

**Deloitte.**

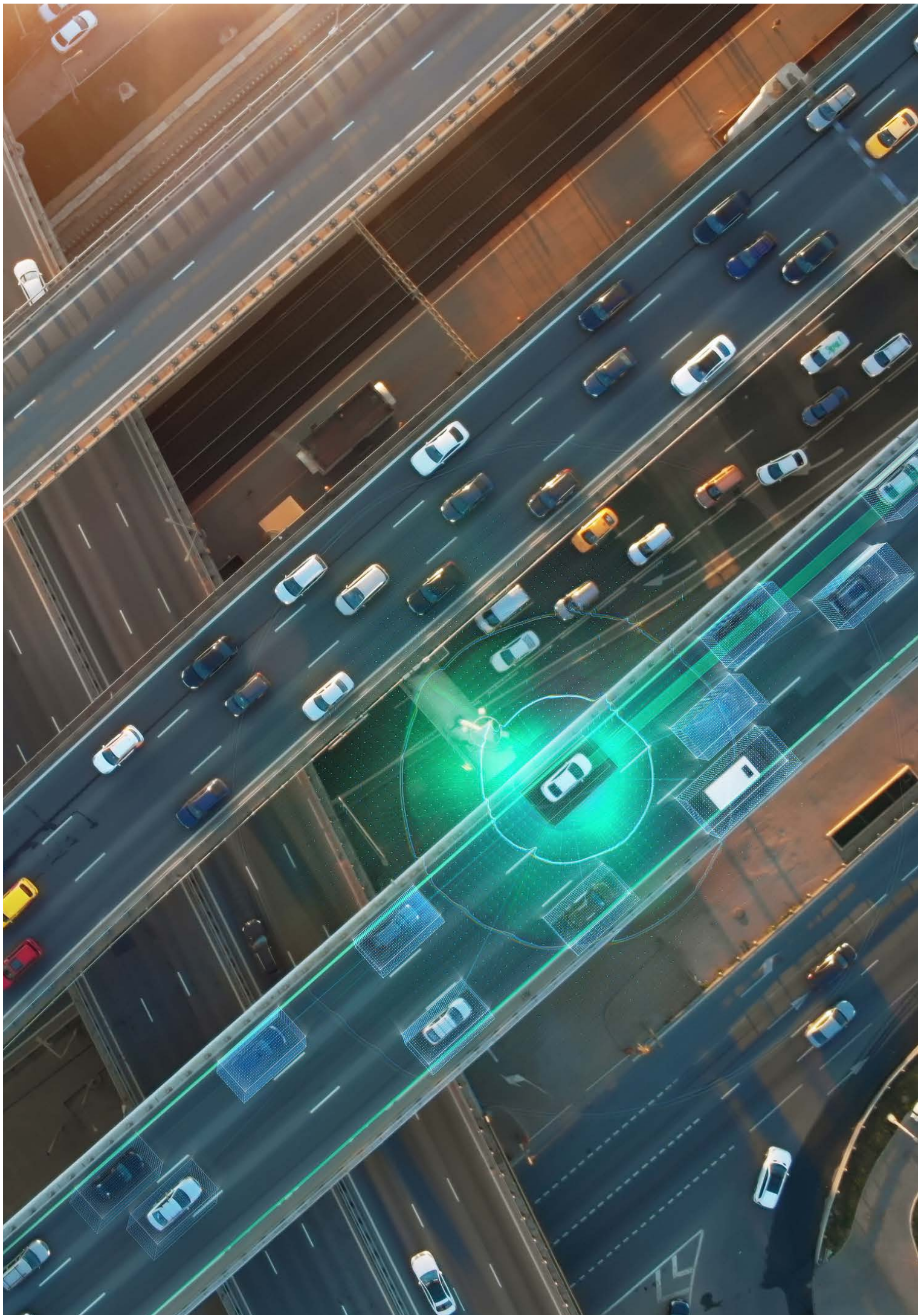
Towards developing  
trustworthy behavior for  
autonomous vehicles using  
reinforcement learning





Executive Summary	05
Introduction	06
Simulative development of a behavior planner using reinforcement learning	11
Evaluation on the autonomous vehicle	16
Conclusion	20
Contacts	22





# Executive Summary

Deloitte's trustworthy AI framework outlines the criteria that AI algorithms must satisfy to earn the trust among society. Making an application more trustworthy is especially relevant in safety-critical applications such as autonomous driving and remains one of the open challenges when it comes to AI-based algorithms deployed in autonomous vehicles. Deloitte highlights which factors are essential for developing trustworthy AI algorithms using supervised or unsupervised learning paradigms in the Trustworthy AI Framework. Reinforcement learning is a well-known learning paradigm that has gained traction in recent years. It enables systems to learn complex driving behavior offline in simulation, which reduces the computational demands of the

decision-making component (Planner) on the vehicle. However, it is not strictly possible to apply the trustworthiness criteria of other learning paradigms to reinforcement learning applications. Deloitte collaborated with fortiss to investigate to what extent the Trustworthy AI Framework is a useful and complete framework to use in the context of reinforcement learning. This whitepaper outlines the obstacles developers encounter when they use reinforcement learning for behavior planning and deploy it in an autonomous research vehicle. In the conclusion, we propose various investigations that will be vital in the future to make the trustworthy AI framework more useful in real-world contexts.

With the upcoming European AI Act, Deloitte engaged with fortiss to assess the robustness of the application of AI in a collision avoidance use-case for autonomous driving by identifying risk factors and executing countermeasures.



# Introduction

Over the last decade, we have seen major advances in highly automated and autonomous driving systems, which are divided into categories according to the SAE levels ([sae\\_international\\_standard\\_2021](#))<sup>1</sup>. Companies such as Waymo, Argo AI and others are working towards L4 urban autonomy with the objective of fully autonomous taxis. Disengagement has gone down steadily in test drives over the past few years, and the first taxi services that do not require a backup safety driver are on the road. Other players have focused on developing L3 autonomy as the future of driver assistance systems, bringing the first L3 driver assistance systems to market. We are expecting similar systems to be launched the coming years.

Novel developments in artificial intelligence (AI) have made significant contributions to these advances. Decision-making and motion planning for autonomous vehicles rely heavily on the field of classical AI and offer a continuation of approaches such as transferring game-theory concepts to model interactions between traffic participants. The last decade has seen dramatic advances in Deep Learning and connectionist AI, which are being used for developing perception and prediction algorithms with supervised or unsupervised learning paradigms. Deep Reinforcement Learning has emerged as a combination of Deep Learning and reinforcement learning. The latter paradigm, by contrast, enables robots to learn optimal decisions by repeating interactions with and receiving rewards from an environment.

In all likelihood, both L3 and L4 systems will require different variants and combinations of AI paradigms to achieve autonomy. However, the use of AI in safety-critical domains is still under investigation. Research and industry are trying hard to apply existing safety engineering methods to AI-based systems. To resolve this challenge, Deloitte interviewed data scientists, computer scientists, mathematicians as well as risk, ethics, and economic experts worldwide and compiled their collective insight in the "Trustworthy AI Framework". The framework provides a summary of the criteria that AI must satisfy to gain human trust, but its current focus is mainly on the trustworthiness of supervised and unsupervised learning paradigms.

This led to Deloitte's collaboration with fortiss to investigate to what extent the Trustworthy AI Framework is sufficiently useful and complete to apply it in reinforcement learning methods. Behavior planning for autonomous driving (AD) is an extensive field of research with a series of promising developments over the past few years ([schwartzing\\_planning\\_2018](#))<sup>2</sup>. Using reinforcement learning to develop behavior planning algorithms has shown promise in simulation ([hart\\_graph\\_2020](#), [bernhard\\_addressing\\_2019](#))<sup>3</sup> and in prototype vehicles ([wang\\_learning\\_2021](#))<sup>4</sup>. This paper goes in a similar direction, attempting to relate these findings to the trustworthy AI framework. It describes a process whereby we develop a behavior planner for an autonomous vehicle (AV) using reinforcement learning and deploy it in real-world

scenarios in a fully functional autonomous driving prototype vehicle. Fig. 1 provides details about the development process. We developed the behavior planner for the rather simplistic Operational Design Domain (ODD) depicted in Fig. 2. The goal is to enable the AV to maneuver around parked cars. The ODD is located near fortiss in Guerickestr. 25 Munich (Germany).

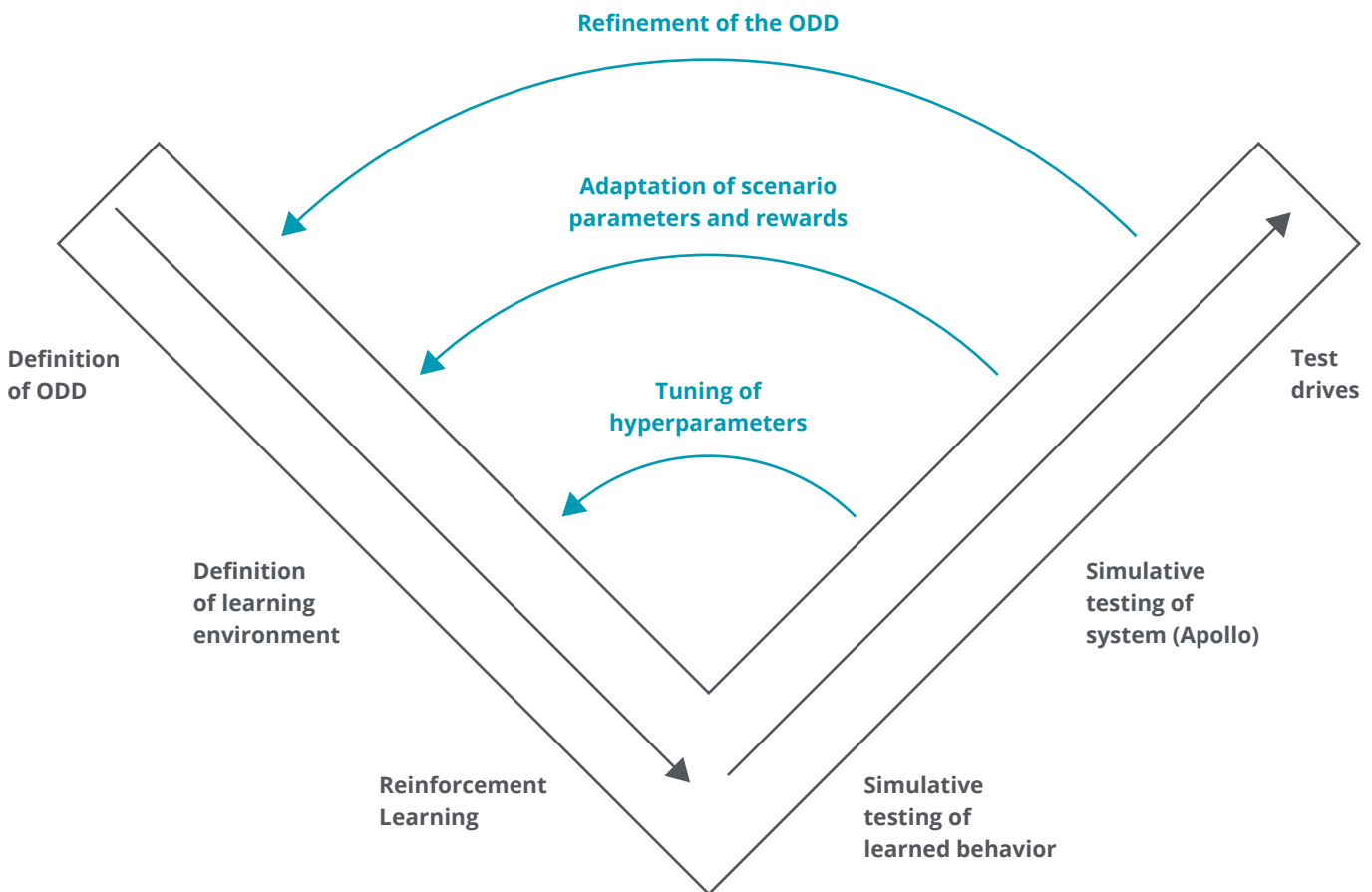
<sup>1</sup> SAE International: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. 2021.

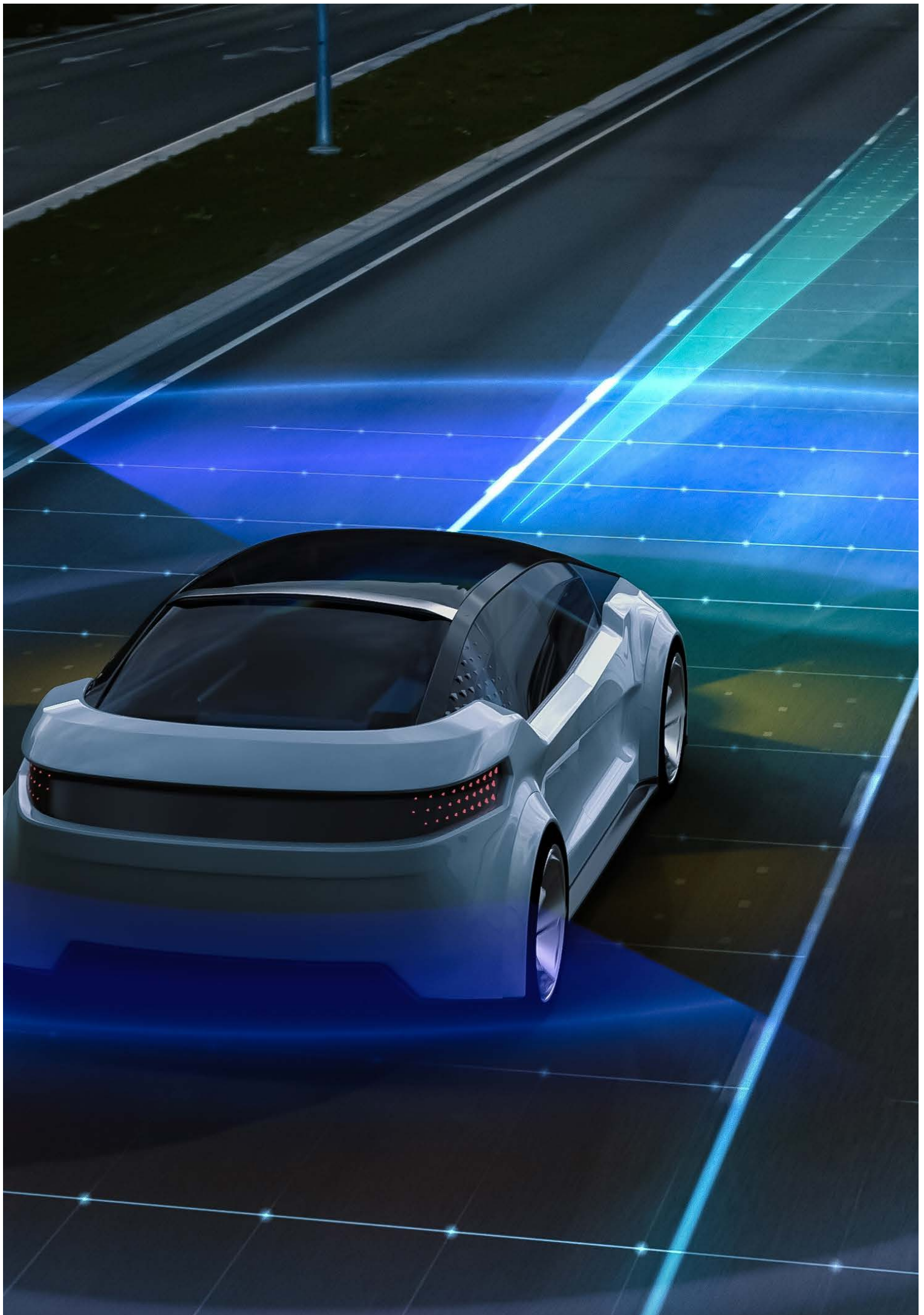
<sup>2</sup> Schwartzing, Wilko and Alonso-Mora, Javier and Rus, Daniela: Planning and Decision-Making for Autonomous Vehicles. 2018. P. 187–210.

<sup>3</sup> Hart, Patrick and Knoll, Alois: Graph Neural Networks and Reinforcement Learning for Behavior Generation in Semantic Environments. 2020. P. 1589–1594.

<sup>4</sup> Wang, Tsun-Hsuan and Amini, Alexander and Schwartzing, Wilko and Giltschenski, Igor and Karaman, Sertac and Rus, Daniela: Learning Interactive Driving Policies via Data-Driven Simulation. 2021.

**Fig. 1 – Overview of the development process used in this paper. The learning environment is designed to model the ODD via specifically designed traffic scenarios and reward specifications. Iteratively, hyperparameters of the reinforcement learning algorithm are fine-tuned until the learned behavior achieves sufficient performance in the simulation based on the performance criteria specified in the ODD. The system performs these training and testing steps in the behavior learning framework BARK-ML. Once these steps have been completed, we test the learned behavior in the autonomous vehicle system architecture using the Apollo OpenSource driving stack. Instances of unsatisfactory system performance lead to adaptations of the scenario and reward parameters. Where performance is satisfactory, we conduct test drives, which eventually result in further ODD refinement before whole development process starts again.**



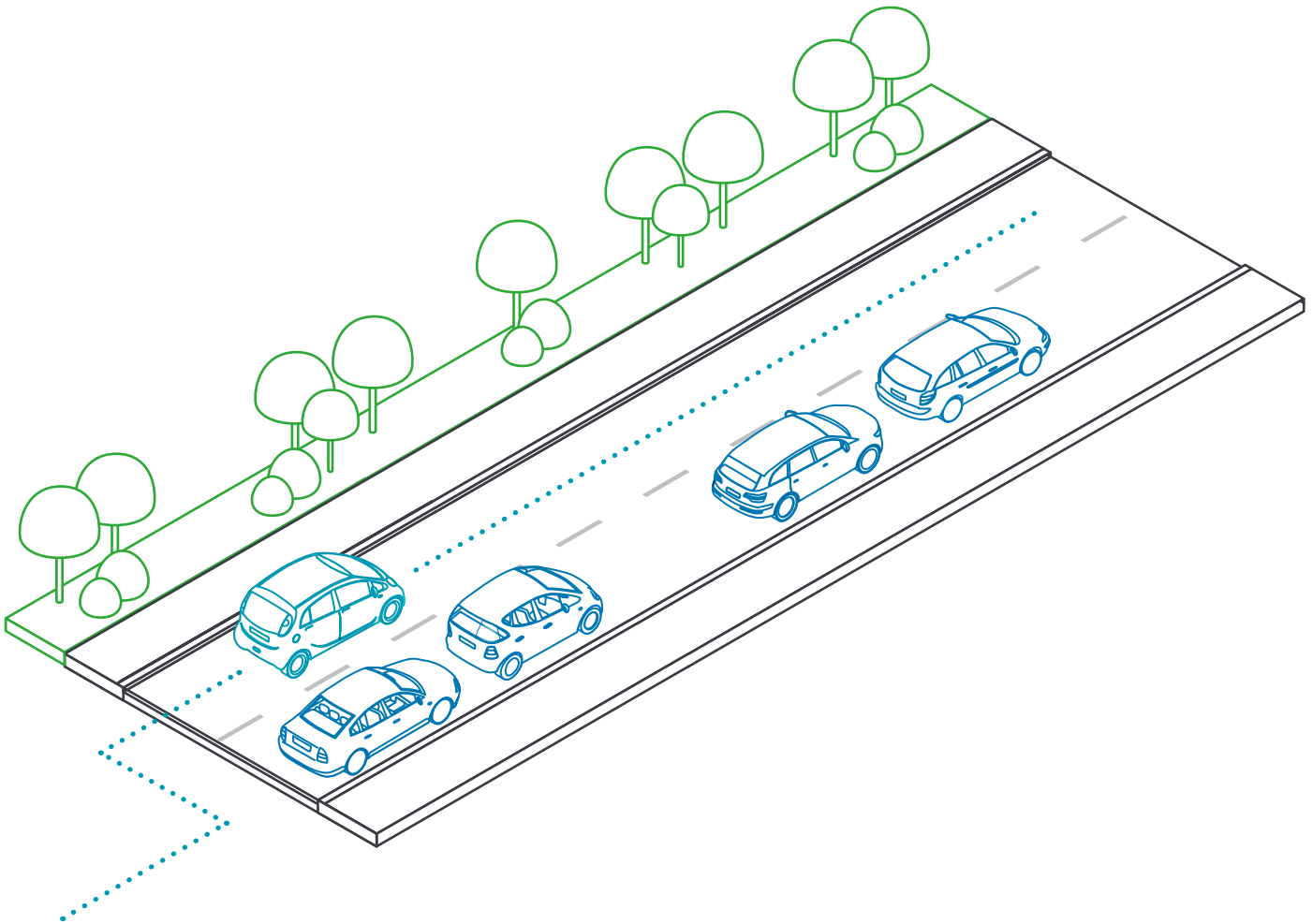




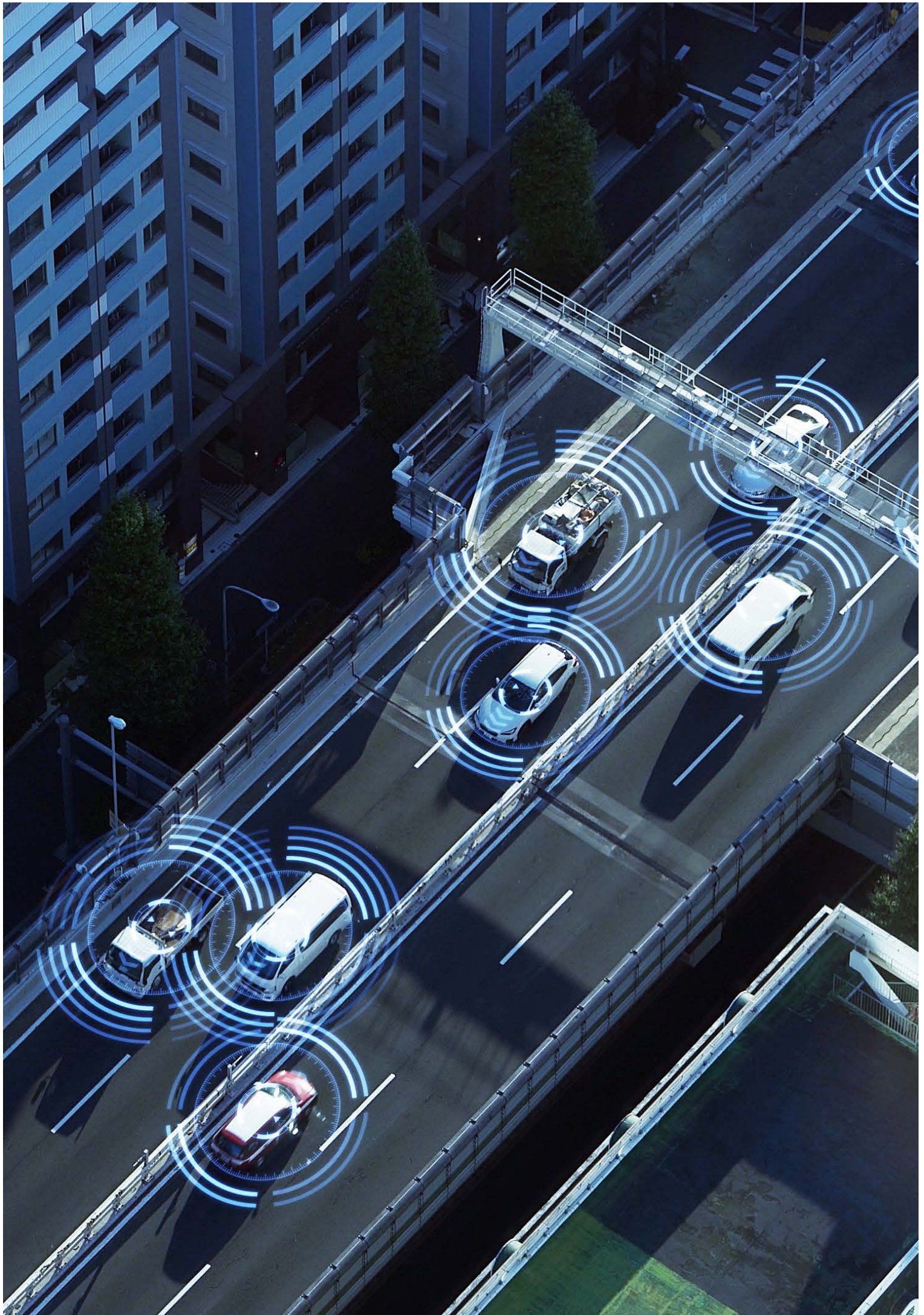
This whitepaper outlines the obstacles we encountered during the development, testing and deployment steps of the Deloitte Trustworthy AI Framework. It concludes with the future investigations we believe will be necessary in order to further increase the framework's relevance. We

begin by describing how we used reinforcement learning to develop a behavior planner. Then we move on to the setup of the prototype vehicle as well as the findings of and insights gained in the test drive and end with our overall conclusion.

**Fig. 2 - : Exemplary real-world conditions within the ODD: The scenario is a narrow two-lane road with parked cars as obstacles at the side. The AV is supposed to start behind a parked vehicle, pull out into the left lane and come to a complete stop at the end of the road. This ODD is located near fortiss in Guerickestr. 25 Munich (Germany).**









# Simulative development of a behavior planner using reinforcement learning

## Overview of the learning process

In the planning module of an AV, the system calculates the trajectory guiding the vehicle through a particular driving scenario. The planned motion must provide a meaningful balance between safety, efficiency, and comfort. This section outlines how we developed the behavior policy that will impact the vehicle's decision in the next time step. The system learns the behavior policy by interacting with the simulated traffic environment using Deep Reinforcement Learning, specifically the actor-critic algorithm presented in Haarnoja Soft 2018.<sup>5</sup> The autonomous vehicle

simulation observes the current state of the environment state repeatedly, selects an action based on the current behavior policy and receives a reward if it transitions to the next environment. Based on the past decisions and rewards, the AV can learn an optimal behavior policy designed to maximize future rewards. Whether the learned behavior policy is applicable will depend on how meaningful the reward function is during learning. On the other hand, the traffic scenarios used during learning must comply with the ODD specifications. We will discuss these difficulties in the following.

We iteratively optimise the reward function for a better-performing Reinforcement model and test the vehicle's decision in a simulative environment.

<sup>5</sup> Haarnoja, Tuomas and Zhou, Aurick and Abbeel, Pieter and Levine, Sergey: Soft Actor-Critic: Off-policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. 2018.



## Obstacles during reward function design

The first step in designing a meaningful reward function is to investigate the behavioral specifications that were made for the AV during the initial design phase. The specifications for the AV behavior in a particular ODD could be:

- **Collision avoidance:** The AV must not provoke any collisions with static objects, e.g., parked cars.
- **Successful maneuvering:** The AV must be able to complete the maneuver
- **Maneuver completion time:** The AV must navigate through the traffic at the same average speed as human drivers in comparable traffic conditions.
- **Driving comfort:** The AV should ensure an appropriate level of driving comfort, veering into uncomfortable steering and acceleration changes only when it is required to avoid a collision.

This is a stripped-down, simplified set of specifications, and yet it demonstrates just how difficult it is to design a reward function that helps the AV learn the behaviors that meet with these requirements.

## Reward Terms

In principle, we can model each specification to contribute to the reward function. There are two options for converting requirements into reward terms.

Sparse reward terms are helpful to model requirements based on the exclusion or inclusion of a single event, e.g., successfully completing a maneuver or provoking a collision. We often choose sparse rewards for high-priority modeling requirements, e.g., collision reduction and maneuver completion. For instance, we assign a huge negative reward to the collision requirement when a collision occurs in simulation. If we use only sparse rewards, we can infer the approximate probability that a particular event occurs for a learned behavior. Analyses like

these can be meaningful if we want to argue that a specific requirement has been met with sufficient accuracy. Using only sparse rewards can, however, keep us from exploring the broader behavior problem during learning, as the AV only rarely receives a reward from the environment and may not successfully complete the learning process.

Continuous reward terms are helpful to model behavioral specifications that will only be satisfactory through continual assessment the AV's actions. For instance, the AV receives an increasingly positive reward for maneuver completion time the closer the speed of the AV comes to the speed of human drivers. The system evaluates and issues these rewards after each action of the AV during the learning process. Apart from modeling requirements, continuous reward terms allow us to explore during learning and accelerate the learning process. They are, however, more sensitive when it comes to fine-tuning the parameters. More importantly, the accuracy with which the learned behavior meets the requirements modeled by sparse reward terms may be obscured when you use continuous reward terms. The strategy of improving exploration by adapting the reward structure is also referred to as reward shaping (ng\_policy\_1999)<sup>6</sup>.

## Design reward function

Two training processes that combine the above reward specifications have been designed and analyzed to better understand these difficulties.

### • Sparse reward

A solely sparse reward definition is specified as

$$R_{sparse} = R_{goal} + R_{collision\_offroad}$$

### • Mixed rewards

Additionally, continuous reward terms are added:

$$R_{mixed} = R_{sparse} + R_{goal\_dist} + R_{desired\_vel}$$

The function  $R_{goal}$  issues a reward if the agent reaches and slows down to within a specified target range at which the driving

maneuver shall end. The term  $R_{collision\_offroad}$  penalizes collisions with other objects or the edge of the road. The continuous reward function  $R_{goal\_dist}$  issues rewards when the AV gets closer to the goal. The continuous reward term  $R_{desired\_vel}$  penalizes deviations from the target velocity.

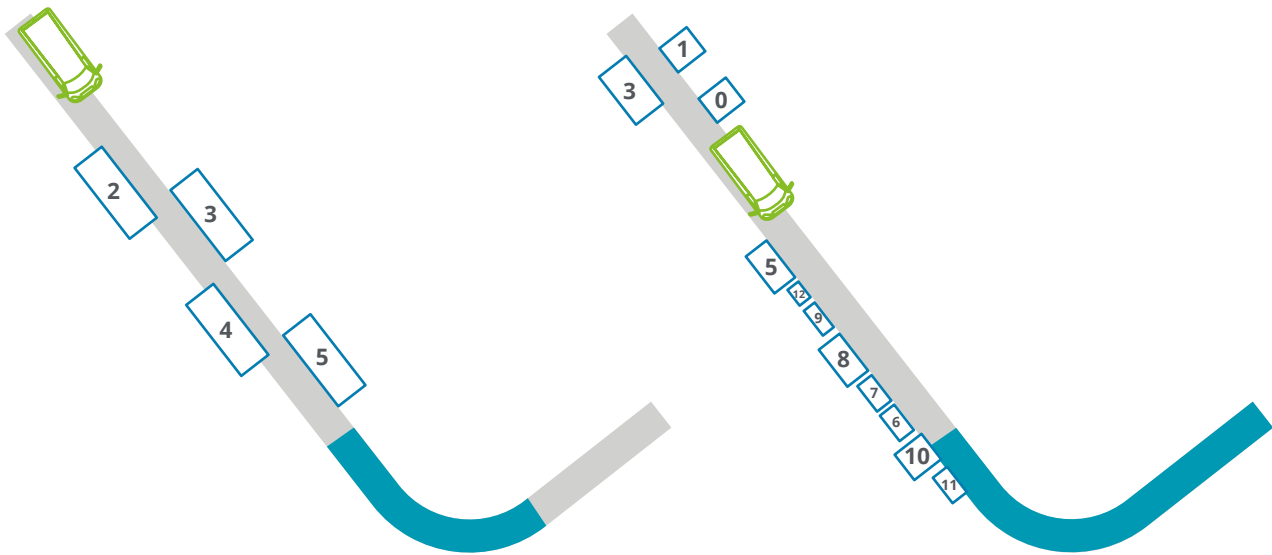
We assess the quality of the learning process with two metrics that measure the extent to which two of the previous behavior requirements have been met. Fig. 3 depicts the success and collision rate throughout training, which relates to the requirements "collision avoidance" and "successful maneuvering". The collision rate decreases in both reward configurations during training. The success rate remains around zero with the sparse reward configuration. The system only achieves a satisfactory success rate when it uses a combination of the reward configurations. However, the goal-directed reward functions,  $R_{goal\_dist} + R_{desired\_vel}$  have an adverse impact on the collision rate. We can only fine-tune the balance between collisions and successful maneuvering empirically; it is not possible to meet the specification a-priori. Requiring empirical analyses impedes the development of RL-based planners in the context of the Trustworthy AI Framework.

<sup>6</sup> Ng, Andrew Y. and Harada, Daishi and Russell, Stuart: In Proceedings of the Sixteenth International Conference on Machine Learning. 1991, P. 278-287.

**Fig. 3 – Performance criteria of a behavior planner over the reinforcement learning process for two different reward configurations. Learning using a mixed reward specification outperforms the sparse reward setting in terms of the share of successful scenario completions (success rate) and the time to complete the maneuver (step count). With mixed reward terms, however, the collision rate rises slightly. Adding additional continuous reward terms obscures the relationship between the specified collision reward and the resulting collision rate. As a result, it remains challenging to strike the right balance between behavioral specifications given by the ODD in the design of the reward system in our efforts to develop trustworthy behavior using reinforcement learning.**



**Fig. 4 – Examples of a scenario obtained with uniform sampling (US) in 4a or variation sampling (VS) in 4b. The red rectangle represents the ego vehicle. The blue region at the end of the road marks the goal area. In US scenarios, the position of objects is more uniformly spread in the ODD. Scenarios generated with VS tend to show only minor variations compared to the situation captured in the test drive.**



### Obstacles during scenario simulation

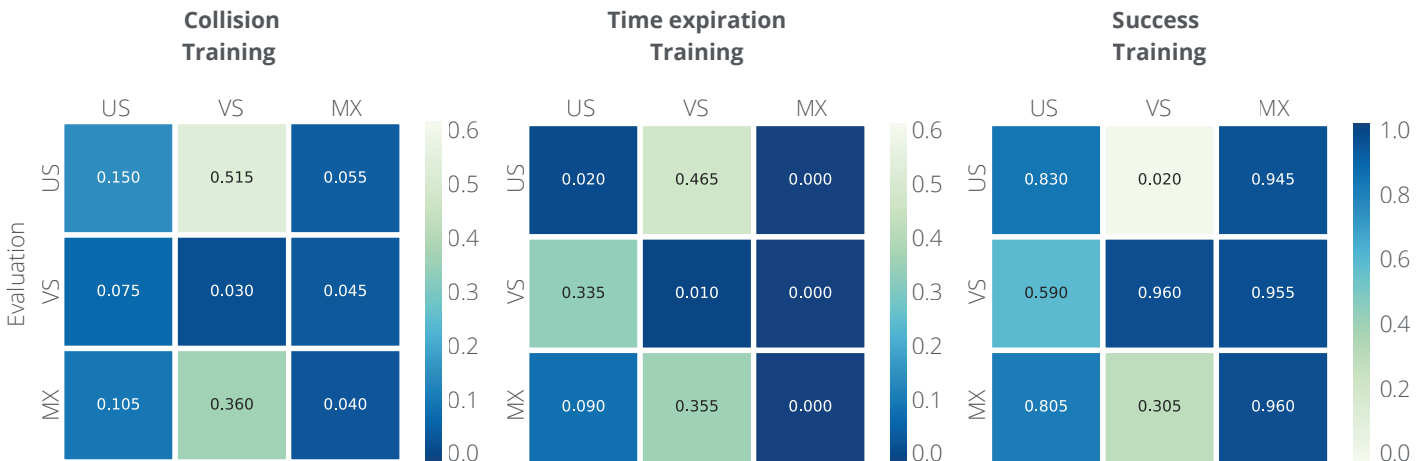
The previous section discussed the obstacles we encountered in designing the reward function in our efforts to meet the behavioral specifications of the ODD. This section deals with the extent to which the system is able to sufficiently cover the scenario specifications of the ODD during both the training and the evaluation stage of the development process. fortiss developed the semantic simulation framework BARK (bernhard\_bark\_2020)<sup>7</sup> to benchmark and build behavior planning algorithms for autonomous driving. The extension BARK-ML allows for training and evaluating behavior planning algorithms using reinforcement learning.

We have analyzed two approaches to generating scenarios for training and evaluation in terms of how useful they are in achieving coverage of the ODD. These are

- **Uniform sampling (US):** This approach uniformly samples scenario properties from property ranges designed to cover the traffic situations in the ODD. For the ODD here, the properties we sampled include the number of, distances between and sizes of surrounding obstacles as well as the AV's initial starting position and velocity. Uniform sampling enables us to generate a set of scenarios that thoroughly cover the ODD. However, it also assumes that all relevant scenario properties can be captured and that they are distributed uniformly in the ODD.
- **Variation sampling (VS):** Manually specifying the scenario properties and their distribution can lead to incomplete coverage of the ODD and a distribution shift towards unlikely scenarios. As a result, it is crucial to capture additional scenario properties by analyzing recorded driving data. The perceived objects within multiple perception frames are combined, while any duplicate information, such as overlapping obstacles, is filtered out. Sampled variations, e.g., object position, size and change of orientation, are added on top of the filtered scene to generate multiple scenarios from a small set of captured test drives. The resulting scenarios are more realistic regarding the probability distributions of scenario properties. They may not, however, contain low-probability, edge-case scenarios within the ODD.



**Fig. 5 – Cross-evaluation comparing behavior planners trained with a specific set of scenarios also to scenarios other scenario sets. In the heatmap, dark colors denote better performance. Generalization is considered high when performances of training and non-training scenarios nearly match. Training on a mixture of scenarios (MX) from uniform sampling (US) and variation sampling (VS) achieves the highest generalization.**



**Evaluation**

We evaluate the generalization capabilities of the above approaches to generating scenarios using a cross-evaluation. First, we train three behavior planners, two of which are based on sets of scenarios generated with the two aforementioned approaches and one of which is based on a mixed approach to scenario generation. Each trained planner is also evaluated against the other approaches during the evaluation process. The resulting performance metrics are presented in Fig. 5. In the following, we consider generalization to be high when the performance of non-training scenarios is close to the performance of the training scenarios. Given this definition of generalization, the results clearly show that generalization is at its lowest when we use VS during training. Training with US shows a moderate level of generalization. We achieve the highest level of generalization when training

with mixed scenarios (MX). Interestingly, the use of MX scenarios during training not only generalizes the trained behavior, but it even surpasses the performance of other trained planners on their training data set. Overall, these are promising insights that will help design sets of scenarios for reinforcement learning that meet the ODD coverage criteria. It remains unclear, however, what role the individual downsides of each approach will play when we combine the scenarios and whether there is a mix of scenarios that would even be able to increase performance during training. There will have to be further analysis of these factors, which we will ultimately include in Deloitte’s AI framework.

# Evaluation on the autonomous vehicle

## System Setup and AV architecture

Technically, a fully autonomous vehicle can be described as a robotic system. Its software architecture follows a classic state-of-the-art sense-plan-act cycle, as shown in Figure 6. Variants exist, but the ingredients are the same.



### Sense

Various sensors are in place to observe the environment around the car. This data is collected, interpreted and merged into a consistent environment model that can be used in subsequent software processes. The model includes static information, such as road layout and geometry, as well as dynamic information, such as position, the velocity and intention of other vehicles and Vulnerable Road User (VRU). We treat this as a black box component with well-defined interfaces. A processing pipeline finds, extracts, segments and classifies objects perceived by the vehicle sensors and estimates a polygonal shape and future predicted motion for each object within the sensor range.



### Plan

Based on the state of the environment, the systems must decide (1) what to do and (2) how to perform the action. (1) is referred to as behavior planning and (2) as motion planning. The planning step calculates a trajectory for the next couple of seconds describing the planned motion of the car that is safe (collision free), comfortable (smooth) and within the dynamically changing environment. A trajectory is a time-dependent motion plan of the agent and contains a consistent, admissible sequence of Cartesian points with dynamic states, such as a velocity. Each time the perceived environment changes, that triggers a new planning sequence to make use of the new data, as the old planning may no longer be valid. This takes place against a receding horizon scheme, in which new trajectories are smoothly faded into the current existing trajectory. This trajectory is the ideal target in terms of how the vehicle will move.

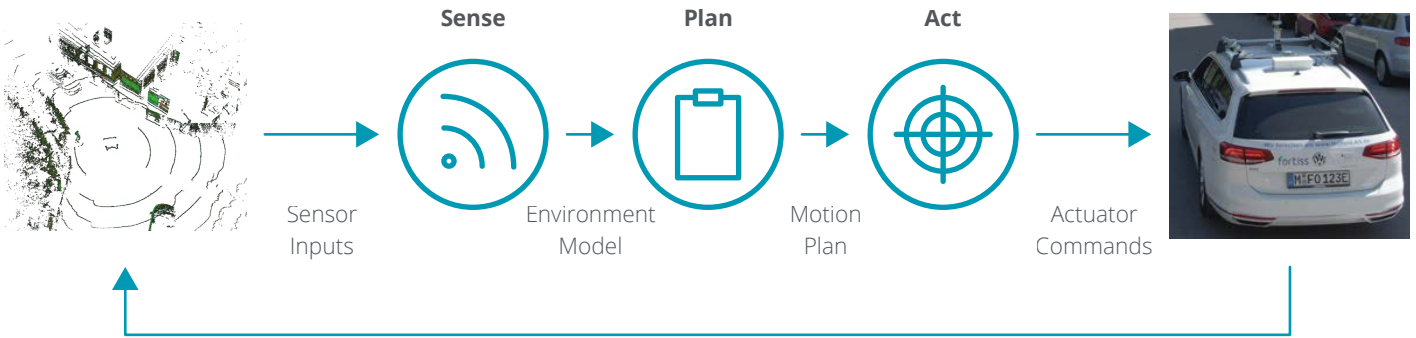


### Act

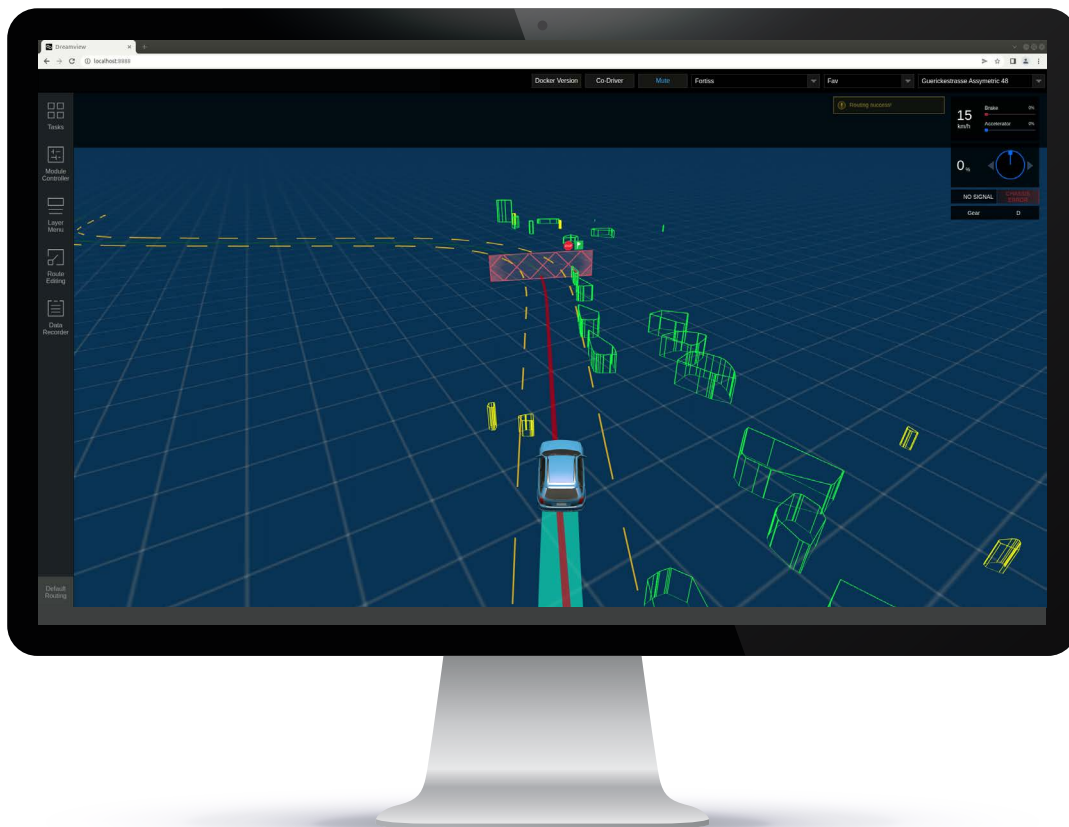
In the Act step, the system hands the planned trajectory over to a tracking controller designed to track the trajectory as closely as possible with respect to the physical limits of the vehicle. This step can also consider emergency situations that are not accounted for in the planning stage. This software module derives the actions for the car to perform (e.g., in terms of actions on the steering wheel or the pedals) at a very high frequency.

The work in this whitepaper focuses on the development of a behavior planner (Plan step). The Fortuna research vehicle implements the whole Sense-Plan-Act loop and employs an adapted version of the Apollo open-source driving stack (kessler\_mixed-integer\_2022)<sup>8</sup>.

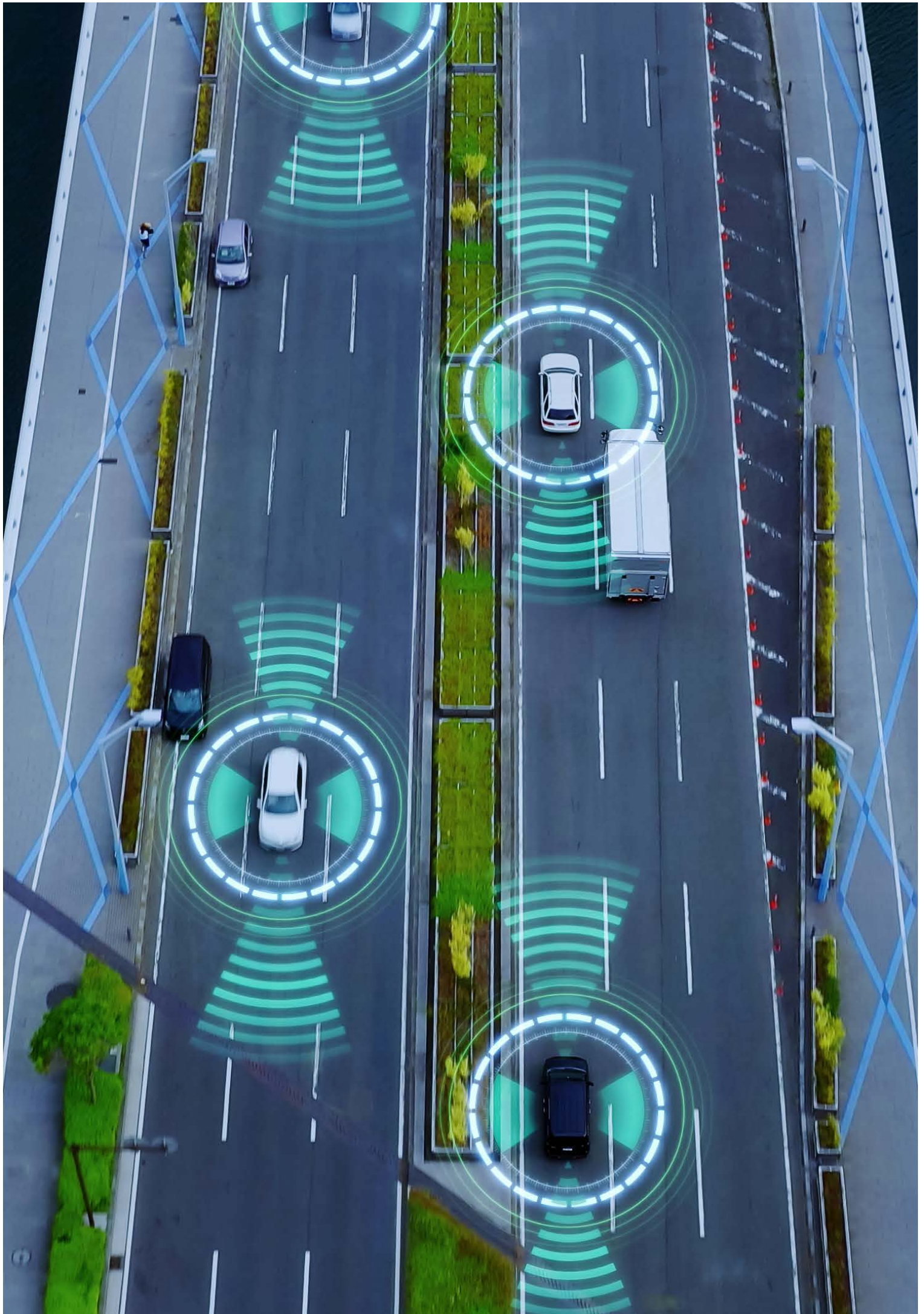
**Fig. 6 – Sense-Plan-Act loop of the Fortuna research vehicle.** Fortuna employs the classic software architecture found in robotics systems. The sensor inputs are processed in an environment model used by the planner to calculate a motion plan for the AV. The controller lets the vehicle act in the environment by transforming the motion plan into steering and acceleration actuator commands.



**Fig. 7 – Results of the test drives.**







### Test drives

When it comes to performing the test drives, we deploy the behavior planner that we developed on the Apollo driving stack. The test drives start in a parked position on the right side of the road behind a larger obstacle, in this case behind a construction trailer located at Guerickestr. 25 Munich (Germany) during the period of the test drives. The AV is meant to swerve autonomously from behind the obstacle to the left side of the road, continue driving on this side, and then come to a full stop at the end of the road.

We obtain two test drive variants by placing a large cardboard box either on the left side of the road or near the construction trailer. The cardboard box decreases the available driving gap, which creates conditions for the AV that push the boundaries of the specifications considered by the ODD during training. These conditions are interesting in terms of evaluating to what extent the training performance is applicable in reality.

The resulting driving scenes of the two test drive variants are depicted in Fig. 7. In both cases, the AV manages to navigate around the cardboard obstacle autonomously. The requirements for the performance of the learned behavior, e.g., keeping the desired velocity, collision avoidance, and reduction of steering variations, are modeled using rewards as defined in Sec. 2.2. The AV also satisfies the qualitative performance criteria during the test drives. It seems, therefore that the performance of the learned behavior obtained in simulation using a mixture of different scenario sets is applicable in a real-world setting. These qualitative observations illustrate the extent to which the development process as outlined in this paper is meaningful. The test drives show that the safety and comfort perceived as a human co-driver in the AV may differ from the safety and comfort criteria used for training in simulation. We believe it is worth investigating these differences and the role they play in the trustworthy AI Framework in the future, which is now possible using the proposed development process.

### Observations on trustworthiness at the system level

When used in the real-world system, we no longer have access to ideal input data, as in simulated scenarios. In other words, the planning system needs to be robust enough to eliminate degraded input data and system performance while still producing valuable results.

First, the perceived state of the environment is different from the actual state of the environment due to the delay in the system's processing time or due to unavoidable errors in perception. As a result, the planned trajectory and especially the starting point of a trajectory will not match the true position of the vehicle. The AV still has to execute a smooth and safe motion. Second, there is also uncertainty in terms of the controller's execution of the trajectory, as actuation and perception errors can make it impossible to strictly track the trajectory. The actual vehicle dynamics may not be admissible for the controller, and deviations in space or time between the planned and executed trajectory may result in uncomfortable or even dangerous actions.

While this list is not meant to be exhaustive, it reminds us that we have to consider dynamic, timing and error characteristics of the cyber-physical system in the AI framework and in the learning process. This paper includes the physical limits of the vehicle in the training process. We also trained model variants with and without accounting for execution delays in the system. As a result, we observe more collisions with obstacles or road boundaries when we evaluate the RL agent trained without execution delay where the delays are set to realistic values. Training the agent with delays achieves high success rates, even in scenarios with longer execution delays. As a result, we believe that system-specific error patterns, such as the ones discussed here, should have to be generally applicable in a general framework for developing AI components.

# Conclusion

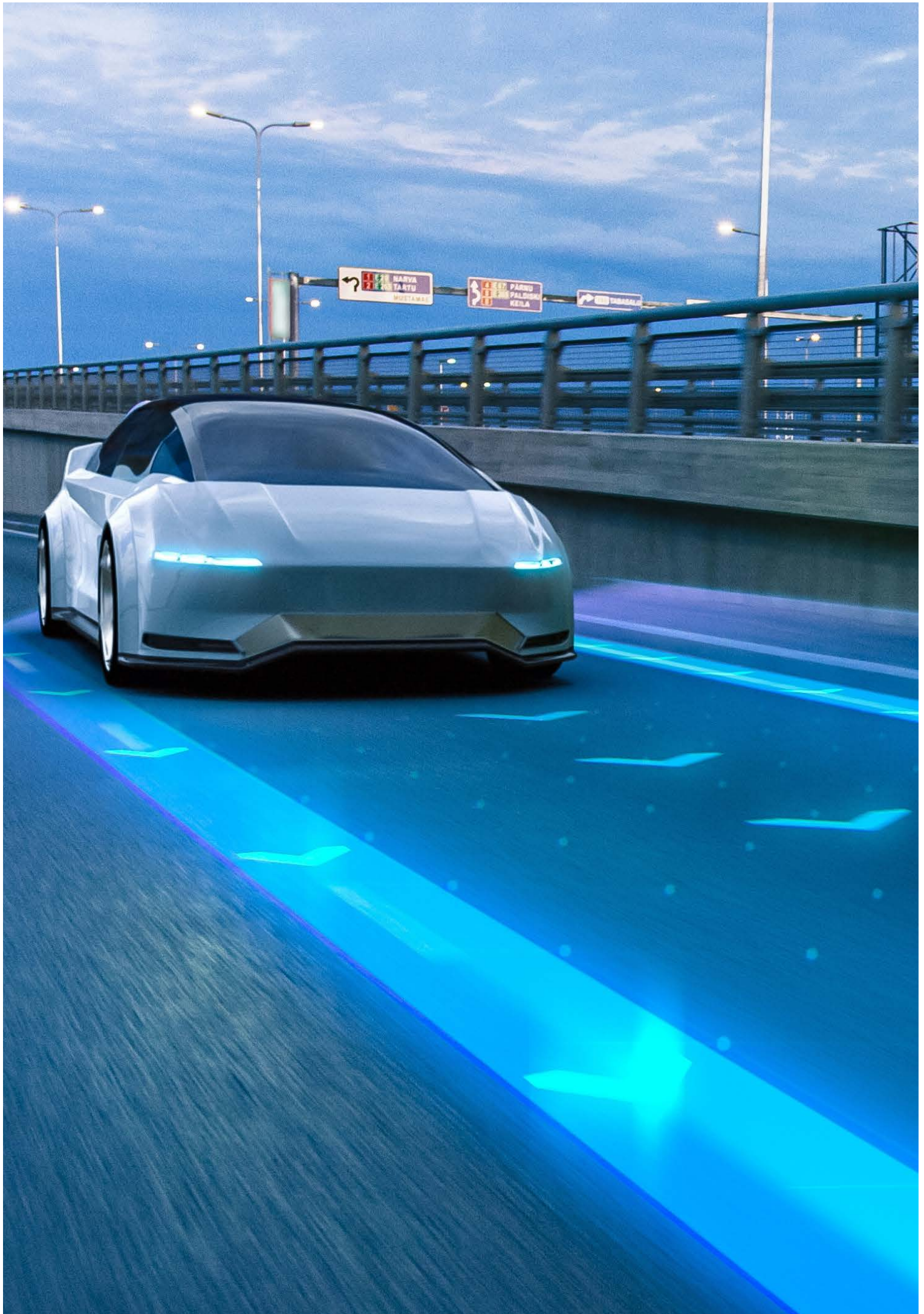
Ensuring that AI-based algorithms are trustworthy in safety-critical applications such as autonomous driving remains a major challenge in research and industry. In this whitepaper, we outline the development process of a behavior planner for an autonomous vehicle using reinforcement learning and its performance in a research vehicle – including an in-depth overview of the challenges that come with to ensure that the developed system is indeed trustworthy.

The majority of this whitepaper is focused on developing a behavior planner using reinforcement learning in simulation. We defined and evaluated two types of reward specifications for reinforcement learning that model behavioral specifications within the ODD. Our evaluation found that we need a better understanding of how to apply the priority of behavior requirements to the reward specification in order to determine how trustworthy the system truly is. The whitepaper also outlines two approaches to defining scenario sets for reinforcement learning with different benefits in terms of ODD coverage. A cross-evaluation of these approaches to scenario generation achieves the highest generalization when a mixed set of scenarios is used during learning. We believe this finding calls for further analysis in the future before we can include it in the Deloitte Trustworthy AI framework.

Based on the evaluation of the behavior planner on the prototype vehicle, our proposed development approach has been validated. It also showed, however, just how vital it is to factor the architecture-specific impacts on the system behavior into the development process for the behavior planner. As a result, we need to find a way to include system-specific error patterns as trustworthiness criteria when we develop behavior planners using reinforcement learning.

Ensuring trustworthiness in a safely-critical application requires coverage of the whole AI lifecycle as well as the system components that deliver input in real time. We showed that increasing ODD coverage has a positive impact on the generalization capabilities of our Reinforcement model leading to better performance in the planner.





# Contacts



**Andreas Herzig**

Partner | Risk Advisory  
Tel: +49 711 16554 7160  
aherzig@deloitte.de



**Dr. David Dang**

Manager | Risk Advisory  
Tel: +49 151 14880524  
dadang@deloitte.de



**Xiangzhong Liu**

Researcher  
Trustworthy Autonomous Systems  
Fortiss  
Tel: +49 89 3603522 182  
xliu@fortiss.org



# Deloitte.

Deloitte refers to one or more of Deloitte Touche Tohmatsu Limited (“DTTL”), its global network of member firms, and their related entities (collectively, the “Deloitte organization”). DTTL (also referred to as “Deloitte Global”) and each of its member firms and related entities are legally separate and independent entities, which cannot obligate or bind each other in respect of third parties. DTTL and each DTTL member firm and related entity is liable only for its own acts and omissions, and not those of each other. DTTL does not provide services to clients. Please see [www.deloitte.com/de/UeberUns](http://www.deloitte.com/de/UeberUns) to learn more.

Deloitte provides industry-leading audit and assurance, tax and legal, consulting, financial advisory, and risk advisory services to nearly 90% of the Fortune Global 500<sup>®</sup> and thousands of private companies. Legal advisory services in Germany are provided by Deloitte Legal. Our professionals deliver measurable and lasting results that help reinforce public trust in capital markets, enable clients to transform and thrive, and lead the way toward a stronger economy, a more equitable society and a sustainable world. Building on its 175-plus year history, Deloitte spans more than 150 countries and territories. Learn how Deloitte’s approximately 415,000 people worldwide make an impact that matters at [www.deloitte.com/de](http://www.deloitte.com/de).

This communication contains general information only, and none of Deloitte GmbH Wirtschaftsprüfungsgesellschaft or Deloitte Touche Tohmatsu Limited (“DTTL”), its global network of member firms or their related entities (collectively, the “Deloitte organization”) is, by means of this communication, rendering professional advice or services. Before making any decision or taking any action that may affect your finances or your business, you should consult a qualified professional adviser.

No representations, warranties or undertakings (express or implied) are given as to the accuracy or completeness of the information in this communication, and none of DTTL, its member firms, related entities, employees or agents shall be liable or responsible for any loss or damage whatsoever arising directly or indirectly in connection with any person relying on this communication. DTTL and each of its member firms, and their related entities, are legally separate and independent entities.