ASSESSING TEST SUITE QUALITY FOR SCENARIO CLASSES IN AUTOMATED DRIVING

Christian Neurohr

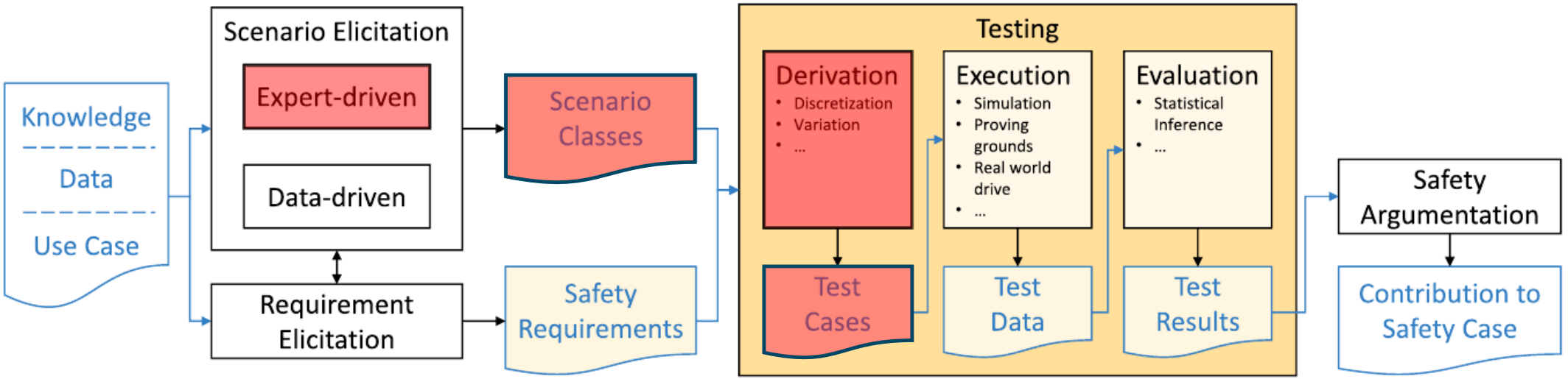
ADAS & Autonomous Vehicle Technology Conference 2025

22.05.2025

Safeguarding Automated Driving

- Impossible to prove safety using distance-based, statistical approaches (Wachenfeld **[1],** Kalra **[2]**)
- Scenario-based development has become state of the art for automated driving systems
- Find scenario classes that adequately cover the operational design domain (ODD)
- Sample test cases from scenario classes until ...
  - the system exhibits no hazardous behavior?
  - no test requirements are violated?
  - all critical scenarios have been tested?
  - the test cases adequately cover the scenario class?

Scenario-Based Testing for Automated Driving



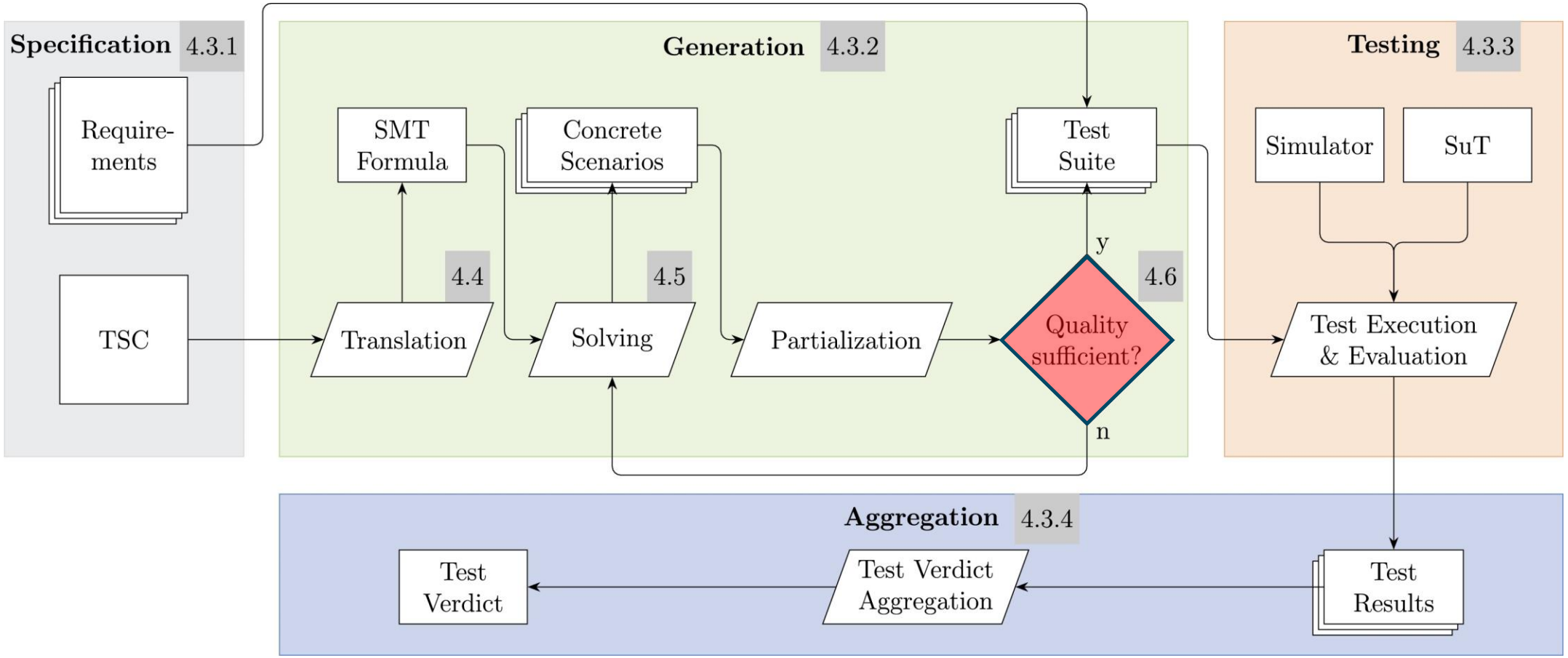
Overview of scenario-based testing for safeguarding automated driving systems **[3]**

Scenario Formalisms – Logical vs. Abstract

Characteristics	**Logical Scenarios**	**Abstract Scenarios**
Approach	Imperative (Command-based)	Declarative (Constraint-based)
Expressiveness	Theoretically Lower; Practically Equivalent	Theoretically Higher; Practically Equivalent
Specification Complexity	Higher	Lower
Description of Degrees of Freedom	Explicit	Implicit
Dimensionality	Finite	Unbounded
Applications	**Logical Scenarios**	**Abstract Scenarios**
Specifying	Data-based	Expert-based
Sampling	Easier (evaluation of parameter ranges)	Harder (sampling temporal formulae)
Monitoring	Unsuitable	Suitable
Reasoning	Unsuitable	Suitable

Comparison of formalisms for specification of scenario classes **[4]**

TSC2CARLA – Flowchart



Abstract scenario-based verification toolchain. Traffic Sequence Charts are translated into SMT problems **[5]**

Question – How can we ...

- determine the quality of a test suite ...
  - consisting of trajectory-level concrete scenarios (models) iteratively obtained by sampling an abstract scenario via constraint solving,
  - generated before test execution, independently of the SuT,
  - without an imperative description of the scenario class,
  - to reflect coverage/saturation on trajectory-level?

- Idea:
  - Use scenario distance measures to compare concrete scenarios **[6]**
    - i.e., a function $d : C(\mathcal{S}) \times C(\mathcal{S}) \to \mathbb{R}_{\geq 0}$
    - Examples: Euclidean distance, Fréchet distance, Dynamic Time Warping, ...
  - Aggregate pairwise scenario distances to a quality function for sets of concrete scenarios $Q : 2^{C(\mathcal{S})} \to \mathbb{R}_{\geq 0}$

Which properties do we want for quality functions?

#1 **Avoid redundancy:** Adding an instance to the test suite that is equal to an already sampled instance leads to a decrease in quality.

#2 **Existence of quality decreasing neighborhoods:** There exist neighborhoods around the instances in a test suite, so that adding an instance that is inside of such a neighborhood leads to a decrease in quality. This property extends the *Avoid redundancy* property.

#3 **High distances increase quality:** If there exists an instance whose minimal distance to the instances in the test suite is higher than the maximal distance between those instances, adding this instance leads to an increase in quality.

#4 **Preference of higher distance instances:** Adding an instance to the test suite that is further away to all already sampled instances than another possible instance leads to a stronger increase in quality.

Desirable properties of quality functions for test suite quality **[5]**

Which properties do we want for quality functions?

#5 **Preference of less sampled regions:** Having clusters of instances in the test suite, adding an instance to the test suite that is equally far away to a cluster than another possible instance to a denser cluster leads to a higher quality.

#6 **Boundedness:** There exist lower and upper bounds for the test suite quality, enabling a better estimation of the actual test suite quality.

#7 **Invariance under permutation:** The order in which instances were added to the test suite has no influence on the quality.

Desirable properties of quality functions for test suite quality **[5]**

Quality Functions – Potential Candidates?

- Notation:
  - $\mathbf{M}$ a set of $n \geq 2$ models (concrete scenarios)
  - $d_{i,j}$ the distance between the i-th and the j-th model

- Average Minimal Distances: $Q_{AMD}(\mathbf{M}) = \frac{1}{n} \sum_{i=1}^{n} \min_{\substack{1 \leq j \leq n \\ i \neq j}} d_{i,j}$

- Nearest Neighbor Entropy: $Q_{NNE}(\mathbf{M}) = \frac{1}{n} \sum_{i=1}^{n} \ln(1 + n \cdot \min_{\substack{1 \leq j \leq n \\ i \neq j}} d_{i,j})$

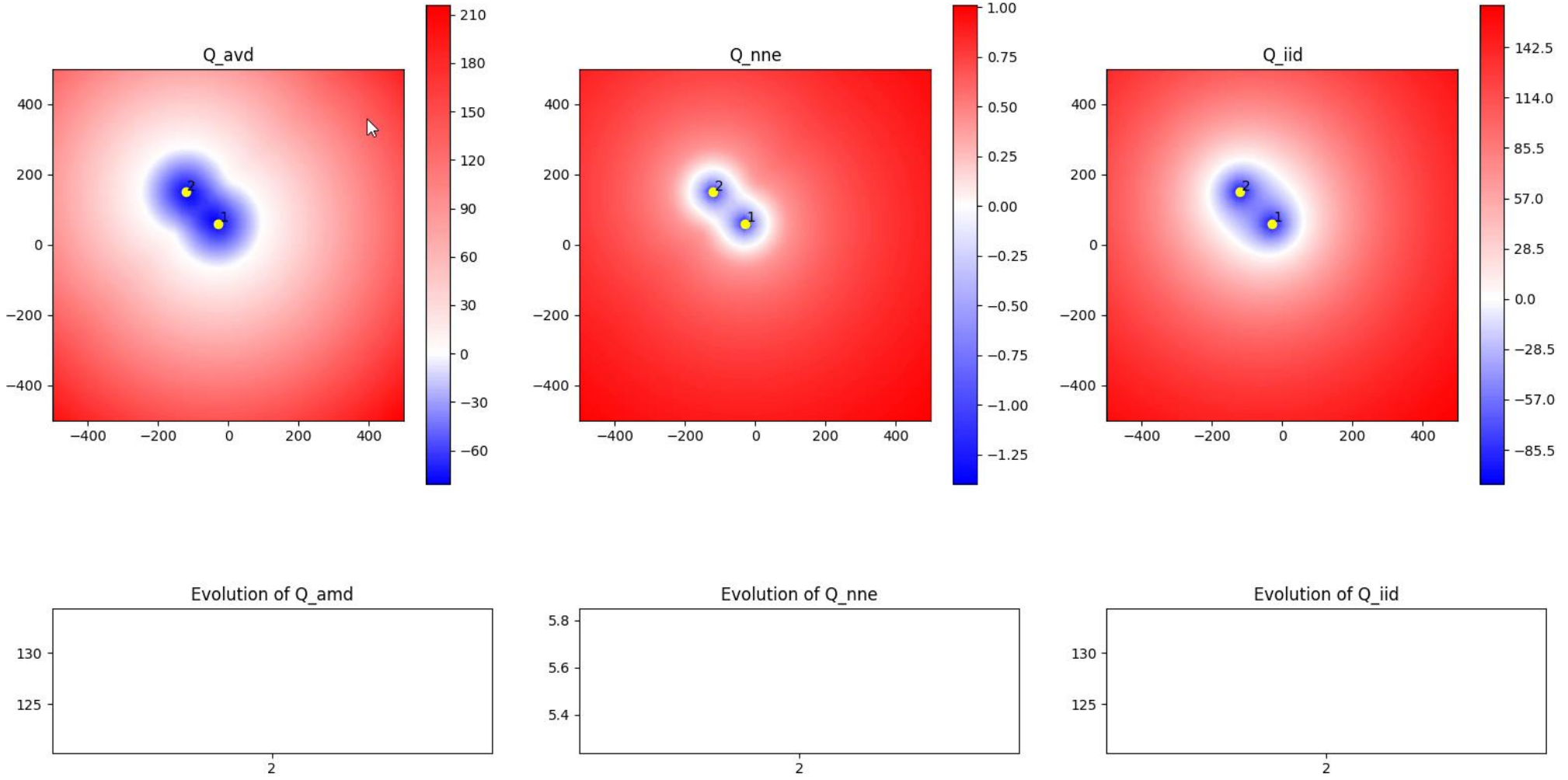
- Inverse Inverse Distances: $Q_{I^2D}(\mathbf{M}) = \frac{n(n-1)}{2 \sum_{i=1}^{k} \sum_{j=1}^{i-1} \frac{1}{d_{i,j}+1}}$

Which properties hold?

	Property	Q_{AMD}	Q_{NNE}	Q_{IID}
#1	No redundancy	☑	☑	☒
#2	Existence of quality decreasing neighborhoods	☑	☑	☒
#3	Preference of less sampled regions	☒	☒	☑
#4	Preference of higher distance models	☑	☑	☑
#5	Existence of quality increasing models	☑	☑	☑
#6	Boundedness	☑	☑	☑
#7	Permutation invariance	☑	☑	☑

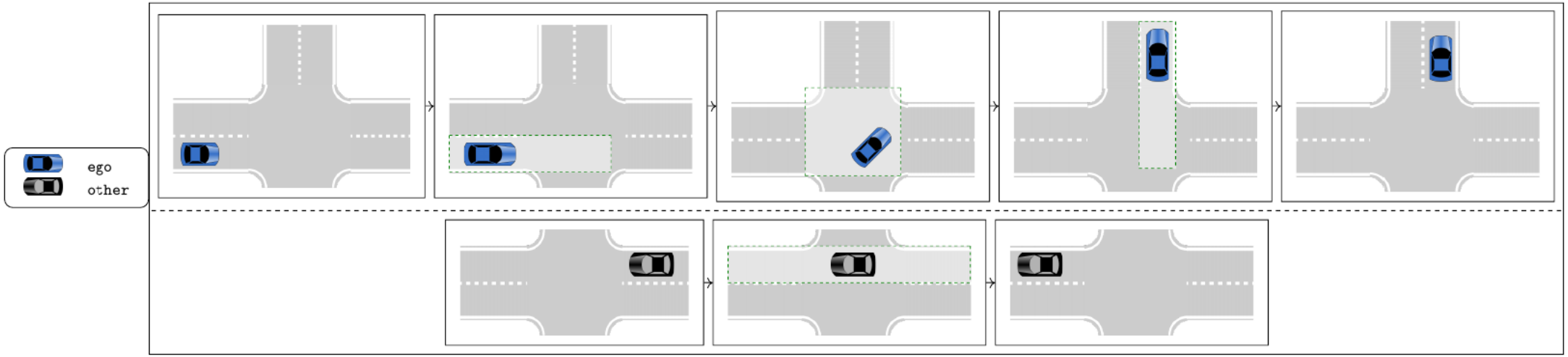
Formal proofs are possible if properties are formalized !

Quality Functions – 2D Experiments



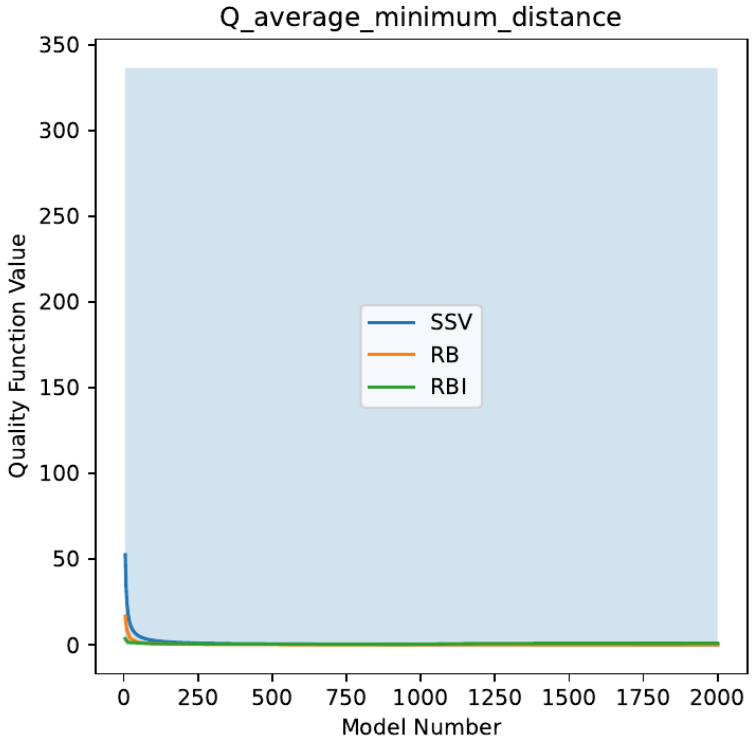
Use Case: Comparison of Sampling Methods (I)

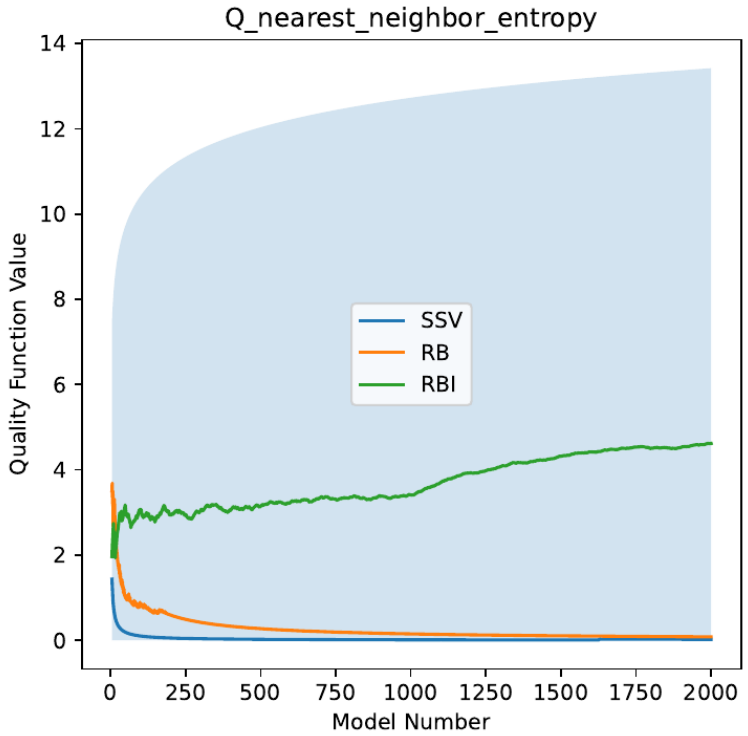
- Consider an abstract scenario specified as an Traffic Sequence Chart (TSC)
  - Unprotected left turn at urban intersection with uncoming traffic

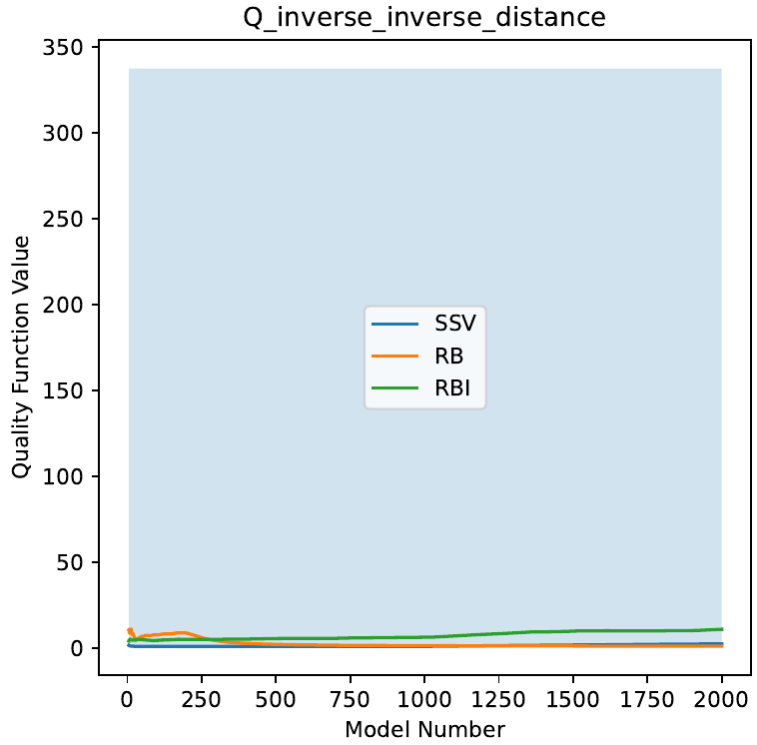


- Sample $n = 2000$ concrete test scenarios using different sampling methods
  - Solver Seed Variation (SSV), Recurvisve Blocking (RB), Recursive Blocking of Invariants (RBI) **[5]**
- Apply scenario distance (here: Dynamic Time Warping) + quality function to compare how well the methods cover the scenario class

Use Case: Comparison of Sampling Methods (II)



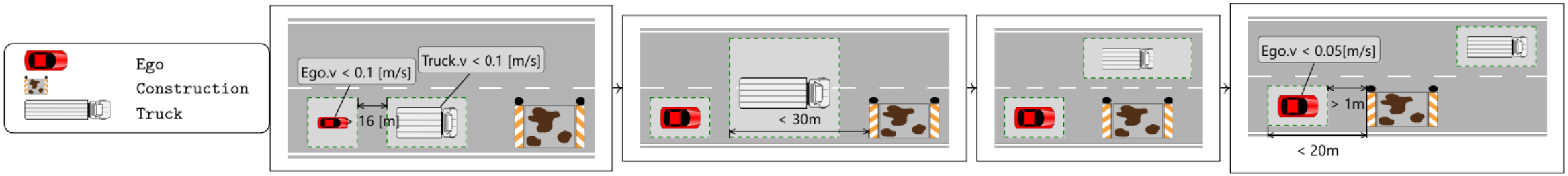


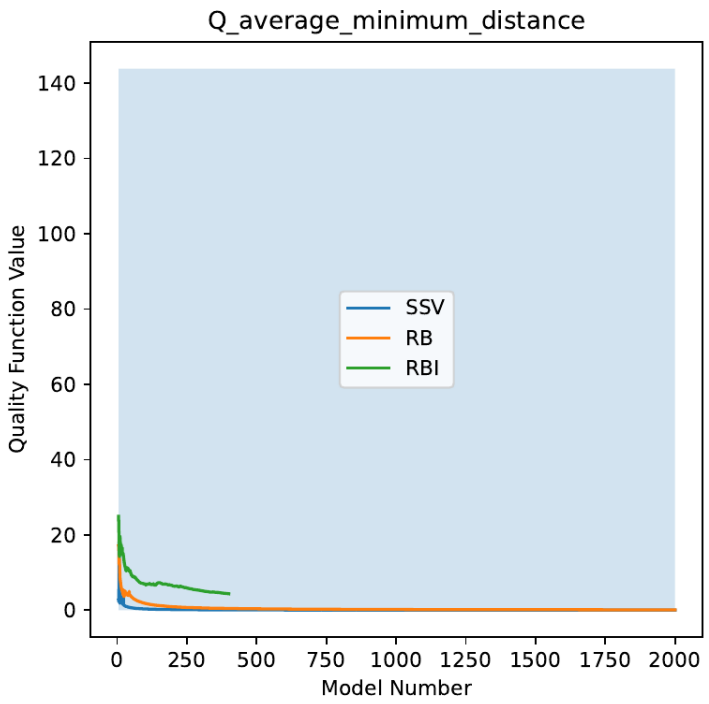


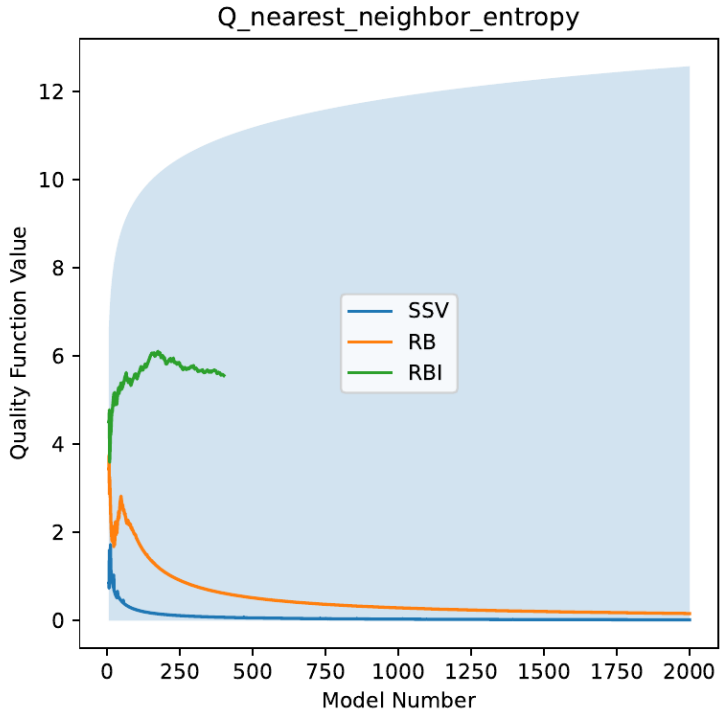
- Q_{AMD} punishes close models too severily
- Q_{NNE} enables appropriate comparism (RBI > RB > SSV)
- Q_{IID} could work, but scale needs improvement

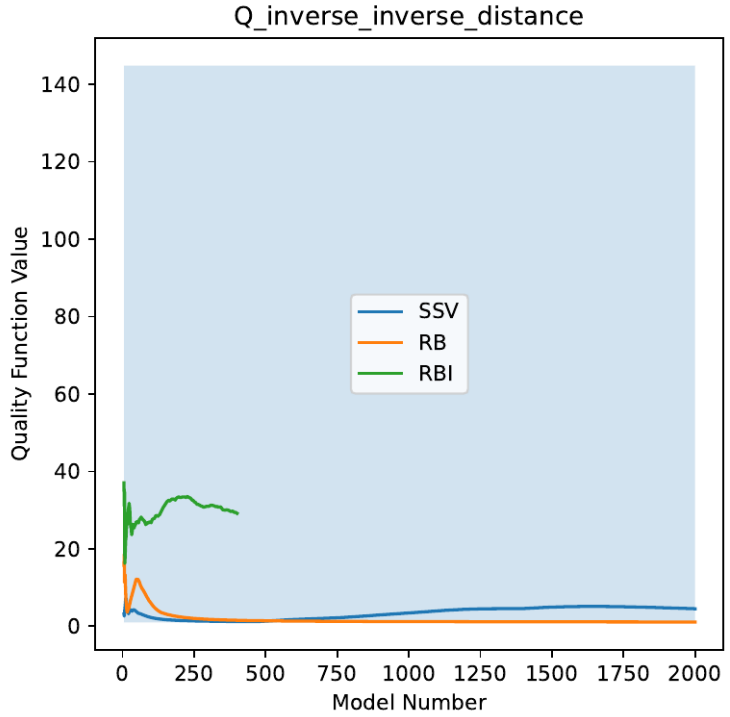
Use Case: Comparison of Sampling Methods (III)

- Truck cuts out of lane in front of construction site









Conclusion and Future Work

- Key takeaways:
  - Novel method to evaluate the quality of test suites for scenario classes
  - Evaluate generated test cases pre-execution, indepently of SuT
  - Originally developed to compare sampling methods for abstract scenarios

- Future work:
  - Find better quality functions, try different scenario distances
  - Improve and formalize desirable properties
  - Formal proofs for properties of quality functions
  - Apply quality functions to post-execution test suites sampled from logical scenarios
  - Intertwine quality function with sampling

Criticality Analysis - Publications

[1] Wachenfeld, Winner, „The release of autonomous vehicles“, 2016, in „Autonomous Driving: Technical, Legal and Social Aspects“

[2] Kalra, Paddock, „Driving to safety:How many miles of driving would it take to demonstrate autonomous vehicles reliability?“, 2016, Transportation research part A: policy and practice

[3] Neurohr et al., „Fundamental Considerations around Scenario-Based Testing for Automated Driving“, 2020, IEEExplore, ResearchGate

[4] Neurohr et al., „On Scenario Formalisms for Automated Driving”, 2025, arXiv, ResearchGate

[5] Borchers et al., „TSC2CARLA: An abstract scenario-based verification toolchain for automated driving systems”, 2024, ScienceDirect, ResearchGate

[6] Braun et al., „A Review on Scenario Similarity Measures for Validation of Highly Automated Driving”, 2023, International Conference on Intelligent Transportation Systems (ITSC)

THANKS FOR THE ATTENTION!

Dr. Christian Neurohr,
German Aerospace Center (DLR) e.V.
Team Lead Criticality Analysis