F. Beringhoff, J. Greenyer, C. Roesener and M. Tichy, "Robust Realization of Test Scenarios for Varying Behavior of the Automated Vehicle Under Test," 2025 IEEE International Automated Vehicle Validation Conference (IAVVC), Baden-Baden, Germany, 2025, pp. 1-8, doi: 10.1109/IAVVC61942.2025.11219575.
https://ieeexplore.ieee.org/abstract/document/11219575

Robust realization of test scenarios for varying behavior of the automated vehicle under test*

1st Felix Beringhoff
Volkswagen AG
Wolfsburg, Germany
felix.beringhoff@volkswagen.de

2nd Joel Greenyer
FHDW Hannover
Hanover, Germany
joel.greenyer@fhdw.de

3rd Christian Roesener
Volkswagen AG
Wolfsburg, Germany
christian.roesener1@volkswagen.de

4th Matthias Tichy
Ulm University
Ulm, Germany
matthias.tichy@uni-ulm.de

***Abstract*—Closed-loop simulation-based testing of automated driving systems creates new challenges as the vehicle-under-test acts autonomously. Prior to the test of an SAE level 4 automated vehicle, its actual behavior and motion are unknown. This makes it hard for test engineers to model test scenarios, as verifying the automated vehicle's correct functionality requires specific test conditions including interactions with other vehicles. We propose the use of an AI-controlled surrounding vehicle (AISV) which intelligently influences the automated vehicle to induce the desired scenario. Considering evolving behavior of the automated vehicle throughout development, we evaluate the AISV's ability to adapt accordingly. Specifically, this is done for two workflows of using a goal-conditioned reinforcement learning method to create the AISV: First, by directly integrating the policy of a trained agent in the actor models of the simulation. This shows success in 86% of our tested cases. Second, by deriving a simpler yet robust, rule-based surrogate model of the reinforcement learning agent. Our feasibility analysis of this second approach shows close to 100% success rates in four tested scenarios, which makes this approach worth for further research. With both approaches, we want to find recommendations for realizing robust test scenarios for varying behavior of the automated vehicle.**

***Index Terms*—automated driving, scenario-based testing, actor behavior models, simulation, goal-conditioned reinforcement learning.**

I. Introduction

A commonly used method to test automated vehicles (AV) is by evaluating the AV's behavior in certain test scenarios. This method is extensively researched [2], [3] and found its way into regulatory frameworks for the verification and validation of AVs [4]. In a test scenario, the AV is exposed to situations relevant to assessing its functionality.

Sources for test scenarios could be, for example, requirement documents, such as those for automated lane keeping systems (ALKS) [5], expert knowledge or driving data [6], [7]. The scenarios have in common that their description usually starts with an initial condition. In the case of ALKS scenarios, the initial conditions are described by velocities and relative distances between the AV and a surrounding vehicle (SV). On some test benches, the tests cannot be initialized directly in these states. Especially when real hardware is in the loop, the initial conditions of the test must evolve realistically from a standstill of the vehicle to avoid implausible signal values for the electronic control units under test.

*The results, opinions and conclusions expressed in this publication are not necessarily those of Volkswagen Aktiengesellschaft.

Current guidelines for modeling test scenarios primarily deal with the phase following an initial condition and recommend to let an OpenSCENARIO simulation engine to control the preceding phase. However, if this is not possible, a test engineer has to script less reusable scenario wrappers for the ramp-up to the initial condition of the scenario. [8] Additionally, as proprietary scenario formats are still relevant in practice [9], the modeling of scenario scripts for the pre- and post-initial condition phase of the test scenario is still relevant to test engineers. This process, however, is often based more on the individual experience of the test engineers than on common guidelines.

Especially when working with a black-box automated driving system (ADS), the test engineer has to go through several iterations of creating, simulating, and modifying the trigger-action-based SV behavior in test scripts (e.g., OpenSCENARIO-like files) until desired initial conditions of the test scenario are fulfilled. This process might be repeated several times as the AV behavior will change over the course of vehicle development project due to software updates of the ADS. Therefore, the realization of test scenarios can become cumbersome, time-consuming, and blocks capacity of testing resources. This is seen by practitioners as one challenge when testing automated vehicles [10].

To ease this process, we report on further development of our goal-conditioned reinforcement learning (GCRL) approach to control one SV of the AV. In this approach, an RL agent learns how to drive in the surroundings of the AV in order to create given situations, e.g., as shown at the top of Fig. 1. The SV enforces a lane change by the AV to create initial conditions which require the AV to be in the left lane. Those situations are defined through goal conditions, including the relative distance and velocity of the AI-controlled surrounding vehicle (AISV) and the AV [1]. In this paper, we train the AISV with different parametrizations of an AV behavior model and show the applicability of the GCRL approach on differently behaving AVs without changing the reward function.

We understand that some test environments require determinism of the actors during the test execution. This is especially the case when testing with real vehicles on real-world test tracks, or when the exact reproducibility of the tests, also in simulations, is crucial. Therefore, we want to analyze the feasibility of scripting a simpler, deterministic surrogate

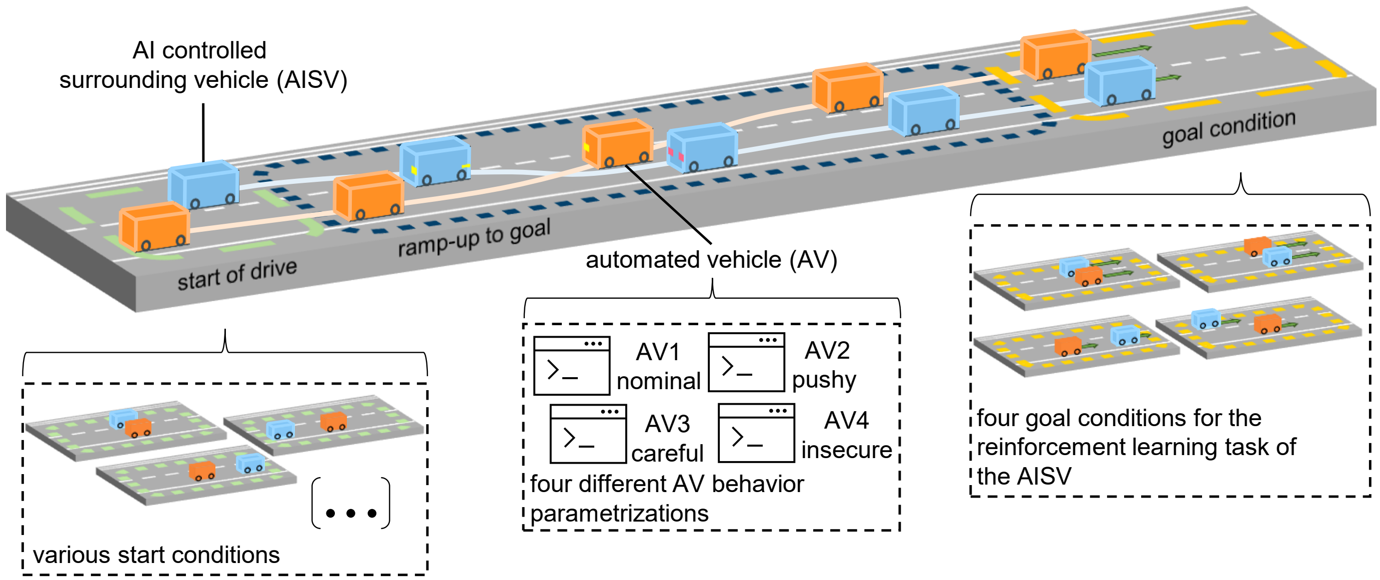


Fig. 1. Concept for the automatic realization of initial conditions of test scenarios through an AI-controlled surrounding vehicle for differently acting AVs, based on our previous work [1] and adapted to the conditions of the experiment from the current paper.

model for the AISV. As a first step, we use an expert-based, creative approach to derive a rule-based model by observing the behavior of the trained RL agents.

The contributions of this paper are:

- experiment results with a GCRL approach to train an SV to realize certain test scenario conditions for differently behaving AVs;
- a feasibility analysis on modeling simpler surrogate agents for the robust scenario realization for differently acting AVs, derived from the observation of the trained RL agents.

The research on robust scenario realization methods should help practitioners to realize test scenarios more efficiently with less scenario-scripting iterations on the test bench itself.

II. RELATED WORK

The behavior of scenario actors is seen as a key for the development of automated vehicles by ADS providers [11]. Depending on the behavior of the scenario actors, scenarios can be clustered in different groups. Wang et al. [12] distinguish between adaptive and non-adaptive scenarios. In non-adaptive scenarios, the actions of scenario actors is fixed and predetermined. The actors do not react to, e.g., the AV. In adaptive scenarios, behavior models for the actors are used. Similarly, Ransiek et al. [13] distinguish between non-reactive actors, which follow predefined trajectories, and reactive actors. Reactive actors can be further split into groups of rule-based actors, which mostly are used to model nominal traffic behavior, and learning based actors, which could also incorporate adversarial agents. Adversarial agents can be trained to enforce critical situations for the AV. In [14], crash-relevant trajectories for an SV are identified and modeled with a GCRL approach. In [15] and [16], the drivable area of the AV is reduced in order find defects of the AV and provoke collisions. Other forms of learning based actors focus on nominal or cooperative driving behavior as in [17], where agents are trained to flow in lane-free traffic without colliding.

In contrast, we train actors, which influence the AV in a desired way. The goal is not necessarily to create crash-prone conditions, but any given initial condition of a test scenario. For that, only a limited number of publications can be found which report from industrial use-cases. Regarding the realization of test scenarios on test tracks, Schöner et al. [18] report on two ways to create trajectories for the moveable objects on test tracks. One is by recording trajectories during manual driving of test drivers. The other way is by iteratively generating scenarios with graphical planning and simulation tools. In the context of resimulations in a software-in-the-loop setting, Scanlon et al. [19] present a multi-step approach to align the trajectory of the ADS with the ones from recorded data. In some cases, e.g., for simulating red light running, it was necessary to simulate a green traffic light as the ADS would not cross a red light. A different approach by the same company uses autoregressive models to inject agents into scenarios [20].

Within our work, we use methods for the modeling adversarial agents to create intended scenario conditions.

III. CONCEPT

To model reactive SVs which aim to achieve a certain scenario goal, we propose to train a reinforcement learning (RL) agent for controlling the SV in a goal-conditioned manner. In goal-conditioned reinforcement learning (GCRL), an agent is trained to solve goal-oriented tasks [21], [22]. The agent is not necessarily specialized for a single task but can perform multiple tasks characterized by different goals. In our

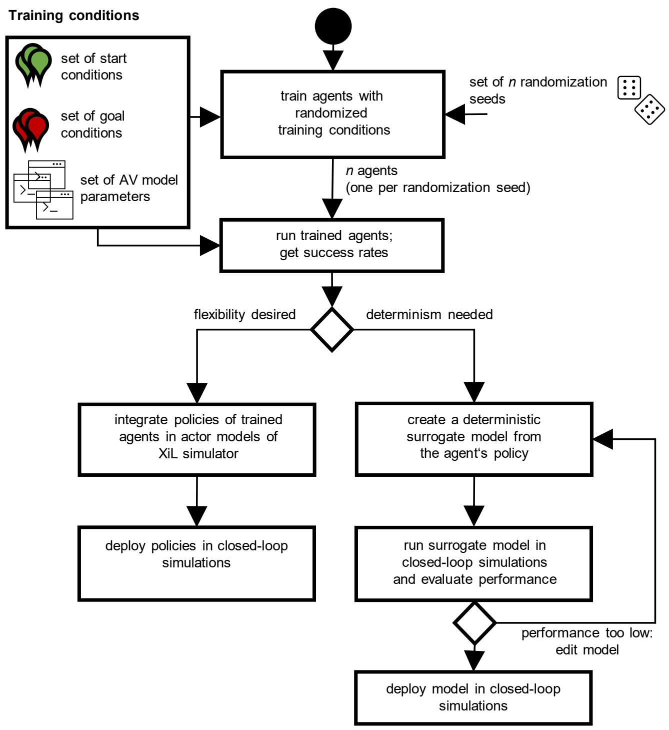


Fig. 2. Two workflows of utilizing the goal-condition reinforcement learning agents for the scenario realization task: one prefers flexibility, the other determinism.

case, the goals consist of scenario parameters, like the relative distance between the SV and the AV, and their targeted lane affiliation. The reward function used for the training of the RL agent only sparsely rewards reaching goal-conditions. This makes the approach applicable for different goal conditions without having to change the reward function. Therefore, the RL agent learns itself a policy by trial and error without any guidance. By varying the behavior of the AV during training, this policy shall be made robust against differently behaving AVs. In the reinforcement learning context, the varying AV behavior is equivalent to a changing environment, which often is a challenge in transfer learning applications [23]. Since our intention is to train the agent in an artificial, simulated environment with an abstract model of an ADS, our approach also faces this challenge when deployed to different simulation environments or test benches with an actual ADS in the loop.

In this paper, we propose two workflows of using our approach as presented in Fig. 2: In workflow 1 (left path in Fig. 2), the policy of the trained agent is embedded in the closed-loop simulation framework and determines new actions based on the current state of the simulation. Workflow 1 has the benefit of being flexible regarding the start states of a scenario and allows for a concatenation of test scenarios. However, this approach comes with limited determinism and limited reproducibility. Although the policy would reproducibly select the same actions with the same observations, unknown states or noise in the observation could lead to unforeseeable actions of the AISV.

With workflow 2 (right path in Fig. 2), we aim to find an explainable surrogate model for the AISV from the RL agent's policy. Workflow 2 is suited when it is not possible to integrate custom traffic agents in the simulation framework or a highly deterministic and reproducible behavior of the actors is required. This can be the case, e.g., in tests on real-world test tracks when safety during the test execution plays a role. For a first feasibility analysis of this workflow, we utilize the simulation results of the RL agents as a teacher for a test engineer who creatively and manually designs the surrogate model for the AISV. By providing example videos and data of the RL agents test runs, the manual scripting process of a deterministic test script should be capable with less iterations. We propose this workflow assuming to automate it by, e.g., agent distillation methods [24] in future work. Similarly to [25], we allow to enhance the performance of the surrogate model through human expert knowledge, as the understandable structure of the surrogate model makes it easy to do so. This is one additional benefit of using an explainable model compared to using a neural network for the AISV policy [25].

IV. EXPERIMENT SETUP

This section describes the setting to train the RL agent for reaching an exemplary set of four goals for a set of four differently behaving AVs. For better understanding, we consider a concrete *task* of an agent as the exercise of reaching a certain goal from a certain start state for one given behavior of the AV. Therefore, a task is described by one combination of elements of the sets of start states, AV behavior types, and goals. In the context of this paper, we define a *scenario* as a tuple of a certain start state and a goal, without specifying the AV behavior.

A. Automated vehicle behavior model

To model differently behaving AVs, we make use of the IDM [26] in combination with the MOBIL lane change decision model [27], and an extension to model patience in lane change decisions [28]. Additionally, we added an accepted velocity difference to the lane change decision. If the leading vehicle is travelling with a velocity of $v_{lead} > v_0(1 - dv_{tol})$, then the AV stays behind the leading vehicle.

The parameter sets should represent four different driving behavior characteristics. Under consideration of the parameter correlation of Kim et al. [29], a nominal, pushy, careful and insecure set is parametrized (see Table I) and visualized in Fig. 3. The AVs use different speed and acceleration profiles leading to different duration until the AV reaches its desired speed from standstill. AV1 and AV2 use the symmetric lane changing rules, while AV3 and AV4 keep to the right lane and only use the left lane for overtaking. The politeness of AV1 leads to lane changes in order to let by a fast approaching car from behind. AV2 immediately changes lanes when a leading car is within its safety headway due to the zero patience in lane change decisions. AV2 has the lowest targeted time headway T of all four AVs. AV3 is hesitant in driving past other vehicles

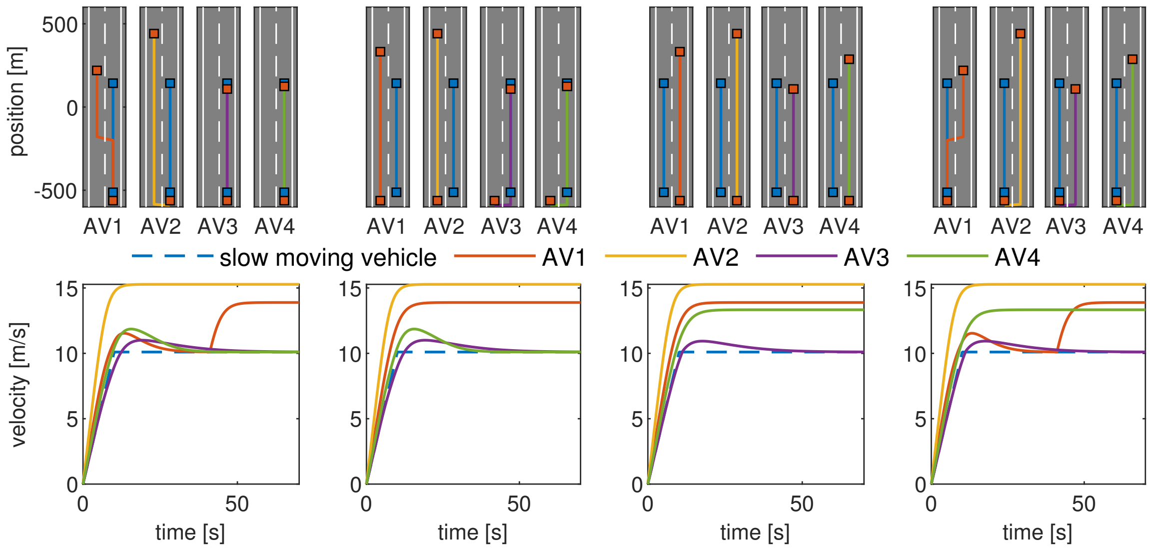


Fig. 3. Four differently behaving AVs confronted with a slower moving vehicle (blue) in four different scenarios (different start states).

TABLE I
AV PARAMTER SETS

Parameters		***AV1***	***AV2***	***AV3***	***AV4***
model	param				
	nickname	nominal	pushy	careful	insecure
IDM	$v_0[km/h]$	50	55	45	48
IDM	$T[s]$	1.27	0.5	2.1	1.0
IDM	$s_0[m]$	2	2	2	2
IDM	$a[m/s^2]$	1.41	2	1	1.2
IDM	$b[m/s^2]$	2.3	3	2	2.5
IDM	$delta[1]$	4	4	4	4
MOBIL	overtaking rule	sym.	sym.	asym.	asym.
MOBIL	$p[1]$	0.8	0	0.8	0
MOBIL	$b_{safe}[m/s^2]$	3	3	3	3
MOBIL	$a_{thres}[m/s^2]$	0	0	0	0
MOBIL	$a_{bias}[m/s^2]$	0.3	0.3	0.3	0.3
MOBIL	$v_{crit}[m/s]$	60	60	20	60
PATIENCE	$dv_{tol}[\%]$	20	0	20	98
PATIENCE	$v_{thres}[m]$	80	0	80	60
VEHICLE	$ax_{max}[m/s^2]$	5	5	5	5
VEHICLE	$ax_{min}[m/s^2]$	-8	-8	-8	-8

TABLE II
OBSERVATIONS OF THE RL AGENT

channel	symbol [unit]
	kinematic states
1	$vx_{AISV}[m/s]$
2	$lane_{AISV}[1]$
3	$x_{rel}[m]$
4	$vx_{rel}[m/s]$
5	$ax_{rel}[m/s^2]$
6	$y_{rel}[m]$
7	$vx_{AV}[m/s]$
8	$lane_{AV}[1]$
	goal
9	$dx_{goal}[m]$
10	$vx_{rel,goal}[m/s]$
11	$lane_{AISV,goal}[1]$
12	$lane_{AV,goal}[1]$
	deviation from goal
13	$dx_{goal} - dx_{rel}[m]$
14	$vx_{rel,goal} - vx_{rel}[m/s]$
15	$lane_{AISV,goal} - lane_{AISV}[1]$
16	$lane_{AV,goal} - lane_{AV}[1]$
	distances and relative velocity to neighboring vehicles
17	$x_{rel,frontmid}$
18	$vx_{rel,frontmid}$
19-28	...
	road
29	$flag_{leftlaneavailable}$
30	$flag_{rightlaneavailable}$

on the right and only does so below 20 km/h. AV4 is very tolerant against slow moving lead vehicles and only overtakes when the lead vehicle drives 98% slower than AV4's desired velocity $v0$.

B. Reinforcement learning agent

The modeled reinforcement learning agent is adapted from our previous work [1] and uses the same seven discrete actions: acceleration of $\{0m/s^2, 1m/s^2, 4m/s^2, -1m/s^2, -4m/s^2\}$ and lane change actions to the left or right lane. The observation space includes the current goal and further driving related signals as shown in Table II.

The reward function is a sparse reward function which rewards reaching the goal state as in [1]. Additionally, at each time step a reward of -1 points is deducted to reward reaching the goal faster. We use the same DQN algorithm and settings for learning as in [1].

a) Selection and modeling of goal conditions: A goal condition is modeled by the target relative distance between AV and AISV $x_{rel,goal}[m]$, the target relative velocity between AV and AISV $vx_{rel,goal}[m/s]$, the target lane of the AISV $lane_{AISV,goal}[1]$, and the target lane of the AV $lane_{AV,goal}[1]$. The tolerance for reaching the dx_{goal} is 4 m, and 1.1 m/s for the $vx_{rel,goal}$. We chose four exemplary goal conditions for our experiments, which shall demonstrate the adjustability of the agent to different goals. All four goal states have a $vx_{rel,goal}$ of 0 m/s. Therefore, we denotate the goal conditions by the differentiating set of $\{x_{rel,goal}, lane_{AISV,goal}, lane_{AV,goal}\}$. Goal 1 ($\{25m, 1, 0\}$ with 1 being the left and 0 the right lane index) can be the initial condition of a test scenario for testing the AVs behavior with approaching vehicles from behind, like in testing of automated lane changing systems. Goal 2 $\{0m, 1, 0\}$ and goal 3 $\{0m, 0, 1\}$ could be the initial conditions for testing the AVs lane merging behavior when the neighboring lane is blocked. Goal 4 $\{-25m, 0, 0\}$ can be the initial condition of a test with a decelerating leading vehicle, as in [4]. We did not include goals which require the AV and the AISV to drive on the left lane as this conditions are unrealistic to achieve with AVs obeying the right-hand drive rule. The number of goals is limited in this paper, which leads to a simpler training setting for the agent and is addressed in the discussion of this paper.

Training and test procedure: The reinforcement learning training environment is modeled in MATLAB/Simulink. During training, a start condition, an AV behavior type, and a goal-condition is randomly selected. A training episode is finished after 700 timesteps of 0.1 seconds, when a collision occurs, or when the goal is reached. The training is either stopped when a reward threshold or a maximum number of 8000 episodes is reached. We independently train three agents with different randomization seed to consider its influence.

For evaluating the agents performance, we simulate the three trained agents in each of the 544 combinations of the training set {start condition, AV behavior type, goal condition} and count the number of times the agent reaches the goal to calculate the success rate.

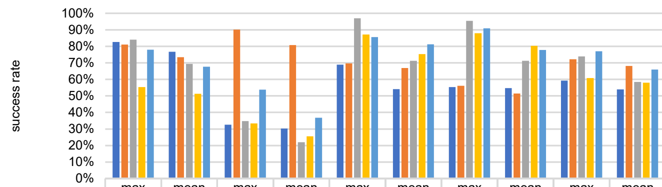
V. TRAINING RESULTS OF THE RL AGENTS FOR DIFFERENTLY BEHAVING AVs

Table III shows the success of the combined best of the three trained agents in realizing the goal from the 34 start states for a given AV behavior type. In 88 % of all tasks, one of the three agents was able to realize the goal-condition successfully. The table shows great differences in the success rates for the different goals and AV behavior types. Especially the success rate for the tasks with AV2 are lower than for the others, indicating that this specific AV behavior leads to a more difficult scenario realization process.

A comparison of the agents trained with varying AV behavior and with specialized agents trained with only one of the four behavior types is shown by Fig. 4. While each specialized agent performs the best when deployed in their trained tasks, all show lower success rates when tested with a differently parametrized AV. This underlines the motivation of the study to make the realization of the goals robust against changes of the AV behavior. The best of the three agents trained with a set

TABLE III
SUCCESS RATES OF THE COMBINED BEST RL AGENT IN THE TRAINING TASKS.

	Goal1 {25m,1,0}	**Goal2** {0m,1,0}	**Goal3** {0m,0,1}	**Goal4** {-25m,0,0}	**total**
AV1	100%	100%	97%	79%	94%
AV2	76%	68%	76%	15%	59%
AV3	100%	100%	100%	100%	100%
AV4	100%	94%	100%	100%	99%
total	94%	90%	93%	74%	88%



trained with:	tested with AV1 max	tested with AV1 mean	tested with AV2 max	tested with AV2 mean	tested with AV3 max	tested with AV3 mean	tested with AV4 max	tested with AV4 mean	tested with varying AV max	tested with varying AV mean
AV1	$\boxed{83\%}$	77%	33%	30%	69%	54%	55%	55%	59%	54%
AV2	81%	73%	$\boxed{90\%}$	81%	70%	67%	56%	52%	72%	68%
AV3	84%	69%	35%	22%	$\boxed{97\%}$	71%	95%	71%	74%	58%
AV4	55%	51%	33%	26%	87%	75%	$\boxed{88\%}$	80%	61%	58%
varying AVs	78%	68%	54%	37%	86%	81%	91%	78%	$\boxed{77\%}$	66%

Fig. 4. Comparison of the success rates of specialized RL agents (trained with one specific AV behavior) and the generalized RL agent (trained with differently behaving AVs). The max and mean values are a result of training three agents with different randomization seeds.

of differently acting AVs shows slightly higher success rates than the ones trained with AV3 and AV2 only and more than 15% higher success rates than AV1 and AV4. Interestingly, this also shows that training with certain agents could favor robustness in the application with other agents.

VI. MODELING A SURROGATE MODEL FOR THE AISV

We opted to create a scenario language independent sequential automata-like model for the scenario script, modeled in MATLAB/Simulink stateflow within our RL training framework. This simple proposed sequential sequence of actions makes the transfer to other scenario language like OpenSCENARIO XML [30] or proprietary formats simple.

We created four different rule-based actor scripts for four different scenarios, shown in Fig. 5. The aim is to find one generalizable script for the SV per scenario which works for all four differently behaving AVs. The Fig. 5 shows the created rule-based actors which are capable to reach the goal in all 16 (4 AVs, 4 scenarios) tested conditions. The rule-based actors use the same seven actions as the AISV. The conditions for proceeding to the next activity are relative velocity, absolute actor velocity, lane indices of actor or AV, relative distance.

In scenario 1, the rule-based agent considers the AV to drive on either lane under free driving and not always on the right. The agent has to force the AV1 and AV2 to drive on the right lane (see Fig. 3 Goal3, AV1, AV2) by decelerating on the left lane until the AV changes to the right lane. If the AV drives

TABLE IV

SUCCESS RATES OF THE GENERALIZED RL AGENTS AND THE RULE-BASED SURROGATE MODELS IN FOUR SCENARIOS WITH 100 RANDOMIZED AV BEHAVIOR PARAMETRIZATIONS.

Scenario	1	2	3	4	total
mean RL agent	83%	67%	46%	65%	65.4%
best RL agent	100%	87%	79%	78%	86%
surrogate models	100%	100%	98%	100%	99.5%

already on the right lane, the condition is immediately true and the agent proceeds with reaching the desired relative distance.

In the scenarios 2-3, the rule-based agent tries to force a lane change by the AV by decelerating down to a standstill (instead of driving just slower) in front of the AV (see Fig. 3 Goal2, AV4). This is modeled by decelerating without defining a minimal velocity, but by defining the lane change action by the AV as condition to proceed with the test script in the *let by* acts of the scenario.

Scripting the agent for scenario 4 is the most challenging due to AV2's defensive behavior against the given goal-condition. The goal condition is usually avoided by AV2 as the AV changes to the left lane as the reaction to closely leading SV. For the AISV, cutting in in front of the AV at the desired distance (see Fig. 3 Goal4, AV2) seems to be the only way to realize the goal condition for AV2.

VII. SUCCESS RATES OF RL AGENTS AND SURROGATE MODELS IN TESTS WITH UNKNOWN AV BEHAVIOR

For testing the robustness against varying behavior of the AV, we deploy the RL agents and the surrogate models in the four scenarios from Fig. 5 with 100 randomized parametrizations of the AV. The parameter values are within the ranges of the values used in training, see Table I.

Tab. IV shows the success rates of the RL agents trained with varying AVs and the rule-based surrogate models in the four scenarios from Fig. 5. The mean performance of the three RL agents is similar to the success rates on the training set. However, the mean success rate of the best performing agent per scenario is with 86% even higher than in the training set. This potentially is due to lesser occurrence of behavior characteristics similar to AV2's behavior in training.

The rule-based agents reach the goal-conditions in 99.5% of all runs. This indicates that it is likely to find a generalizable automata-like script for the SV which is capable to cope with a wide range of differently behaving AVs.

VIII. DISCUSSION

Our experiment aimed to show the adaptability of our GCRL approach to diverse AV behaviors and propose a new workflow for distilling explainable and robust scenario scripts for applications with high demands on determinism.

Regarding the GCRL approach, we could show that using the sparse reward function allows an exchange of the AV behavior without having to make adaptions to the RL part of the simulation framework. The AISV can be trained with

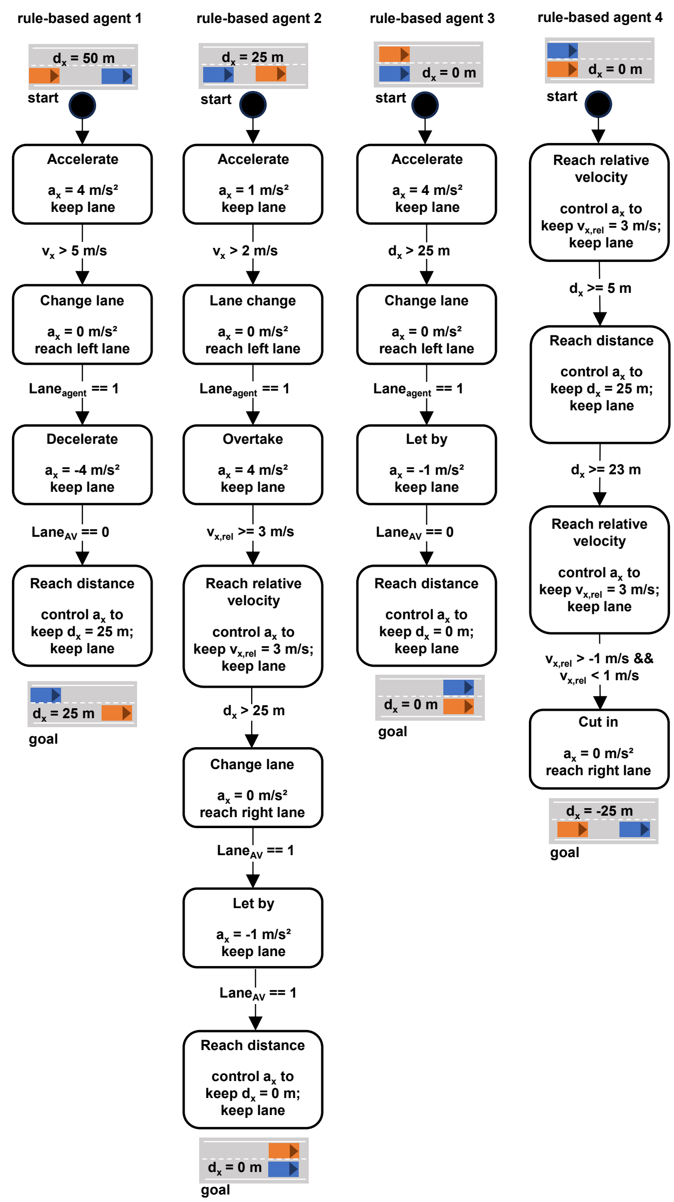


Fig. 5. Four rule-based agents for four different scenario; designed to be robust against varying behavior of the AV.

differently behaving AVs and still shows high success rates when deployed. The RL agents, however, are not yet able to solve all the presented tasks. Anyhow, the agent found ways to realize all goal-conditions for all AVs at least from some of the start states. To improve the success rates, newest GCRL methods could increase performance by using experience replay to increase sample efficiency [21].

Regarding the modeling of a surrogate model for the AISV, we initially proposed a manual process to evaluate the feasibility of this approach. The amount of goals and AV types used in this study was limited for practicality reasons, which limits the validity of our study. However, it should be considered as a starting point for further research as our results show great success in modeling simple yet robust automata-like

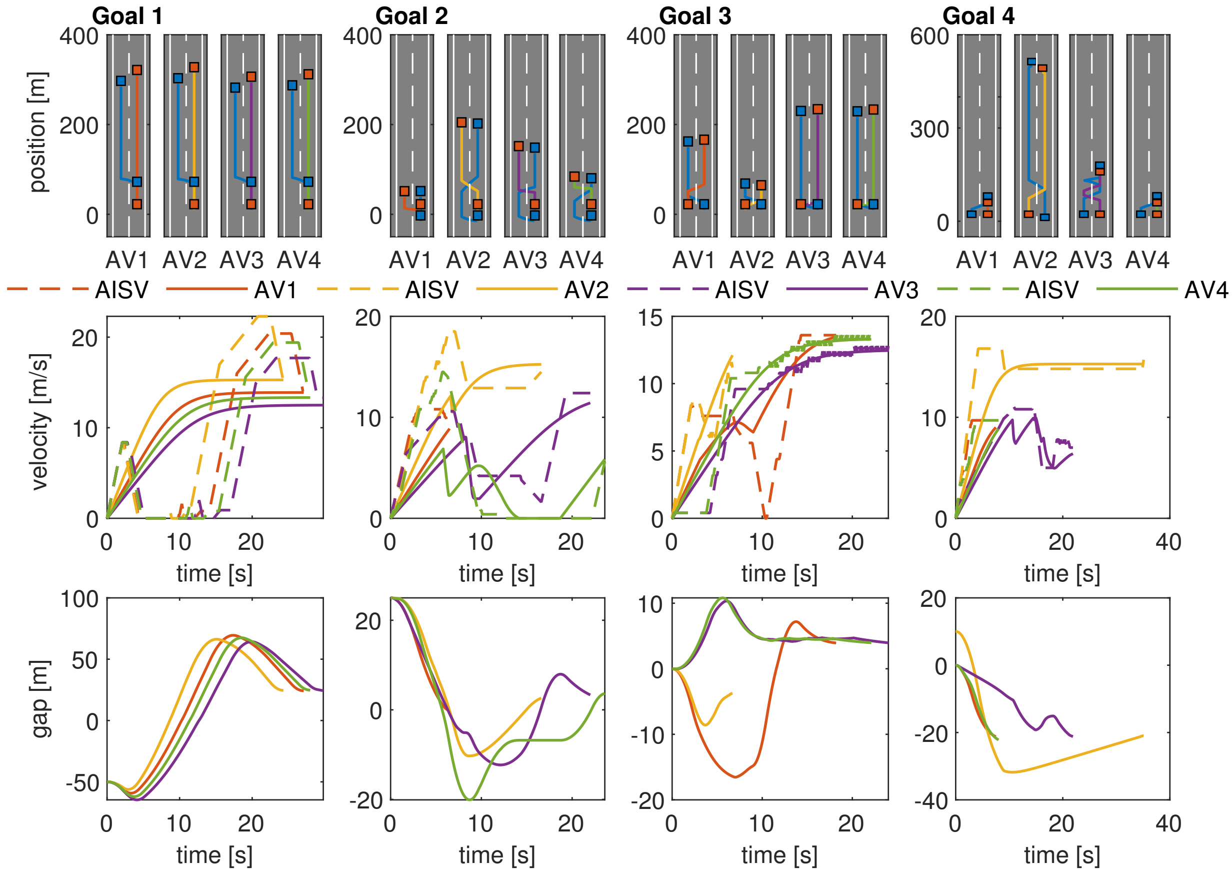


Fig. 6. Ways of realizing the four goal conditions by the AISV.

scripts for the AISV. An automatic approach for the agent distillation should be considered for scaling the approach to more scenarios. Furthermore, finding a generic *skeleton* for the surrogate model could be an outcome of future work. In our case, it was possible to find one generic structure of the rule-based agents per scenario which works for almost all the tested AV behaviors. This also opens up the possibility of applying process mining methods or directly learning interpretable agents for the AISV instead of learning neural networks. However, in regard of more complex scenarios with several agents and more complex goal conditions, the GCRL approach might be easier to scale.

Due to the white box AV model used for our study, it would be possible to directly derive the rule-based agents by a static analysis of the AV model, as the parameters already suggest how to trigger certain actions by the AV. However, our approach shows a method which is independent of the internal structure of the AV model by first training an RL agent and then deriving rules from its behavior instead of from the AV's behavior. Therefore, the accuracy of the AV model used for training of the RL agent is important, further underlining the motivation of making the scenario realization robust against differently behaving AVs. For example, modeling patience in the AV behavior has a crucial impact in the behavior of the AISV. This leads the AISV to learn to brake to a stop in order to provoke lane changes by the AV. We deployed the policy of an RL agent to a HiL testbench used in practice and trialed it to realize test conditions for a prototypical stack of an SAE level 4 ADS. Here, the AISV trained with a patient AV model was able to nudge the AV to change lanes while others were not. This means that observations from the ADS under test should feed back into the AV model used to train the AISV.

The goal conditions in the present study focused on urban speed ranges, aligning with the operational domains of current robotaxi applications. Although the goal conditions are yet simple, they can already be used for evaluating basic functionality of robotaxi features like inner urban lane change behavior or rerouting features. However, this simplicity also limits the application of the approach in more complex urban scenarios for validation testing, indicating a need for further development. Our future work focuses on the extension of our approach on multi-agent scenarios, investigating the feasibility of centralized and decentralized multi-agent reinforcement learning methods, but also explainable methods. One further consideration is to control only the most important objects in the near distance to the vehicle-under-test by the proposed agents while other actors follow a nominal or scripted behavior. This makes it easy to extend the scenarios by vulnerable road users on non-driveable lanes although they are not yet considered in our current RL training setup.

IX. CONCLUSIONS

In this paper, we presented two methods for the robust realization of test conditions for differently behaving AVs. Starting point for both methods is the training of goal-conditioned reinforcement learning agents for controlling an SV. The success rate of the best AI-controlled vehicle in reaching its goals is decent, but cannot guarantee a successful realization of test conditions in all scenario settings. Certain behavior characteristics of the AV lead to a prevention of certain situations, which makes the realization of this scenario conditions very difficult. However, 88 % of the training tasks could be solved by at least one of the three trained agents, indicating that the RL agents are capable of finding ways to realize even those edge-cases. In a second step, we analyzed the feasibility of deriving a simpler surrogate model for the AISV. This approach enhances determinism, reproducibility, and transferability to different simulation frameworks. Our results indicate that it is possible to define generic scenario procedures and rules that work across differently behaving AVs. The rule-based agents even outperform the RL agents in the tested tasks, however, modeling the rule-based agents utilized expert knowledge and required manual effort.

Future work should focus on the extension of the GCRL approach to multi-agent problems in which manual and iterative scenario scripting rapidly becomes time-robbing and therefore requires automated methods. Furthermore, directly learning interpretable models for the AISV for simple scenarios should be considered as our results indicate the feasibility of using simple and interpretable representations of the AISV policy for simple scenario conditions.

REFERENCES

[1] F. Beringhoff, J. Greenyer, C. Roesener, and M. Tichy, "Realizing scenario-based verification tests of automated vehicles with an ai-controlled surrounding vehicle in a practice-relevant context," 2024.

[2] 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. 87 456–87 477, 2020.

[3] D. Nalic, T. Mihalj, M. Baeumler, M. Lehmann, A. Eichberger, and S. Bernsteiner, "Scenario based testing of automated driving systems: A literature survey," 2020.

[4] UNECE, GRVA and IWG on VMAD, "New assessment/test method for automated driving," 2022, wP.29, 187th session.

[5] UNECE, "Un-regulation no. 157." [Online]. Available: https://unece.org/transport/documents/2021/03/standards/un-regulation-no-157-automated-lane-keeping-systems-alks

[6] B. Schütt, J. Ransiek, T. Braun, and E. Sax, "1001 ways of scenario generation for testing of self-driving cars: A survey," in *2023 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2023, pp. 1–8.

[7] C. Roesener, F. Fahrenkrog, A. Uhlig, and L. Eckstein, "A scenario-based assessment approach for automated driving by using time series classification of human-driving behaviour," in *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*. IEEE, 2016, pp. 1360–1365.

[8] ASAM e.V., "Asam openscenario, user guide, 9.1 writing reusable scenarios," https://www.asam.net/static_downloads/public/asam-openscenario/2.0.0/user-guide/writing_reusable_scenarios.html [2025-04-08].

[9] L. Eisemann, N. Neis, J. Beyerer, and J. Maucher, "Repetition, replay, replica - a survey on the understanding of resimulation in the context of automated driving," in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, 2024, pp. 529–536.

[10] F. Beringhoff, J. Greenyer, C. Roesener, and M. Tichy, "Thirty-one challenges in testing automated vehicles: Interviews with experts from industry and research," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 360–366.

[11] N. Montali, J. Lambert, P. Mougin, A. Kuefler, N. Rhinehart, M. Li, C. Gulino, T. Emrich, Z. Yang, S. Whiteson, B. White, and D. Anguelov, "The waymo open sim agents challenge," in *Advances in Neural Information Processing Systems Track on Datasets and Benchmarks*, 2023.

[12] Z. Wang, J. Ma, and E. M. Lai, "A survey of scenario generation for automated vehicle testing and validation," *Future Internet*, vol. 16, no. 12, p. 480, 2024.

[13] J. Ransiek, P. Reis, and E. Sax, "Adversarial and reactive traffic agents for realistic driving simulation," *arXiv preprint arXiv:2409.14196*, 2024.

[14] J. Ransiek, J. Plaum, J. Langner, and E. Sax, "Goose: Goal-conditioned reinforcement learning for safety-critical scenario generation," *arXiv preprint arXiv:2406.03870*, 2024.

[15] M. Althoff and S. Lutz, "Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1326–1333.

[16] M. Klischat and M. Althoff, "Generating critical test scenarios for automated vehicles with evolutionary algorithms," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 2352–2358.

[17] A. Karalakou, D. Troullinos, G. Chlkiadakis, and M. Papageorgiou, "Deep reinforcement learning reward function design for autonomous driving in lane-free traffic," *Systems*, vol. 11, no. 3, 2023.

[18] H.-P. Schöner and W. Hurich, "Testing with coordinated automated vehicles," in *Handbook of Driver Assistance Systems*, H. Winner, S. Hakuli, F. Lotz, and C. Singer, Eds. Cham: Springer International Publishing, 2016, pp. 261–276.

[19] J. M. Scanlon, K. D. Kusano, T. Daniel, C. Alderson, A. Ogle, and T. Victor, "Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain," *Accident Analysis & Prevention*, vol. 163, p. 106454, 2021.

[20] R. Mahjourian, R. Mu, V. Likhosherstov, P. Mougin, X. Huang, J. Messias, and S. Whiteson, "Unigen: Unified modeling of initial agent states and trajectories for generating autonomous driving scenarios," 2024.

[21] M. Liu, M. Zhu, and W. Zhang, "Goal-conditioned reinforcement learning: Problems and solutions," 2022. [Online]. Available: https://arxiv.org/abs/2201.08299

[22] L. P. Kaelbling, "Learning to achieve goals," in *IJCAI*, vol. 2. Citeseer, 1993, pp. 1094–1098.

[23] S. Padakandla, "A survey of reinforcement learning algorithms for dynamically varying environments," *ACM Comput. Surv.*, vol. 54, no. 6, Jul. 2021. [Online]. Available: https://doi.org/10.1145/3459991

[24] Y. Bekkemoen, "Explainable reinforcement learning (xrl): a systematic literature review and taxonomy," *Machine Learning*, vol. 113, no. 1, pp. 355–441, 2024.

[25] A. M. Roth, J. Liang, and D. Manocha, "Xai-n: Sensor-based robot navigation using expert policies and decision trees," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 2053–2060.

[26] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[27] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," *Transportation Research Record*, vol. 1999, no. 1, pp. 86–94, 2007.

[28] X. Gong, Y. Guo, Y. Feng, J. Sun, and D. Zhao, "Evaluation of the energy efficiency in a mixed traffic with automated vehicles and human controlled vehicles," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1981–1986.

[29] J. Kim and H. S. Mahmassani, "Correlated parameters in driving behavior models: Car-following example and implications for traffic microsimulation," *Transportation Research Record*, vol. 2249, no. 1, pp. 62–77, 2011.

[30] ASAM e.V., "Asam openscenario xml." [Online]. Available: https://www.asam.net/standards/detail/openscenario-xml/