DeepScenario: An Open Driving Scenario Dataset for Autonomous Driving System Testing

Chengjie Lu
Simula Research Laboratory
Oslo, Norway
chengjielu@simula.no

Tao Yue
Simula Research Laboratory
Oslo, Norway
taoyue@ieee.org

Shaukat Ali
Simula Research Laboratory
Oslo, Norway
shaukat@simula.no

***Abstract*—With the rapid development of autonomous driving systems (ADSs), testing ADSs under various environmental conditions has become a key method to ensure the successful deployment of ADS in the real world. However, it is impossible to test all the scenarios due to the inherent complexity and uncertainty of ADSs and the driving tasks. Further, testing of ADSs is expensive regarding time and computational resources. Therefore, a large-scale driving scenario dataset consisting of various driving conditions is needed. To this end, we present an open driving scenario dataset** DeepScenario**, containing over 30*K executable* driving scenarios, which are collected by 2880 test executions of three driving scenario generation strategies. Each scenario in the dataset is labeled with six attributes characterizing test results. We further show the attribute statistics and distribution of driving scenarios. For example, there are 1050 collision scenarios, in 917 scenarios there were collisions with other vehicles, 105 and 28 with pedestrians and static obstacles, respectively. Target users include ADS developers who need to validate their systems under various environmental conditions.**

***Index Terms*—autonomous driving system testing, driving scenario, open source, dataset**

I. Introduction

Autonomous Driving Systems (ADSs) are critical systems that require extensive quality assurance before they can operate safely in the real world. To test ADSs, autonomous driving simulation is often adopted to generate driving scenarios, where an ADS is deployed in a virtual environment and tested under various driving conditions as it interacts with the environment. However, in practice, ADS testing is very expensive since, due to the complexity and uncertainty of ADSs themselves and their operating environments, the number of possible driving scenarios for testing ADSs is infinite [1]. Furthermore, generating critical driving scenarios is very costly with regard to time costs and computational resources, which mainly come from the use of simulation for searching and evaluating critical scenarios [2]. These challenges lead to the need for an open driving scenarios dataset that has the potential to facilitate the development and evaluation of ADSs.

Recently, several benchmark suites and datasets for training, testing, and evaluating Deep Neural Network (DNN) models in ADSs [3], [4], [5], [6], [7] have been proposed, which is mainly constructed using stereo sensor data (e.g., images and LiDAR point clouds). These datasets usually target on single ADS modules such as visual recognition and object detection models and aim to improve visual understanding of various driving conditions [4], [8]. However, data from such datasets are scenes that describe snapshots of the environment's state without involving sufficient time series information.

This paper introduces DeepScenario, an open driving scenario dataset consisting of more than 30*K* driving scenarios, which focuses on system-level ADS testing. Each driving scenario in this dataset describes the environment state changes over a period of time in which an ADS operates and is tested regarding its ability to operate safely or comfortably. The contributions of this paper are summarized as follows: (1) We present DeepScenario dataset with more than 30*K executable* driving scenarios. To create the dataset, we have executed three driving scenario generation strategies for 2880 times, on four roads and four real-world weather conditions; (2) To facilitate the recording and executing of driving scenarios, we provide a toolset for automatically collecting and replaying scenarios; (3) To characterize driving scenarios with test execution results, we labeled each scenario with six attributes indicating the extent of safety/comfort violation; (4) Based on the six attributes, we further show statistics of the driving scenarios, including distributions of driving scenarios regarding collision object types, the speed at the time of collision, etc.

Dataset availability. The dataset is available in our GitHub online repository[1], containing DeepScenario dataset and DeepScenario toolset usage examples of collecting and replaying driving scenarios. We also plan to release the dataset in a permanent repository, such as in Zenodo, once it is accepted.

II. Methodology

Fig. 1 presents the overview of the scenario dataset generation process. As the figure shows, to generate driving scenarios, several *Test Setups* need to be specified. Then we execute an *Environment Configuration Framework* proposed in *DeepCollision* [9], which generates critical driving scenarios and tests an ADS by configuring its operating environment. After the test executions, the test results are further used for *Dataset Creation*. We describe each component in detail as follows:

A. Environment Configuration Framework

As shown in Fig. 1, the framework integrates three *Configuration Strategies* to support the environment configuration

[1] https://github.com/Simula-COMPLEX/DeepScenario

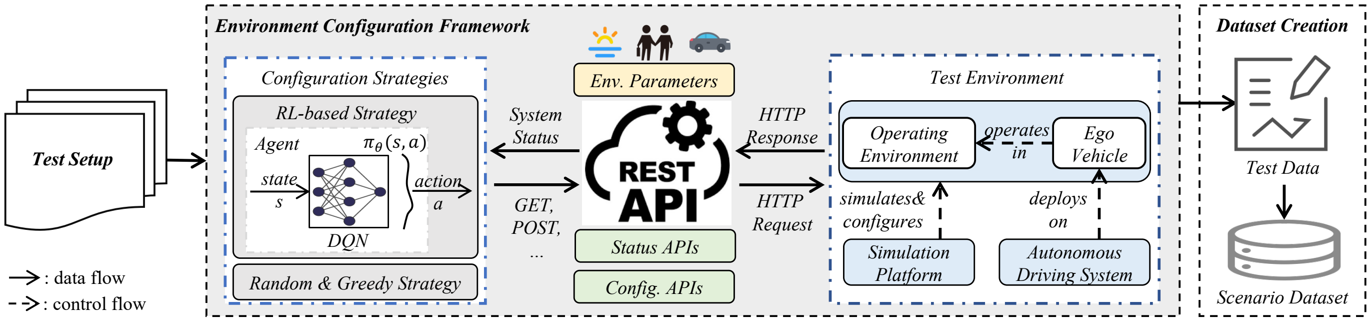

Fig. 1. DeepScenario Dataset Generation Process

generation process, which are a reinforcement learning (RL)-based Strategy (*RLS*), a Random Strategy (*RS*), and a Greedy Strategy (*GS*). *RLS* employs Deep Q-Learning [10] as the RL solution, where an agent constantly interacts with the operating environment and learns environment configurations through observations (states), actions, and rewards. As two comparison baselines for the *RLS*, *RS* and *GS* were also integrated into the framework to generate environment configurations randomly and greedily, respectively. We extracted a list of configurable environment parameters (i.e., *Env. Parameters*) and implemented the invocations of these parameters as *REST APIs* [11] for configuring the environment, which includes *Status APIs* that obtains the status of the vehicle and its operating environment through *HTTP Response*, and *Configuration APIs* that configure the operating environment through *HTTP Request*. The *Test Environment* is a virtual environment, which employs a *Simulation Platform* (e.g., LGSVL [12]) to simulate and configure an operating environment, in which the *ego vehicle* controlled by an *ADS* (e.g., Apollo [13]) operates. More details about the framework can be found in the paper [9].

B. Test Setup and Execution

Test Setup. Before executing the framework, several test setups need to be specified manually. These test setups include setting up driving tasks, weather conditions, and setups for configuration strategies. Specifically, we define driving tasks using four different roads (*R1...R4*) in San Francisco with various road structures [14]. To test an ADS under various weather conditions, we introduced real-world weather data as well as its changes over time of four different days (i.e., rainy day (*RD*), rainy night (*RN*), sunny day (*SD*), sunny night (*SN*)) into the simulation. The weather data is selected from OpenWeather [15], which is an open-source online weather database and can be accessed via weather APIs. In terms of setups for the three strategies, we selected time to collision (*TTC*), distance to obstacles (*DTO*), and *Jerk* as the safety/comfort measures and defined three reward functions R_{TTC}, R_{DTO}, and R_{Jerk} as guidelines of the three environment configuration strategies. Smaller *TTC*/*DTO* values indicate higher safety risk, while larger *Jerk* values indicate less comfort.

Test Execution. We ran each *RLS*, *RS*, and *GS* 20 times on each test setup. In total, we obtained test data of 2880 executions (20 runs × four roads × three strategies × three reward functions × four real-world weather data).

C. Dataset Creation

Preprocessing Test Data. Test executions resulted in over 38*K* test data, including information on test results and the generated driving scenarios. We performed preprocessing to remove duplicated scenarios from the original data to reduce the unnecessary effort when performing further analysis. Finally, we obtained a dataset containing about 33*K* scenarios.

Labeling Scenario with Attributes. After collecting and preprocessing driving scenarios, we labeled each driving scenario with several attributes, like collision, *TTC*, *DTO*, which characterize driving scenarios. The detailed description of driving scenarios and their attributes will be discussed in Section IV.

III. Scenario Definition and Tools

A. Scenario Definition

A driving scenario S describes the temporal development between several scenes, where a scene describes a snapshot of the environment [16]. We define a driving scenario S as a tuple with several scenes: $S =< scene_1, scene_2, ..., scene_n, T >$, where T is the time period that S spans and n denotes the number of scenes in S. A scene sc is defined as a 3-tuple: *sc* = <*ego story*, *obstacle story*, *environment state*>, where 1) *ego story* denotes the operations and kinetic parameters of the autonomous vehicle; 2) *obstacle story* denotes the operations and kinetic parameters of the static and dynamic obstacles; 3) *environment state* includes weather conditions, time of day, and the driving tasks. To facilitate the recording and replaying of driving scenarios, we further developed a Domain Specific Language (DSL) for scenario representation and evaluation based on the scenario definition. The scenario format is $.deepscenario$, which is an XML-based file format.

B. Scenario Toolset

We developed *ScenarioCollector* to automatically collect driving scenarios. *ScenarioCollector* has already been integrated into the environment configuration framework and is able to collect and store the driving scenarios in scenario file format. *ScenarioRunner* was developed to support replaying driving scenarios by taking scenario files as inputs. Specifically, it has two ways of replaying a driving scenario. First, it can exactly replay the behaviors of the ego vehicle and obstacles by reading their kinetic parameters. By selecting and replaying driving scenarios in this way, *ScenarioRunner* can facilitate further analysis and diagnoses. Furthermore, *ScenarioRunner* can also integrate different *ADSs*. In this way, the behaviors of the ego vehicle are not replayed by *ScenarioRunner* but controlled by the *ADS*, and the behaviors of obstacles can still be accurately replayed. This way enables testing various *ADSs* by integrating ADSs in the replayed driving conditions.

IV. DeepScenario Dataset Description

A. Dataset Overview

Table I presents the statistics of DeepScenario dataset across strategies and rewards. As the table shows, we have created the dataset with 33530 driving scenarios, among which 6703 are generated by *RLS*, which are less than the number of scenarios generated by *RS* (i.e., 13565) and *GS* (i.e., 13262). When looking at each strategy on each reward setting, *RLS* generates fewer scenarios on all the reward settings compared to *RS* and *GS*. This is because test execution finishes earlier in *RLS*, which is usually caused by the occurrence of collisions. We can also see that the differences among the three reward functions are not obvious.

TABLE I
Statistics of DeepScenario dataset across Strategies and Rewards*

Strategy	R_{TTC}	R_{DTO}	R_{Jerk}	$Total_{Strategy}$
RLS	2025	1954	2724	6703
RS	4522	4640	4403	13565
GS	4366	4434	4462	13262
$Total_{Reward}$	10913	11028	11589	**33530**

[*] *RLS: RL-based Strategy; RS: Random Strategy; GS: Greedy Strategy.*

B. Scenario Distribution

We show scenario distribution in DeepScenario across roads (*R1*...*R4*) and real-world weather data (*RD*, *RN*, *SD*, *SN*).

Distribution Across Roads. In the dataset, the number of scenarios generated on each road is 8555 (*R1*), 8590 (*R2*), 8619 (*R3*), 7765 (*R4*), respectively. Furthermore, we show the distribution of scenarios across roads on different strategies and reward settings in Fig. 2. As the figure shows, *R4* has the least or second least number of scenarios for *RLS*, and the least scenarios were generated on *R4* for both *RS* and *GS*. This is because *R4* has more complicated road structures (e.g., sidewalks, two-lane) than others, which is more prone to collisions or traffic jams; therefore, test execution ends earlier.

Distribution Across Weather Conditions. As for the distribution across weather conditions, there is not much difference observed between the four kinds of weather conditions, which are 8378 (*RD*), 8399 (*RN*), 8291 (*SD*), 8462 (*SN*), respectively. One plausible explanation is that the ADS (i.e., Apollo) we used is advanced enough to handle different weather conditions. Fig. 3 further shows the distribution of scenarios across weather conditions on different strategies and reward settings, which is consistent with the above observations.

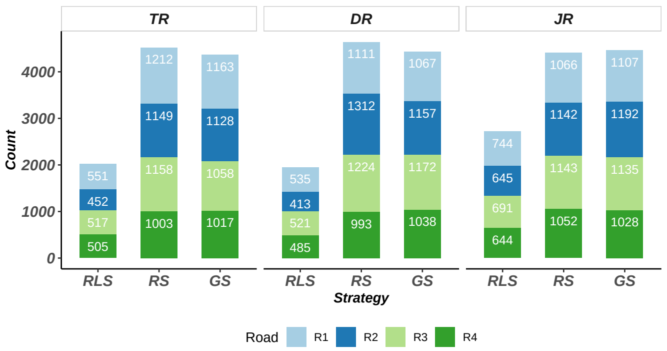

Fig. 2. Distribution of scenarios across roads. (*RLS*: *RL-based Strategy*, *RS*: *Random Strategy*, *GS*: *Greedy Strategy*; *TR*: R_{TTC}, *DR*: R_{DTO}, *JR*: R_{Jerk})

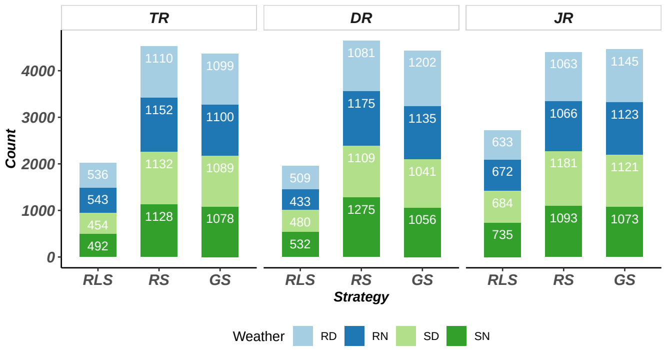

Fig. 3. Distribution of scenarios across weathers. (*RLS*: *RL-based Strategy*, *RS*: *Random Strategy*, *GS*: *Greedy Strategy*. *TR*: R_{TTC}, *DR*: R_{DTO}, *JR*: R_{Jerk}). *RD: Rainy Day*; *RN: Rainy Night*; *SD: Sunny Day*; *Sunny Night*.

C. Driving Scenario Attributes

To characterize driving scenarios in the dataset, we associate each scenario with six attributes, which are calculated based on the test results of driving scenarios. Specifically, for a scenario $\mathcal{S}$, its attributes can be classified into two types which are defined as follows. ***Reward Attributes*** are attributes related to the safety/comfort measures used to define reward functions, and *TTC*, *DTO*, and *Jerk* are the three reward attributes, which measure the extent of safety/comfort when driving in $\mathcal{S}$. ***Collision Attributes*** are collision-related attributes, and we have defined three collision attributes: *Collision (COL)* is a Boolean attribute indicating if the ego vehicle collided with obstacles in $\mathcal{S}$. *Collision-Type (COLT)* is an enumerated attribute that shows the type of obstacle the ego vehicle collided with. Concretely, *COLT* has three possible values, which are *Non-player character (NPC) Vehicle*, *Pedestrian*, and *Static Obstacle*. *Speed-At-Collision (SAC)* is an attribute that records the speed at which the ego vehicle in $\mathcal{S}$ collided (if it happened) with the obstacle.

D. Scenario Attribute Statistics

Using scenario attributes, we can select safety/comfort-critical scenarios and analyze them to support diagnoses of ADSs. Hence, we further present the statistics of scenario attributes.

Statistics of Reward Attributes. Fig. 4 shows the distribution and mean values (red dots) of reward attributes. As the figure shows, regarding the statistics across different strategies, we can observe that *RLS* outperformed *RS* and *GS* in terms of all three reward attributes. As for the statistics across different reward functions, one can see that R_{TTC} achieved the best performance regarding *TTC* and *Jerk* attributes. And R_{DTO} performed the best regarding *DTO*.

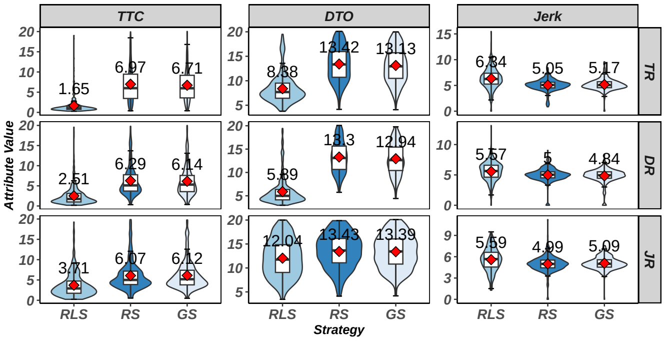

Fig. 4. Statistics of reward attributes across strategies and reward functions. Smaller *TTC*/*DTO* values indicate higher safety risk, while larger *Jerk* values indicate less comfort. (*RLS*: *RL-based Strategy*, *RS*: *Random Strategy*, *GS*: *Greedy Strategy*; *TR*: R_{TTC}, *DR*: R_{DTO}, *JR*: R_{Jerk})

Statistics of Collision Attributes. Table II presents the statistics of collision attributes across strategies and reward functions, where *#C* is the number of collision scenarios, i.e., *COL* is *True*, and we show the average *SAC* in the table. As shown in Table II, there are a total of 1050 collision scenarios generated, among which 700 were generated by *RLS*, while only 158 and 192 were generated by *RS* and *GS*, respectively. The statistics across strategies show that *RLS* achieved the highest *#C* with the highest *SAC*, while *RS* performed the worst. As for the statistics of *COLT*, in total, 917 scenarios collided with NPC vehicles, while only 105 and 28 collided with pedestrians and static obstacles, respectively. Further, when the ego vehicle collides with pedestrians, *SAC* achieved the maximum, and *SAC* is the minimum when colliding with static obstacles. Regarding the statistics across reward functions, R_{TTC} obtained the highest *#C* (i.e., 370) with the second highest *SAC* (i.e., 4.57 m/s), while R_{Jerk} achieved the highest *SAC* (i.e., 4.65 m/s) with the least *#C* (i.e., 319).

V. Dataset Usage And Limitations

A. Dataset Usage

DeepScenario dataset can be used in various ADS development contexts and can facilitate the development and validation of ADSs. We present the usages of our dataset as follows.

Testing ADSs to detect system-level failures. As described earlier, we developed *ScenarioRunner* for running scenarios. By running scenarios, we can deploy an ADS in the environment and perform testing to identify system-level failures of the ADS. Moreover, various ADSs or various versions of an ADS can be integrated with *ScenarioRunner*, which can be tested by running scenarios.

Analyzing scenarios for further diagnoses. By using attributes of the driving scenarios, we can easily select critical scenarios from the dataset that caused higher safety/comfort risk or collisions of the ego vehicle. The selected scenarios can be replayed with *ScenarioRunner*, which can facilitate further analysis and diagnoses of ADSs. For example, we can analyze collision scenarios with respect to collision type and identify key environmental factors for different types of collisions.

TABLE II
Statistics of Collision Attributes Across Strategies and Reward Functions*

ST	*COLT*	R_{TTC}		R_{DTO}		R_{Jerk}		$\#C_{ST}$	SAC_{ST}
		#C	*SAC*	*#C*	*SAC*	*#C*	*SAC*		
RLS	*NPC*	250	4.97	222	3.76	153	5.83	**625**	**4.75**
	PED	2	6.27	21	5.04	27	5.27	**50**	**5.21**
	STA	4	6.08	5	3.30	16	3.39	**25**	**3.80**
	SUM	**256**	**5.00**	**248**	**3.86**	**196**	**5.55**	**700**	**4.75**
RS	*NPC*	36	3.31	47	3.17	53	2.87	**136**	**3.09**
	PED	5	3.49	8	4.54	9	3.88	**22**	**4.03**
	STA	0	N/A	0	N/A	0	N/A	**0**	**N/A**
	SUM	**41**	**3.34**	**55**	**3.37**	**62**	**3.02**	**158**	**3.22**
GS	*NPC*	60	3.81	44	3.87	52	3.24	**156**	**3.64**
	PED	13	3.63	11	3.83	9	4.56	**33**	**3.95**
	STA	0	N/A	3	0.63	0	N/A	**3**	**0.63**
	SUM	**73**	**3.78**	**58**	**3.70**	**61**	**3.43**	**192**	**3.64**
	Total	**370**	**4.57**	**361**	**3.76**	**319**	**4.65**	**1050**	**4.32**

* *ST: Strategy; COLT: Collision Type. NPC: NPC Vehicle; PED: Pedestrian; STA: Static Obstacle. RLS: RL-based Strategy; RS: Random Strategy; GS: Greedy Strategy. #C: Number of collision scenarios; SAC (m/s): the speed at collision.*

Selecting and prioritizing scenarios for regression testing. In practice, generating critical testing scenarios is very expensive in terms of time costs and computational resources, and as ADS evolves, regression testing for multiple versions will become even more expensive [17]. By using search techniques, we can further select critical scenarios from DeepScenario and prioritize them to support regression testing of ADSs.

B. Limitations

DeepScenario dataset was created by test executions in a virtual environment. Therefore, although we have mechanisms such as realistic constraints and the introduction of real-world weather data, to ensure the realism of scenarios, our dataset may be limited by the capabilities of the simulator applied. However, the fidelity of an autonomous driving simulation can be enhanced by using real-time co-simulation techniques [18] and further improve the realism of our dataset.

VI. Conclusion

This paper presents DeepScenario, an open driving scenario dataset that can facilitate the development and validation of autonomous driving. DeepScenario dataset contains a total of 33530 scenarios, among which 1050 are collision scenarios. Further, each scenario is characterized by six attributes, showing the extent of safety/comfort violation and collision information. We also provide a DeepScenario toolset for automatically collecting and replaying driving scenarios. Finally, we show several usages of our dataset in various ADS development contexts, and we believe that autonomous driving research will benefit from DeepScenario for testing and developing ADSs.

References

[1] X. Zhang, J. Tao, K. Tan, M. Torngren, J. M. G. Sanchez, M. R. Ramli, X. Tao, M. Gyllenhammar, F. Wotawa, N. Mohan, *et al.*, "Finding critical scenarios for automated driving systems: A systematic mapping study," *IEEE Transactions on Software Engineering*, 2022.

[2] A. Stocco, B. Pulfer, and P. Tonella, "Mind the gap! a study on the transferability of virtual vs physical-world testing of autonomous driving systems," *IEEE Transactions on Software Engineering*, pp. 1–13, 2022.

[3] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361, IEEE, 2012.

[4] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3061–3070, 2015.

[5] G. Yang, X. Song, C. Huang, Z. Deng, J. Shi, and B. Zhou, "Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 899–908, 2019.

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[7] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4040–4048, 2016.

[8] C. Vogel, K. Schindler, and S. Roth, "Piecewise rigid scene flow," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1377–1384, 2013.

[9] C. Lu, Y. Shi, H. Zhang, M. Zhang, T. Wang, T. Yue, and S. Ali, "Learning configurations of operating environment of autonomous vehicles to maximize their collisions," *IEEE Transactions on Software Engineering*, vol. 49, no. 1, pp. 384–402, 2023.

[10] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[11] R. T. Fielding, *Architectural styles and the design of network-based software architectures*. University of California, Irvine, 2000.

[12] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, *et al.*, "Lgsvl simulator: A high fidelity simulator for autonomous driving," in *2020 IEEE 23rd International conference on intelligent transportation systems (ITSC)*, pp. 1–6, IEEE, 2020.

[13] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, "Baidu apollo em motion planner," *arXiv preprint arXiv:1807.08048*, 2018.

[14] K. Czarnecki, "Operational world model ontology for automated driving systems–part 1: Road structure," *Waterloo Intelligent Systems Engineering Lab (WISE) Report, University of Waterloo*, 2018.

[15] OpenWeather, "Weather forecasts, nowcasts and history in a fast and elegant way." https://openweathermap.org/.

[16] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, "Defining and substantiating the terms scene, situation, and scenario for automated driving," in *2015 IEEE 18th international conference on intelligent transportation systems*, pp. 982–988, IEEE, 2015.

[17] C. Lu, H. Zhang, T. Yue, and S. Ali, "Search-based selection and prioritization of test scenarios for autonomous driving systems," in *International Symposium on Search Based Software Engineering*, pp. 41–55, Springer, 2021.

[18] Q. Chen, T. Wang, C. Lu, T. Yue, and S. Ali, "Enhancing the realism of autonomous driving simulation with real-time co-simulation," in *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, pp. 659–667, 2022.