



Review

A Survey of Scenario Generation for Automated Vehicle Testing and Validation

Ziyu Wang , Jing Ma and Edmund M-K Lai *

Department of Data Science and Artificial Intelligence, School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand; ziyu.wang@aut.ac.nz (Z.W.); jing.ma@aut.ac.nz (J.M.)

* Correspondence: edmund.lai@aut.ac.nz

Abstract: This survey explores the evolution of test scenario generation for autonomous vehicles (AVs), distinguishing between non-adaptive and adaptive scenario approaches. Non-adaptive scenarios, where dynamic objects follow predetermined scripts, provide repeatable and reliable tests but fail to capture the complexity and unpredictability of real-world traffic interactions. In contrast, adaptive scenarios, which adapt in real time to environmental changes, offer a more realistic simulation of traffic conditions, enabling the assessment of an AV system's adaptability, safety, and robustness. The shift from non-adaptive to adaptive scenarios is increasingly emphasized in AV research, to better evaluate system performance in complex environments. However, generating adaptive scenario is more complex and faces challenges. These include the limited diversity in behaviors, low model interpretability, and high resource requirements. Future research should focus on enhancing the efficiency of adaptive scenario generation and developing comprehensive evaluation metrics to improve the realism and effectiveness of AV testing.

Keywords: autonomous vehicles; driving scenarios; adaptive tests; automatic scenario generation



Citation: Wang, Z.; Ma, J.; Lai, E.M.-K. A Survey of Scenario Generation for Automated Vehicle Testing and Validation. *Future Internet* **2024**, *16*, 480. <https://doi.org/10.3390/fi16120480>

Academic Editor: Paolo Bellavista

Received: 19 November 2024

Revised: 19 December 2024

Accepted: 19 December 2024

Published: 23 December 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Reducing traffic accidents has long been a key focus of research and development. Studies have shown that a significant proportion of vehicle accidents are caused by human errors [1,2]. To mitigate these errors, various advanced driver assistance systems (ADAS) have been developed and integrated into vehicles over the past several decades. As automation technology continues to evolve, fully autonomous vehicles (AVs) are becoming a reality. As a result, major car manufacturers and new companies have been actively developing AV technology [3].

The primary goal of AV testing is to ensure safety. Testing AV systems in various traffic conditions and scenarios is crucial to verify that AVs can operate safely in complex environments [4]. Apart from identifying potential risks and ensuring the reliability of AVs, rigorous testing establishes public trust, removing concerns that limit their widespread adoption. In addition to ensuring safety, testing must also evaluate a vehicle's ability to perform specific driving tasks, thereby assessing its overall capabilities.

There are three primary methods of conducting AV testing—on-road tests, closed-field tests, and simulation tests [5]. On-road testing evaluates AVs on real-world roads, providing realistic and unpredictable scenarios [6]. However, safety-critical situations are rare in real-world driving, making on-road testing inefficient for comprehensive safety validation. One study [5] showed that proving that AVs are as safe as human-driven cars would require driving hundreds of millions of miles, or even billions of miles, of on-road testing. Closed-field testing takes place in controlled environments specifically designed to simulate real-world traffic and road conditions. This method offers a controlled experimental setting where hazardous scenarios can be tested safely, improving efficiency [6]. However, this cannot fully replicate the complexities of real-world conditions, such as traffic dynamics,

weather variations, and road environments, which may limit its ability to fully evaluate AV performance. Simulation testing, also known as scenario-based testing, uses computer-generated environments to test AV systems. Simulation testing allows autonomous driving systems to be tested in a safe and controlled simulation of a wide range of traffic conditions, road scenarios, and vehicle behaviors. Compared with closed-field testing and on-road testing, simulation testing does not require physical vehicles, making it more resource-efficient. It enables early detection of issues during algorithm development and allows for the testing of highly dangerous situations [6]. Among the methods, simulation testing stands out as the most convenient and advantageous. It can generate a variety of driving scenarios, enables real-time parameter adjustments to suit different AV systems, and supports rapid iteration and optimization of autonomous driving systems. Moreover, testing in a virtual environment also allows developers to reproduce specific scenarios to analyze system performance and make improvements. In addition, simulation-based scenario testing has become the most widely used and promising approach for AV testing. This survey focuses on the simulation testing method, specially the generation of test scenarios for AVs.

In the generation of simulation test scenarios, many current approaches use predefined or scripted behaviors for movable objects such as the background vehicles (BVs), pedestrians, and other road users. BVs refer to all vehicles in the scenario other than the AV under test. These BVs follow preset behaviors and do not make real-time decisions based on the current environment, leading to what are referred to as non-adaptive scenarios. In contrast, adaptive scenarios are those that allow movable objects to make real-time decisions based on changing conditions. To ensure comprehensive, effective, and realistic testing of AVs, non-adaptive scenarios are no longer sufficient, and the generation of adaptive test scenarios has become the primary focus.

This survey aims to introduce the changing trends in scenario generation methods and identifies the types of scenarios that require attention. The structure of the survey is as follows: Section 2 introduces key terminology in the field of scenario generation. Section 3 presents the two major types of scenarios: non-adaptive and adaptive, and their respective generation methods. Section 4 analyzes and discusses non-adaptive and adaptive generation methods and points out future research directions. Finally, Section 5 gives the conclusions of this survey.

2. Terminology

This section defines terms directly relevant to the field of scenario generation.

2.1. Operational Design Domain (ODD)

According to the Society of Automotive Engineers (SAE) J3016 [7], the operational design domain (ODD) of a driving automation system is defined as the “Operating conditions under which a given driving automation system, or feature thereof, is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics” [8]. In other words, the ODD specifies the particular conditions and environmental scope within which the autonomous driving system can operate, thereby establishing constraints on the driving environment for a given test scenario.

To ensure autonomous vehicles function effectively in their intended environments, it is crucial that their validation includes training data and testing that cover all relevant operating conditions. A common approach to achieving this involves restricting the vehicle’s operational environment to a subset of scenarios that a human driver might encounter. This approach, known as adopting an operational design domain (ODD) [9], helps define the system’s operational scope. An ODD can also limit the road environment, AV behavior, and vehicle states [10].

An ODD can also impose restrictions on the road environment, vehicle behavior, and vehicle states [10]. In [10], the road environment for autonomous driving was categorized into urban, rural, or highway settings; intersections, roundabouts, tunnels, construc-

tion zones, parking lots, narrow alleys, and narrow roads; as well as specific conditions like rainy, snowy, foggy weather, and nighttime driving, based on road types, structural elements, temporary modifications, traffic flow, weather, and visibility. Beyond road environment limitations, an ODD may also specify constraints on the behavior and state of autonomous vehicles, such as speed limits, restrictions on certain maneuvers like reversing, and limitations related to vehicle loading.

According to [11], six top-level categories are used to define and classify ODDs, which are physical infrastructure, operational constraints, objects, connectivity, environmental conditions, and zones. There are various types of ODDs, encompassing scenarios such as car-following; lane changing; turning around; overtaking; navigating intersections; driving on roundabouts, highways, and urban roads; operating in construction zones and tunnels; maneuvering in confined spaces; driving in adverse weather conditions; and responding to emergencies, among others. Consider lane changing as an example. On a straight two-lane urban road marked by a dotted line, with clear weather and no pedestrians, vehicles are traveling in both lanes. The autonomous vehicle (AV) is in one lane, while another vehicle (BV) occupies the other lane. Both vehicles are moving at a speed of 60 km/h. In this scenario, the AV intends to change lanes from its current lane into the lane occupied by the BV. Figure 1 provides an illustration of this ODD. Another example is a car-following ODD, illustrated in Figure 2. In this scenario, on a straight, two-lane highway with dotted line markings and cloudy weather, there are no pedestrians, and vehicles in both lanes are traveling at 100 km/h. In one of the lanes, an AV follows a BV at a constant speed.

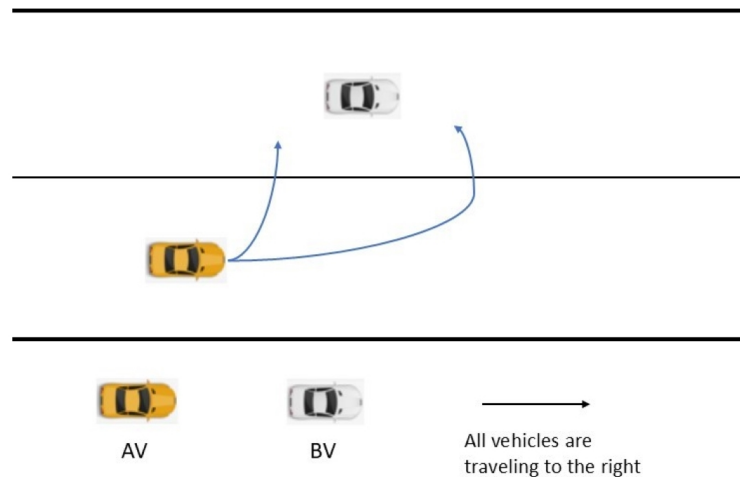


Figure 1. Lane-changing ODD representation.

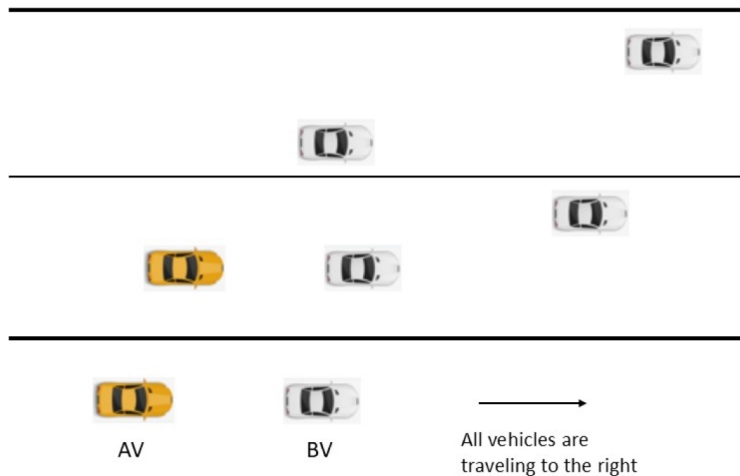


Figure 2. Car-following ODD Representation.

2.2. Description of Scenario

A scene represents a snapshot of a world state, which includes the road configuration, static objects, dynamic objects, and environmental conditions, but excludes unobservable states [10]. According to [12], a scenario provides a dynamic representation of an autonomous vehicle and its surrounding dynamic and static environment over time, capturing the interaction between the vehicle and its surrounding environment. The specific conditions within a scenario can be characterized using probability distribution and probability density functions, such as the number of vehicles, their positions, and speeds.

A scenario consists of both static and dynamic elements. Static elements include road configurations, such as lane types, the number of lanes, lane markings, the initial positions and speed of vehicles, weather conditions, etc. Dynamic elements include the presence of movable objects, such as pedestrians or animals, and other vehicles. These elements may be non-adaptive, which means that their movements are predetermined, or adaptive if the speed and trajectory of their movements are dependent on the behavior of other dynamic elements. In subsequent discussions, in order to keep the scenario illustrations simple, the only dynamic elements considered are the BVs. It should be noted that the scenario generation methods reviewed do not place restrictions on the kinds of dynamic objects to be included.

2.3. Scenario Types

As described in [13], the classification of scenarios into functional, logical, and concrete levels provides a structured framework for scenario generation, progressively increasing the detail and specificity of the scenario parameters. The functional view serves as a high-level description, typically written in natural language by human experts, capturing the conceptual essence of traffic situations such as “a vehicle overtakes another on a curved highway under rainy conditions.” The logical view makes these descriptions more specific by providing parameter ranges and constraints for critical elements, enabling the generation of diverse and representative instances within defined bounds. For example, the overtaking scenario could include variables such as the range of vehicle speeds allowed and the road curvature. Finally, the concrete view provides fully parameterized instantiations of logical scenarios, detailing precise values for all elements, to ensure reproducibility and direct execution in simulation environments. For instance, the initial speeds and positions of the vehicles would be specified. This hierarchical approach supports scenario generation across development phases, allowing for both broad exploration of possible situations and rigorous, repeatable testing of autonomous vehicle systems under controlled and varied conditions. Figure 3 illustrates this categorization.

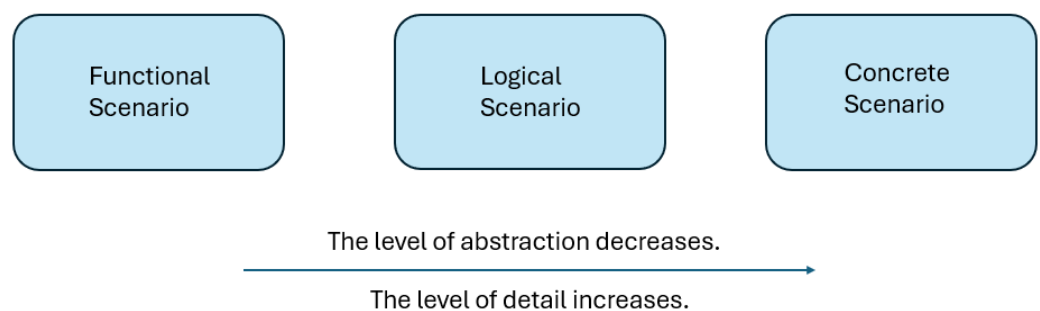


Figure 3. Test scenario category.

3. Scenario Generation Methods

There are two types of test scenarios for AVs—non-adaptive and adaptive. In a non-adaptive test scenario, the movements of all movable objects and their behaviors are predefined. In contrast, the movements of movable object such as BVs are not predetermined in adaptive test scenarios. They are modeled to react to changes in the environment. Therefore, if the behavior of the AV being tested is changed, then the movements of the

BVs, for instance, will change accordingly. Hence, adaptive test scenarios are more realistic and closer to the real world.

Figure 4 shows a taxonomy of scenario generation methods. In this review, we divide the generated scenario into non-adaptive and adaptive scenarios. There are three types of methods to generate non-adaptive scenarios, namely knowledge-based generation, data-driven generation, and scenario library generation. There are also three types of methods for generating adaptive scenarios, namely reinforcement learning-based methods, importance sampling-based methods, and imitation learning-based methods. More details of these methods are reviewed.

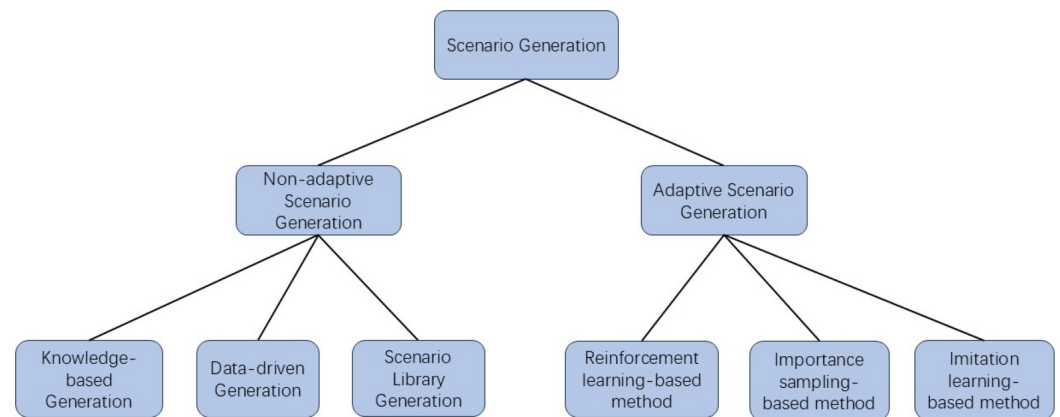


Figure 4. Taxonomy of scenario generation methods.

3.1. Non-Adaptive Test Scenario Generation Methods

Non-adaptive test scenarios involve testing autonomous vehicles in environments where the movements and behaviors of movable objects are simulated using predefined rules or scripts. They follow fixed actions that are unaffected by external influences during the testing process [14].

Non-adaptive scenarios have several notable disadvantages. As highlighted in [14], traffic participants in non-adaptive test scenarios lack interaction awareness, failing to capture the complexity of the human-like social behavior observed in real-world driving environments. Furthermore, since the behaviors of traffic participants are predefined based on the autonomous vehicle (AV) being tested, hard-coded driving behaviors result in simulation environments with limited interaction categories, reducing the efficiency of testing. Additionally, predefined scenarios often focus on specific test objectives, such as assessing the comfort of the AV under test, and are rarely designed for multi-objective testing. Given that AV intelligence involves multiple performance metrics, non-adaptive scenarios struggle to address such diverse evaluation requirements. Moreover, non-adaptive scenarios are typically reconstructed from driving data collected by sensor-equipped vehicles over time. As a result, if certain risky scenario patterns are absent from the dataset, they are overlooked in the evaluation, making the results highly dependent on the dataset's completeness [15]. The following subsections present a review of the literature on non-adaptive scenarios.

3.1.1. Knowledge-Based Generation Method

The first widely recognized method for generating test scenarios was knowledge-based generation, which involves utilizing existing knowledge and data to create scenarios. The knowledge and data can originate from various sources, such as traffic regulations, road conditions, vehicle behaviors, environmental factors, accident records, and other related knowledge domains.

The study by [16] used big data technology [17] to generate functional scenarios, using human natural language descriptions to define the generated scenarios. They performed text weight analysis on crash descriptions from accident records to extract high-frequency keywords related to crash locations, traffic operations, maneuvers, and triggering events,

which were then combined to construct new test scenarios. However, since [16] only generated functional scenarios, non-player characters (NPCs) in the scenario such as background vehicles (BVs), did not respond to the AV's behavior. Additionally, the scope of the generated scenarios was narrow, concentrating on crashes at main roads regulated by traffic lights, specifically at intersections and their approach links. Furthermore, all generated scenarios were crash-related, significantly limiting their utility for comprehensive and effective testing of autonomous vehicle performance.

In [18], a Markov decision process (MDP) [19] framework combined with deep reinforcement learning [20] was employed to generate test scenarios. Static scenario elements that included parameters such as the number of lanes, road curvature, and traffic density, collectively defined the physical layout of the scenario. The movement trajectories of BVs were generated through the use of an intelligent driver model (IDM) and a MOBIL lane-changing model. The IDM ensured safe following distances by modeling acceleration based on headway and relative velocity, while the MOBIL evaluated the safety and incentive criteria for lane changes. However, the trajectories did not change in response to the behavior of the AV. The study used the highway-environment simulation package to create the simulation environment, as this package enables the simulation of various driving tasks, such as lane changing, parking, and navigating intersections. To introduce interference, the researchers increased the traffic density and added randomization factors, training an autonomous agent in this environment. In addition, a baseline model was trained in a conventional environment to compare its performance with the model trained under hazardous conditions.

Both articles used domain knowledge to generate scenarios, but they differed in their approaches. The study by [18] primarily focused on dangerous driving behaviors generated through simulations and experiments, highlighting the system's adaptive capabilities. In contrast, ref. [16] relied on big data technology and existing accident data to generate scenarios, drawing on the extraction of historical knowledge.

3.1.2. Data-Driven Generation Methods

Data-driven methods refer to using real-world driving data, such as traffic accident records and vehicle sensor data, to generate test scenarios. Typically, big data analytics are employed to extract valuable scenario insights from large datasets.

The article in [21] introduced a framework called "Accelerated Deployment" designed for the safe and efficient testing of AVs that are not yet in production on public streets. This approach involved strategically selecting testing environments based on metrics such as safety risks, learning potential, and deployment costs. Accelerated evaluation, as discussed in [22,23], involves increasing the frequency of observed failures through an importance sampling mechanism [24], which adjusts the sampling distribution to "skew" the evaluation scenarios. Furthermore, the criticality of a scenario was assessed by analyzing the AV's capability to avoid risk events in various deployment settings. The probability of the AV encountering or engaging in a risk event at any given moment was quantified using the cumulative intensity function.

The work in [25] proposed a neural autoregressive model to generate realistic test scenarios. Based on the current state of the AV and a high-definition map of its surrounding environment, the ConvLSTM [26] architecture, an extension of the LSTM [27], was used to sequentially generate the trajectories and driving velocities for vehicles, pedestrians, and cyclists. The modeling process begins by classifying participants into categories using neural networks [28] to identify their respective types. Convolutional neural networks (CNNs) [29] are then used to model locations based on these categories. For vehicles and cyclists, directed bounding box modeling is applied, followed by velocity modeling to complete the representation of other traffic participants, excluding the AV itself.

The study in [30] proposed an editing framework based on reinforcement learning [31], which considers scenario generation as an optimization problem, aimed at identifying the most risky scenario within realistic constraints. This approach utilizes deep reinforcement

learning (DRL) to edit scenarios dynamically. The state space represents the scenario as unordered sets, incorporating a traffic agent feature matrix along with static environmental features. The reinforcement learning agent can perform various operations to modify scenarios, such as perturbing the trajectories of existing traffic agents, reconfiguring agent paths, introducing new agents, and marking undrivable regions. The reward function combines a risk model and a plausibility model. The risk model evaluates risk by calculating the number of feasible driving plans for autonomous vehicles in the scenario. The fewer plans, the higher the risk of the scenario. The plausibility model ensures the generated scenarios align with real-world constraints using conditional variational autoencoders (CVAE) [32], penalizing unrealistic data. In addition, the policy network uses a graph neural network (GNN) [33] to calculate the embedding of the scenario, and then outputs the next editing action. The reinforcement learning algorithm was optimized using proximal policy optimization (PPO) [34] to generate diverse security-critical scenarios.

3.1.3. Scenario Library Generation Method

Another approach to AV testing involves creating a library of test scenarios. A scenario library comprises a diverse collection of traffic scenarios designed to test and evaluate the performance of AV systems. This library includes a wide range of driving conditions, road configurations, traffic behaviors, and weather scenarios, effectively simulating the various situations vehicles may encounter in real-world environments.

In [35,36], a general framework for generating a library of test scenarios under different operational design domains (ODDs), CAV models, and performance metrics was proposed. The test scenario library was defined as a collection of critical scenarios, with criticality assessed using a newly introduced metric called criticality. This metric is calculated as the product of exposure frequency and maneuver challenge. Exposure frequency is estimated using naturalistic driving data (NDD), while maneuver challenge is determined through a surrogate model (SM) of the connected autonomous vehicle (CAV). The core concept of critical scenario identification involves using optimization methods to locate local critical scenarios and subsequently explore related scenarios in the surrounding space. Additionally, the study introduced an auxiliary objective function to guide the search direction and employed a multi-start optimization approach along with a seed-filling technique to comprehensively search for critical scenarios.

The study in [37] built upon the works of [35,36]. The libraries generated in the earlier studies were found to have certain limitations. They may be unable to accommodate certain scenario types, fail to be applicable to specific connected autonomous vehicle (CAV) models, or fall short in evaluating certain performance metrics. High-dimensional or complex dynamic scenarios, such as those involving intricate interactions between multiple vehicles and variable environmental conditions, are particularly challenging due to computational constraints and the curse of dimensionality. Additionally, the framework relies on surrogate models calibrated from naturalistic driving data to represent CAV behavior. While it is generally effective, this approach may fail to capture the unique features of proprietary or specialized CAV systems, limiting its applicability to a broader range of models. Furthermore, while the framework effectively addresses safety and functionality as key performance indicators, it lacks higher-level evaluation metrics such as mobility and ride comfort. Discrepancies between the surrogate model (SM) and the actual CAV could result in suboptimal scenario libraries, reducing its overall effectiveness.

In [37], the generated scenario library was optimized to create a customized test scenario library tailored to a specific CAV model through an adaptive process. To address performance discrepancies between the surrogate model (SM) and the CAV, Bayesian optimization techniques and classification-based Gaussian process regression were employed. In addition, the most informative scenarios were selected for testing in each iteration, allowing the SM to be dynamically updated and the custom library to be progressively refined. Ultimately, these customized libraries enable more efficient testing and evaluation of CAVs.

In [6], a functional test scenario library generation framework for CAV testing was proposed. This framework consists of static road libraries and dynamic scenarios. The static road library is formed by extracting roads from OpenStreetMap (OSM) [38] and then hierarchically clustering them. Dynamic scenarios are modeled by formulating the test scenario generation problem as a partially observable Markov decision process (POMDP) [39], while incorporating input values from sensors and modeling them using reinforcement learning. When modeling dynamic scenarios, the first requirement is to determine the purpose of the test, because different test purposes require different test scenarios. Then, a reward function is defined according to the test purpose, and scenarios of interest can be determined through the reward function. Finally, a comprehensive test scenario library is generated by combining the static road library and dynamic scenarios.

The scenarios described above are non-adaptive, meaning dynamic elements like background vehicles (BVs) or pedestrians are not individually modeled and lack the ability to make independent behavioral decisions in response to AV actions. Such non-adaptive scenarios have limited value for testing autonomous vehicles. The next section will focus on the modeling of adaptive scenarios.

3.2. Adaptive Test Scenario Generation Methods

To overcome the limitations of non-adaptive test scenarios, the development of adaptive test scenarios has become a key research focus in recent years. An adaptive test scenario for AVs involves the autonomous learning and decision-making of surrounding elements, such as other vehicles, pedestrians, and animals, based on the real-time situation. In these scenarios, the entities within the test environment possess some degree of intelligence and learning capabilities, with their movements and actions generated autonomously through models, algorithms, or learning processes. The behavior of these elements can be influenced by external factors, such as the actions of other vehicles, traffic conditions, and obstacles.

Consider a two-lane road scenario where AV and BV are traveling on separate lanes. A cut-in scenario occurs when the AV attempts to change into the lane occupied by the BV. This is illustrated in Figure 5.

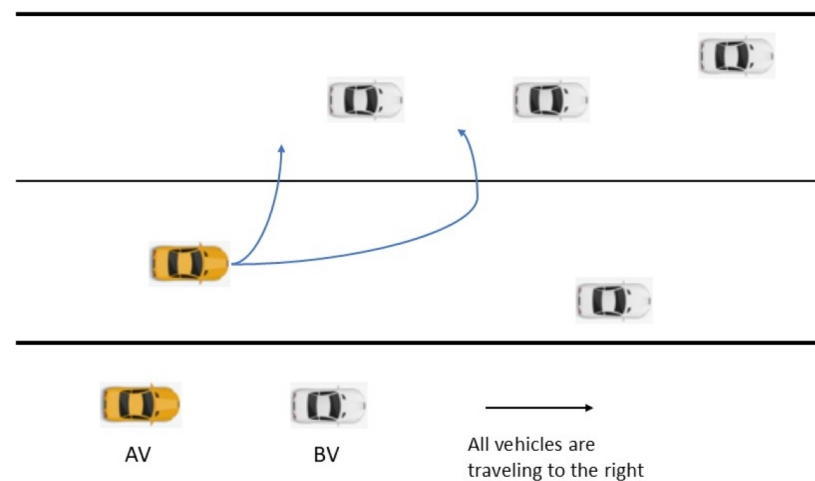


Figure 5. Cut-in scenario representation.

In a non-adaptive scenario, the BV is typically assumed to be traveling at a constant speed. Consequently, the BV's motion is easily predictable and so this test does not really test the AV's decision-making ability [40]. The assumption that the BV travels at a constant speed throughout the process is a very strong one and is very unrealistic.

In order to overcome the limitations of non-adaptive scenarios described above, adaptive scenarios need to be used. In such scenarios, the environment the AV experiences becomes more dynamic. Using a cut-in scenario as an example, the strong assumption that the speed of the BV remains constant will be removed in an adaptive scenario. In other

words, the BV has the freedom to drive at any speed within the legal speed limit of the road, and it can accelerate or decelerate at any time throughout the cut-in or lane-changing process. The only restriction is that BV remains within the speed limit, which allows for a more controlled and predictable testing environment for the AV.

Since adaptive scenarios involve more variables than non-adaptive ones, they are more difficult to design. Appropriate designing of adaptive test scenarios should be governed by the purpose of the tests. For our example, the purpose is to test whether the AV can adaptively perform lane-changing operations based on changes in the BV's behavior. Passing this test could entail one of three possible outcomes—the AV cut-in in front of the BV, the AV changed lane behind the BV, and the lane-changing was abandoned if it was not safe enough to do so. An additional criterion for determining success could be that the position of the AV relative to the BV after the lane change is above a certain safety threshold. This would be a more comprehensive and realistic test for this ODD.

In subsequent subsections, we will focus on the literature on adaptive scenarios. Adaptive test scenario generation methods can be categorized into three categories. These are reinforcement-learning-based methods, importance-sampling-based methods, and imitation-learning-based methods.

3.2.1. Reinforcement-Learning-Based Methods

The reinforcement-learning-based approach is to use a BV to automatically generate challenging driving scenarios. The agent interacts with a virtual environment, assesses the current scenario, takes appropriate actions, and receives rewards based on the AV's performance. Through continuous optimization of its strategy, the agent generates scenarios that push the AV system's performance to its limits. This method allows for the exploration of complex, diverse, and extreme test conditions, enhancing the test coverage and improving system safety.

In [14], a set of human-like social-driven models were built to generate evolving test scenarios. The study combined level-k game theory [41] and DRL frameworks based on previous work to train game-like driving policies. Level-k game theory refers to the hypothesis that participants in a game predict the behavior of others based on different levels of reasoning and formulate corresponding strategies, where each level of reasoning is based on the expectation of lower-level behavior [42]. In the study, level-k game theory was used to build the interaction logic between intelligent driver models and DRL was used to evolve the interaction between them, while the driving model was given game-like driving policies of competitive, mutual, and cooperative by shaping the reward function. Among these, a competitive driving policy means that the driving behavior of the car executing this driving policy is very aggressive. In a lane-changing scenario, it will constantly try to change lanes. In a car-following scenario, it will always try to follow the car at a shorter distance. This driving policy simulates drivers who behave aggressively in the real world. A mutual driving policy simulates the driving behavior of ordinary human drivers in the real world. Under this driving policy, if the vehicle finds that there are other vehicles in front trying to change lanes into its own lane, the vehicles with the mutual driving policy will slow down appropriately to allow the vehicle in front to change lanes more safely. If it is found that the vehicle in front is still driving slowly after changing lanes, the vehicles with the mutual driving policy will try to change to other lanes to restore a normal driving speed. The vehicle behavior of a cooperative driving policy is relatively more conservative. Vehicles executing this driving policy will generally not actively change lanes in lane changing scenarios, unless there are special circumstances. When other vehicles change lanes to their lane, they will obviously slow down to avoid the lane-changing vehicles. The TD3 algorithm [43] was then used to realize continuous control of the car's following behavior and human-like social driving policies for ego vehicle decision-making. Finally, these driving models were used in BVs to generate evolving scenarios.

The generated test scenarios were highly efficient for AV testing and the generated driving environment was realistic and challenging. However, the paper only used three

driver models, which does not fully represent the diversity of human driving behaviors, may limit the range of scenarios that can be generated, and may not fully capture the complexity of real-world driving scenarios. Additionally, the generated driving policies lacked interpretability. The paper used deep reinforcement learning to train the driver models, which can make policies difficult to interpret and understand. This can make it challenging to identify the specific factors that influenced the background vehicle behavior in a generated scenario.

A sampling method of interpretable, operable corner cases was proposed in [44]. This was based on the idea of sampling a small number of challenging scenarios from all possible scenarios to test autonomous vehicles. The article pointed out that AVs can be flexibly tested through a large number of interactions between the AV and surrounding vehicles (SVs), and the different behaviors and strategies of the SVs play a crucial role in this process. The different driving behaviors of the SVs constitute hypothetical factors in the test scenarios, so a better description of the driving behaviors of SVs is needed. The article used a utility function to model and generate aggressive, conservative, and normal driving policies by adjusting the hyperparameters of the utility function, because a utility function-based method can combine multiple driving decisions to model driving behaviors, and therefore can be more easily extended to describe rich and complex behaviors. However, this method may have limitations in capturing the full range of interactions between an AV and surrounding vehicles. This may affect the comprehensiveness of the test scenario, as some subtle driving behaviors and interactions may not be fully represented.

In [45], an interactive critical scenario generation method based on in-depth crash data was proposed. First, this method extracted the crash data that occurred at an intersection from the China In-depth Mobility Safety Study-Traffic Accident (CIMSS-TA) database to generate an initialized scenario state, including the relative position, speed, direction, driving intention, and acceleration of the ego vehicle (the AV under test) and the objective vehicle. A conditional tabular generative adversarial network (CTGAN) was then used to extrapolate the scenario, to increase its diversity. Then, the TD3 algorithm was used as the driving algorithm for the BV. This makes use of Baidu's "Apollo" platform, which is an open-sourced autonomous driving ecosystem that provides a suite of tools and algorithms for the development and deployment of AVs. It supports key functionalities such as perception, planning, control, and localization. In the study, Apollo was deployed within the SVL simulator to perform three distinct driving tasks: turning left, going straight, and turning right at intersections, reconstructed from crash data. These tasks allowed the system to engage in dynamic interactions with the BV controlled by the TD3 algorithm. Through these interactions, critical and realistic scenarios were created, challenging Apollo's decision-making and control algorithms. The authors also developed a simulator platform called RL-Scenario and used this platform to train an RL agent model. Crash rate, generalized-time-to-collision (GTTC), and post-encroachment time (PET) were the evaluation metrics used for analysis.

The proposed method is based on real traffic accident data, especially intersection accident data, so the generated scenarios reflect high-risk situations that may occur on actual roads. In addition, this method uses a CTGAN, which can generate diverse test scenarios and expand the coverage of scenarios. The TD3 reinforcement learning algorithm is used to control the objective vehicle, so that it can dynamically adjust its behavior according to the actions of the ego vehicle. This interactive scenario generation can effectively simulate complex driving situations. However, since this method relies on historical accident data, and accidents themselves are low-probability events, although it can generate high-risk scenarios, it may ignore some potential risk scenarios that have not yet occurred, especially those based on natural driving data (NDD) scenarios. And although the scenario library was expanded through a CTGAN, some generated scenarios may be too extreme and may not fully comply with actual road conditions. In addition, this method was mainly verified on the Baidu Apollo, and different autonomous driving systems use different control algorithms, which means that this method may lack adaptability to other autonomous driving

platforms and have poor versatility. Finally, this method only considers the interaction between a single objective vehicle and an ego vehicle, and does not involve complex interaction scenarios between multiple vehicles. In reality, traffic conditions are often more complex, so it is necessary to consider generating scenarios involving multiple data sources and interactions between multiple vehicles.

There have been some articles that clustered the generated scenarios after using reinforcement learning to train the driving strategy of BVs, because clustering can help better manage the generated corner cases and be better applied in the subsequent testing of autonomous vehicles.

The study in [46] proposed a unified framework for generating corner cases for decision-making systems in autonomous vehicles. Since the real-world traffic environment is high-dimensional, the article proposed using a Markov decision process (MDP) to describe the traffic environment, and used a deep Q-network (DQN) [47] in DRL to learn the optimal behavior policy of a BV, thereby making the BV behave more aggressively when interacting with the CAV, and therefore systematically generating corner cases in complex driving environments. In order to better manage the generated corner cases and apply them in subsequent tests, the article used principal component analysis (PCA) [48] to reduce the dimensions of the corner cases and extract the principal features, and then used K-means [49] and DBSCAN [50] algorithms to cluster the corner cases to identify the most valuable corner cases.

The proposed method could comprehensively evaluate the performance of an AV in various scenarios by generating and analyzing corner cases, which can help identify potential vulnerabilities of the AV system and improve the security of the system. Furthermore, the method can help improve the performance of AVs using feature extraction and clustering techniques to identify valuable corner cases, which can reveal specific challenges and critical situations. Despite that, the training of deep reinforcement learning models for generating corner cases can be very time-consuming and computationally resource-intensive, especially for complex scenarios. Second, the proposed method has limited applicability to different driving environments and scenarios. It is currently limited to highway driving environments and may not be directly applicable to other driving scenarios such as urban driving environments. Therefore, further adjustment and expansion are needed to achieve a wider range of applications. Finally, this method uses a DRL model to generate corner cases, and the DRL model may lack interpretability, making it difficult to understand the AV's decision-making process in various scenarios.

Another study that used clustering techniques to classify generated risk scenarios was [51]. The article used deep reinforcement learning to generate adversarial environments for the evaluation of an ego vehicle. In order to better simulate the mixed cooperation and competition interaction between AVs and environment vehicles, the article designed a non-zero-sum reward function based on domain knowledge and traffic rules, and expressed the scenarios as an MDP. During training, ensemble reinforcement learning was used to collect various risk scenarios. After training, a non-parametric Bayesian approach was used to cluster the generated risk scenarios to increase the collision rate between the AVs and environment vehicles.

The framework proposed with this method is adaptive, so it can generate a temporal adversarial environment based on the behavior of the vehicle being tested, thereby providing a dynamic and responsive evaluation method, so the generated scenarios are more realistic. In addition, by designing a non-zero-sum reward function, a mixed cooperative and competitive interaction relationship between the ego vehicle and the environment vehicle is better simulated, making the scenario more realistic. However, this method runs the risk of generating adversarial policies that overfit specific scenarios or behaviors, thus potentially limiting the evaluation of AV performance.

In addition to mainly using reinforcement learning methods to generate test scenarios, there have also been some methods that combined reinforcement learning methods with importance sampling (IS) techniques, which will be discussed in the following subsection.

3.2.2. Importance-Sampling-Based Method

Importance sampling techniques generate test scenarios by modifying the sampling probability distribution [52]. This is used to effectively generate representative test scenarios, especially those corner cases or extreme scenarios that are harder to encounter but are critical to AV system performance. The main purpose of using importance sampling is to increase the efficiency of finding rare or high-risk scenarios in testing, thereby reducing the number of tests required.

The work in [53] used naturalistic driving environments (NDE) to generate a naturalistic and adversarial driving environment (NADE). First, the Markov decision process (MDP) was used to establish the generation of a NDE model, and then the natural distribution of vehicle maneuvers was calculated from naturalistic driving data, and vehicle maneuver samples were extracted from it to generate the vehicle's driving behavior. Crude Monte Carlo (CMC) and importance sampling were then used to adjust for the natural distribution of vehicle maneuvers in NDE. IS was used to adjust the distribution of a small subset of variables that are important for rare events, while CMC was used for the remaining variables. The reason for this is that AV tests can be regarded as a rare event estimation problem with high-dimensional variables, and CMC handles high-dimensional problems well, while IS handles rare events well. Therefore, combining the two could solve the AV testing problem effectively. For the determination of variables for rare events, the approach used RL to achieve this. It identified principal other vehicles (POVs) by learning the challenges of background vehicle maneuvers for the tested AV, and then used the above-mentioned IS for maneuver distribution adjustment.

By making sparse but intelligent adjustments to the natural driving environment, this method accelerates the evaluation process by orders of magnitude compared to the prevalent natural driving environment. In addition, the generated environment ensures that the test results of AVs do not deviate from the natural driving environment. This means that the assessment results, such as the rates of the different types of accidents, are statistically accurate and reflect real-world driving scenarios. Nevertheless, the case study used in the article was simplified. The article focused on highway driving and limited actions, which may limit the generalization of the research results to more complex driving scenarios. Therefore, further research and testing in more complex driving environments are needed to fully evaluate the applicability of the proposed method. Additionally, the proposed approach lacked perceptually relevant tests in generated natural and adversarial driving environments, such as evaluating the performance of AVs in different weather conditions, which are crucial for a comprehensive evaluation.

3.2.3. Imitation-Learning-Based Method

Imitation learning (IL) [54] is a technique that generates test scenarios by learning the behavior of humans or other experts. In generation based on imitation learning, the generation method usually combines imitation learning with other techniques to generate test scenarios.

The work in [55] proposed a dynamic test scenario generation method based on conditional generative adversarial imitation learning (CGAIL) [56], which generates test scenario by modeling environmental vehicles as agents with human behavior and simulating the interaction process between autonomous vehicles and environmental vehicles. The study modeled the driving scenario as a Markov decision process (MDP), including a state set, an action set, a policy set, a transition probability function, and a reward set. The hierarchical Dirichlet process hidden semi-Markov model (HDP-HSMM) method was used to cluster the expert demonstration data. The CGAIL method was then used to generate dynamic test scenarios based on scenario category labels. The CGAIL model includes a generator and a discriminator. The generator outputs the actions of environmental vehicles based on the current state and scenario labels, and the discriminator is used to distinguish between expert demonstrations and generated samples. Then, the environmental vehicle is modeled as an agent, the interaction process between it and the autonomous vehicle is

simulated, and the performance of the autonomous vehicle in different scenarios is tested. The autonomous vehicles are evaluated by evaluating their different collision rates, failure rates, and success rates in different scenarios.

This method can generate adaptive test scenarios and model environmental vehicles as agents with human behavior, simulating the time sequence interaction process between autonomous vehicles and environmental vehicles, providing a more realistic evaluation environment than static testing. Moreover, by clustering real traffic environments and integrating scenario category labels, this method can generate test scenarios with diverse and multi-modal characteristics, improving the test coverage and effectiveness. However, the training process for generative adversarial networks (GAN) and CGAIL models requires a large amount of computing resources and time, and therefore may not be suitable for situations with limited resources. Additionally, the quality and diversity of the generated scenarios are highly dependent on the richness and representativeness of the training data. If the training data are insufficient, the generated scenarios may not cover all possible driving situations.

In general, the test scenario generation process of AVs can be summarized as the process shown in Figure 6. This is the framework for generating test scenarios, showing the entire process of generating a test scenario.

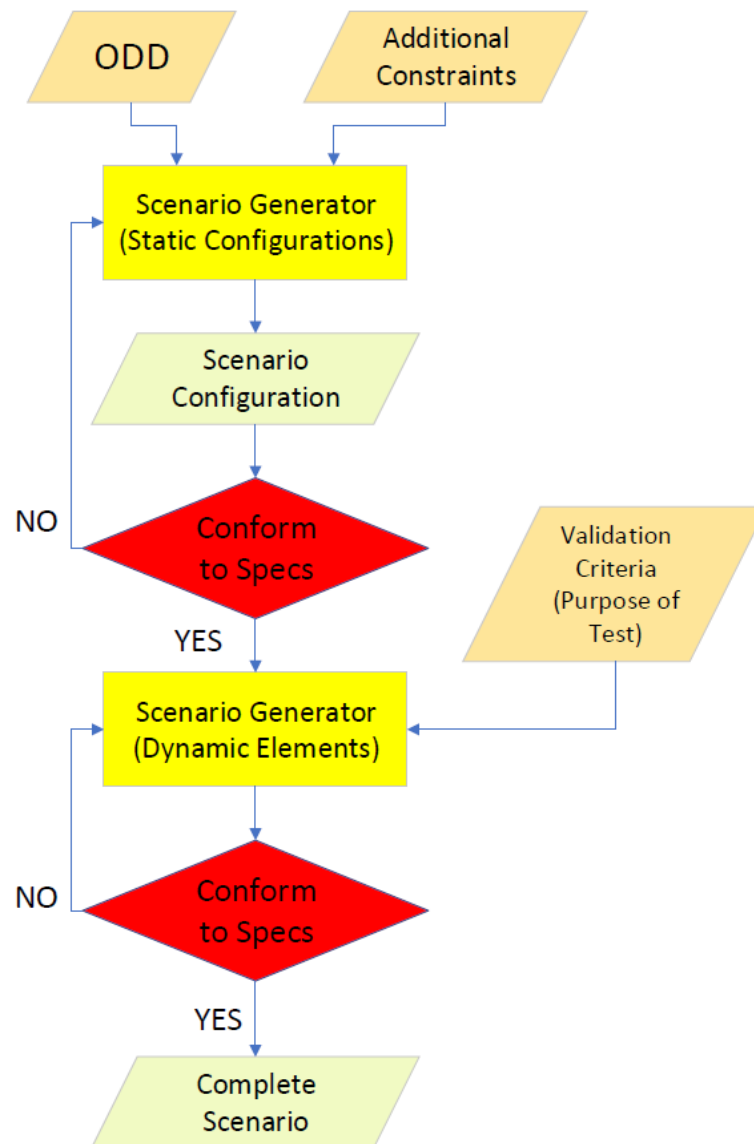


Figure 6. Adaptive scenario generation framework.

This framework clearly shows how to generate test scenarios for autonomous vehicles. First, we need to determine the type of ODD, whether it is car-following ODD, lane-changing ODD, U-turn ODD, or other types of ODD. Then, the additional constraints of the scenario are determined, such as whether the road is an urban road, a highway, a rural road, or other; how many lanes there are; how many vehicles there are, etc. This information is then used to generate static objects in the scenario, which is the “static configuration” in the framework, and the output result is the “scenario configuration” in the framework. Then, we judge this to see if the output result meets the scenario specifications we defined at the beginning. If not, then return to the generation stage to regenerate. If it meets the specifications, then enter the next stage of dynamic element generation. The generation of dynamic elements is based on the static elements we just generated. It is crucial that these dynamic elements serve the purpose of the test, which is reflected by the validation criteria. Only with clear and precise validation criteria can the behavior of the dynamic elements be designed. For example, the purpose of our test is to test whether the AV can avoid collision with an unpredictable BV and complete the lane-changing task in a highway lane-changing ODD. After determining the purpose of this test, we can start to design the BV’s motion behavior, such as making the BV’s behavior more aggressive and making it more difficult for the AV to drive and change lanes. Therefore, it is crucial to determine the purpose of the test. After adding the test purpose (validation criteria), dynamic elements can be designed and generated. After the generation has been completed, it is checked whether the generation result meets the test purpose we defined. If it does not meet the test purpose, it is regenerated. If it does, the entire generation is completed, and our final generation result is the output.

The scenario generator shown in Figure 6 was designed to systematically produce testing scenarios by parameterizing environmental, vehicular, and behavioral elements according to predefined operational design domain (ODD) and additional constraints. The main parameters required for its operation include environmental parameters, such as road type, lane configuration, and weather conditions; vehicular parameters, including the dynamics of the autonomous vehicle and background vehicles (e.g., velocity, acceleration, and relative position); and behavioral parameters, which define interaction patterns like lane changes, overtaking, or yielding behaviors. Validation of the scenario generator relies on ensuring consistency with real-world data and verifying that the generated scenarios align with the defined criticality measures, such as exposure frequency and maneuver challenges. By incorporating these parameters, the generator can produce a diverse set of realistic and challenging scenarios tailored to evaluating autonomous vehicle performance across safety, functionality, and other relevant metrics. This systematic approach ensures comprehensive coverage of potential driving conditions, while maintaining fidelity to real-world scenarios.

4. Discussion

In this survey, we noted that in AV test scenario generation, scenarios can be classified into non-adaptive and adaptive types. A key characteristic of a non-adaptive scenario is that the movement of dynamic objects (such as vehicles, pedestrians, animals, etc.) follows predetermined rules or scripts and does not adapt to the current environment or road conditions. While such scenarios allow for repeatable experiments and reliable data comparisons, they fall short in capturing the complexity and unpredictability of real-world driving. Non-adaptive scenarios do not fully represent how vehicles or pedestrians dynamically respond to actual traffic environments, which limits their effectiveness in assessing AVs’ reactions to emergencies and complex interactions.

In contrast, research on adaptive scenario generation aligns more closely with the needs of AV system development and testing. In adaptive scenarios, the behavior of objects adjusts and responds in real time to changes in the environment, providing a more realistic simulation of varying traffic conditions on actual roads. This approach enables the capture of complex interaction scenarios, including collaborative or adversarial behaviors

among dynamic objects. Such flexibility makes adaptive scenarios particularly valuable for evaluating the robustness, safety, and adaptability of AV systems.

Consequently, the current trend in test scenario generation research is a gradual shift from non-adaptive to adaptive scenarios. While non-adaptive scenarios serve as a foundational testing tool, their limitations have led researchers to increasingly focus on adaptive scenario generation, which allows for a more thorough evaluation of AV systems' performance in complex, changing traffic environments. In this survey, we discussed several adaptive scenario generation methods. However, these methods still present challenges, such as the limited diversity in the generated BV driving strategies, low interpretability of models due to reinforcement learning, limited adaptability of BV models, and significant resource and time requirements for model training. This suggests that future research should emphasize not only the efficient generation of adaptive scenarios but also the establishment of effective evaluation metrics to assess generation methods from multiple perspectives. Such advancements are essential for more realistically replicating real-world driving conditions during testing and ensuring that autonomous driving technology can handle the unpredictability and diversity of practical applications.

5. Conclusions

This survey provided a systematic review and analysis of adaptive scenario generation methods for the testing and validation of autonomous vehicles. The evolutionary trend from non-adaptive to adaptive scenarios indicates that traditional static scripted scenarios are increasingly inadequate for meeting the testing requirements of complex, dynamic, and highly interactive environments, as testing demands continue to grow. Looking forward, research should delve deeper into refining existing scenario classification by introducing subclasses, extending beyond the basic functional, logical, and concrete frameworks to more hierarchical and comprehensive categories. Additionally, the scenario generation process should adapt to the advancements in multi-vehicle cooperation and vehicular networking technologies to simulate higher-dimensional traffic ecosystems. Furthermore, there is a need to enhance the modeling and control strategy design of complex dynamic scenarios involving multiple autonomous vehicles, exploring collaborative decision-making, resource sharing, and safety redundancy mechanisms under abnormal conditions to achieve more realistic, flexible, and robust testing environments. Building on this foundation, researchers must continuously improve evaluation metrics and measurement standards, while also focusing on enhancing the interpretability and scalability of models. This will ensure effective support for real-world applications and industrial-grade validation testing requirements under diverse conditions.

Author Contributions: The original draft was prepared by Z.W.; review and editing contributions from supervisors: J.M. and E.M.-K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. National Highway Traffic Safety Administration. 2015 Motor Vehicle Crashes: Overview. *Traffic Saf. Facts Res. Note* **2016**, *2016*, 1–9.
2. European Commission. *2018 Road Safety Statistics: What Is Behind the Figures?*; European Commission: Brussels, Belgium, 2018.
3. Ding, W.; Xu, C.; Arief, M.; Lin, H.; Li, B.; Zhao, D. A survey on safety-critical driving scenario generation—A methodological perspective. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 6971–6988 [[CrossRef](#)]
4. Koopman, P.; Wagner, M. Autonomous vehicle safety: An interdisciplinary challenge. *IEEE Intell. Transp. Syst. Mag.* **2017**, *9*, 90–96. [[CrossRef](#)]
5. Kalra, N.; Paddock, S.M. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transp. Res. Part A Policy Pract.* **2016**, *94*, 182–193. [[CrossRef](#)]
6. Zhu, Y.; Wang, J.; Guo, X.; Meng, F.; Liu, T. Functional testing scenario library generation framework for connected and automated vehicles. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 9712–9724. [[CrossRef](#)]

7. SAE International. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *SAE Int.* **2018**, *4970*, 1–5.
8. On-Road Automated Driving (Orad) Committee. Taxonomy & Definitions for Operational Design Domain (ODD) for Driving Automation Systems. 2021. Available online: <https://www.sae.org/standards/content/j3259/> (accessed on 15 October 2024).
9. Koopman, P.; Fratrick, F. How many operational design domains, objects, and events? In Proceedings of the Safe AI 2019: AAAI Workshop on Artificial Intelligence Safety, Honolulu, HI, USA, 27–28 January 2019.
10. Czarnecki, K. Operational design domain for automated driving systems. In *Taxonomy of Basic Terms*; Waterloo Intelligent Systems Engineering (WISE) Lab, University of Waterloo: Waterloo, ON, Canada, 2018.
11. Thorn, E.; Kimmel, S.C.; Chaka, M.; Hamilton, B.A. *A Framework for Automated Driving System Testable Cases and Scenarios*; Technical Report; United States Department of Transportation, National Highway Traffic Safety Administration: Washington, DC, USA, 2018.
12. Zhang, Y.; Sun, B.; Li, Y.; Zhao, S.; Zhu, X.; Ma, W.; Ma, F.; Wu, L. Research on the Physics–Intelligence Hybrid Theory Based Dynamic Scenario Library Generation for Automated Vehicles. *Sensors* **2022**, *22*, 8391. [[CrossRef](#)]
13. PEGASUS Project. Scenario Description and Knowledge-Based Scenario Generation. Available online: https://www.pegasusprojekt.de/files/tmp1/Pegasus-Abschlussveranstaltung/05_Scenario_Description_and_Knowledge-Based_Scenario_Generation.pdf (accessed on 12 June 2024).
14. Ma, Y.; Jiang, W.; Zhang, L.; Chen, J.; Wang, H.; Lv, C.; Wang, X.; Xiong, L. Evolving testing scenario generation method and intelligence evaluation framework for automated vehicles. *arXiv* **2023**, arXiv:2306.07142.
15. Schelter, S.; Lange, D.; Schmidt, P.; Celikel, M.; Biessmann, F.; Grafberger, A. Automating large-scale data quality verification. *Proc. VLDB Endow.* **2018**, *11*, 1781–1794. [[CrossRef](#)]
16. So, J.J.; Park, I.; Wee, J.; Park, S.; Yun, I. Generating traffic safety test scenarios for automated vehicles using a big data technique. *KSCE J. Civ. Eng.* **2019**, *23*, 2702–2712. [[CrossRef](#)]
17. Sagioglu, S.; Sinanc, D. Big data: A review. In Proceedings of the 2013 International Conference on Collaboration Technologies and Systems (CTS), San Diego, CA, USA, 20–24 May 2013; pp. 42–47.
18. Rana, A.; Malhi, A. Building safer autonomous agents by leveraging risky driving behavior knowledge. In Proceedings of the 2021 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI), Virtual Event, 3–5 March 2021; pp. 1–6.
19. Puterman, M.L. Markov decision processes. In *Handbooks in Operations Research and Management Science*; Elsevier: Amsterdam, The Netherlands, 1990; Volume 2, pp. 331–434.
20. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [[CrossRef](#)]
21. Arief, M.; Glynn, P.; Zhao, D. An accelerated approach to safely and efficiently test pre-production autonomous vehicles on public streets. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2006–2011.
22. Zhao, D.; Peng, H.; Bao, S.; Nobukawa, K.; LeBlanc, D.J.; Pan, C.S. Accelerated evaluation of automated vehicles using extracted naturalistic driving data. In Proceedings of the 24th International Symposium of Vehicles on Road and Tracks, Gothenburg, Sweden, 17–21 August 2015.
23. Zhao, D.; Lam, H.; Peng, H.; Bao, S.; LeBlanc, D.J.; Nobukawa, K.; Pan, C.S. Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 595–607. [[CrossRef](#)] [[PubMed](#)]
24. Glynn, P.W.; Iglehart, D.L. Importance sampling for stochastic simulations. *Manag. Sci.* **1989**, *35*, 1367–1392. [[CrossRef](#)]
25. Tan, S.; Wong, K.; Wang, S.; Manivasagam, S.; Ren, M.; Urtasun, R. Scenegen: Learning to generate realistic traffic scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 892–901.
26. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; Woo, W.-C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 802–810.
27. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
28. Yegnanarayana, B. *Artificial Neural Networks*; PHI Learning Pvt. Ltd.: Delhi, India, 2009.
29. O’Shea, K. An Introduction to Convolutional Neural Networks. *arXiv* **2015**, arXiv:1511.08458.
30. Liu, H.; Zhang, L.; Hari, S.K.S.; Zhao, J. Safety-Critical Scenario Generation via Reinforcement Learning Based Editing. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 14405–14412.
31. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285. [[CrossRef](#)]
32. Doersch, C. Tutorial on variational autoencoders. *arXiv* **2016**, arXiv:1606.05908.
33. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [[CrossRef](#)]
34. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.

35. Feng, S.; Feng, Y.; Yu, C.; Zhang, Y.; Liu, H.X. Testing scenario library generation for connected and automated vehicles, part I: Methodology. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 1573–1582. [[CrossRef](#)]
36. Feng, S.; Feng, Y.; Sun, H.; Bao, S.; Zhang, Y.; Liu, H.X. Testing scenario library generation for connected and automated vehicles, part II: Case studies. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 5635–5647. [[CrossRef](#)]
37. Feng, S.; Feng, Y.; Sun, H.; Zhang, Y.; Liu, H.X. Testing scenario library generation for connected and automated vehicles: An adaptive framework. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 1213–1222. [[CrossRef](#)]
38. Haklay, M.; Weber, P. Openstreetmap: User-generated street maps. *IEEE Pervasive Comput.* **2008**, *7*, 12–18. [[CrossRef](#)]
39. Monahan, G.E. State of the art—A survey of partially observable Markov decision processes: Theory, models, and algorithms. *Manag. Sci.* **1982**, *28*, 1–16. [[CrossRef](#)]
40. Kuutti, S.; Bowden, R.; Jin, Y.; Barber, P.; Fallah, S. A survey of deep learning applications to autonomous vehicle control. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 712–733. [[CrossRef](#)]
41. Jin, Y. Does level-k behavior imply level-k thinking? *Exp. Econ.* **2021**, *24*, 330–353. [[CrossRef](#)]
42. Crawford, V.P.; Iriberri, N. Level-k auctions: Can a nonequilibrium model of strategic thinking explain the winner’s curse and overbidding in private-value auctions? *Econometrica* **2007**, *75*, 1721–1770. [[CrossRef](#)]
43. Fujimoto, S.; Hoof, H.; Meger, D. Addressing function approximation error in actor-critic methods. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1587–1596.
44. Ge, J.; Xu, H.; Zhang, J.; Zhang, Y.; Yao, D.; Li, L. Heterogeneous driver modeling and corner scenarios sampling for automated vehicles testing. *J. Adv. Transp.* **2022**, *2022*, 8655514. [[CrossRef](#)]
45. Wei, Z.; Huang, H.; Zhang, G.; Zhou, R.; Luo, X.; Li, S.; Zhou, H. Interactive Critical Scenario Generation for Autonomous Vehicles Testing Based on In-depth Crash Data Using Reinforcement Learning. *IEEE Trans. Intell. Veh.* **2024**. [[CrossRef](#)]
46. Sun, H.; Feng, S.; Yan, X.; Liu, H.X. Corner case generation and analysis for safety assessment of autonomous vehicles. *Transp. Res. Rec.* **2021**, *2675*, 587–600. [[CrossRef](#)]
47. Roderick, M.; MacGlashan, J.; Tellex, S. Implementing the deep q-network. *arXiv* **2017**, arXiv:1711.07478.
48. Abdi, H.; Williams, L.J. Principal component analysis. *Wiley Interdiscip. Rev. Comput. Stat.* **2010**, *2*, 433–459. [[CrossRef](#)]
49. Krishna, K.; Murty, M.N. Genetic K-means algorithm. *IEEE Trans. Syst. Man Cybern. Part B* **1999**, *29*, 433–439. [[CrossRef](#)]
50. Khan, K.; Rehman, S.U.; Aziz, K.; Fong, S.; Sarasvady, S. DBSCAN: Past, present and future. In Proceedings of the the Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014), Bangalore, India, 17–19 February 2014; pp. 232–238. [[CrossRef](#)]
51. Chen, B.; Chen, X.; Wu, Q.; Li, L. Adversarial evaluation of autonomous vehicles in lane-change scenarios. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 10333–10342. [[CrossRef](#)]
52. Tokdar, S.T.; Kass, R.E. Importance sampling: A review. *Wiley Interdiscip. Rev. Comput. Stat.* **2010**, *2*, 54–60. [[CrossRef](#)]
53. Feng, S.; Yan, X.; Sun, H.; Feng, Y.; Liu, H.X. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nat. Commun.* **2021**, *12*, 748. [[CrossRef](#)]
54. Hussein, A.; Gaber, M.M.; Elyan, E.; Jayne, C. Imitation learning: A survey of learning methods. *ACM Comput. Surv. (CSUR)* **2017**, *50*, 21.
55. Jia, L.; Yang, D.; Ren, Y.; Qian, C.; Feng, Q.; Sun, B.; Wang, Z. A dynamic test scenario generation method for autonomous vehicles based on conditional generative adversarial imitation learning. *Accid. Anal. Prev.* **2024**, *194*, 107279. [[CrossRef](#)]
56. Ho, J.; Ermon, S. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems 29*; Curran Associates Inc.: Red Hook, NY, USA, 2016.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.