d}x}\right|,$$

where $z = f_\theta^{-1}(x)$ follows the known base distribution. By learning a complex mapping f_θ, NF can approximate multimodal distributions and capture intricate dependencies between variables.

NF methods, such as RealNVP [86], Masked Autoregressive Flows [87], and Neural Spline Flows [88], have gained popularity due to their scalability and ability to model high-dimensional densities. However, they require computationally expensive training and careful selection of model depth, transformations, and optimization hyperparameters.

In summary, KDE is best suited for low-dimensional data (typically $d \leq 3$), where a simple and interpretable density estimation method is sufficient. Its non-parametric nature makes it easy to implement, computationally efficient, and requires no training, making it a practical choice when speed and clarity are priorities. However, as dimensionality increases, KDE becomes inefficient due to the curse of dimensionality, leading to inaccurate estimates. In contrast, NF scale better with higher dimensions where complex dependencies exist between variables. By leveraging deep learning and invertible transformations, NF can model intricate PDFs without restrictive assumptions. While NF provides greater flexibility and expressiveness, it comes at the cost of higher computational complexity and the need for training. As a result, KDE is the preferred approach for simpler, low-dimensional problems, whereas NF is the better choice for applications requiring scalability and rich density modelling despite its increased computational demands.

5.4 (Dis)similarity of scenarios

A fundamental consideration when defining scenario similarity is the level of abstraction at which comparisons occur. Scenario relationships can be analysed at three primary levels:

4. Abstract level – High-level categorization based on general driving behaviours (e.g., merging, overtaking, emergency braking).
5. Logical level – Structural comparisons using event sequences, parameter constraints, and manoeuvre definitions.

6. Concrete level – Fine-grained comparisons incorporating numerical parameters, vehicle trajectories, and temporal execution data.

Based on the categorization above, the definition of similarity metrics varies across different levels. At the abstract level, similarity is mainly determined by semantic relationships, emphasizing intent and functional categorization of scenarios. In contrast, lower-level assessments rely more on quantitative metrics, such as numerical parameter comparisons. At the logical level, factors like event sequencing and trajectory alignment play a crucial role.

Given that scenario descriptions contain tags, regardless of the abstraction level, these tags can serve as a universal similarity metric across all levels. By leveraging tag-based comparisons, we can establish a consistent approach to scenario similarity assessment. We introduce tag-based similarity by formally defining tag overlap using the so-called Tag Similarity Score (TSS). The TSS is defined as the Jaccard similarity coefficient (intersection over union) between two sets of tags, A and B:

$$\mathrm{TSS} = \frac{|\mathrm{Tags}(A) \cap \mathrm{Tags}(B)|}{|\mathrm{Tags}(A) \cup \mathrm{Tags}(B)|}$$

For example, if $A = \{\mathrm{Highway}, \mathrm{Merging}, \mathrm{Daytime}\}$ and $B = \{\mathrm{Highway}, \mathrm{LaneChange}\}$, we have $\mathrm{TSS} = 1/4$.

To ensure a meaningful Tag Overlap Index, the existence of a common ontology at the federative layer, which provides a Uniform Resource Identifier (URI) for each tag, is essential. This ontology ensures that tags are standardized across different databases, preventing mismatched or inconsistent tagging that could otherwise distort similarity measurements.

5.4.1 Abstract-level similarity

For abstract-level classification, Natural Language Processing (NLP) techniques can be employed to analyse scenario descriptions. Ontology-based classification, semantic embeddings, and clustering techniques help establish functional categories, reducing ambiguity in scenario relationships. For abstract-level comparisons, we propose the use of Sentence Bidirectional Encoder Representations from Transformers (SBERT) [89] to encode scenario descriptions into numerical representations. The similarity between scenarios is then computed using cosine similarity [90].

SBERT generates fixed-size dense vector embeddings using a pre-trained transformer model fine-tuned on semantic textual similarity tasks. These embeddings capture sentence-level meaning and can be directly compared using cosine similarity. Alternatively, OpenAI's embedding models (such as text-embedding-ada-002) produce highly contextualized representations based on larger training data and more recent architecture. They are particularly strong in capturing semantics, causal reasoning, and longer sequences of events due to broader token-level attention during embedding generation [91].

5.4.2 Logical-level similarity

Logical-level scenario similarity in OpenSCENARIO requires a structured approach due to variations in scenario definitions and external file references. To ensure a fair comparison, we first identify common parameters between two scenarios, extracting only those that exist in both. Parameter similarity is computed using Jaccard overlap on numerical ranges, measuring shared constraints such as speed limits and lateral object offsets. For two scenarios with common parameter ranges $[a_1, b_1]$ and $[a_2, b_2]$, the Parameter Similarity Score (PSS) is:

$$\mathrm{PSS} = \frac{|\max(b_1, b_2) - \min(a_1, a_2)|}{|\min(b_1, b_2) - \max(a_1, a_2)|}$$

5.4.3 Concrete-level similarity

Two different approaches are presented. The first approach focusses on the comparisons of the trajectories of the scenario actors, while the second approach calculates the similarity between two scenarios based on the most critical scene.

5.4.3.1 Concrete-level similarity based on trajectories

At the concrete level, scenario similarity depends on the source and the type of data available [92], different types of scenarios exist:

- Recorded scenarios consist of logged time-series data, such as vehicle trajectories collected from real-world driving.
- Executable scenarios contain scripted instructions for traffic participants, specifying the manoeuvres and actions they must perform in a simulation.

Trajectory-based similarity metrics are only applicable when detailed trajectory data is available, typically in recorded scenarios. These trajectories can be compared using spatiotemporal trajectory metrics, such as: Dynamic Time Warping (DTW) [93], Hausdorff distance [94] or Fréchet distance.

The fundamental distinction between these methods lies in their treatment of spatial alignment and temporal correspondence. For example, DTW accommodates non-linear temporal variations, whereas Hausdorff and Fréchet distances primarily capture geometric similarity. To further elucidate these differences, the metrics are computed and visualized in the context of a simple takeover scenario, as shown in Figure 9, which highlights the key contrasts between the methods.

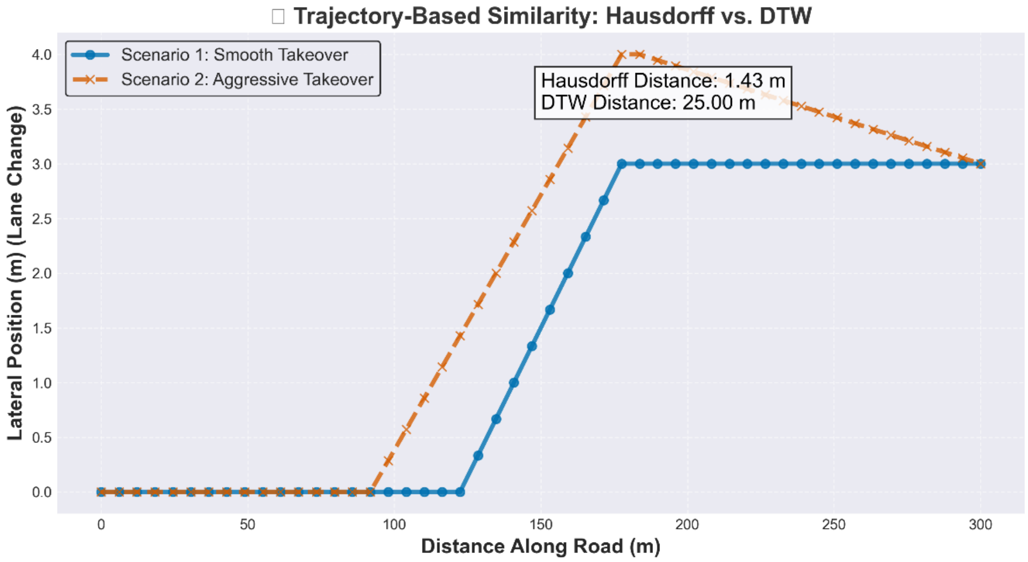


Figure 9: Comparison of the Hausdorff and DTW distances between two takeover scenarios.

Instead of comparing full trajectory there is possibility of inspecting critical moments. Mahadikar et al. [95] introduce a dissimilarity metric that prioritizes the most safety-critical scenes rather than the entire trajectory. Instead of comparing the whole path, they focus on the point of maximum risk (e.g., the minimum distance between vehicles before a crash).

As a similarity metric for executable scenarios, we propose a hierarchical tree-based approach for comparing OpenSCENARIO files by analysing both structural organization and content similarity. By parsing OpenSCENARIO files into nested dictionary representations, we can evaluate scenario similarity based on tree structure (branches and nodes) and leaf values (parameters and attributes). Structural similarity is computed using tree-edit distance or graph-based techniques, while content similarity relies on exact matching, Jaccard similarity (for attributes), and range-based comparisons (for parameters). The final similarity score combines these two metrics, ensuring meaningful comparisons even when scenarios contain dynamically assigned parameters or cross-file references (e.g., OpenDRIVE road networks and vehicle catalogues).

In practice, Python libraries such as DeepDiff can be leveraged to extract and compute differences between highly nested OpenSCENARIO structures. By applying Jaccard-based metrics, we can quantify similarity using measures such as shared nodes over total nodes, effectively capturing both structural overlap and content-based differences.

For a given scenario set, the aggregated similarity score can be computed using a weighted summation over all valid metrics:

$$S = \sum_i w_i S_i,$$

where w_i is the weight assigned to the i-th similarity metric, S_i, ensuring adaptability based on scenario characteristics.

5.4.3.2 Concrete-level similarity based on most critical scene

The literature review on dissimilarity metrics showed that many previous works compute dissimilarity by comparing entire trajectories across two scenarios. Here, a novel method for computing scenario dissimilarity is proposed which prioritizes the most safety-critical scene, while still accounting for the complete trajectory information of involved actors [95]. The most safety-critical scene may be defined considering different safety metrics. Below, we consider the distance of the ego vehicle to other scenario actors for scene criticality. The distance of ego vehicle to other actors is a suitable measure of scene criticality as it directly signifies imminent collision risk. Thus, the scene with the minimum distance of ego vehicle to other actors is the most safety-critical scene in the scenario.

Some assumptions and guidelines are devised for scenario dissimilarity:

1. Scenarios are different when operational conditions are different, e.g., road layout, weather, static environment, etc.
2. Scenarios are different when the actors in a scenario are different.
3. The trajectories followed by actors in a scenario can be abstracted by using a grid cell sequence on top of a road layout. Figure 10 (A) shows such an abstraction. Two scenarios are different when the grid cell sequences traversed by actors are different. Thus, the two scenarios shown in Figure 10 (B) and (C) are dissimilar.
4. The variables of interest when describing dissimilarity depend on the application use case [96]. For example, while comparing scenarios for testing a motion planning system, important features could be locations and velocities of all actors at the instant of criticality. In contrast, if the goal is to evaluate the perception system, features such as vehicle colour or size might also be important.

These pre-conditions define the boundaries for the dissimilarity metric definition and demonstrate how the metric definition could be adapted for different application use cases. For the metric formulation and the application example in this study, a motion planning system use case is considered. Thus, the variables of interest considered are positions, orientations, and velocities of the involved actors in the scenario. The subsequent section explains how the dissimilarity metric is formulated with respect to these variables of interest, considering the most safety-critical scene.

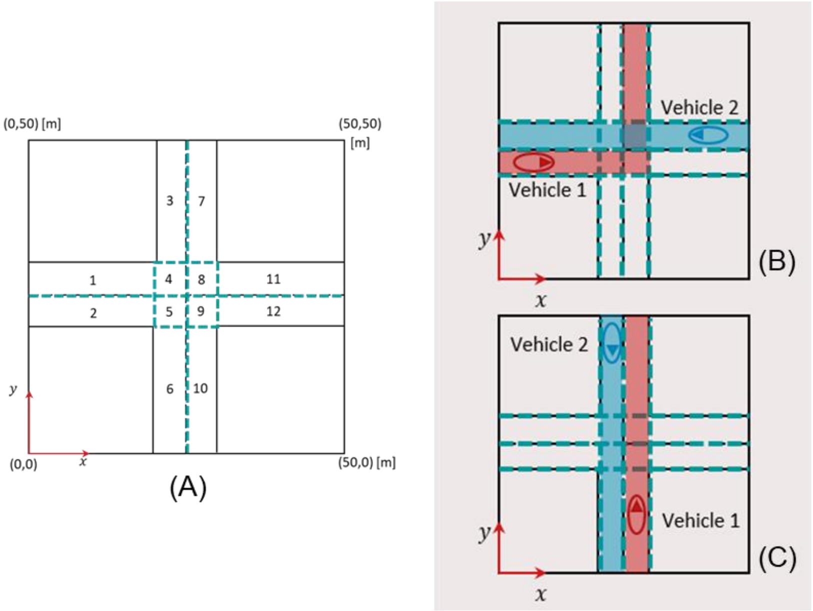


Figure 10: (A) shows the grid cell abstraction on top of the road network. (B) and (C) are two scenarios where the grid cells traversed by actor trajectories are different, and therefore the two scenarios are considered dissimilar.

Metric formulation

To quantify dissimilarity at the most safety-critical scene, a combination of discrete and continuous features can be considered. Discrete features have distinct, unordered values and can be categorized into groups. An example of a discrete feature could be the actor type associated with the safety-critical scene. These features, if different between scenarios, lead to completely different scenarios. Continuous features represent an infinite number of values within a defined range, for example, the values of positions or velocities of actors at the instant of maximum criticality. Thus, the dissimilarity when considering continuous features is also more continuous.

While considering the interaction of actors in the most safety-critical scene, two discrete features are: (1) actor type to which the minimum distance is observed, and (2) the grid cell which the ego vehicle occupies in the most safety-critical scene. If either of the discrete features differs between scenario 1 and scenario 2, the scenarios are considered dissimilar:

$$\Delta\mathrm{ActorType} = \begin{cases} 0 & \text{if } \mathrm{ActorType}_1 = \mathrm{ActorType}_2 \\ 1 & \text{otherwise} \end{cases} \in \{0,1\},$$

$$\Delta\mathrm{GridCell} = \begin{cases} 0 & \text{if } \mathrm{GridCell}_1 = \mathrm{GridCell}_2 \\ 1 & \text{otherwise} \end{cases} \in \{0,1\}.$$

The orientation angles of the scenario actors at maximum criticality are selected as continuous features for dissimilarity. Two angles are considered as shown in Figure 11: the relative heading angle θ_{rel} and the potential collision angle ϕ_{c} in the ego vehicle frame. These angles are selected due to their influence on safety. The θ_{rel} defines the orientation difference between two actors, which determines how they approach each other. The ϕ_{c} indicates the location of the other vehicle in the ego-vehicle frame and provides insight into the potential severity of collision as well as the responsibility of the ego vehicle in the incident. To quantify the magnitude of dissimilarity, the cosine similarity index [90] is used to obtain a normalized value between 0 and 1.

$$\begin{aligned}&\Delta\mathrm{Heading} = \Delta\mathrm{H} = 0.5(1-\cos(\theta_{\mathrm{rel1}} - \theta_{rel2})) \in [0,1]\\&\Delta\mathrm{CriticalPointAngle} = \Delta\mathrm{C} = 0.5(1-\cos(\phi_{c1} - \phi_{c2})) \in [0,1]\\&\Delta\mathrm{Combined} = w_1\Delta\mathrm{H} + w_2\Delta\mathrm{C},\end{aligned}$$

where $w_1 = w_2 = 0.5$ can be adjusted to prioritize $\Delta\mathrm{H}$ over $\Delta\mathrm{C}$ or vice versa. Finally, the dissimilarity computed using discrete and continuous features are combined to a final dissimilarity value between two scenarios:

$$\Delta(\mathrm{Scenario\ 1}, \mathrm{Scenario\ 2}) = \max(\Delta\mathrm{ActorType}, \Delta\mathrm{GridCell}, \Delta\mathrm{Combined}).$$

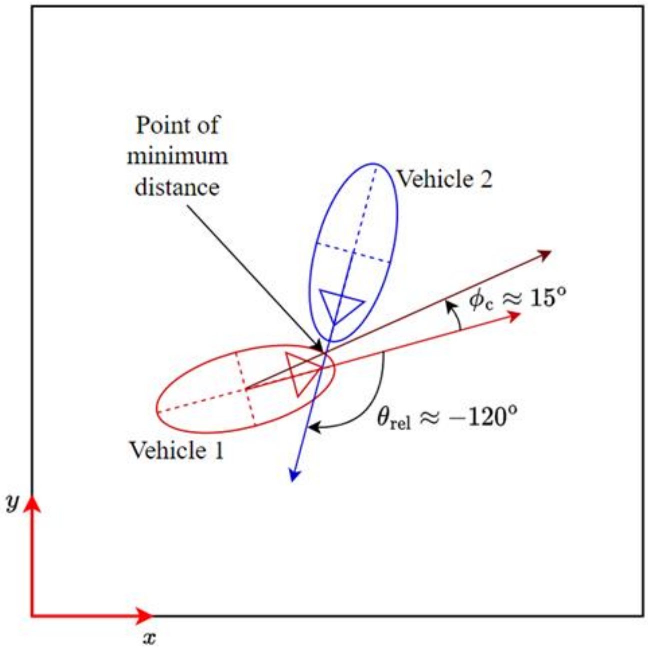


Figure 11: The relative heading angle θ_{rel} and the potential collision angle ϕ_c are the selected continuous features for dissimilarity.

5.5 Coverage

Two different approaches for measuring coverage are proposed. The first type focuses on the coverage of an ODD by a set of scenarios. The second approach focuses on the coverage of the parameter values of a set of test scenarios given the underlying distribution of these parameter values.

5.5.1 ODD coverage by scenarios

The following text on coverage metrics is taken from [73], where coverage is defined as the degree to which a set of scenarios observed in real-world data cover an ODD. To further distinguish the metrics that are proposed later in this section, two types of coverage are considered, both aiming to answer different questions:

- Type I: Do the collected scenarios cover all relevant aspects of an ODD?
- Type II: Do the collected scenarios cover all relevant aspects that are in the driving data?

Four different coverage metrics are proposed. The first metric is the tag-based coverage, which addresses coverage type I. The other three metrics, i.e., time-based coverage, actor-based coverage, and actor-over-time-based coverage, address coverage type II.

5.5.1.1 Tag-based coverage

A shared keyword structure established across the database enables tag-based analysis which becomes an efficient method to evaluate the coverage and balance of scenario across the database. By systematically assigning tags to all scenarios, it becomes possible to assess whether certain attributes are underrepresented or entirely absent within the database. This approach ensures that the SCDB is diverse, well-distributed, and capable of supporting different kinds of robust analysis across various use cases.

Figure 12 illustrates how a tag-based approach can be used to evaluate the coverage of a defined ODD within a database. A set of tags representing the ODD is used as input to query the database, and the resulting data is then analysed to assess how comprehensively the input ODD attributes are covered.

One of the key advantages of using a structured taxonomy is the ability to perform coverage analysis at different levels of the taxonomy. This approach allows for a detailed evaluation of each level within the taxonomy to verify that for a given ODD input, scenarios containing all the queried attributes are included in the database. For instance, if “particulates” are included in the ODD input but completely absent from the database (as indicated in red in Figure 12), complete coverage of the ODD cannot be achieved. At a more granular level, if a parent attribute such as “natural illumination” is part of the input ODD, tag-based coverage analysis can help determine whether all its corresponding child attributes of “daytime,” “night time,” and “low ambient lighting”, are also represented in the database. As highlighted in red in Figure 12, the child attribute “low ambient lighting” is missing from the query results, indicating a gap in coverage. By leveraging tag-based analysis, the SCDB can systematically identify and address such gaps, ensuring that it provides full coverage of ODD requirements.

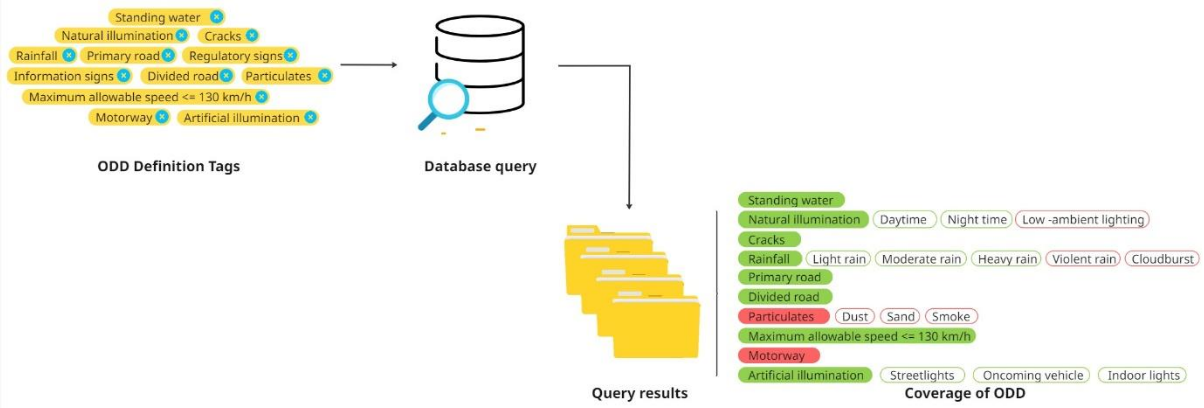


Figure 12: Coverage of ODD attributes by scenarios within the database.

Beyond identifying missing attributes, this granular assessment can gauge if each level within the taxonomy is well-represented. For example, consider an ODD that is designed to handle all intensities of rainfall. The database may contain 1,000 scenarios labelled with “rainfall”, indicating that wet weather conditions are well-documented. However, a deeper analysis might reveal that only 12 of those scenarios specifically describe “violent rainfall” intensity, which is a critical factor for testing vehicle performance in extreme weather conditions. Without this level of analysis, the database may appear comprehensive at a high level but could lack specific scenarios, which may be crucial for ensuring safety in real-world applications.

By systematically analysing the distribution of tags, we can take corrective measures, such as curating new scenarios to address gaps to achieve a well-balanced database that can provide complete coverage of an ODD input.

The following tag-based coverage metrics evaluate not only the coverage of all tags but also the distribution of tags across various types of scenarios. Before introducing this tag-based coverage metric, we need to distinguish scenarios from Scenario Categories (SCs) [32]. Here, a scenario refers to a quantitative description of the relevant characteristics of the ego vehicle, its activities and/or goals, its static environment, and its dynamic environment. In contrast, an SC refers to a qualitative description of the ego vehicle, its activities and/or goals, its static environment, and its dynamic environment. For example, the SC “cut in” comprises all possible cut-in scenarios. Scenarios may further be enriched with tags, e.g., a scenario belonging to the SC “cut in” may have the tag “actor at left” to indicate that there is an actor at the left side of the ego vehicle that prevents the ego vehicle from changing lane to the left.

Let $\mathcal{L}$ denote a set of tags and let $\mathcal{C}$ denote a set of SCs. Note that the set of tags should be based on the relevant aspects of an ODD, whereas the set of SCs could be based on the coverage type II metrics presented later. For the tag-based coverage, we make use of the function $N(L, C)$, which returns the number of scenarios that belong to SC C and contain the tag L. Continuing the previous example, in case we have 10 cut-in scenarios with an actor at the left of the ego vehicle, we would have $N(\text{Actor at left, Cut-in}) = 10$. The tag-based coverage metric is defined as follows:

$$\text{Coverage}_{\text{Tag}} = \frac{1}{n|\mathcal{L}||\mathcal{C}|} \sum_{L\in\mathcal{L}} \sum_{C\in\mathcal{C}} \min\big(n, N(L, C)\big),$$

where $n \in \mathbb{Z}^+$ and $|\cdot|$ denotes the cardinality, e.g., $|\mathcal{L}|$ equals the number of (distinct) tags. In case $\text{Coverage}_{\text{Tag}}(1) = 1$, each tag is associated to at least one scenario of each SC.

For this coverage metric, three choices need to be made:

1. The SCs belonging to $\mathcal{C}$. The SCs should cover the ODD. The set of SCs could be based on relevant literature [97, 98], though we suggest using other coverage metrics to justify that the set of SCs is complete. As mentioned before, the metrics for coverage type II may be used.

2. The tags belonging to $\mathcal{L}$. The tags should follow from the ODD description. When defining the ODD in accordance with the ISO 34503 standard [99], the corresponding tags listed in the ISO 34504 standard [100] may be used.

3. The required number of tags per SC, n. Minimally, $n = 1$, but to achieve more accurate statistics, if may be required to choose a higher value for n. To determine n, other metrics be used, e.g., see [70, 45, 71, 72, 101].

To obtain more accurate statistics of the scenarios belonging to an SC, it may be desired to have at least several scenarios of each SC with a certain tag. In that case, a larger value of n may be chosen.

Note that different tag-based coverage metrics can be defined if different sets of tags are considered. For example, one may choose to calculate the tag-based coverage with $\mathcal{L}$ consisting of tags related to environmental conditions, such as weather and lighting conditions, and with another set of tags consisting of scenery attributes, such as different types of roads.

5.5.1.2 Time-based coverage

The time-based coverage metric answers the question of whether all timestamp in the data is covered by one or more scenarios. Let $\mathcal{T}$ denote the set of all timestamps in the data set. For the time-based coverage, we introduce the function $M(t)$, which returns the number of scenarios at time t. Note that it may be possible that scenarios happen in parallel, e.g., a leading vehicle decelerating and another vehicle overtaking the ego vehicle. The time-based coverage metric is defined as follows:

$$\mathrm{Coverage}_{\mathrm{T}}(n) = \frac{1}{n|\mathcal{T}|} \sum_{t \in \mathcal{T}} \min\big(n, M(t)\big),$$

with $n \in \mathbb{Z}^{+}$. In case $\mathrm{Coverage}_{\mathrm{T}}(1) = 1$, all timestamps in the data are covered by at least one scenario. To account for the number of scenarios that can occur in parallel, one can increase the value of n.

5.5.1.3 Actor-based coverage

The actor-based coverage metric answers the question of whether every relevant actor is covered by at least one scenario. Let $\mathcal{A}$ denote the set of relevant actors. Here, the term “relevant” could be defined using some conditions. For example, $\mathcal{A}$ could contain all actors that are at some point in time within a certain distance of the ego vehicle. Alternatively, $\mathcal{A}$ could contain all emergency vehicles in the data set, etc. Let $\mathcal{B}$ denote the set of actors that are part of at least one scenario. Then, the actor-based coverage metric is defined as follows:

$$\mathrm{Coverage}_{\mathrm{A}}(\mathcal{A}) = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A}|}.$$

5.5.1.4 Actor-over-time-based coverage

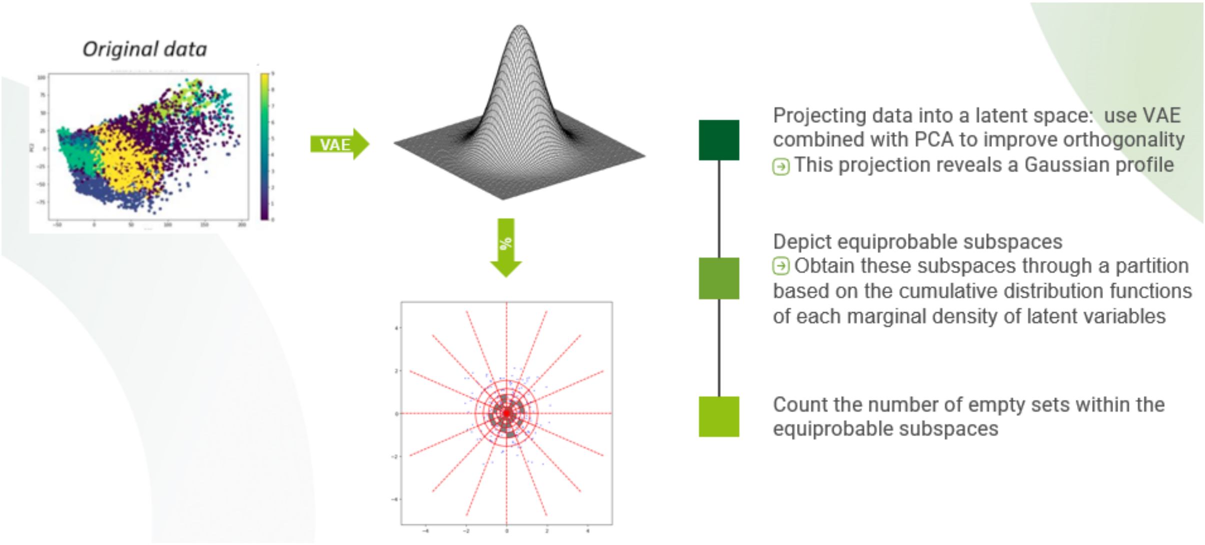
Achieving $\mathrm{Coverage}_{\mathrm{A}}(\mathcal{A}) = 1$ means that all actors of the set $\mathcal{A}$ are part of at least one scenario. However, it does not consider the temporal aspect of when these actors are part of a scenario. For example, it could be the case that an actor is near the ego vehicle – and thus part of $\mathcal{A}$ – but only part of a scenario once this vehicle is far away. To accommodate the time aspect, we introduce the fourth coverage metric; the actor-over-time-based coverage.

Let $\mathcal{T}_a$ denote the set of timestamps at which the actor $a \in \mathcal{A}$ satisfies the conditions that makes this actor part of $\mathcal{A}$. Furthermore, let $K(a,t)$ be the number of scenarios at time t that contain actor a. Then, the actor-over-time-based coverage is defined as follows:

$$\text{Coverage}_{\text{AT}}(\mathcal{A}) = \frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}}\frac{1}{|\mathcal{T}_a|}\sum_{t\in\mathcal{T}_a}\min\bigl(1, K(a,t)\bigr).$$

5.5.2 Parameter space coverage

The following method is based on [102, 103]. The proposed method (see Figure 13) is based on an innovative statistical approach designed to optimize the selection of testing scenarios for autonomous vehicle safety. Initially, high-dimensional scenario data—composed of numerous continuous variables—is transformed into a latent space using a Factor Variational Autoencoder (FVAE) [104]. This transformation is enhanced with cyclic KL annealing [105] and principal component analysis to achieve nearly independent, Gaussian-distributed latent variables. In this latent space, the continuous data is partitioned into equiprobable [106] subspaces by computing the cumulative distribution functions of each marginal variable. This discretization enables the selection of a representative subset of samples that faithfully capture the overall statistical characteristics of the original data, thereby dramatically reducing the number of tests required.



Projecting data into a latent space: use VAE combined with PCA to improve orthogonality
➔ This projection reveals a Gaussian profile

Depict equiprobable subspaces
➔ Obtain these subspaces through a partition based on the cumulative distribution functions of each marginal density of latent variables

Count the number of empty sets within the equiprobable subspaces

Figure 13: The main steps for determining the parameter space coverage.

A major challenge in this process is managing the complexity inherent in real-world data. The high number of variables — and their frequent interdependencies — complicates the discretization of the scenario space. This complexity can lead either to an oversimplification of the data or to an exponential increase in the number of required samples, thereby compromising the faithful representation of diverse scenarios and posing significant challenges to maintaining both statistical rigor and computational feasibility.

Finally, the method includes a mechanism to identify gaps or under-represented regions within the latent space. When such areas are detected, synthetic scenarios are generated using the same deep generative model, ensuring comprehensive coverage of the scenario space. This step not only guarantees that critical cases are not overlooked but also further enhances the efficiency of the testing process.[1]

[1] This solution is a Vedecom-MOOVE4.0 project outcome brought as background to SUNRISE.

Observations from the method reveal that the latent space transformation effectively captures the essential features of high-dimensional driving data while drastically reducing the number of required test scenarios. The equiprobable partitioning ensures that each subspace is statistically representative, and the subsequent selection process reliably identifies both typical and critical cases. Notably, the approach's ability to detect gaps in the scenario space — and to fill these with synthetic data — demonstrates a robust mechanism for maintaining comprehensive coverage. In conclusion, the proposed approach offers a cost-effective and scalable solution for autonomous driving safety testing. It leverages advanced deep learning techniques and rigorous statistical methods to improve testing precision, reduce computational overhead, and ultimately enhance the reliability of safety evaluations for ADSs.

VEDECOM's Intellectual Property.

6 APPLICATION

This chapter presents demonstrations of the metrics that have been developed in SUNRISE and presented in Chapter 5. This chapter adheres to the structure established in Chapter 5.

6.1 Testing purpose

As outlined in Section 5.1, three distinct metrics have been developed for testing purposes: scenario relevance, scenario criticality, and scenario complexity. This section provides examples of how these metrics are applied.

6.1.1 Scenario relevance

This section demonstrates the application of the scenario quality metric defined in Section 5.1.1. The process involves four key steps: scenario selection, risk estimation, acceptance criteria definition, and relevance evaluation. These steps help identify and prioritize high-risk scenarios from a scenario database, ensuring that tests are both relevant and effective in enhancing system safety and robustness.

Step 1 - Scenario Selection

The first step in applying the metric is to select a subset of scenarios from the available SCDB for a specific ODD. For this study, data from the Safety Pool SCDB are used, which contains over 200,000 predefined driving scenarios for testing and validation, including 3,096 that are freely available. To ensure relevance to safety testing, the selection focuses on junction scenarios (intersections and roundabouts), which are known for their complexity and high accident occurrence. Within the freely available dataset, 1,065 scenarios fall into this category, forming the primary dataset for risk assessment.

Step 2 - Risk Estimation

With the ODD scenarios defined, the next step is to assess the risk of each scenario in the dataset. This is done using the risk estimation approach from [78], which quantifies risk based on CPs. Examples of CPs are:

- CP_{17}: Intersecting planned trajectories of TPs
- CP_{90}: Strong braking manoeuvre of ego/non-ego-TP
- CP_{118}: Road weather
- CP_{134}: Occluded vehicle

A sample of CP IDs by associated risk is provided in Table 3.

To estimate the overall risk of a scenario, it is first necessary to identify which CPs are present. This process is automated through a script that analyses scenario definitions, extracting relevant information based on common keywords and tags. As a result, 65 out of 166 CPs were identified in the Safety Pool SCDB, while the remaining CPs could not be detected due to incomplete scenario descriptions.

Table 3: Sample table of Criticality Phenomena (CPs) by associated risk.

CP identifier	$\mathrm{Risk}(\mathrm{CP},\mathrm{Accident},\mathrm{Severity} \geq 1 \mid \mathrm{OD}) \times 10^{9}$
CP_{17}	455.34
CP_{23}	68.47
CP_{26}	178.93
CP_{40}	77.31
CP_{41}	456.55
CP_{48}	203.11
CP_{53}	7.76
CP_{90}	475.70
CP_{118}	199.86
CP_{134}	67.70

Since some scenarios contain multiple CPs, their individual risks must be aggregated to determine the overall scenario risk. This is achieved using Eq. (1), which combines the risk contributions of all identified CPs within each scenario. An example applied to a scenario would be as follows:

$$\begin{aligned}
\mathrm{Risk}(\mathrm{scenario}_1) &= 1 - \prod_{i=1}^{n} \left(1 - \mathrm{Risk}(\mathrm{CP},\mathrm{Accident},\mathrm{Severity} \geq 1 \mid \mathrm{OD})_i\right) \\
&= 1 - \big((1 - \mathrm{RiskCP}_{26}) \cdot (1 - \mathrm{RiskCP}_{40}) \cdot (1 - \mathrm{RiskCP}_{41}) \cdot (1 - \mathrm{RiskCP}_{54})\big) \\
&= \big(1 - (1 - 178.93 \cdot 10^{-9}) \cdot (1077.31 \cdot 10^{-9}) \cdot (1 - 456.55 \cdot 10^{-9}) \\
&\cdot (1 - 7.76 \cdot 10^{-9})\big) \\
&= 7.21 \cdot 10^{-7}
\end{aligned}$$

Step 3 - Acceptance Criteria Definition

With a risk value estimated for each scenario, the next step is to establish an acceptance criterion to filter out lower-risk scenarios, ensuring that the selection process prioritizes the most critical scenarios for testing. To achieve this, a risk threshold is defined – for instance, selecting only scenarios with a risk value higher than the mean plus one standard deviation of the ODD scenario risk distribution.

When this threshold is applied to the selected ODD scenarios, the data set is significantly reduced, narrowing it down from approximately 1,000 scenarios to 200 scenarios, as shown in Figure 14.

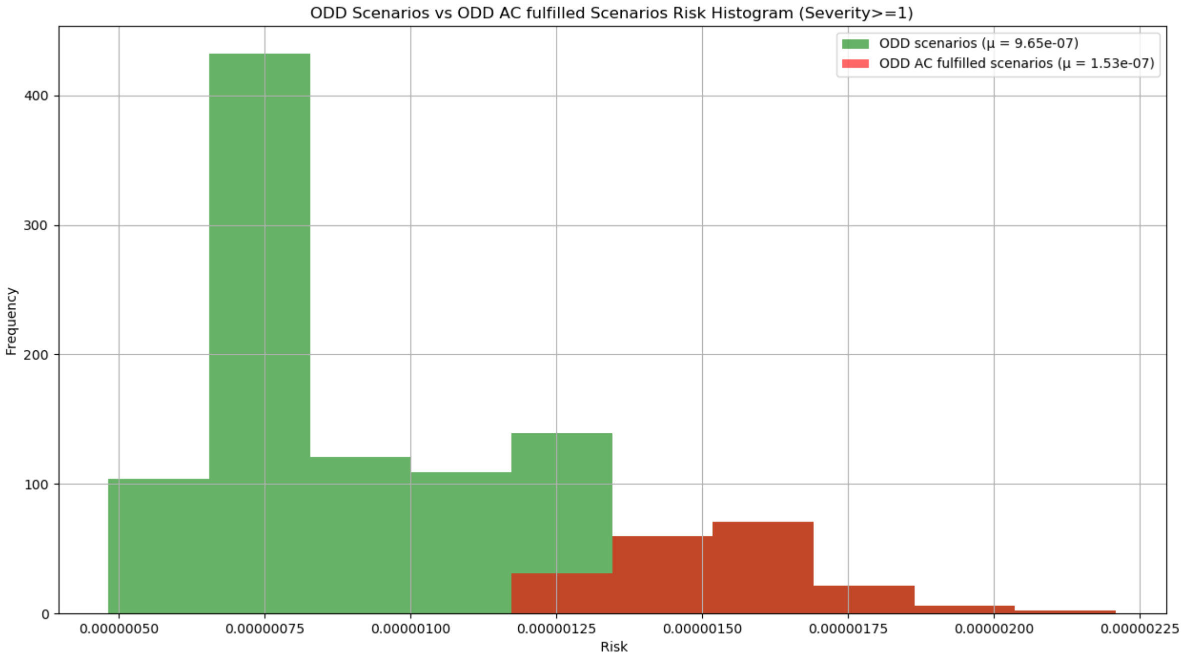


Figure 14: Risk of ODD scenarios, where the colour indicates whether the acceptance criteria are met.

From this refined set, a test scenario sample is randomly selected, ensuring that the final selection consists of scenarios that meet the predefined risk criteria while maintaining variability within the dataset.

Step 4 - Relevance Evaluation

The final step is to assess the relevance of the test scenario sample by comparing it to the reference distribution, which consists of the scenarios that meet the acceptance criteria. This is done using a Mann-Whitney U test [107], a non-parametric statistical test suitable for comparing distributions that do not necessarily follow a normal distribution.

The null hypothesis is defined as: *the mean risk of the test sample is lower than that of the reference distribution*. To determine whether the selected scenarios are relevant, the p-value is used as the evaluation metric. A p-value lower than 5 % leads to the rejection of the null hypothesis, confirming that the test sample accurately reflects the most critical scenarios.

To analyse the robustness of this method, experiments were conducted using different sample sizes. The results, summarized in Table 4, show that most test samples were rejected due to having a p-value greater than 5 %, indicating that they were not sufficiently relevant. However, a subset of test samples, highlighted in bold in Table 4, passed the evaluation and are considered strong candidates for proving ground testing.

Figure 15 illustrates the test results for the sample size of 20 and sample 5, which passed the evaluation, while Figure 16 shows the results for the sample size of 10 and sample 1, which did not pass.

Table 4: Test sample p-value results.

Sample size	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
10	0.5444	0.3196	**0.0267**	0.2515	0.1707
20	0.0784	0.8846	0.0551	0.0987	**0.0310**
30	0.3487	0.3087	0.9179	**0.0429**	0.8968

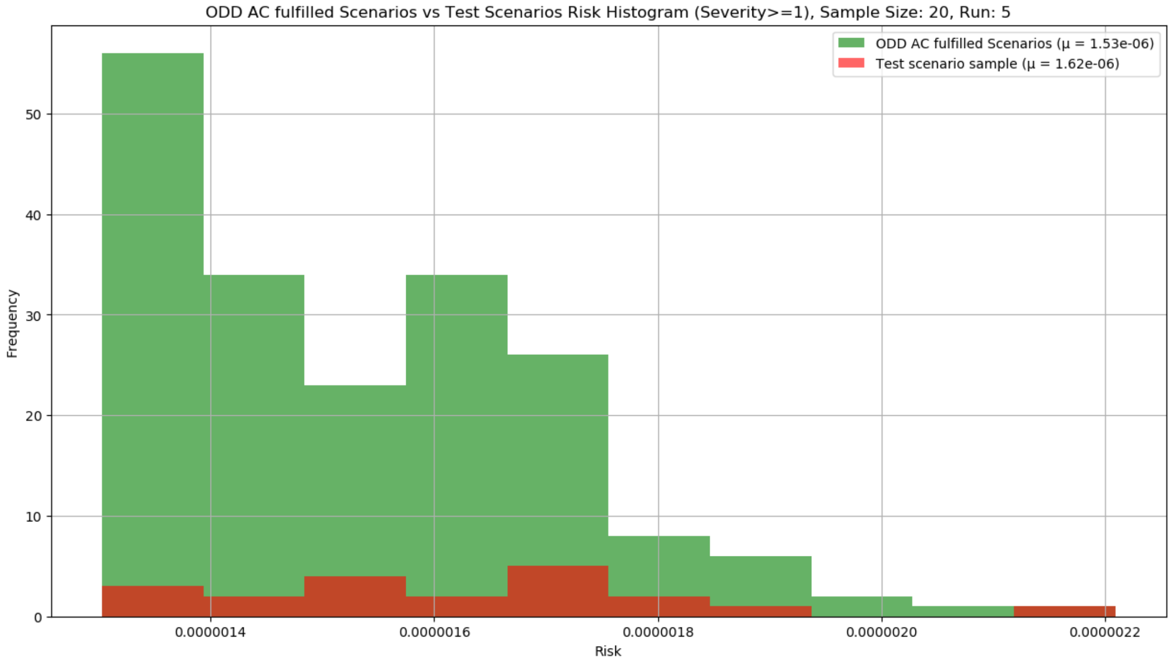


Figure 15: Scenarios for which ODD acceptance criteria are met, with in red the 20 scenarios that are selected with sample 5.

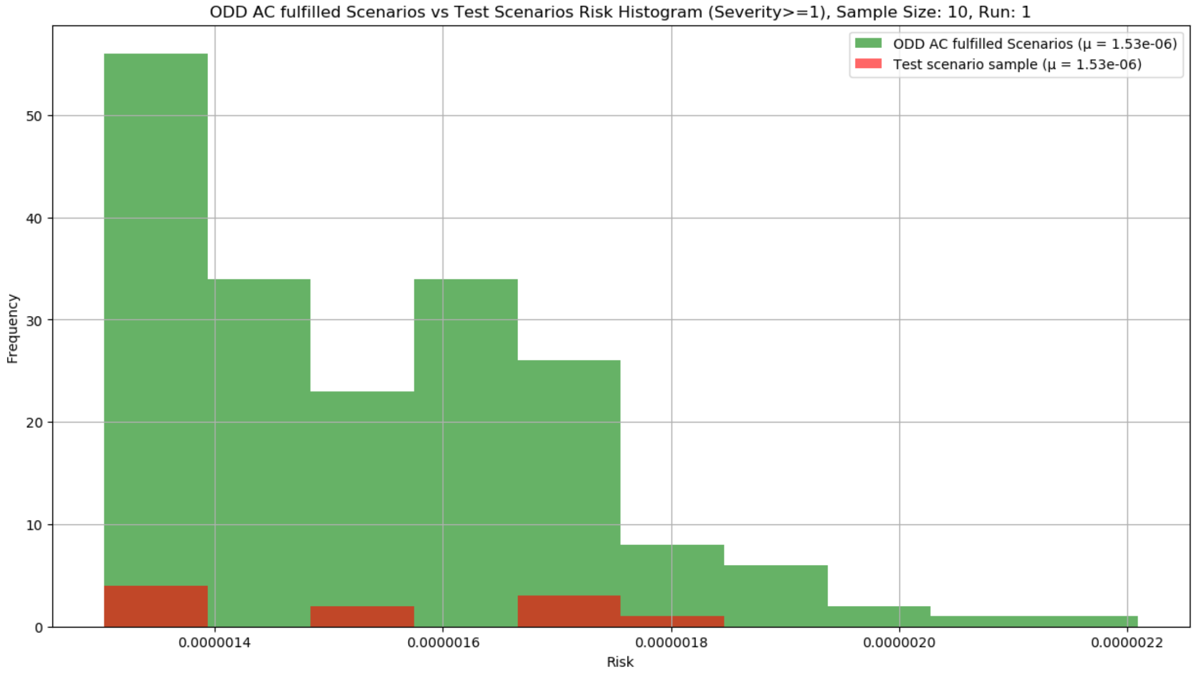


Figure 16: Similar as Figure 15, but with a sample size of 10 and sample 1.

6.1.2 Scenario criticality

This section discusses the application of the methodology described in Section 5.1.2 for the assessment of concrete scenario criticality within a logical scenario parameter space, upon execution of the scenarios. The logical scenario of a darting out pedestrian as described in UC1.3 of SUNRISE was used for the demonstration of the method. The logical scenario is parameterized using three key parameters. Interesting scenarios are generated in a two-step approach:

Step 1: to generate scenarios that are critical for each subsystem independently based on the different metrics' combinations for each SUT (leading to the use of different objective functions, namely the ego speed, the pedestrian speed, and the distance between ego and pedestrian when the latter starts crossing. Specifically, the following ranges of the three parameters were used:

- Ego speed: [30, 50] km/h
- Pedestrian speed: [0.5, 2] m/s
- Distance ego-pedestrian: [10, 50] m

Two SUTs were considered in this application: the driving SUT (end-to-end system) and the collective perception SUT. Criticality metrics from the respective SUT driving and the perception classes were selected. Specifically, the TTC as criticality metric for the driving SUT and the minimum detected distance of detection (DD) of the pedestrian as criticality metric for the perception. On the scenario level, the minimum TTC (TTC_{min}) and the maximum DD (DD_{max}) from all frames were used. The pass/fail thresholds were defined as follows:

- TTC_{min} threshold $\gamma_{\text{ttc}} = 1$ s
- DD_{max} threshold $\gamma_{\text{dd}} = 0.4 \cdot \left({}^{v_{\text{ego}}}/_{10}\right)^2 + 4$ (this threshold considers a simple formula of distance to break as a function of the vehicle speed)

The application is divided into three steps described in detail below.

Step 1: Random sampling within logical scenario parameter space

In this step, a random set of concrete scenarios from the logical scenario space are allocated and executed in simulation, and the criticality metric of interest (TTC_{min} or DD_{max}) is logged. This dataset is used for the pre-training of a Gaussian Process (GP) algorithm to predict the outcome of concrete scenarios as pass/fail.

Step 2: Online training and oversampling close to boundary

In this step, the pre-trained GP is used for the generation of samples close to the boundary with the aim to learn better the difficult boundary space where the model confidence is low. Each sampled scenario is executed, and the criticality metric is calculated. The GP is retrained after every 50 new samples until convergence. Figure 17 illustrates the random sampling from step 1 and the boundary sampling from step 2 along with the pass/fail boundary for the end-to-end driving SUT (TTC_{min})

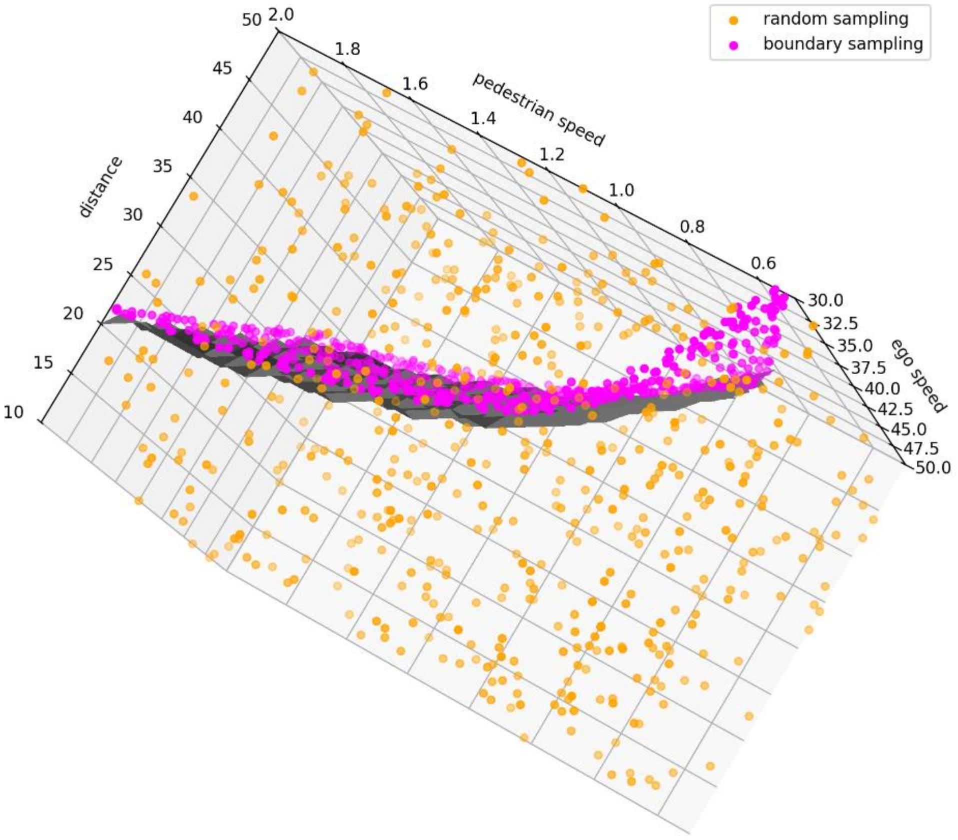


Figure 17: Random and boundary sampling from steps 1 and 2, respectively.

For different SUTs, the scenarios deemed as critical depend on the used metric and reflect the criticality with respect to a specific SUT. This is illustrated in Figure 18, where the pass and fail regions for the driving SUT and the CP SUT are superimposed. For better visualization, a 2D slice of the 3D parameter space is shown with given ego speed $v_{ego} = 42\,km/h$. The region where the outcome differs between the two SUTs is denoted with pink colour. As observed, this region is significant and highlights the need for assessment of criticality with a variety of metrics covering different SUTs of interest. Figure 18 also shows that the samples selected during step 2 fall indeed close to the respective boundary for both cases. This also demonstrates that uncertainty and scenario selection for further testing is also SUT dependent.

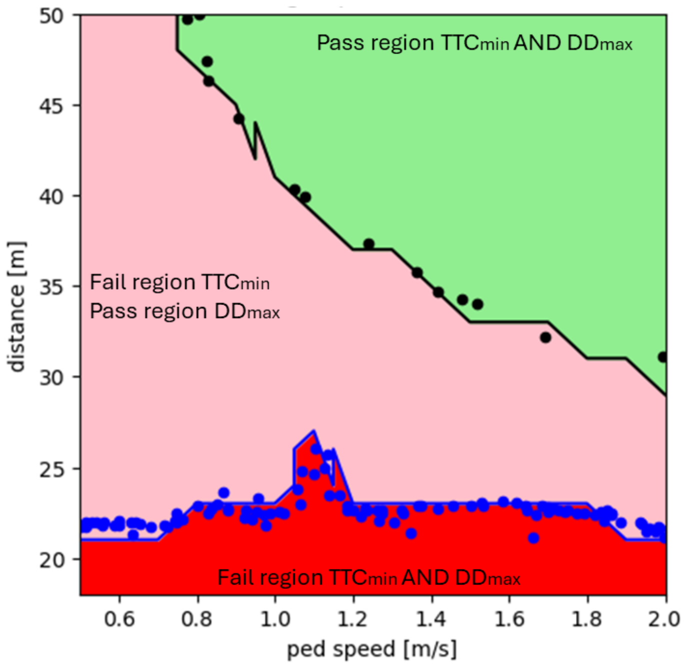


Figure 18: Superimposed pass and fail regions for driving and CP SUTs for a 2D slice with given $v_{\text{ego}} = 42\ \text{km/h}$. Green area denotes region where both systems pass, red area denotes region where both systems fail and pink area denotes region where driving SUT fails and CP SUT passes. Pass/fail boundaries for driving SUT (black) and CP SUT (blue) and boundary samples generated by the GP algorithm in step 2.

Step 3: Critical scenario assessment and parameter space coverage

After the two-step training phase, the GP algorithm is queried on the whole logical scenario parameter space to provide an estimate of the pass/fail regions and boundary along with confidence levels of its estimations. This permits the characterisation of the logical scenario space, the identification of critical scenarios, as well as the identification of scenarios that need to be further investigated on hybrid simulation or testing grounds (scenarios with low confidence level). Figure 19 demonstrates that the additional boundary oversampling (step 2) enhances the estimation confidence of the GP algorithm (from 92.1% to 96.3%), hence increases the coverage of the logical scenario parameter space. Orange regions denote low prediction confidence by the GP algorithm and are tagged as uncertain. The uncertain region is shown to be reduced after boundary oversampling.

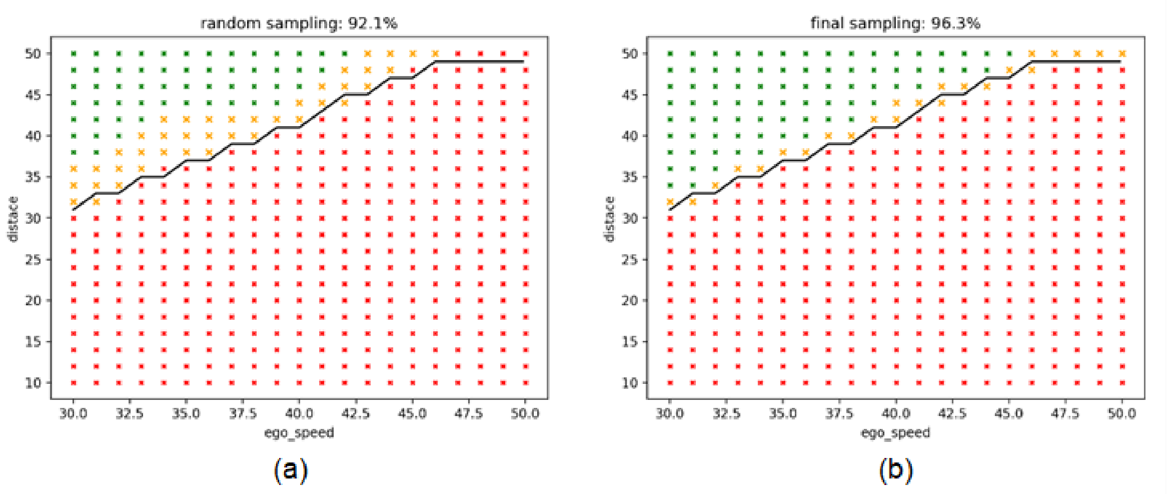


Figure 19: Coverage of logical scenario parameter space for a 2D slice with given $v_{\mathrm{ped}} = 0.9\ \mathrm{m}/s$. Prediction of pass (green) and fail (red) regions using the GP algorithm after (a) random sampling and (b) random sampling and boundary oversampling. Orange regions denote uncertain scenarios.

The above results demonstrate that our method can be used for the characterization of scenario criticality and the discovery of critical scenarios. Moreover, they illustrate that scenario criticality analysis needs to consider a specific SUTs and depends strongly on the selected metrics.

6.1.3 Scenario complexity

In the context of SAF, the Static Scenario Complexity Metric (C_{scene}) defined in Section 4.1.3 has been applied to a representative set of scenarios from SAF Use Case 2.1: Traffic Jam Assist (TJA).

All the scenarios provided were written in OpenSCENARIO format and represent typical traffic situations relevant for TJA system validation. The selected scenarios cover a variety of driving events, including:

- Speed adaptations due to traffic or speed limits
- Lane keeping and entering curves
- Target vehicle cut-in and cut-out manoeuvres
- Emergency braking situations
- Pedestrian crossings
- Stationary obstacles on the lane

For the complexity assessment, each scenario file was processed to extract the necessary input factors for the static complexity calculation. The input features extracted for each scenario can be described as follows:

- Environmental Conditions: Weather, illumination, and road surface conditions, as defined in the OpenSCENARIO <Environment> section.

- Traffic Participant Configuration and Types: Number and type of active traffic participants, limited to ego and one target actor for consistency with UC 2.1's actor configuration. <Entities> section of the scenario file provides the total number and categories of the traffic participant information.
- Ego Speed: The ego vehicle's initial speed was determined from the initial scenario actions.
- Initial Longitudinal Distance: The initial distance between the ego and target actors was extracted from the scenario's <Init> block, using their respective LanePosition's coordinates.

The bar chart in Figure 20 shows the complexity level of the SAF UC 2.1 scenarios with the default parameter values.

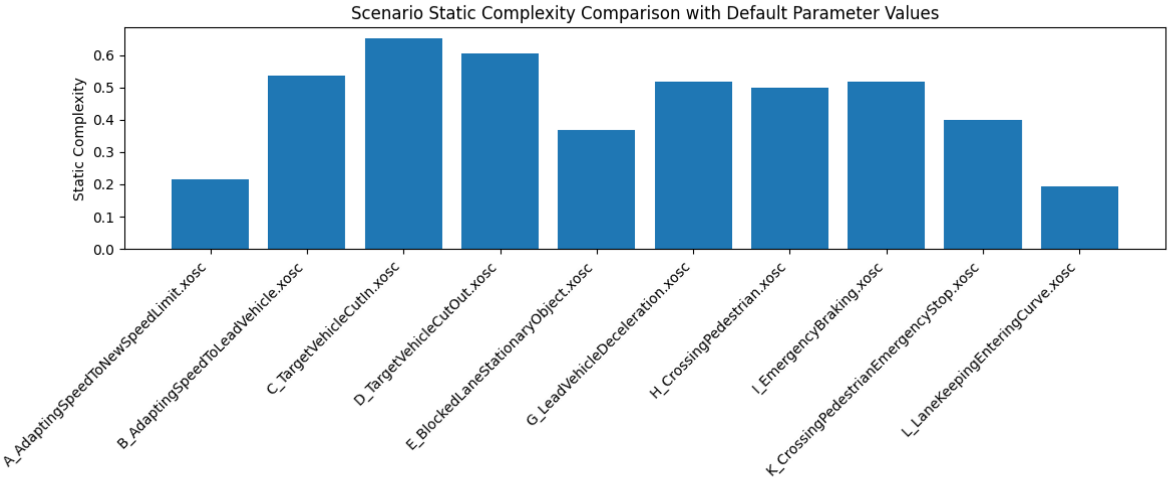


Figure 20: Static Scenario Complexity Comparison of UC 2.1 TJA Scenarios.

According to the resulting bar chart, scenarios without a target vehicle entity generally exhibit lower static complexity scores (e.g., adapting to new speed limit generally exhibit lower static complexity scores compared to scenarios involving dynamic interactions with target vehicles). This trend reflects the influence of traffic participant type and behaviour on the overall complexity calculation, where scenarios with moving vehicle targets introduce higher levels of interaction potential. Consequently, scenarios with only stationary objects or vulnerable road users such as pedestrians tend to receive lower C_{scene} values, highlighting their relatively reduced situational and operational complexity in the context of Traffic Jam Assist system validation.

6.2 Scenario description

This section follows the same structure as Section 5.2, meaning that first the application of the guidelines of Section 5.2.1 are discussed. Next, the application of a consistent taxonomy is addressed. This section ends with a demonstration of the scenario description completeness metric.

6.2.1 Scenario description guidelines

In this section, we will utilize the developed guidelines to assess the completeness of Use Case 2.1, which focuses on testing a "Traffic Jam AD System" [108]. We examine how the guidelines from Section 5.2.1 apply to Test 2-A. The primary goal of Test 2-A is to evaluate whether the system can adjust to a new speed limit that is lower than the current speed of the AD vehicle while maintaining a safe distance from the vehicle ahead.

In the initial step, we verify whether the given scenario conforms to the specified scenario format. The scenario is provided as an OpenSCENARIO XML file, which must adhere to the XML schema defined by this standard. OpenSCENARIO supplies an .xsd file that outlines the required schema. The scenario file for this use case is based on OpenSCENARIO version 1-1. Upon checking, it has been confirmed that the file complies with the schema set forth by ASAM. Consequently, the first guideline has been successfully met.

The second guideline stipulates that the scenario's content must be internally plausible. For instance, the initial positions of all vehicles should be realistic. Initially, the given scenario had identical starting positions for all vehicles, which would result in inaccurate starting conditions. However, within the scenario itself, these initial positions are correctly established for the initialization and are consistent with the use case description. Additionally, both the positions and speeds of all actors are coherent within the scenario description. The speed reduction specified in the scenario file does not conflict with any other provided information. Therefore, this guideline is satisfied, as all information within the scenario is plausible.

The third guideline mandates that the use case specifies a minimum level of information required within the scenario. For this use case, the following details must be included, which were derived from the test description [108]:

- an ego vehicle;
- the initial speed of the ego vehicle;
- the adapted speed of the ego vehicle, which is lower than its initial speed;
- a target vehicle positioned in front of the ego vehicle;
- a target vehicle with an initial speed identical to that of the ego vehicle; and
- a target vehicle with an adapted speed matching that of the ego vehicle, which is lower than its initial speed

These requirements represent the essential information needed to fulfil the use case description. Upon reviewing the scenario file, all necessary information is present, and the specified values align with those outlined in the use case. Consequently, the third guideline is adhered to.

In conclusion, the scenario file for Use Case 2.1 Test 2-A successfully adheres to all outlined guidelines, demonstrating its completeness as a test case in relation to the use case description. Initially, the scenario file was verified to conform to the OpenSCENARIO XML schema, ensuring structural integrity and compliance with ASAM standards. Subsequently, the scenario's content was evaluated for internal plausibility, confirming that all initial positions and speed settings of vehicles were realistic and consistent with the use case requirements.

Finally, a thorough examination confirmed that all essential information specified by the third guideline was present within the scenario file.

6.2.2 Consistent use of taxonomy

To ensure compliance with a shared vocabulary, the scenarios within the SCDB must align with the governing taxonomy adopted by the database. For example, if the SCDB follows the taxonomy defined in ISO 34503, this implies adherence to a standardized structure for representing ODD attributes. The consistent use of shared keywords and attribute hierarchies is essential for maintaining semantic coherence, interoperability, and traceability across scenarios contributed by diverse stakeholders.

Compliance with this common terminology can be assessed at any level of abstraction, depending on the level at which a scenario is described. For example, consider a logical scenario described in SDL (Scenario Description Language) Level 2 format [109, 110] as seen in Figure 21, which includes structured definitions of elements such as environmental conditions, road layout and traffic participants. When scenario descriptions deviate from the defined taxonomy – ISO 34503 in this case, several challenges arise. Most notably, it introduces inconsistencies in terminology that deviate from the standardized structure followed by the rest of the SCDB. This misalignment can significantly impact downstream processes such as search, filtering, automated tagging, and coverage analysis, leading to incomplete or misleading results.

Take, for example, the scenery description shown in Figure 21, where the attribute "Puddle" is assigned under the parent category "Drivable area surface". According to the ISO 34503 taxonomy, "Standing water" is the appropriate terminology for a phenomenon that occurs when water accumulates due to a depression in the drivable area. Moreover, this attribute is typically categorized under a different branch of the taxonomy – “Drivable area induced surface conditions”. Such discrepancies not only hinder semantic interoperability but also prevent accurate mapping between scenario content of the database and ODD requirements as defined in a search criterion.

It is important to emphasize that the taxonomy applies not just at the leaf node level but across the entire hierarchical structure, from high-level categories down to specific attribute values. This would also allow for consistency across units. Ensuring correct placement and naming of attributes throughout this hierarchy is essential for maintaining a logically sound and searchable SCDB. This, in turn, directly supports the SAF by enabling traceability between scenario descriptions and system operational limits, enabling trustworthy safety claims.

In conclusion, both the language used in scenario descriptions and the taxonomy to which they conform must be consistent across the SCDB. This dual consistency enables traceable, structured scenario descriptions that support reliable search and analysis. It also allows for the systematic addition of new attributes through structured extensions, avoids naming mismatches for existing attributes, and ensures that no branch of the taxonomy is inadvertently omitted. This foundational consistency is essential for enabling robust coverage analysis as described later and for supporting the SAF through transparent alignment between scenario description data and the system’s operational boundaries.

```
SCENERY ELEMENTS:
DO: Map - roads and junctions network [Network1] as:
Junctions: None
Roads:
R1:
START
        Road type [Motorway] as [R1] with zone as [N/A] AND speed limit of [30] in an [Urban]
        environment with
        Number of lanes [3] as [R1.L1, R1.L2, R1.L3]
        Road traffic direction [Right-handed]
        Lane type [Traffic lane]
        Lane markings [Broken line]
        Horizontal road geometry [Straight]
        Vertical road geometry [Level plane]
        Drivable area surface [Puddle]
        Length [9000 to 11000] AND Lane width [3.4 to 3.7]
END
```

Figure 21: Snippet of a logical scenario in SDL Level 2 format [109, 110].

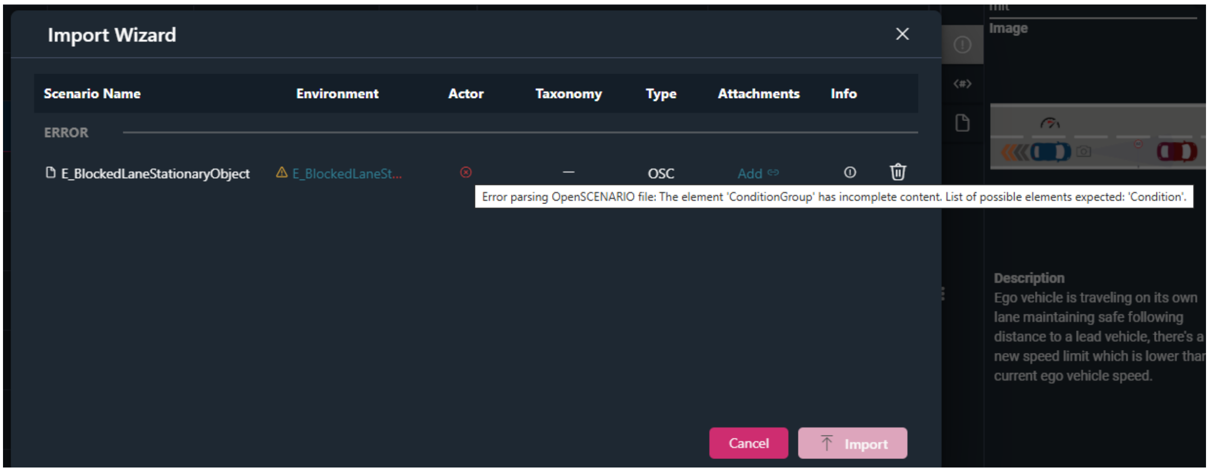
6.2.3 Scenario description completeness

Completeness evaluation of SAF UC 2.1 Traffic Jam Assist Scenarios in AVL SCENIUS SCDB

The application of the Scenario Completeness Evaluation in AVL Scenius SCDB begins at the scenario import phase.

Initially, the extension of the scenario file is verified (.xosc), followed by an XSD Schema validation. An XML Schema Definition (.xsd) file is an is an XML-based grammar that precisely describes the shape, data types, and rules an XML document must follow, so computers can validate, exchange, and auto-process that data with confidence.

AVL Scenius uses the official XSD Schema from ASAM to validate the structure and grammar of each imported OpenSCENARIO file. If the imported scenario file does not comply with the XSD Schema, the invalid elements are highlighted, and the scenario upload is rejected, as illustrated in Figure 22. This validation step represents the initial assessment of the core completeness criterion for the Scenario Artifact. All UC 2.1 scenarios are fully compliant with the ASAM-provided XSD Schema.



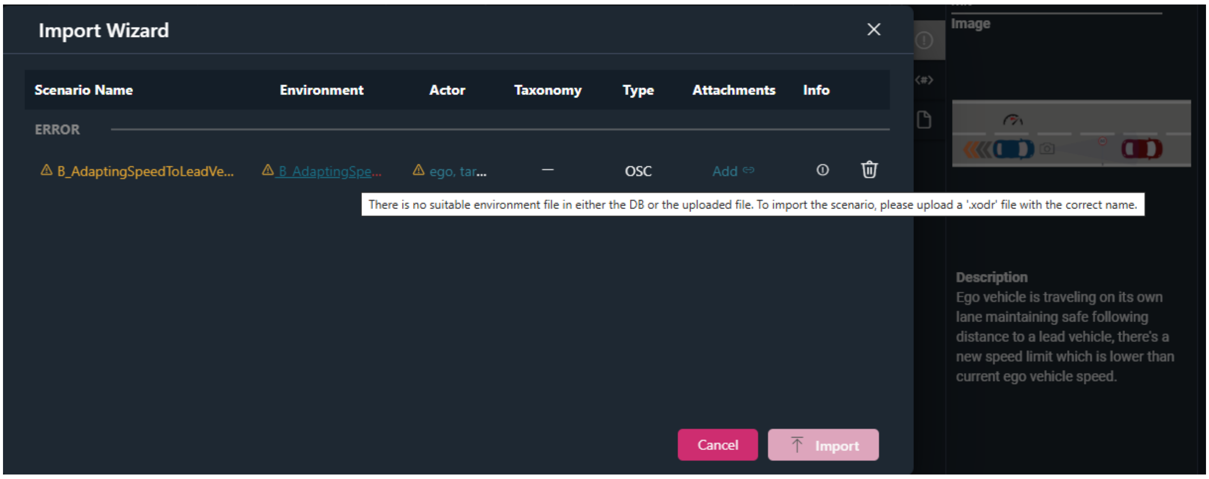
Scenario Name	Environment	Actor	Taxonomy	Type	Attachments	Info
ERROR						
E_BlockedLaneStationaryObject	⚠ E_BlockedLaneSt...	⊗	–	OSC	Add	ⓘ

Error parsing OpenSCENARIO file: The element 'ConditionGroup' has incomplete content. List of possible elements expected: 'Condition'.

Description
Ego vehicle is traveling on its own lane maintaining safe following distance to a lead vehicle, there's a new speed limit which is lower thar current ego vehicle speed.

Figure 22: Invalid OpenSCENARIO file import in AVL SCENIUS (as an example).

The second core completeness metric, Environment Definition, is also assessed during the scenario import process. If the referenced environment file does not exist in the SCDB, the user is prompted at this stage to upload the missing environment file. The corresponding error message displayed to the user is shown in Figure 23. The import process cannot proceed until the required environment file is uploaded to the database. In the case of UC 2.1 scenarios, all relevant OpenDRIVE files were included, and this completeness criterion was successfully fulfilled.



Scenario Name	Environment	Actor	Taxonomy	Type	Attachments	Info
ERROR						
⚠ B_AdaptingSpeedToLeadVe...	⚠ B_AdaptingSpe...	⚠ ego, tar...	–	OSC	Add	ⓘ

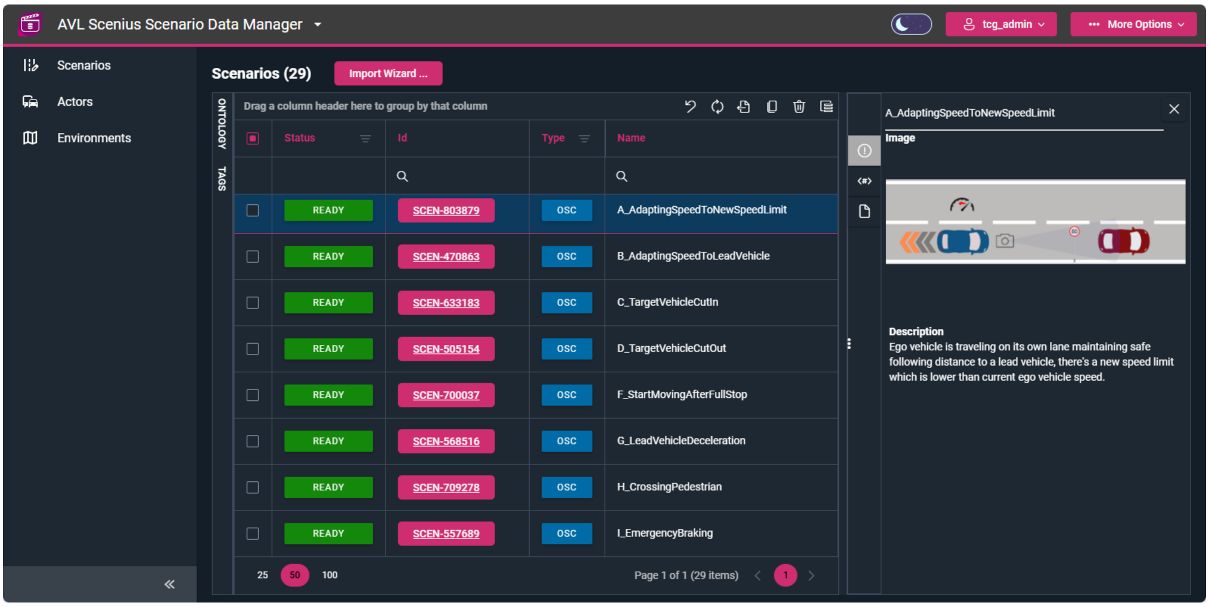
There is no suitable environment file in either the DB or the uploaded file. To import the scenario, please upload a '.xodr' file with the correct name.

Description
Ego vehicle is traveling on its own lane maintaining safe following distance to a lead vehicle, there's a new speed limit which is lower than current ego vehicle speed.

Figure 23: Environment not found error in the database.

The final core completeness criterion, Scenario Parameters, is assessed once the upload process is complete. If no parameters are declared within the OpenSCENARIO file, the user is notified that the scenario is marked as "Incomplete" due to missing parameters.

All SAF UC 2.1 scenarios include parameter declarations, and thus no such warning has been issued. AVL Scenius SCDB also supports the display of metadata and illustrative media, as illustrated in Figure 24. Scenario descriptions and illustrative media can either be included directly in the uploaded scenario package or added through the user interface after the initial upload. For the UC 2.1 scenarios, both descriptions and images are available within deliverable D7.1 [108] and they can be used within Scenius SCDB.


AVL Scenius Scenario Data Manager
tcg_admin
More Options
Scenarios
Actors
Environments
Scenarios (29)
Import Wizard ...
Drag a column header here to group by that column
ONTOLOGY
TAGS
A_AdaptingSpeedToNewSpeedLimit
Image

☐	Status	Id	Type	Name
☐	READY	SCEN-803879	OSC	A_AdaptingSpeedToNewSpeedLimit
☐	READY	SCEN-470863	OSC	B_AdaptingSpeedToLeadVehicle
☐	READY	SCEN-633183	OSC	C_TargetVehicleCutIn
☐	READY	SCEN-505154	OSC	D_TargetVehicleCutOut
☐	READY	SCEN-700037	OSC	F_StartMovingAfterFullStop
☐	READY	SCEN-568516	OSC	G_LeadVehicleDeceleration
☐	READY	SCEN-709278	OSC	H_CrossingPedestrian
☐	READY	SCEN-557689	OSC	I_EmergencyBraking

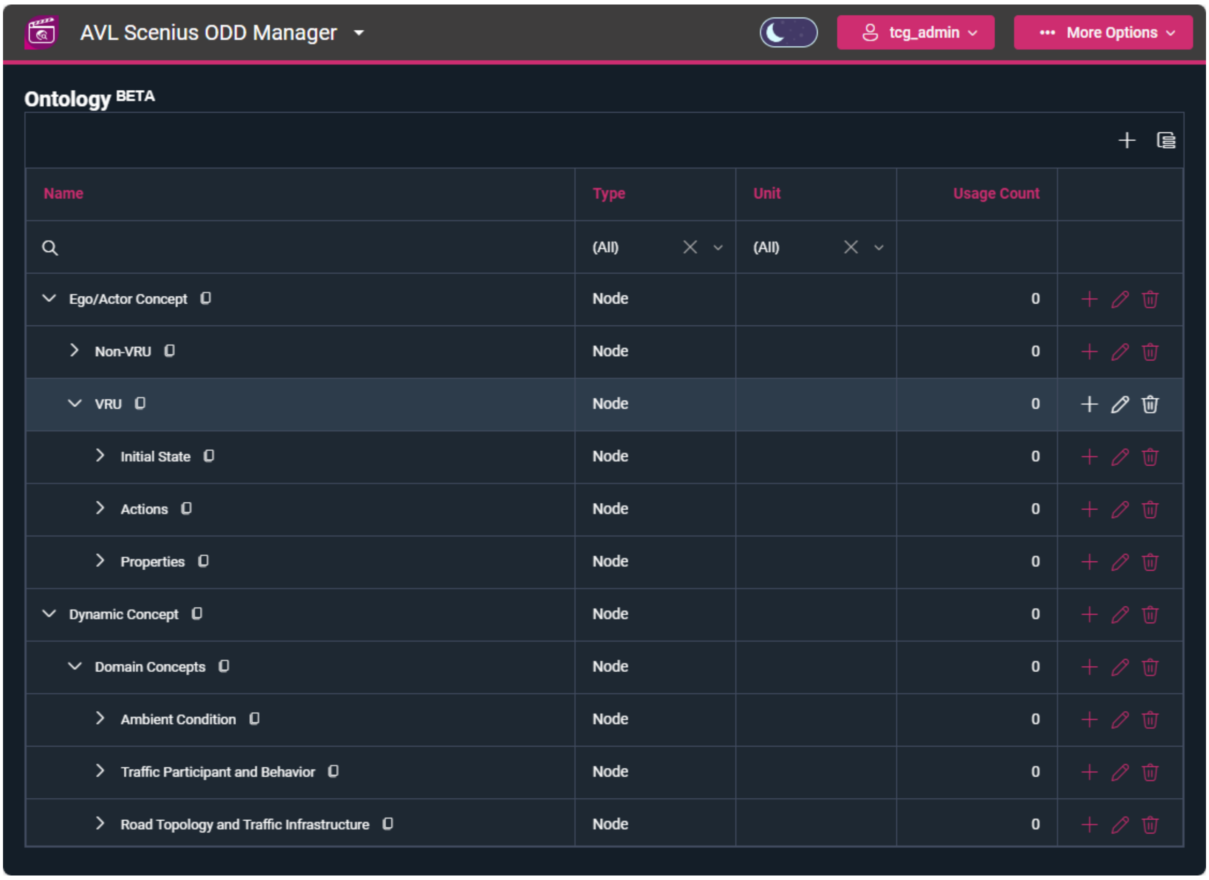
25 50 100 Page 1 of 1 (29 items) 1

Description
Ego vehicle is traveling on its own lane maintaining safe following distance to a lead vehicle, there's a new speed limit which is lower than current ego vehicle speed.

Figure 24: SAF UC 2.1 Scenarios on AVL Scenius Scenario Data Manager.

AVL Scenius assesses the presence of taxonomy by verifying whether a valid taxonomy definition file has been supplied alongside the scenario. If no taxonomy file is provided, the user can assign taxonomy elements from those predefined within the SCDB, as shown in Figure 25.

The UC 2.1 scenarios did not include a taxonomy file; however, they provided a clear ODD definition. Using this ODD information, elements from the master ontology can be assigned directly to the scenarios.



Ontology BETA

Name	Type	Unit	Usage Count
	(All)	(All)	
Ego/Actor Concept	Node		0
Non-VRU	Node		0
VRU	Node		0
Initial State	Node		0
Actions	Node		0
Properties	Node		0
Dynamic Concept	Node		0
Domain Concepts	Node		0
Ambient Condition	Node		0
Traffic Participant and Behavior	Node		0
Road Topology and Traffic Infrastructure	Node		0

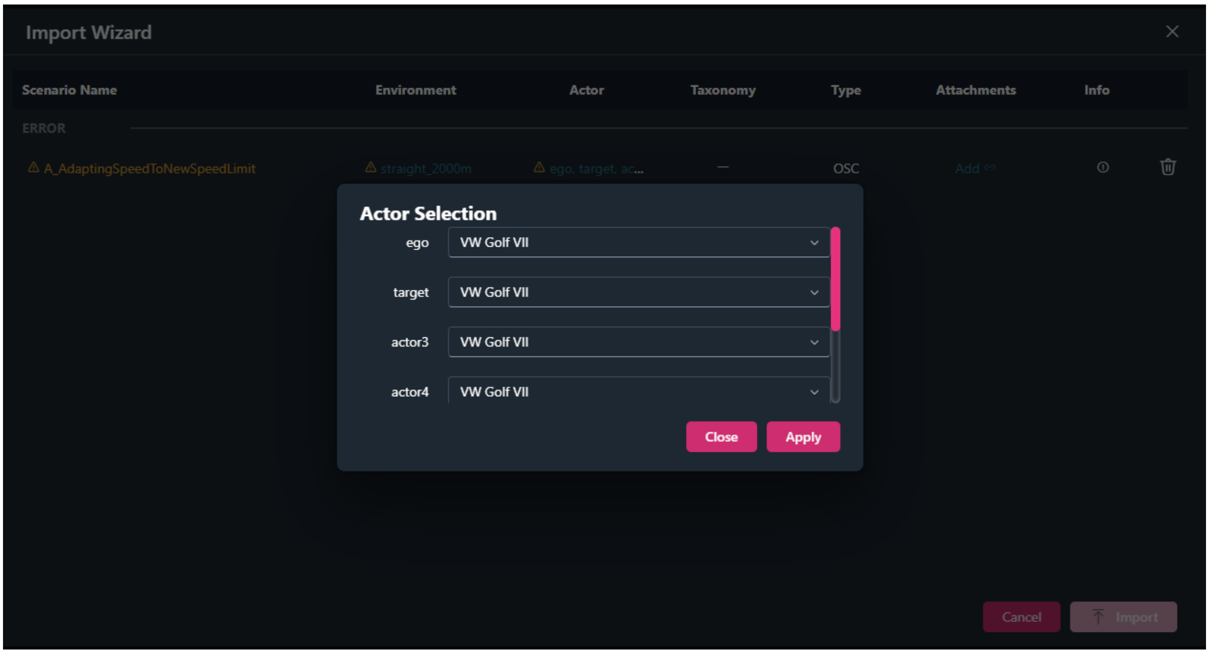
Figure 25: AVL Scenius ODD Manager – Ontology.

The presence of an end condition is assessed by verifying whether a StopTrigger element has been defined as a child element of the Storyboard element within the OpenSCENARIO file. An example snippet illustrating a valid StopTrigger definition is provided below:

```
<StopTrigger>
  <ConditionGroup>
    <Condition delay="0.0" conditionEdge="rising" name="Time limit">
      <ByValueCondition>
        <SimulationTimeCondition rule="greaterThan" value="60"/>
      </ByValueCondition>
    </Condition>
  </ConditionGroup>
</StopTrigger>
```

In the UC 2.1 scenarios, the StopTrigger elements were properly defined within the Storyboard element, thereby successfully meeting the End Condition check.

AVL Scenius SCDB verifies that each referenced actor is explicitly defined and categorized according to established entity types. If the actor definition is not made correctly the user will be prompted to choose an actor which was defined in the Scenius SCDB – Figure 26. For the SAF UC 2.1 scenarios, all actors have been explicitly defined, thus meeting this completeness criterion successfully.



Scenario Name	Environment	Actor	Taxonomy	Type	Attachments	Info
ERROR						
⚠ A_AdaptingSpeedToNewSpeedLimit	⚠ straight_2000m	⚠ ego, target, ac...	–	OSC	Add	ⓘ

Actor Selection

ego: VW Golf VII

target: VW Golf VII

actor3: VW Golf VII

actor4: VW Golf VII

Figure 26: User is prompted to choose actors for the scenario.

The completeness evaluation applied to SAF UC 2.1 scenarios within AVL Scenius confirmed that all scenarios meet both the core and descriptive completeness criteria. Each scenario successfully passed checks related to scenario artifact validity, environment definition availability, parameter declarations, taxonomy classification, presence of descriptive information and illustrative media, explicit end condition definitions, and actor entities. Consequently, all SAF UC 2.1 scenarios are classified as “complete” within the Scenius SCDB.

6.3 Scenario exposure

To illustrate the use of KDE and NF to estimate the scenario exposure at parameter level, the HighD data set was selected for the experiment due to its extensive coverage of over 40,000 km of naturalistic driving data and its high accuracy. This data includes trajectories of cars and trucks at six different locations on German motorways, captured using drone video footage.

To generate scenario data, each of the more than 100,000 vehicles in the data set is treated as an ego vehicle once. This process creates over 100,000 smaller data sets, each containing a single ego vehicle and its relative trajectory data, assuming the ego vehicle can see all surrounding vehicles within a 100-meter radius. Each smaller data set ends when the ego vehicle is 100 meters from its final position to prevent the sudden disappearance of vehicles from the drone's view. Consequently, vehicles with trajectories shorter than 100 meters are excluded as ego vehicles. This process resulted in 109,986 data sets with a single ego vehicle.

For illustrating the estimation of the exposure, parameter values of cut-in scenarios are used. In total, 2,992 cut-in scenarios are found in the data. Four different parameters are considered:

1. The initial speed of the ego vehicle when the other vehicle initiates the cut-in manoeuvre.
2. At the same moment, the speed of the vehicle that performs the cut-in manoeuvre.

3. The average lateral speed of the vehicle performing the cut-in manoeuvre during the cut-in manoeuvre.
4. The distance between the ego vehicle and the vehicle performing the cut-in manoeuvre when the cut-in manoeuvre starts.

KDE and NF have been used to estimate the joint distribution of these four parameters. Figure 27 shows the result of the estimation of the PDF. Note that this figure shows the marginal distributions based on the estimate joint distribution. The bars show the histograms of the data, and the lines denote the estimated distributions. For the first three parameters, the difference between KDE and NF is not substantial. However, for the fourth parameter, KDE appears to overestimate the probability density near the edges. This is a common phenomenon with KDE. This problem is less pronounced for NF.

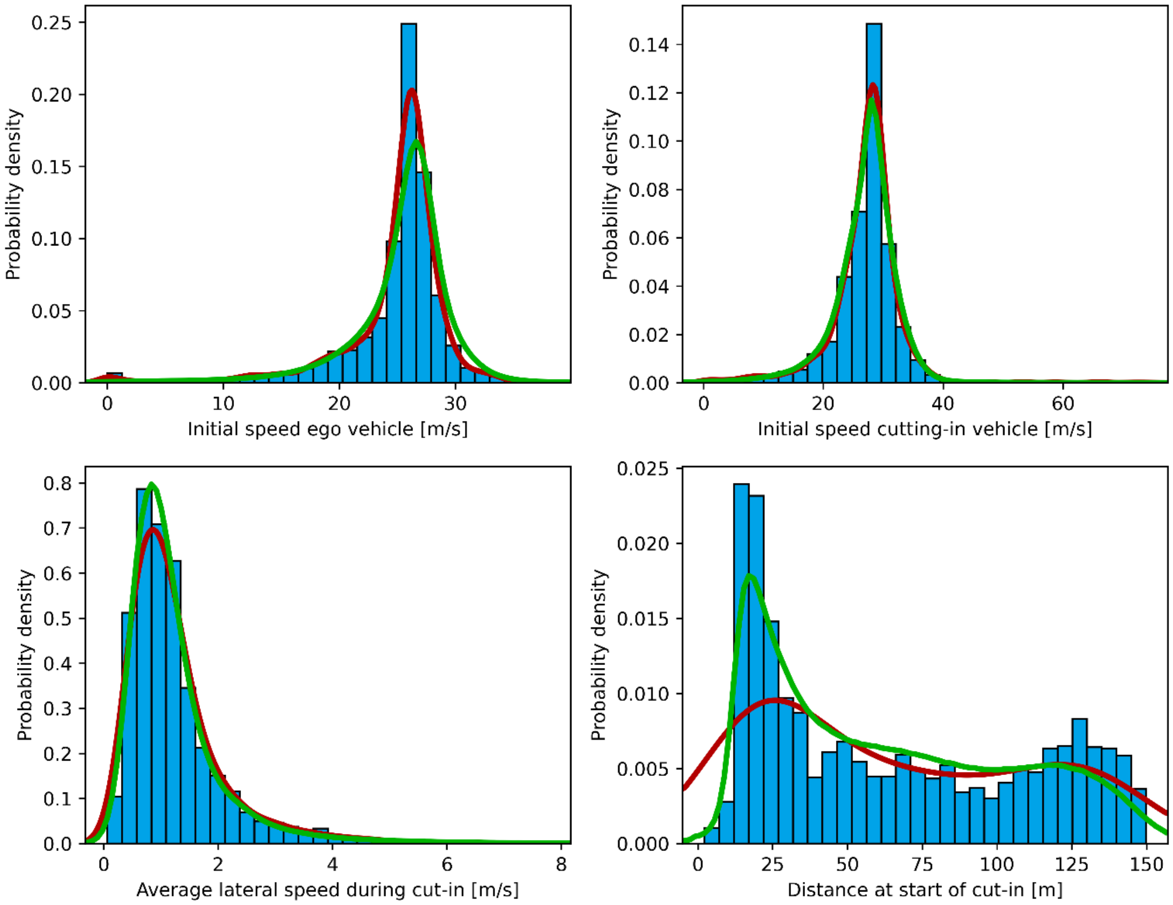


Figure 27: Four parameters of cut-in scenarios. The histograms display the original data, while solid lines represent the marginal probability distributions of the estimated PDF. Green lines show KDE estimations, and red lines show NF estimations.

6.4 (Dis)similarity of scenarios

Section 5.4 distinguishes between abstract-level, logical-level, and concrete-level scenario descriptions. This section applies the metrics proposed in Section 5.4, organized according to these scenario-level descriptions.

6.4.1 Abstract-level similarity

A collection of 600 abstract scenarios have been developed to demonstrate and evaluate the effectiveness of our semantic similarity metrics. The creation process followed a structured, multi-stage methodology:

1. Scenario Generation Using Language Models: We used large language models like ChatGPT to create new scenarios based on a set of initial examples. These new scenarios follow a structured format (SDL) [109] and are designed to be consistent with the original examples in terms of content and style. A similar two-step scenario generation process – starting from abstract descriptions and refining them using language models is described in [111].
2. Ontology Alignment: To ensure conceptual coherence and relevance, we imposed constraints based on the SUNRISE ontology, guiding the structure and content of the scenarios toward a unified semantic framework.
3. Cleaning and Validation: The resulting dataset was rigorously reviewed through automated filters and manual inspection to eliminate redundancy, ensure clarity, and maintain high-quality linguistic and semantic standards.

Once finalized, the scenarios were encoded into numerical embeddings using transformer-based language models (SBERT). These numerical embeddings used to compute pairwise semantic similarity across all scenarios. The similarity results were visualized using a heatmap (Figure 28), and the derived similarity matrix also served as a distance metric for clustering. The scenarios were then reordered based on clustering outcomes, making it easier to visually identify clusters of conceptually similar scenarios.

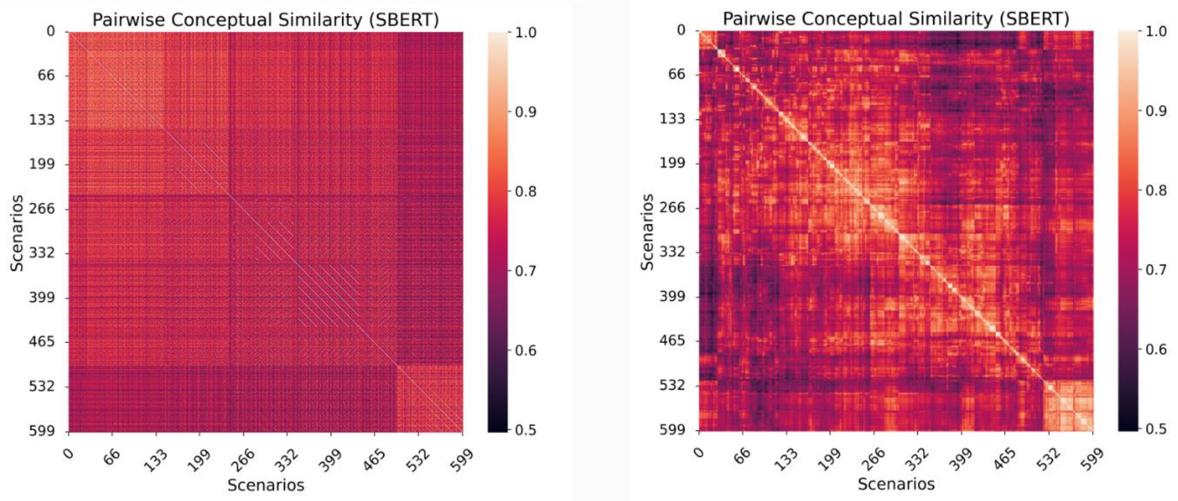


Figure 28: Similarity matrix between randomly distributed scenarios (left) and sorted scenarios (right).

In addition to sentence-level semantic analysis, each scenario is also associated with a list of ontology-based tags derived from the SUNRISE ontology, representing high-level conceptual elements present in the scene. These tag sets allow for a complementary similarity analysis using the Jaccard index, which measures the degree of overlap between two tag sets based on the ratio of their intersection to their union. By calculating pairwise Jaccard similarity scores across all 600 scenarios, we generated a second similarity matrix that captures conceptual alignment based purely on shared ontological classes. Comparing this tag-based similarity matrix with the SBERT-based semantic similarity matrix revealed a strong positive correlation (Pearson r = 0.71), indicating that semantic embedding models align closely with structured ontological representations. This correlation is visualized in Figure 29, which highlights the consistency between sentence-level meaning and ontology-based abstraction across the dataset.

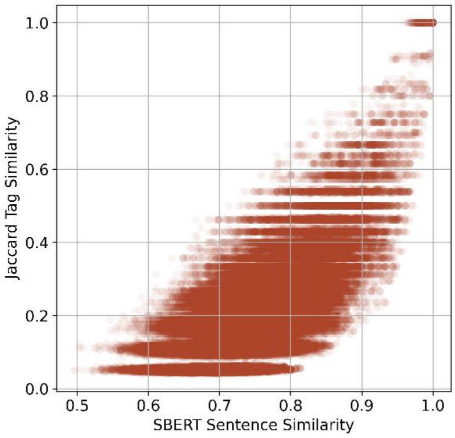


Figure 29: The correlation between SBERT and Jaccard similarities.

6.4.2 Logical-level similarity

In the evaluation of logical-level scenario similarity, scenario parameters can be systematically classified into three primary categories: dynamic, scenery, and environmental. This categorization aligns with the structured approach found in SDL formats, particularly for scripted scenarios, where each actor's behaviour is explicitly defined through sequenced manoeuvre phases. Among these, dynamic parameters – such as speed, acceleration, relative distance, and manoeuvre type – carry the highest semantic importance, as they determine the behavioural trajectory, risk evolution, and functional objective of the test. In contrast, scenery, and environmental parameters (e.g., road geometry, weather conditions) set the operational context but have a secondary influence on the core functional behaviour.

The reliability of PSS as a similarity metric depends heavily on structural alignment between the scenarios. The metric is most meaningful when the scenarios share the same manoeuvre sequence structure – i.e., a similar number of phases and manoeuvre types (e.g., cut-in vs. cut-in, or overtaking vs. overtaking). Comparing scenarios across fundamentally different behaviours (such as a merging manoeuvre versus a pedestrian crossing) is less informative, as the internal semantics and testing objectives differ. In such cases, numerical overlaps in parameters might be misleading, failing to reflect true functional similarity.

For a practical example with PSS Calculation, we consider two logical SDL2 scenarios (Figure 30):

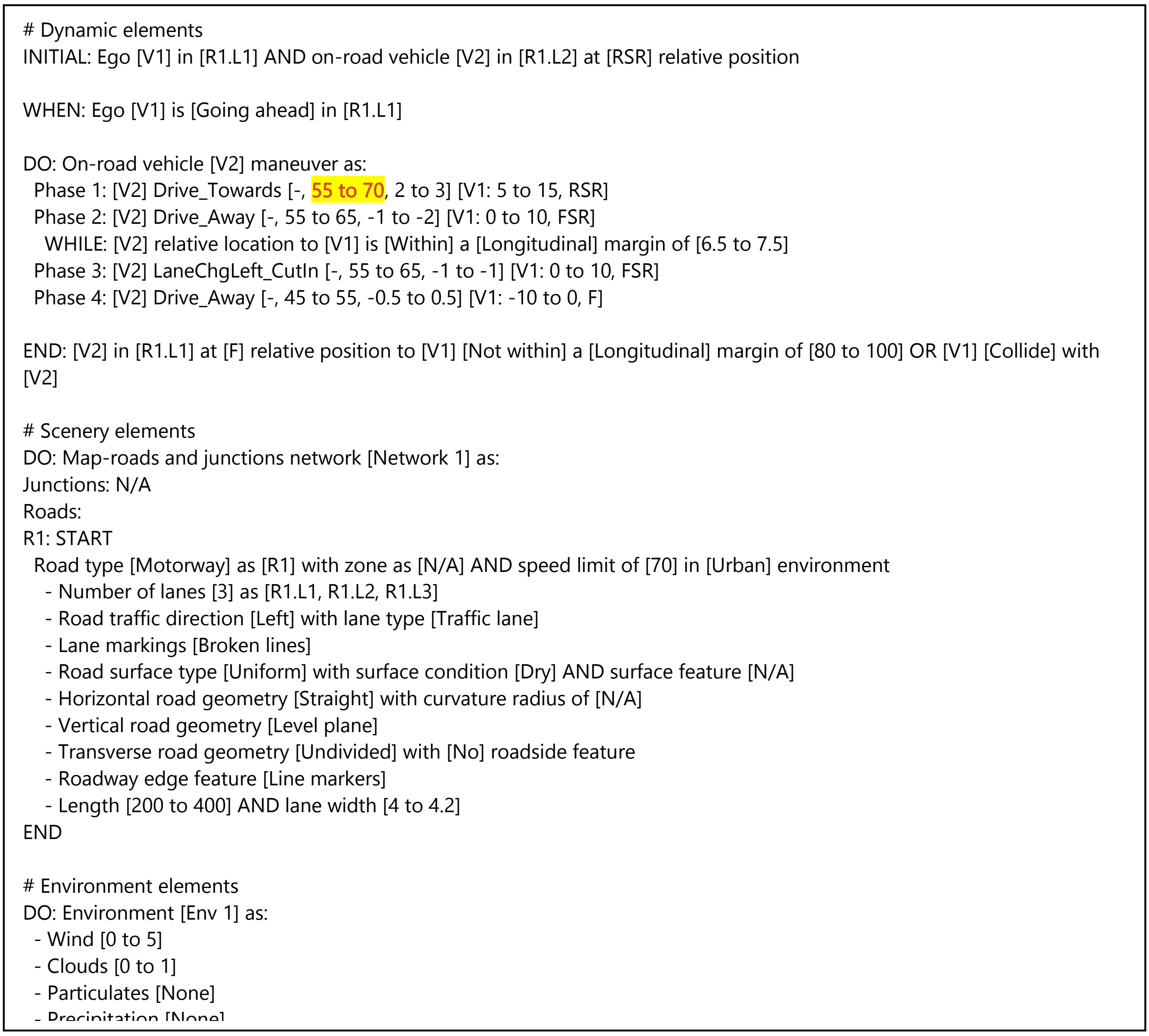
- An on-road vehicle other than the ego vehicle (labelled V2 in Figure 30) approaches speed range during Phase 1: ([55, 70]) km/h

Suppose we wish to compare it to a second scenario where:

- The approach speed is ([50, 60]) km/h

$$\mathrm{Overlap} = [55{,}60], \quad \mathrm{Union} = [50{,}70],$$

$$\mathrm{PSS}_{\mathrm{speed}} = \frac{5}{20} = 0.25.$$



```
# Dynamic elements
INITIAL: Ego [V1] in [R1.L1] AND on-road vehicle [V2] in [R1.L2] at [RSR] relative position

WHEN: Ego [V1] is [Going ahead] in [R1.L1]

DO: On-road vehicle [V2] maneuver as:
  Phase 1: [V2] Drive_Towards [-, 55 to 70, 2 to 3] [V1: 5 to 15, RSR]
  Phase 2: [V2] Drive_Away [-, 55 to 65, -1 to -2] [V1: 0 to 10, FSR]
    WHILE: [V2] relative location to [V1] is [Within] a [Longitudinal] margin of [6.5 to 7.5]
  Phase 3: [V2] LaneChgLeft_CutIn [-, 55 to 65, -1 to -1] [V1: 0 to 10, FSR]
  Phase 4: [V2] Drive_Away [-, 45 to 55, -0.5 to 0.5] [V1: -10 to 0, F]

END: [V2] in [R1.L1] at [F] relative position to [V1] [Not within] a [Longitudinal] margin of [80 to 100] OR [V1] [Collide] with
[V2]

# Scenery elements
DO: Map-roads and junctions network [Network 1] as:
Junctions: N/A
Roads:
R1: START
  Road type [Motorway] as [R1] with zone as [N/A] AND speed limit of [70] in [Urban] environment
    - Number of lanes [3] as [R1.L1, R1.L2, R1.L3]
    - Road traffic direction [Left] with lane type [Traffic lane]
    - Lane markings [Broken lines]
    - Road surface type [Uniform] with surface condition [Dry] AND surface feature [N/A]
    - Horizontal road geometry [Straight] with curvature radius of [N/A]
    - Vertical road geometry [Level plane]
    - Transverse road geometry [Undivided] with [No] roadside feature
    - Roadway edge feature [Line markers]
    - Length [200 to 400] AND lane width [4 to 4.2]
END

# Environment elements
DO: Environment [Env 1] as:
  - Wind [0 to 5]
  - Clouds [0 to 1]
  - Particulates [None]
  - Precipitation [None]
```

Figure 30: Logical scenario description (in SDL Level 2 format [109, 110]) used as an example for the similarity metric. The parameter of interest is highlighted.

Thus, even though the scenarios share similar a structure (Figure 30), the dynamic PSS between them is approximately 0.25, indicating a moderate level of similarity but highlighting clear differences in dynamic risk and timing.

6.4.3 Concrete-level similarity

Two different approaches are presented. The first approach focusses on the comparisons of the trajectories of the scenario actors, while the second approach calculates the similarity between two scenarios based on the most critical scene.

6.4.3.1 Concrete-level similarity based on trajectories

In this step, a set of nine logical-level scenarios was selected from the Safety Pool scenario database. These scenarios each involve two interacting entities and were categorized into three manoeuvre classes: Lane Change Left (LCL), Lane Change Right (LCR), and Stop (STP), with three scenarios per category. All scenarios were encoded in OpenSCENARIO format and executed using the Esmini OpenSCENARIO simulation engine, which was used to generate fully instantiated concrete trajectories for each actor over time.

The resulting spatiotemporal trajectories were then visualized and are shown in Figure 31. To quantify similarity between scenarios at the concrete level, we applied two trajectory-based metrics introduced in Section 5.4.3.1: DTW and the Hausdorff distance. These metrics were used to compute pairwise similarity between the nine scenarios, producing two corresponding similarity matrices, which are presented in Figure 32.

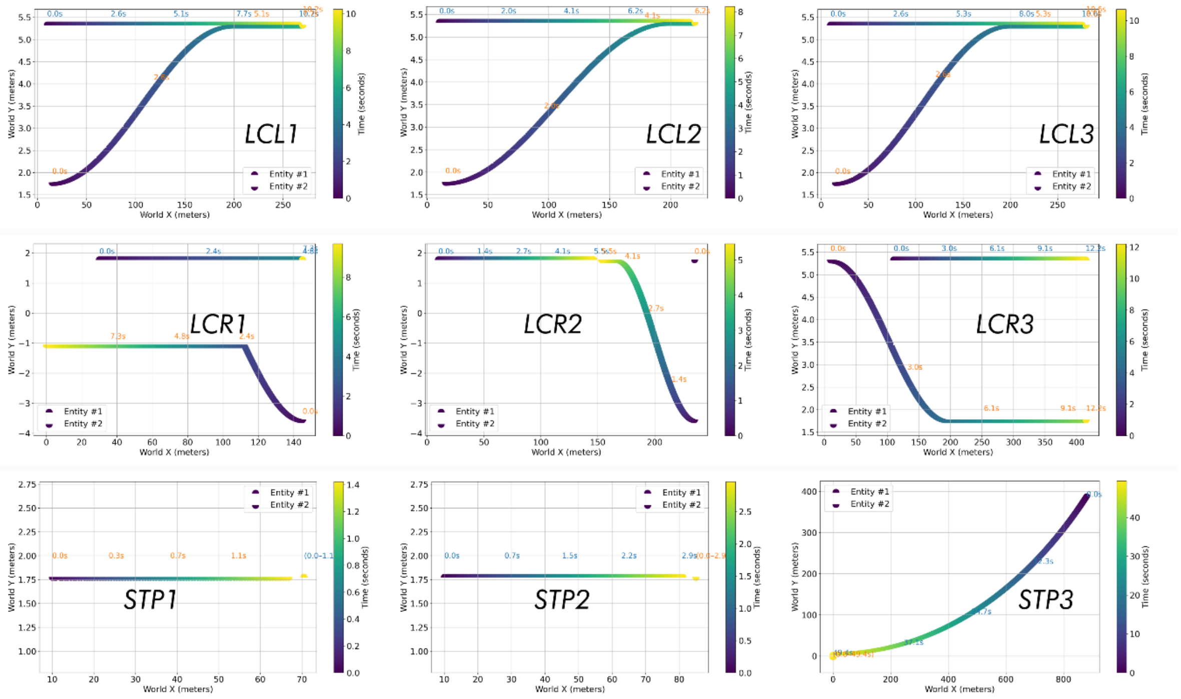


Figure 31: Spatial trajectories (x, y) of nine scenarios, coloured using a timestamp-based colormap to visualize temporal evolution across each manoeuvre.

As expected, both DTW and Hausdorff metrics yielded higher similarity values among scenarios within the same manoeuvre class, and lower similarity across different classes. This confirms that trajectory-based similarity effectively captures the behavioural structure of manoeuvre-specific categories. Notably, within the Stop (STP) group, one scenario – STP3, which takes place on an extended road segment – exhibited distinct trajectory characteristics. The DTW metric was particularly sensitive to this variation, successfully distinguishing STP3 from the other Stop scenarios based on timing and spatial dynamics, even though they share the same manoeuvre label.

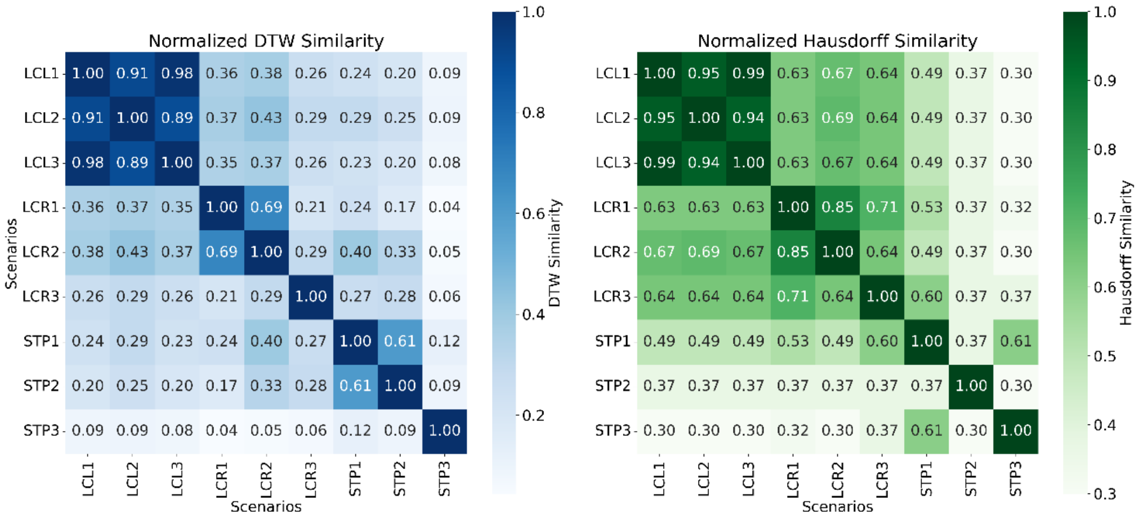


Figure 32: Similarity matrices computed using Dynamic Time Warping (DTW) and Hausdorff distance for the nine scenario trajectories. Higher similarity scores are observed within manoeuvre groups (LCL, LCR, STP), while inter-group comparisons yield lower similarity.

6.4.3.2 Concrete-level similarity based on most critical scene

The following example is based on the work in [112].The chosen application example is a scenario dataset of 1,000 scenarios, generated using an optimization methodology [113]. The optimization problem is aimed at discovering scenarios which are highly “unexpected” for the ego vehicle in its interaction with the chosen target vehicle. The purpose of using dissimilarity metrics for this application example is to extract a representative set of scenarios from this scenario dataset which score high on the “unexpectedness” metric but are also different from each other. These scenarios can then be used as a representative set for early testing of different versions of the ADS algorithms. The diversity requirement is unmet when dissimilarity metrics are not used, as the scenarios scoring highest on unexpectedness metric may be very similar to each other. This is because the optimization engine, upon finding a good scenario, will try minor variations around that scenario to find even ‘better’ scenarios.

The setup is shown in Figure 33. The ego vehicle and a target vehicle drive on fixed path(s) but can have trajectory variations, e.g., speed. Two other vehicles are in the scenario, which may directly or indirectly affect unexpectedness for the ego vehicle. These vehicles can have varied paths, as shown in the figure, as well as trajectory variations. Unexpectedness for the ego vehicle may occur for example due to target vehicle being obstructed from the ego vehicle view by Vehicle 3 or Vehicle 4, or if the target vehicle suddenly accelerates or brakes while crossing the intersection.

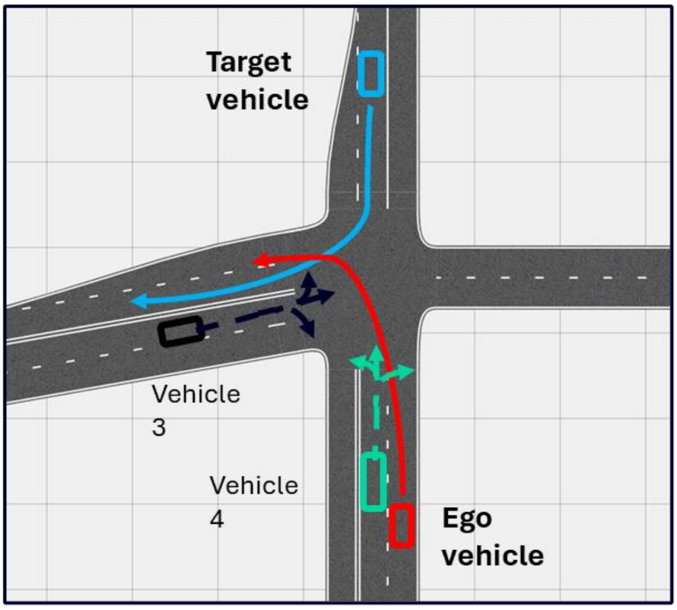


Figure 33: The problem space for the optimization problem. The ego and target vehicle have fixed paths with trajectory parameters. Vehicle 3 and Vehicle 4 have discrete set of possible paths as well as trajectory parameters.

Using the dissimilarity metric, clusters of similar scenarios are identified, after which the scenario with the best objective score in each cluster are extracted. Twelve clusters are identified based on discrete features only, while an additional sixteen clusters are identified by additionally using continuous features. Figure 34 shows how scenarios, within a cluster based on discrete features, is further clustered based on the continuous features. Figure 35 shows the scenarios with the highest objective score within their respective clusters.

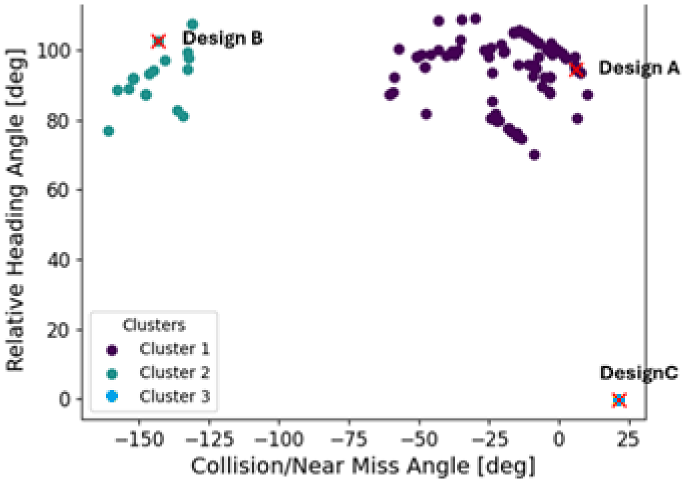


Figure 34: Sub-clustering of scenarios based on continuous features within one cluster as extracted based on discrete features. Design A, B, and C are the scenarios with the highest objective score in their respective clusters.

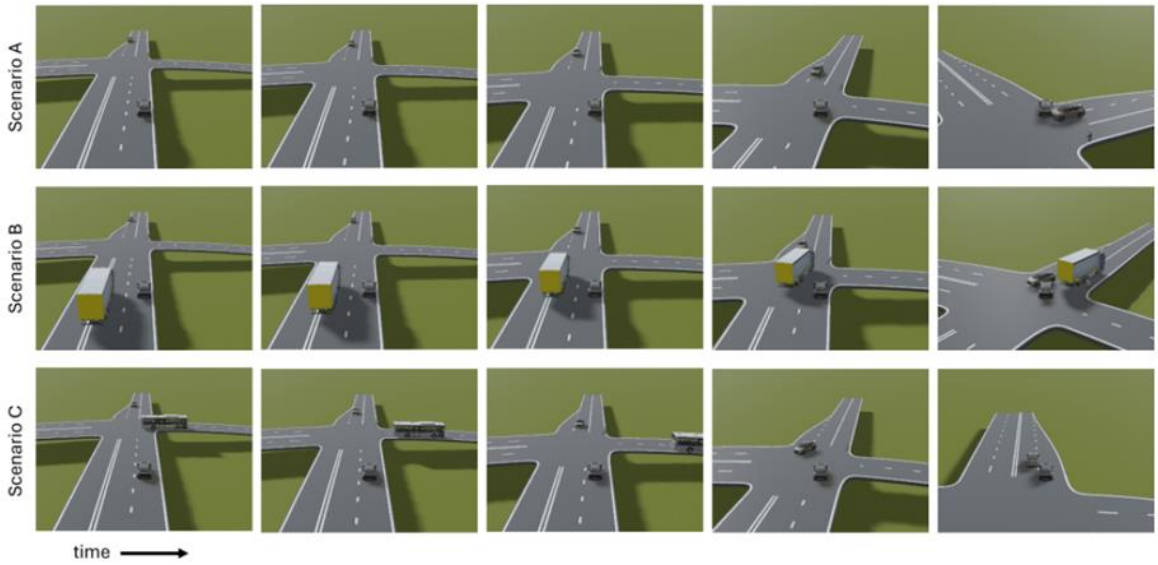


Figure 35: Three dissimilar scenarios which share similar discrete features but have different continuous features in the most safety-critical scene.

6.5 Coverage

In Section 5.5, two different approaches are presented for measuring coverage: coverage of an ODD by scenarios and coverage of the parameter space of a logical scenario. This section presents applications of these metrics.

6.5.1 ODD coverage by scenarios

This section describes applications of the tag-based coverage, time-based coverage, actor-based coverage, and actor-over-time-based coverage.

6.5.1.1 Tag-based coverage

With a structured taxonomy in place and shared keywords consistently used across the SCDB, it becomes possible to perform systematic tag-based coverage analysis. This analysis helps identify whether the SCDB comprehensively represents relevant attributes across the ODD and behaviour specifications or if there are gaps or underrepresented conditions.

Let’s assume that the SCDB contains 1,000 scenarios and follows the ISO 34503 taxonomy which provides a standardized hierarchical structure to represent ODD attributes. While a high-level view might suggest that rainfall is sufficiently represented across the dataset, a deeper analysis, by drilling down into lower levels of the taxonomy, reveals that no scenarios capture "Extreme rain" as an attribute. Such a gap could have significant implications for safety validation if the automated system is intended to operate under such conditions.

Table 5 presents an evaluation of weather condition coverage based on this structured, using a tag-based approach. While a high-level view might suggest that rainfall is sufficiently represented across the dataset, a deeper analysis, by drilling down into lower levels of the taxonomy, reveals that no scenarios capture "Extreme rain" as an attribute. Such a gap could have significant implications for safety validation if the automated system is intended to operate under such conditions.

Table 5: Tag-based coverage analysis of weather conditions.

Weather Attribute	Number of scenarios tagged	Coverage in SCDB
Ambient air temperature	1000	100%
Wind	1000	100%
No wind	80	8%
Low wind	543	54%
Medium wind	287	28%
High wind	90	9%
Rainfall	670	67%
Rainfall type	20	2%
Dynamic	7	<1%
Convective	11	1%
Orographic	2	<1%
Rainfall intensity	670	67%
No rain	62	6%
Light rain	385	38%
Medium rain	223	22%
Extreme rain	0	0%
Snowfall	30	3%
No snow	0	0%
Light snow	3	<1%
Moderate snow	26	2%
Heavy snow	1	<1%

Moreover, this tag-based coverage analysis can be extended to directly compare against a specified ODD definition. By aligning the tag set used in the SCDB with the attributes listed in the ODD input, one can assess whether the database can cover the system's operational requirements. For example, if the input ODD definition includes the ability to operate in conditions with extreme rainfall, the coverage analysis clearly highlights where new scenario development is necessary. It helps safety engineers understand the conditions that the system may not be exposed to in virtual testing and helps scenario developers understand exactly where the SCDB needs to be extended to meet ODD requirements. Ultimately, this method enhances the transparency and accountability of the SAF by offering quantifiable insight into how well the SCDB aligns with the operational boundaries of the system under test.

To also illustrate the tag coverage metric $\text{Coverage}_{\text{tag}}$, a different experiment has been set up. For this, the same data as for the application of the exposure metrics (Section 6.3) has been utilized. Table 6 presents the 10 scenario categories considered in this study, summarizing the activities of the ego vehicle and the main actors, who are essential for the scenario to occur. Other actors may also participate in the scenario, such as a vehicle overtaking the ego vehicle in the leading vehicle cruising scenario. The scenarios are automatically extracted based on the activities of the ego vehicle and the main actors, following the approach outlined in. Table 6 also shows the number of scenarios found for each scenario category.

Table 6: Description of the 10 scenario categories that are considered for the application of coverage metrics.

Symbol	Name	Count
C_1	Leading vehicle cruising	102,308
C_2	Leading vehicle accelerating	22,296
C_3	Leading vehicle decelerating	20,351
C_4	Approaching slower vehicle	5,052
C_5	Cut-in in front of ego vehicle	2,992
C_6	Cut-out in front of ego vehicle	3,069
C_7	Changing lane with vehicle behind	2,156
C_8	Merging into an occupied lane	819
C_9	Ego vehicle overtaking vehicle	38,147
C_{10}	Vehicle overtaking ego vehicle	40,307

Eighteen tags are considered for coverage, as shown in Table 7 and Table 8. The first two tags correspond to the vehicle types in the HighD data set. Tags L_3 to L_{10} refer to a vehicle's initial position relative to the ego vehicle. Tags L_{11} and L_{12} indicate if an actor is much slower or faster than the ego vehicle. The other tags describe longitudinal (L_{13} to L_{15}) and lateral (L_{16} to L_{18}) activities of surrounding vehicles. For example, tag L_1 is used once for 5 cars around the ego vehicle.

Table 7 and Table 8 list the scenarios containing each tag. Some tags, such as tag in scenarios C_1, C_2, C_3, and C_6, are inherently included. Figure 36 shows tag-based coverage resulting from Table 7 and Table 8, revealing how coverage varies with different tag sets. For any tag set, $\text{Coverage}_{\text{Tag}}(10) = 1$, meaning each tag appears in at least 10 scenarios per scenario category. When considering tags L_1, L_2, and L_{10} to L_{14}, $\text{Coverage}_{\text{Tag}}(100) = 1$. As n increases, $\text{Coverage}_{\text{Tag}}(n)$ decreases. Low counts for SCs C_7 and C_8 lead to fewer occurrences of their tags.

Table 7: Counts of tags per scenario category with the corresponding scenario categories listed in Table 6. This table continuous with Table 8.

Symbol	Tag	C_1	C_2	C_3	C_4
L_1	Car	102,111	22,292	20,341	5,050
L_2	Truck	81,475	19,406	17,454	4,234
L_3	Same lane in front	102,308	22,296	20,351	5,052
L_4	Same lane rear	37,281	11,248	14,377	2,386
L_5	In front left lane	70,385	17,664	15,934	3,860
L_6	In front right lane	49,139	9,190	8,871	2,443
L_7	At side left lane	4,852	2,052	1,552	228
L_8	At side right lane	4,850	1,284	1,162	201
L_9	Rear left lane	32,294	12,730	13,183	2,243
L_{10}	Rear right lane	31,462	8,052	8,741	1,777
L_{11}	Slower ($\Delta v < -5\ \mathrm{m/s}$)	54,750	13,873	14,138	4,021
L_{12}	Faster ($\Delta v > 5\ \mathrm{m/s}$)	41,061	9,032	8,046	1,798
L_{13}	Cruising	102,308	22,296	20,351	5,043
L_{14}	Accelerating	57,081	22,296	7,652	2,554
L_{15}	Decelerating	58,144	9,107	20,351	3,419
L_{16}	Keeping lane	102,308	22,296	20,351	5,052
L_{17}	Changing lane left	6,771	1,405	1,759	384
L_{18}	Changing lane right	4,154	1,127	982	339

Table 8: Continuation of Table 7.

Symbol	C_5	C_6	C_7	C_8	C_9	C_{10}
L_1	2,992	3,067	2,147	819	37,996	40,305
L_2	2,273	2,624	1,915	734	34,652	31,999
L_3	1,188	3,069	834	480	29,295	29,171
L_4	1,339	1,597	1,006	351	23,666	24,913
L_5	1,857	2,377	980	578	37,850	14,773
L_6	2,208	1,187	820	476	12,388	32,625
L_7	161	205	40	17	870	1,151
L_8	166	95	44	20	1,216	12,83
L_9	1,245	1,750	1,205	366	24,979	12,760
L_{10}	1,387	807	721	275	11,063	37,005
L_{11}	1,348	2,369	1,528	591	35,107	7,480
L_{12}	1,831	957	931	403	8,124	37,569
L_{13}	2,964	3,051	2,142	816	37,935	39,660
L_{14}	1,610	1,481	1,260	516	21,270	24,039
L_{15}	1,794	1,833	1,287	607	21,132	24,804
L_{16}	2,992	3,068	2,156	819	38,147	40,307
L_{17}	2,090	2,101	32	13	2,545	2,668
L_{18}	987	1,073	12	15	1,794	1,741

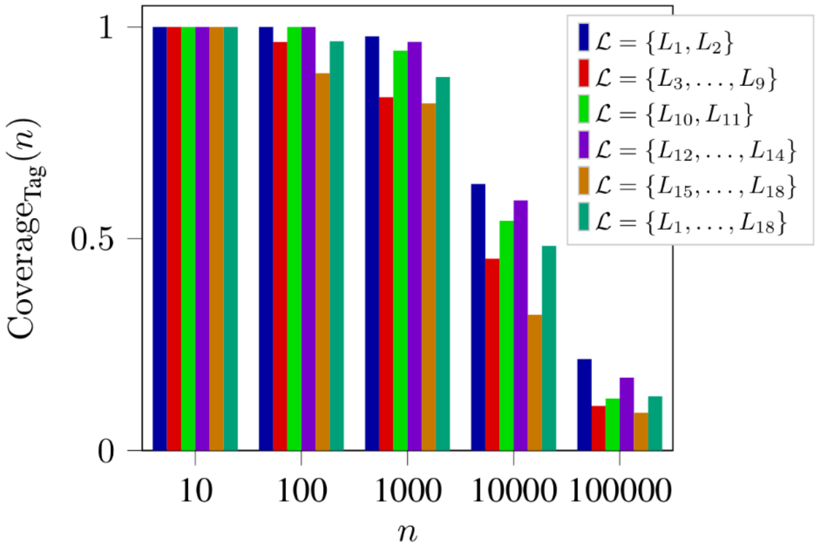


Figure 36: Results of the tag-based coverage.

6.5.1.2 Time-based coverage

To calculate time-based coverage, we treat each dataset's time instants with an ego vehicle separately, then combine the results. Figure 37 shows that around 75 % of these instances are covered by at least one scenario, leaving 25 % uncovered. This gap needs investigation to check for missing important scenario categories. In this 25 %, there might be no actor fitting any of the actors described by the scenario categories or no other actor present. This is confirmed when we add the SC "ego vehicle has no leading vehicle", as this would result in $\mathrm{Coverage}_T(1) = 1$.

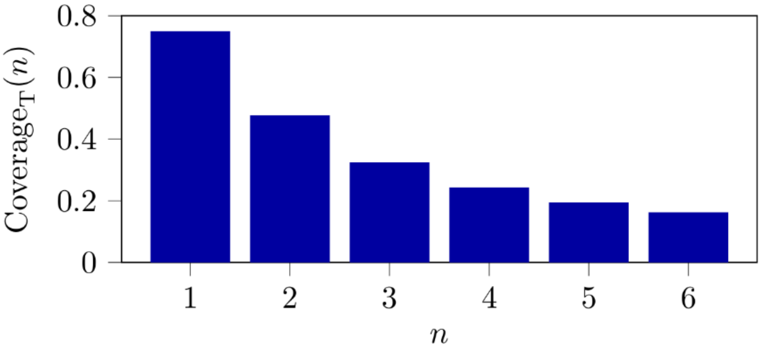


Figure 37: Results of the time-based coverage.

6.5.1.3 Actor-based coverage

Figure 38 illustrates actor-based coverage for different actor sets. Imagine a box around the ego vehicle; $\mathcal{A}$ includes all actors within this box at any time. By varying the box size, different values of $\mathrm{Coverage}_\mathrm{A}(\mathcal{A})$ are shown. For $\mathcal{B}$ (actors part of a scenario), only main actors that are necessary for the scenario annotation are considered. For example, for a scenario of SC “leading vehicle cruising”, only the ego vehicle and the leading vehicle are part of $\mathcal{B}$, while all other surrounding vehicle are not. Including all scenario actors would make actor-based coverage practically similar to time-based coverage.

Considering only actors ahead of the ego vehicle in its lane (blue solid line), $\mathrm{Coverage}_\mathrm{A}(\mathcal{A}) = 1$ for vehicles within 10 m. In other cases, the coverage is lower. Further investigation is needed to understand why some nearby actors are not main actors, even if they are within 15 m. In this study, non-main actors are often vehicles in front of the vehicle the ego follows, particularly in traffic jams.

Changing the width of the box around A substantially drops $\mathrm{Coverage}_\mathrm{A}(\mathcal{A})$. Vehicles in adjacent lanes are considered only if overtaking or being overtaken by the ego vehicle (C_9 and C_{10}). Vehicles two lanes away are not considered main actors for any SC, resulting in lower green lines compared to red lines. Extending the box towards the back of the ego vehicle also lowers actor-based coverage, explained by only one SC (C_8) considering a main actor behind the ego for the entire scenario duration.

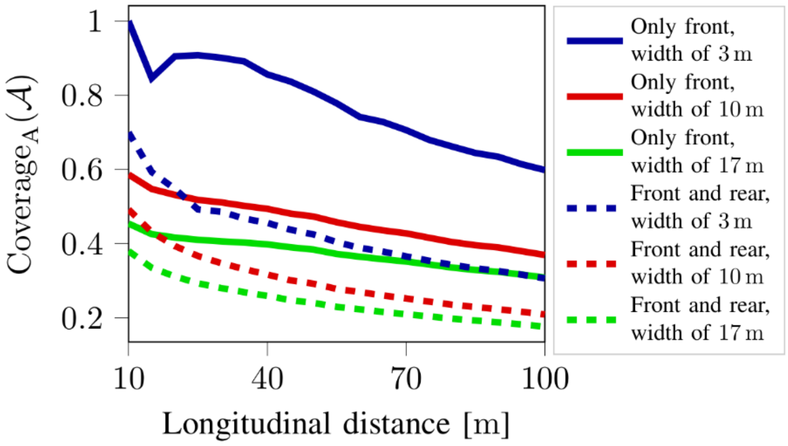


Figure 38: Results of the actor-based coverage. For the actor set $\mathcal{A}$, every actor is considered that is at some point in time within a certain longitudinal distance of the ego vehicle, varying from 10 m to 100 m (x-axis), and within a lateral distance, varying between 1.5 m (blue), 5.0 m (red), and 8.5 m (green). For the solid lines, $\mathcal{A}$ only contain actors in front, while for the dashed lines, $\mathcal{A}$ also contain rear actors within the specified longitudinal distance.

6.5.1.4 Actor-over-time-based coverage

Figure 39 shows actor-over-time-based coverage, similar to Figure 38. For actors in the same lane ahead of the ego vehicle (blue lines), results are almost identical. This suggests these actors are main participants when within the imaginary box. Other lines indicate coverage is slightly lower, meaning that relevant actors are not covered for the entire time they are within the box's boundaries.

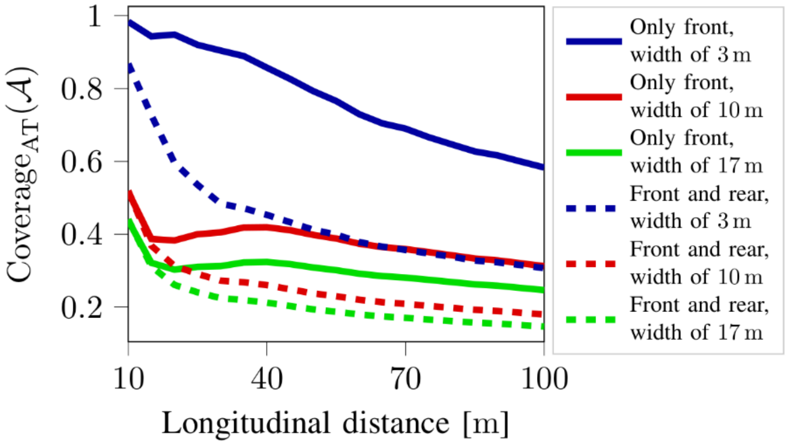


Figure 39: Results of the actor-over-time-based coverage. See Figure 38 for a further explanation.

6.5.2 Parameter space coverage

To illustrate the parameter space coverage, we base our study on the ADScene dataset, processed through multiple steps to extract structured driving scenarios from raw sensor data (cf. Figure 40). The pipeline includes filtering anomalies (e.g., ghost objects), sensor fusion (e.g., ID unification, speed calculation), and trajectory predictions (e.g., lane changes, as in [114]). From this, a high-level data model is built to represent road actors, infrastructure, ego dynamics, and context.

We define expert-based rules using high-level parameters (HLPs) to trigger scenario identification, leading to a structured scenario catalogue. From this catalogue, we selected three real-world scenarios to evaluate our method.

For each scenario, a primary variable set defines its core structure, and two additional variables measure how the dimensionality affects coverage. These are captured at a key moment (e.g., braking, lane change) to assess variable completeness and scenario description quality.

The following scenarios are considered:

- UC1 – "Closest In-Path Vehicle (CIPV) (Brakes" (Figure 41): Sudden braking by the leading vehicle. Main metrics include ego speed, obstacle distance, deceleration, and time to collision. Despite a low recall (40 %), the scenario appears frequently. Supplementary variables (e.g., lateral position) help assess information gaps in a broader space.
- UC2 – "Ego Lane Change" (Figure 42): Lateral manoeuvre without obstacle. Main variables: ego speed, lateral speed, manoeuvre duration. With 83 % recall and many instances, this scenario serves as a strong benchmark.
- UC3 – "Lane Change to Overtake" (Figure 43): Overtaking manoeuvre with focus on time to collision and relative distances. Although recall is low (45 %) and data is limited, it is key for testing robustness under rare conditions.

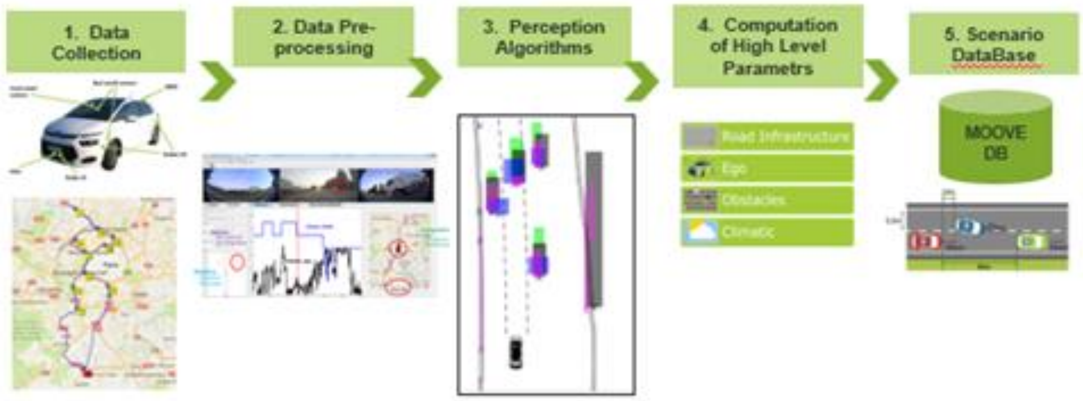


Figure 40: Data workflow of ADScene.

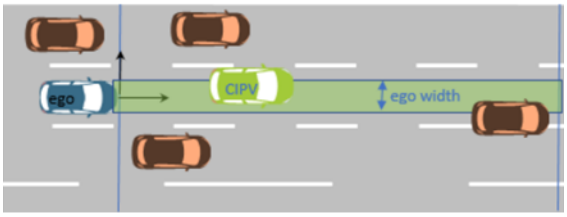


Figure 41: Scenario "CIPV brakes" (UC1).

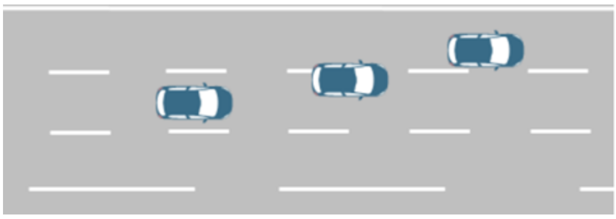


Figure 42: Scenario "ego change lane" (UC2).

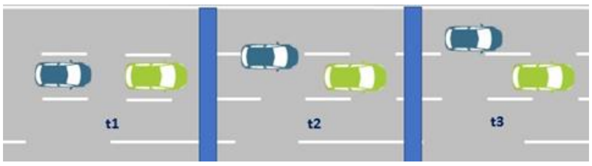


Figure 43: Scenario "ego change lane to overtake" (UC3).

Our approach assumes a Gaussian latent space. We validate this using the Mardia test [115], assessing skewness and kurtosis for each use case, see Table 9. High p-values across all UCs suggest alignment with a 2D Gaussian model. This supports the relevance of our method, especially for UC1 and UC2 with larger sample sizes. UC3's smaller dataset also passes the test, though with lower statistical power.

Table 9: Results of Mardia test for assessing normality of the latent space for each scenario.

Scenario	p-value for skewness	p-value for Kurtosis
1	0.99	0.58
2	0.99	0.67
3	0.99	0.76

We estimate space coverage based on different sample sizes, see Table 10. As expected, coverage decreases as required sample size increases. UC3 yields the lowest rates due to its small sample and low recall. Between UC1 and UC2, both with similar sample sizes, UC2 shows higher completeness, which aligns with its better recall and more consistent data.

Table 10: Coverage rate of scenarios in percentage.

Prescribed sample size	Scenario 1	Scenario 2	Scenario 3
900	100	100	67.51
1,600	98.94	99.94	63.62
2,500	97.24	98.52	58.52
3,600	96.89	97.25	48.22
4,900	95.47	96.08	32.94
6,400	91.80	94.48	32.94
8,100	86.60	91.96	27.38

A key finding appears at $n = 8{,}100$, where UC1 and UC2 show a 5% difference in coverage. To confirm, we downsample to 5,000 samples. Results remain consistent: UC2 retains better coverage across all tested values of, see Figure 44.

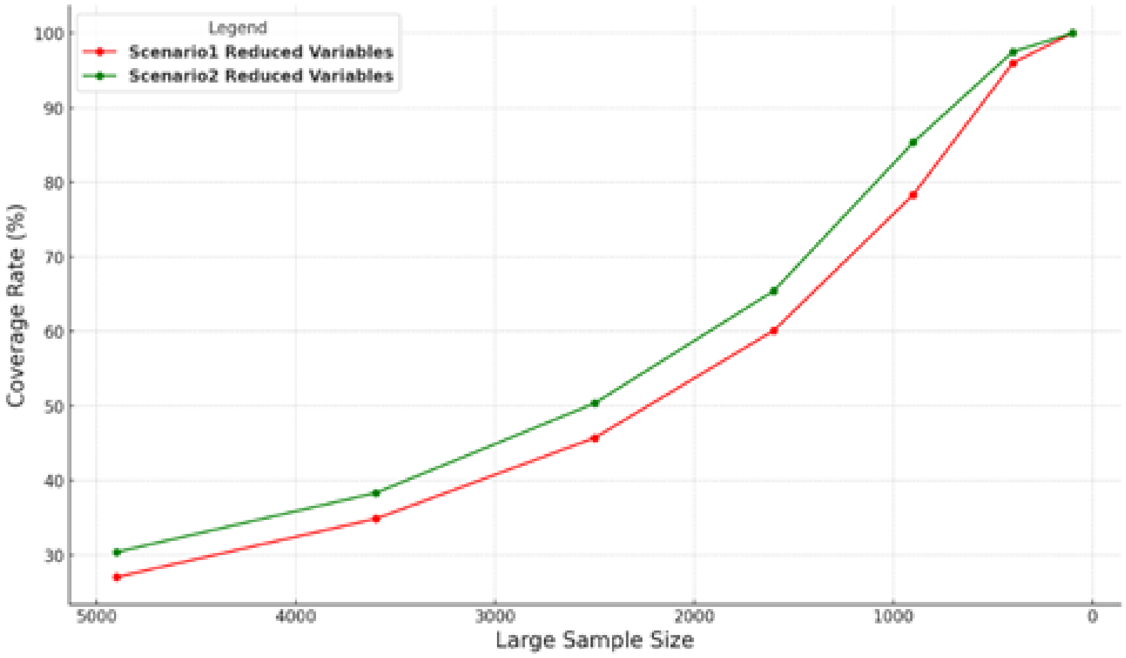


Figure 44: Comparison of coverage rate between UC1 and UC2 for different number of samples after sample reduction.

We also test coverage after removing supplementary variables, see Figure 45. Fewer variables reduce latent dimensionality, leading to higher coverage – this validates our method's sensitivity to the effective complexity of the scenario description.

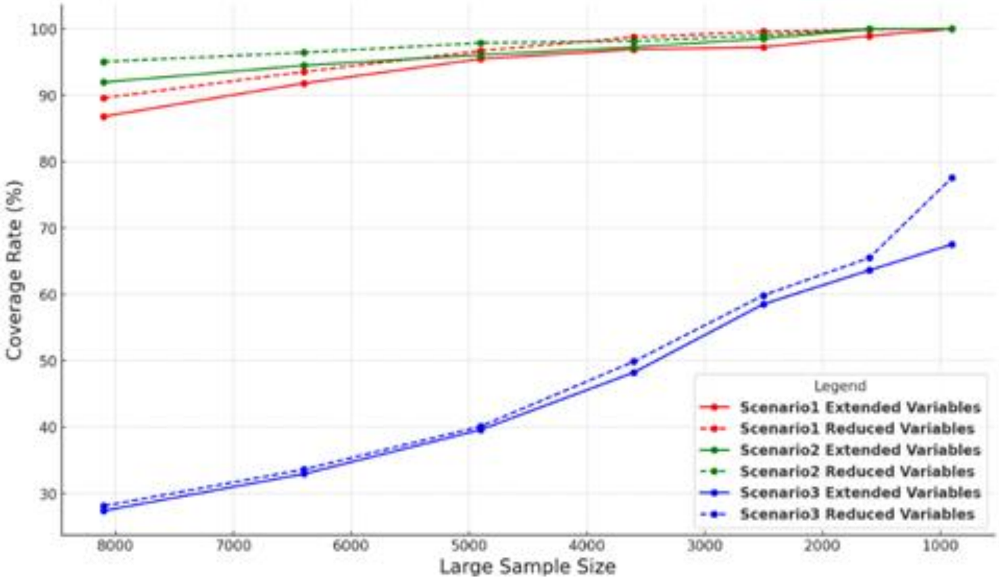


Figure 45: Impact of variable removal on scenario coverage rates.

In conclusion, our method provides a coherent framework to evaluate scenario completeness under varying conditions: data size, variable count, and recall. It offers a mathematically grounded tool to assess gaps in scenario datasets, supporting strategic planning for data collection in real-world applications.

It also holds promise for extension to raw time-series, infrastructure data, and integration into adaptive ODD-specific scenario generation workflows. Ultimately, this contributes to more cost-efficient and representative testing campaigns.

7 CONCLUSIONS AND FUTURE DIRECTIONS

Summary of key findings

This deliverable introduced a comprehensive framework of **quality metrics** to evaluate the **contents of Scenario Databases (SCDBs)**, which are essential components of the **Safety Assurance Framework (SAF)** developed within the SUNRISE project. These metrics were organized into several categories – **testing purpose**, **scenario description (incl. completeness)**, **scenario exposure**, **(dis)similarity**, **coverage**, and **general SCDB metrics** (see Chapter 3). They were grounded in an extensive literature review (Chapter 4) and further developed within SUNRISE (Chapter 5), providing practical and scalable methods for quantifying scenario quality across multiple dimensions. These metrics were also validated and demonstrated through partner applications using real-world datasets and SCDBs (Chapter 6), supporting their usability and effectiveness in a variety of test and validation contexts.

Alignment with project goals

The metrics presented in this deliverable are central to achieving the SUNRISE project's goal of enabling **safe, efficient, and harmonised testing and validation of Connected, Cooperative, and Automated Mobility (CCAM) systems**. Specifically, they serve as core enablers for the Safety Assurance Framework (SAF) (see Chapter 2), which aims to consolidate diverse validation approaches under a unified methodology. By providing standardised and traceable quality indicators, these metrics enhance the ability to evaluate the robustness and representativeness of test scenarios and their alignment with Operational Design Domains (ODDs).

Actionable outcomes

The deliverable identifies several key actionable outcomes for project stakeholders:

- The scenario relevance and criticality metrics (Chapter 5.1) allow testers to identify and prioritise high-risk or safety-critical scenarios, ensuring that validation focuses on the most impactful driving situations.
- The scenario description and completeness metrics (Chapter 5.2) verify that scenarios meet required detail levels for simulation and validation.
- The exposure metrics (Section 5.3) enable quantification of exposure for risk estimations.
- The scenario (dis)similarity metrics (Chapter 5.4) enable database refinement by removing redundant scenarios and ensuring a diverse and representative scenario set.
- The coverage metrics (Chapter 5.5), including tag-based, parameter-space, and actor-over-time coverage, support strategic scenario selection, scenario generation, and gap analysis.

These outcomes are ready for integration into SCDB tools and can be adopted by partners and stakeholders as standard procedures for database assessment and improvement.

Linkage to other project tasks

Deliverable D5.3 is closely connected with several other SUNRISE deliverables and tasks:

- Deliverable D2.3 (Definition of SAF) incorporates many of the metrics defined here into the broader SAF safety argumentation and workflow (see Chapter 2).
- Deliverable D3.4 [1] (Subspace Creation Methodology) leverages the coverage and similarity metrics introduced in this document for scenario generation and scenario space exploration (Chapter 5.4 and 5.5).
- Deliverable D5.1 [2] (Requirements for Data Framework and SCDB content) is substantiated by this work, as the metrics in D5.3 offer a quantitative basis to assess compliance with SCDB requirements, particularly the metrics outlined in Section 5.2 of D5.3.

Stakeholder relevance

The metrics developed in this deliverable are valuable to a wide range of internal and external stakeholders:

- SCDB owners and developers can apply the metrics from Chapter 4 and 5 to validate and improve database content, ensuring fitness for use across a wide variety of applications (Chapter 6).
- Tool developers can integrate the metrics into SCDB interfaces, APIs, and GUIs, enhancing transparency and traceability of scenario quality assessments.
- Test engineers and validation authorities benefit from metrics that enable rigorous, repeatable, and explainable scenario-based testing aligned with regulatory frameworks and real-world requirements.
- Policy makers, regulators, and independent assessors, like consumer testing organisations, can use the metrics to establish objective criteria for type approval processes and future standardisation efforts.

Future work

Building on the findings of this deliverable, several directions for future research and development are proposed:

- Standardisation of the defined metrics across the European CCAM validation landscape, facilitating international harmonisation and interoperability.
- Integration of metrics into automated scenario generation tools and AI-based test pipelines, improving the efficiency and coverage of test scenario sets.
- Establishing appropriate pass/fail thresholds for the proposed metrics. While current metrics enable comparisons between different scenarios or sets of scenarios, defining specific thresholds for compliance with requirements poses a challenge. For example, provided that 100 % coverage may be unfeasible, determining the minimum acceptable percentage that should be reached is essential. Future research could aim to identify these thresholds and ensure they align with compliance standards.

Key Recommendations

Based on the deliverable’s findings, the following recommendations are proposed:

- SCDB stakeholders should apply the provided metrics as a baseline for quality control, with emphasis on the scenario description metrics.
- SUNRISE partners should align SCDB content evaluation with the SAF workflows by using exposure, relevance, similarity, and coverage metrics to guide scenario selection and testing.
- Tool developers are encouraged to implement the metrics in analysis dashboards to assist test engineers in selecting balanced and effective scenario sets.
- Standardization organisations and regulators should consider formal adoption of these metrics as part of future CCAM validation protocols, contributing to a coherent and harmonised European testing ecosystem.

Closing Remarks

This deliverable represents a significant step toward formalising the **quality evaluation of SCDB contents**, a critical pillar in the development of a European **Safety Assurance Framework** (SAF) for CCAM systems. By providing stakeholders with explainable, quantifiable, and actionable metrics, SUNRISE contributes to safety assurance in the context of automated mobility.

8 REFERENCES

[1] J. Beckmann, J. M. Torres Camara, E. Kaynar, E. de Gelder, E. Daskalaki, B. Hillbrand, T. Amler, A. Thorsén, P. Irvine, M. Kirchengast, T. Menzel, F. Hadj Selem, and X. Zhang, “SUNRISE D3.4: Report on subspace creation methologogy,” European Union, Tech. Rep., 2025, in preparation.

[2] F. Alakkad, X. Xhang, S. Khastgir, E. de Gelder, S. van Montfort, J. Lorente Mallada, E. van Hassel, M. Grabowski, J. Beckmann, X. Boabén, A. Nogués, S. Vidal, T. Menzel, I. Panagiotopoulos, and A. Ballis, “SUNRISE D5.1: Requirements for CCAM safety assessment data framework content,” Tech. Rep., 2024.

[3] O. Thorsén, Anders Bartels, C. Berger, T. Bouraffa, C. Kaya, M. Muro, A. Bolovinou, I. Panangiotopoulos, A. Farooqui, M. Skoglund, P. Stålberg, D. Becker, E. Arnoux, O. Op den Camp, P. Ben Nejma, GhadaIrvine, and A. Bruto da Costa, “SUNRISE D3.1: Report on baseline analysis of existing methodology,” Tech. Rep., 2024.

[4] ECE/TRANS/WP.29/2022/59/Rev.1, “Proposal for the 01 series of amendments to un regulation no. 157 (automated lane keeping systems),” Standard, 2022. [Online]. Available: https://unece.org/sites/default/files/2022-05/ECE-TRANS-WP.29-2022-59r1e.pdf

[5] ISO/FDIS 34505, “Road Vehicles – Test scenarios for automated driving systems – Scenario evaluation and test case generation,” Standard, 2024. [Online]. Available: https://www.iso.org/-standard/78954.html

[6] E. de Gelder and O. Op den Camp, “Tagging real-world scenarios for the assessment of autonomous vehicles,” in *38th FISITA World Congress*, no. F2020-PIF-048, 2020. [Online]. Available: https://arxiv.org/abs/2012.01081

[7] B. Huber, S. Herzog, C. Sippl, R. German, and A. Djanatliev, “Evaluation of virtual traffic situations for testing automated driving functions based on multidimensional criticality analysis,” in *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–7.

[8] D. Baumann, R. Pfeffer, and E. Sax, “Automatic generation of critical test cases for the development of highly automated driving functions,” in *IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, 2021, pp. 1–5.

[9] T. F. Koné, E. Bonjour, E. Levrat, F. Mayer, and S. Géronimi, “An approach to guide the search for potentially hazardous scenarios for autonomous vehicle safety validation,” *Applied Sciences*, vol. 13, no. 11, p. 6717, 2023.

[10] E. de Gelder, H. Elrofai, A. Khabbaz Saberi, O. Op den Camp, J.-P. Paardekooper, and B. De Schutter, “Risk quantification for automated driving systems in real-world driving scenarios,” *IEEE Access*, vol. 9, pp. 168953–168970, 2021.

[11] E. de Gelder, A. Khabbaz Saberi, and H. Elrofai, “A method for scenario risk quantification for automated driving systems,” in *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, 2019. [Online]. Available: https://www-esv.nhtsa.dot.gov/-Proceedings/26/26ESV-000129.pdf

[12] ISO 34502, “Road Vehicles – Test scenarios for automated driving systems – Engineering framework and process of scenario-based safety evaluation,” Standard, 2022. [Online]. Available: https://www.iso.org/standard/78951.html

[13] J. Cai, W. Deng, H. Guang, Y. Wang, J. Li, and J. Ding, “A survey on data-driven scenario generation for automated vehicle testing,” *Machines*, vol. 10, no. 11, p. 1101, 2022.

[14] J. Jansson, “Collision avoidance theory: With application to automotive collision mitigation,” Ph.D. dissertation, Linköping University Electronic Press, 2005.

[15] J. C. Hayward, “Near miss determination through use of a scale of danger,” Tech. Rep. TTSC-7115, 1972. [Online]. Available: https://onlinepubs.trb.org/Onlinepubs/hrr/1972/384/384-004.pdf

[16] W. Wachenfeld, P. Junietz, R. Wenzel, and H. Winner, “The worst-time-to-collision metric for situation identification,” in *IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 729–734.

[17] J. Eggert, “Predictive risk estimation for intelligent ADAS functions,” in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 711–718.

[18] M. M. Minderhoud and P. H. Bovy, “Extended time-to-collision measures for road traffic safety assessment,” *Accident Analysis & Prevention*, vol. 33, no. 1, pp. 89–97, 2001.

[19] A. Varhelyi, “Drivers’ speed behaviour at a zebra crossing: A case study,” *Accident Analysis & Prevention*, vol. 30, no. 6, pp. 731–743, 1998.

[20] B. L. Allen, B. T. Shin, and P. J. Cooper, “Analysis of traffic conflicts and collisions,” *Transportation Research Board*, vol. 667, pp. 67–74, 1978.

[21] M. Brannstrom, J. Sjoberg, and E. Coelingh, “A situation and threat assessment algorithm for a rear-end collision avoidance system,” in *IEEE Intelligent Vehicles Symposium*, 2008, pp. 102–107.

[22] B. Yue, S. Shi, S. Wang, and N. Lin, “Low-cost urban test scenario generation using microscopic traffic simulation,” *IEEE Access*, vol. 8, pp. 123398–123407, 2020.

[23] S. Cafiso, A. G. Garcia, R. Cavarra, and M. R. Rojas, “Crosswalk safety evaluation using a pedestrian risk index as traffic conflict measure,” in *3rd International Conference on Road safety and Simulation*, vol. 15, 2011.

[24] F. Cunto and F. F. Saccomanno, “Calibration and validation of simulated vehicle safety performance at signalized intersections,” *Accident analysis & prevention*, vol. 40, no. 3, pp. 1171–1179, 2008.

[25] M. Schreier, V. Willert, and J. Adamy, “An integrated approach to maneuver-based trajectory prediction and criticality assessment in arbitrary road environments,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2751–2766, 2016.

[26] M. Issler, Q. Goss, and M. I. Akbas, “Complexity evaluation of test scenarios for autonomous vehicle safety validation using information theory,” *Information*, vol. 15, no. 12, p. 772, 2024.

[27] T. Liu, C. Wang, Z. Yin, Z. Mi, X. Xiong, and B. Guo, “Complexity quantification of driving scenarios with dynamic evolution characteristics,” *Entropy*, vol. 26, no. 12, p. 1033, 2024.

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

[29] J. Zhou, L. Wang, and X. Wang, “Online adaptive generation of critical boundary scenarios for evaluation of autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 6372–6388, 2023.

[30] F. Gao, J. Duan, Y. He, and Z. Wang, “A test scenario automatic generation strategy for intelligent driving systems,” *Mathematical Problems in Engineering*, vol. 2019, no. 1, p. 3737486, 2019.

[31] Association for Standardization of Automation and Measuring Systems, “OpenSCENARIO,” 2021, accessed September, 2024. [Online]. Available: https://www.asam.net/standards/detail/-openscenario/

[32] E. de Gelder, J.-P. Paardekooper, A. Khabbaz Saberi, H. Elrofai, O. Op den Camp, S. Kraines, J. Ploeg, and B. De Schutter, “Towards an ontology for scenario definition for the assessment of automated vehicles: An object-oriented framework,” *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, pp. 300–314, 2022.

[33] C. Chang, D. Cao, L. Chen, K. Su, K. Su, Y. Su, F.-Y. Wang, J. Wang, P. Wang, J. Wei, G. Wu, X. Wu, H. Xu, N. Zheng, and L. Li, “MetaScenario: A framework for driving scenario data description, storage and indexing,” *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1156–1175, 2022.

[34] G. Bagschik, T. Menzel, A. Reschka, and M. Maurer, “Szenarien für Entwicklung, Absicherung und Test von automatisierten Fahrzeugen,” in *11. Workshop Fahrerassistenzsysteme. Hrsg. von Uni-DAS e. V*, 2017, pp. 125–135.

[35] T. Menzel, G. Bagschik, and M. Maurer, “Scenarios for development, test and validation of automated vehicles,” in *IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1821–1827.

[36] P. Arcaini, X.-Y. Zhang, and F. Ishikawa, “Less is more: Simplification of test scenarios for autonomous driving system testing,” in *IEEE Conference on Software Testing, Verification and Validation (ICST)*, 2022, pp. 279–290.

[37] C. Neurohr, L. Westhofen, M. Butz, M. Bollmann, U. Eberle, and R. Galbas, “Criticality analysis for the verification and validation of automated vehicles,” *IEEE Access*, vol. 9, pp. 18016–18041, 2021.

[38] E. de Gelder and O. Op den Camp, “How certain are we that our automated driving system is safe?” *Traffic Injury Prevention*, vol. 24, no. sup1, pp. S131–S140, 2023.

[39] J.-P. Paardekooper, S. Montfort, J. Manders, J. Goos, E. de Gelder, O. Op den Camp, A. Bracquemond, and G. Thiolon, “Automatic identification of critical scenarios in a public

dataset of 6000 km of public-road driving," in *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, 2019. [Online]. Available: https://www-esv.nhtsa.dot.gov/-Proceedings/26/26ESV-000255.pdf

[40] A. S. Hakkert, L. Braimaister, and I. Van Schagen, "The uses of exposure and risk in road safety studies," Tech. Rep. R-2002-12, 2002. [Online]. Available: https://swov.nl/sites/default/-files/publicaties/rapport/r-2002-12.pdf

[41] O. Gietelink, "Design and validation of advanced driver assistance systems," Ph.D. dissertation, Delft University of Technology, 2007. [Online]. Available: http://resolver.tudelft.nl/uuid:b2f0e7f6-6255-4932-8b5e-d3ef67cd81ec

[42] S. Feng, X. Yan, H. Sun, Y. Feng, and H. X. Liu, "Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment," *Nature Communications*, vol. 12, no. 748, pp. 1–14, 2021.

[43] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[44] Y.-C. Chen, "A tutorial on kernel density estimation and recent advances," *Biostatistics & Epidemiology*, vol. 1, no. 1, pp. 161–187, 2017.

[45] E. de Gelder, J.-P. Paardekooper, O. Op den Camp, and B. De Schutter, "Safety assessment of automated vehicles: How to determine whether we have collected enough field data?" *Traffic Injury Prevention*, vol. 20, no. S1, pp. 162–170, 2019.

[46] E/ECE/TRANS/505/Rev.3/Add.156, "Uniform provisions concerning the approval of vehicles with regard to automated lane keeping systems," Standard, 2021. [Online]. Available: https://-unece.org/sites/default/files/2021-03/R157e.pdf

[47] H. Nakamura, H. Muslim, R. Kato, S. Préfontaine-Watanabe, H. Nakamura, H. Kaneko, H. Imanaga, J. Antona-Makoshi, S. Kitajima, N. Uchida, E. Kitahara, K. Ozawa, and S. Taniguchi, "Defining reasonably foreseeable parameter ranges using real-world traffic data for scenario-based safety assessment of automated vehicles," *IEEE Access*, vol. 10, pp. 37743–37760, 2022.

[48] H. Muslim, S. Endo, H. Imanaga, S. Kitajima, N. Uchida, E. Kitahara, K. Ozawa, H. Sato, and H. Nakamura, "Cut-out scenario generation with reasonability foreseeable parameter range from real highway dataset for autonomous vehicle assessment," *IEEE Access*, vol. 11, pp. 45349–45363, 2023.

[49] E. de Gelder and O. Op den Camp, "A quantitative method to determine what collisions are reasonably foreseeable and preventable," *Safety Science*, vol. 167, p. 106233, 2023.

[50] B. Zhu, P. Zhang, J. Zhao, and W. Deng, "Hazardous scenario enhanced generation for automated vehicle testing based on optimization searching method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7321–7331, 2021.

[51] Z. Zhong, G. Kaiser, and B. Ray, "Neural network guided evolutionary fuzzing for finding traffic violations of autonomous vehicles," *IEEE Transactions on Software Engineering*, vol. 49, no. 4, pp. 1860–1875, 2022.

[52] L. Ries, P. Rigoll, T. Braun, T. Schulik, J. Daube, and E. Sax, "Trajectory-based clustering of real-world urban driving sequences with multiple traffic objects," in *IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 1251–1258.

[53] V. Nguyen, S. Huber, and A. Gambi, "SALVO: Automated generation of diversified tests for self-driving cars from existing maps," in *IEEE international conference on artificial intelligence testing (AITest)*, 2021, pp. 128–135.

[54] Q. Lin, W. Wang, Y. Zhang, and J. M. Dolan, "Measuring similarity of interactive driving behaviors using matrix profile," in *American Control Conference (ACC)*, 2020, pp. 3965–3970.

[55] J. Kerber, S. Wagner, K. Groh, D. Notz, T. Kühbeck, D. Watzenig, and A. Knoll, "Clustering of the scenario space for the assessment of automated driving," in *IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 578–583.

[56] F. Kruber, J. Wurst, and M. Botsch, "An unsupervised random forest clustering technique for automatic traffic scenario categorization," in *21st International conference on intelligent transportation systems (ITSC)*, 2018, pp. 2811–2818.

[57] T. A. Wheeler and M. J. Kochenderfer, "Critical factor graph situation clusters for accelerated automotive safety validation," in *IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 2133–2139.

[58] T. Zohdinasab, V. Riccio, A. Gambi, and P. Tonella, “Deephyperion: Exploring the feature space of deep learning-based systems through illumination search,” in *30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2021, pp. 79–90.

[59] H. Tian, Y. Jiang, G. Wu, J. Yan, J. Wei, W. Chen, S. Li, and D. Ye, “MOSAT: Finding safety violations of autonomous driving systems using multi-objective genetic algorithm,” in *30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2022, pp. 94–106.

[60] T. Braun, L. Ries, M. Hesche, S. Otten, and E. Sax, “Maneuver-based visualization of similarities between recorded traffic scenarios.” in *DATA*, 2022, pp. 236–244.

[61] A. Piziali, *Functional verification coverage measurement and analysis*. Springer Science & Business Media, 2007.

[62] R. Alexander, H. Hawkins, and D. Rae, “Situation coverage — a coverage criterion for testing autonomous robots,” Tech. Rep., 2015. [Online]. Available: https://eprints.whiterose.ac.uk/-88736/

[63] H. Araujo, M. R. Mousavi, and M. Varshosaz, “Testing, validation, and verification of robotic and autonomous systems: A systematic review,” *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 2, pp. 1–61, 2023.

[64] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, “Survey on scenario-based safety assessment of automated vehicles,” *IEEE Access*, vol. 8, pp. 87456–87477, 2020.

[65] P. Weissensteiner, G. Stettinger, S. Khastgir, and D. Watzenig, “Operational design domain-driven coverage for the safety argumentation of automated vehicles,” *IEEE Access*, vol. 11, pp. 12263–12284, 2023.

[66] C. Amersbach and H. Winner, “Defining required and feasible test coverage for scenario-based validation of highly automated vehicles,” in *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 425–430.

[67] F. Hauer, T. Schmidt, B. Holzmüller, and A. Pretschner, “Did we test all scenarios for automated and autonomous driving systems?” in *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 2950–2955.

[68] A. J. Alnaser, A. Sargolzaei, and M. I. Akbas, “Autonomous vehicles scenario testing framework and model of computation: On generation and coverage,” *IEEE Access*, vol. 9, pp. 60617–60628, 2021.

[69] E. de Gelder, O. Op den Camp, J. Broos, J.-P. Paardekooper, S. van Montfort, S. Kalisvaart, and H. Goossens, “Scenario-based safety assessment of automated driving systems,” Tech. Rep., 2024. [Online]. Available: https://www.tno.nl/en/newsroom/papers/scenario-based-safety-assessment/

[70] W. Wang, C. Liu, and D. Zhao, “How much data are enough? A statistical approach with case study on longitudinal driving behavior,” *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 2, pp. 85–98, 2017.

[71] L. Hartjen, R. Philipp, F. Schuldt, and B. Friedrich, “Saturation effects in recorded maneuver data for the test of automated driving,” in *13. Uni-DAS eV Workshop Fahrerassistenz und automatisiertes Fahren*, 2020, pp. 74–83. [Online]. Available: https://www.uni-das.de/images/-pdf/fas-workshop/2020/FAS_2020_HARTJEN.pdf

[72] C. Glasmacher, M. Schuldes, H. Weber, N. Wagener, and L. Eckstein, “Acquire driving scenarios efficiently: A framework for prospective assessment of cost-optimal scenario acquisition,” in *IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 1971–1976.

[73] E. de Gelder, M. Buermann, and O. Op den Camp, “Coverage metrics for a scenario database for the scenario-based assessment of automated driving systems,” in *IEEE International Automated Vehicle Validation Conference*, 2024.

[74] T. Laurent, S. Klikovits, P. Arcaini, F. Ishikawa, and A. Ventresque, “Parameter coverage for testing of autonomous driving systems under uncertainty,” *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 3, pp. 1–31, 2023.

[75] Z. Tahir and R. Alexander, “Coverage based testing for V&V and safety assurance of self-driving autonomous vehicles: A systematic literature review,” in *IEEE International Conference On Artificial Intelligence Testing (AITest)*, 2020, pp. 23–30.

[76] ISO 21448:2022, “Road Vehicles – Safety of the intended functionality,” Standard, 2022. [Online]. Available: https://www.iso.org/standard/77490.html

[77] SAE J3016, “Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles,” Tech. Rep., 4 2021.

[78] S. Babisch, C. Neurohr, L. Westhofen, S. Schoenawa, and H. Liers, “Leveraging the GIDAS database for the criticality analysis of automated driving systems,” *Journal of Advanced Transportation*, vol. 2023, no. 1, p. 1349269, 2023.

[79] W. Damm and R. Galbas, “Exploiting learning and scenario-based specification languages for the verification and validation of highly automated driving,” in *1st International Workshop on Software Engineering for AI in Autonomous Systems*, 2018, pp. 39–46.

[80] ISO 26262, “Road Vehicles – Functional Safety,” Standard, 2018. [Online]. Available: https://-www.iso.org/standard/68383.html

[81] B. A. Turlach, “Bandwidth selection in kernel density estimation: A review,” Tech. Rep., 1993. [Online]. Available: https://www.researchgate.net/publication/-2316108_Bandwidth_Selection_in_Kernel_Density_Estimation_A_Review

[82] A. Z. Zambom and R. Dias, “A review of kernel density estimation with applications to econometrics,” *International Econometric Review (IER)*, vol. 5, no. 1, pp. 20–42, 2013. [Online]. Available: http://www.era.org.tr/makaleler/13120083.pdf

[83] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. CRC press, 1986.

[84] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, 1992.

[85] D. Rezende and S. Mohamed, “Variational inference with normalizing flows,” in *International Conference on Machine Learning*, 2015, pp. 1530–1538. [Online]. Available: https://-proceedings.mlr.press/v37/rezende15.pdf

[86] L. Dinh, J. Sohl-Dickstein, and S. Bengio, “Density estimation using real NVP,” *arXiv preprint arXiv:1605.08803*, 2016.

[87] G. Papamakarios, T. Pavlakou, and I. Murray, “Masked autoregressive flow for density estimation,” *Advances in Neural Information Processing Systems*, vol. 30, pp. 1–10, 2017.

[88] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, “Neural spline flows,” in *Advances in Neural Information Processing Systems*, 2019. [Online]. Available: https://-proceedings.neurips.cc/paper_files/paper/2019/file/7ac71d433f282034e088473244df8c02-Paper.pdf

[89] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT-networks,” in *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[90] “Cosine similarity.” [Online]. Available: https://en.wikipedia.org/w/index.php?title=Cosine similarity&oldid=1163816200

[91] Z. Nussbaum, J. X. Morris, A. Mulyar, and B. Duderstadt, “Nomic embed: Training a reproducible long context text embedder,” *Transactions on Machine Learning Research*, 2025. [Online]. Available: https://openreview.net/forum?id=IPmzyQSiQE

[92] T. Braun, J. Fuchs, F. Reisgys, L. Ries, J. Plaum, B. Schütt, and E. Sax, “A review of scenario similarity measures for validation of highly automated driving,” in *IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 689–696.

[93] M. Müller, “Dynamic time warping,” *Information Retrieval for Music and Motion*, pp. 69–84, 2007.

[94] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, “Comparing images using the Hausdorff distance,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, 1993.

[95] B. B. Mahadikar, N. Rajesh, K. T. Kurian, E. Lefeber, J. Ploeg, N. van de Wouw, and M. Alirezaei, “Formulating a dissimilarity metric for comparison of driving scenarios for automated driving systems,” in *IEEE Intelligent Vehicles Symposium (IV)*, 2024, pp. 1091–1098.

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

[97] W. G. Najm, J. D. Smith, and M. Yanagisawa, “Pre-crash scenario typology for crash avoidance research,” Tech. Rep. DOT HS 810 767, 4 2007. [Online]. Available: https://rosap.ntl.bts.gov/-view/dot/6281/dot_6281_DS1.pdf

[98] E. de Gelder, O. Op den Camp, and N. de Boer, “Scenario categories for the assessment of automated vehicles,” Tech. Rep., 2020, version 1.7. [Online]. Available: http://cetran.sg/wp-content/uploads/2020/01/REP200121_Scenario_Categories_v1.7.pdf

[99] ISO 34503, “Road Vehicles – Test scenarios for automated driving systems – Taxonomy for operational design domain for automated driving systems,” Standard, 2023. [Online]. Available: https://www.iso.org/standard/78952.html

[100] ISO 34504, “Road Vehicles – Test scenarios for automated driving systems – Scenario categorization,” International Organization for Standardization, Standard, 2024. [Online]. Available: https://www.iso.org/standard/78953.html

[101] C. Glasmacher, H. Weber, and L. Eckstein, “Towards a completeness argumentation for scenario concepts,” in *IEEE Intelligent Vehicles Symposium (IV)*, 2024.

[102] F. Hadj Selem, G. Ben Nejma, W. Kheriji, L. Durville, R. M. C, S. Geronimi, and E. Arnoux, “Developing a methodology to assess data completeness of driving scenarios for testing autonomous vehicles: A focus on ODD-specific objectives,” in *Driving Simulation Conference*, 2024.

[103] F. Hadj Selem, G. Ben Nejma, W. Kheriji, L. Durville, and A. E, “Which scenarios must be tested for safety in automated driving: To cut testing budget,” in *30th Intelligent Transportation Systems World Congress*, 2024.

[104] H. Kim and A. Mnih, “Disentangling by factorising,” in *International Conference on Machine Learning*, 2018, pp. 2649–2658. [Online]. Available: https://proceedings.mlr.press/v80/-kim18b.html

[105] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin, “Cyclical annealing schedule: A simple approach to mitigating KL vanishing,” in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 240–250.

[106] N. Weber, D. Frerichs, and U. Eberle, “A simulation-based, statistical approach for the derivation of concrete scenarios for the release of highly automated driving functions,” in *AmE 2020-Automotive meets Electronics; 11th GMM-Symposium*. VDE, 2020, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9094569

[107] “Mann-whitney *U* test.” [Online]. Available: https://en.wikipedia.org/wiki/-Mann%E2%80%93Whitney_U_test

[108] I. Panagiotopoulos, B. Hillbrand, P. Weissensteiner, M. Muro, T. Menzel, A. Thorsén, M. Skoglund, and G. Stettinger, “Sunrise d7.1: Ccam use cases validation requirements,” Tech. Rep., 2023.

[109] X. Zhang, S. Khastgir, and P. Jennings, “Scenario description language for automated driving systems: A two level abstraction approach,” in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 973–980.

[110] WMG, “Safety pool scenario database – sdl.” [Online]. Available: https://docs.safetypooldb.ai/-docs/sdl

[111] S. Tang, Z. Zhang, J. Zhou, L. Lei, Y. Zhou, and Y. Xue, “Legend: A top-down approach to scenario generation of autonomous driving systems assisted by large language models,” in *39th IEEE/ACM International Conference on Automated Software Engineering*, 2024, pp. 1497–1508.

[112] N. Dokania, T. Singh, E. Lefeber, J. Ploeg, and M. Alirezaei, “Implementing a dissimilarity metric for scenarios categorization and selection for automated driving systems,” in *11th IFAC International Symposium on Advances in Automotive Control*, 2025.

[113] T. Singh, E. van Hassel, A. Sheorey, and M. Alirezaei, “A systematic approach for creation of SOTIF’s unknown unsafe scenarios: An optimization based method,” in *WCX SAE World Congress Experience*, 2024.

[114] A. Bracquemond and G. Thiolon, “MOOVE project: Recognition of road scenes by the data collected at the output of the sensors of the autonomous vehicles,” in *28th Aachen Colloquium Automobile and Engine Technology*, 2019.

[115] M. Zhou and Y. Shao, “A powerful test for multivariate normality,” *Journal of Applied Statistics*, vol. 41, no. 2, pp. 351–363, 2014.