A Survey on Safety-Critical Driving Scenario Generation – A Methodological Perspective

Wenhao Ding, Chejian Xu, Mansur Arief, Haohong Lin, Bo Li, Ding Zhao

***Abstract*—Autonomous driving systems have witnessed significant development during the past years thanks to the advance in machine learning-enabled sensing and decision-making algorithms. One critical challenge for their massive deployment in the real world is their safety evaluation. Most existing driving systems are still trained and evaluated on naturalistic scenarios collected from daily life or heuristically-generated adversarial ones. However, the large population of cars, in general, leads to an extremely low collision rate, indicating that safety-critical scenarios are rare in the collected real-world data. Thus, methods to artificially generate scenarios become crucial to measure the risk and reduce the cost. In this survey, we focus on the algorithms of safety-critical scenario generation in autonomous driving. We first provide a comprehensive taxonomy of existing algorithms by dividing them into three categories: data-driven generation, adversarial generation, and knowledge-based generation. Then, we discuss useful tools for scenario generation, including simulation platforms and packages. Finally, we extend our discussion to five main challenges of current works – fidelity, efficiency, diversity, transferability, controllability – and research opportunities lighted up by these challenges.**

***Index Terms*—Autonomous vehicles, Safety, Robustness, Deep Generative Models**

I. Introduction

ARTIFICIAL Intelligence (AI) has been widely used in software products such as facial recognition [1] and voice-print verification [2]. But as AI continues to grow and research has expanded to physical products like autonomous vehicles (AVs), the question of safety is now at the forefront of this cutting-edge field. The reason why intelligent physical systems are much harder to be deployed is that our world is complicated and long-tailed, causing too much uncertainty to the intelligent agents. The driving skills would take several months to learn, even for us humans, due to the complex traffic scenarios. Therefore, the AVs should be trained and evaluated on lots of different scenarios to demonstrate their safety and capability of dealing with diverse situations [3], [4].

According to the 2020 disengagement report from the California Department of Motor Vehicle [5], there were at least five companies (Waymo, Cruise, AutoX, Pony.AI, Argo.AI) that made their AVs drive more than 10,000 miles without disengagement. These results are usually recorded in normal driving scenarios without risky situations. It is a great achievement that current AVs are successful in normal cases trained by hundreds of millions of miles of training. However, we are still not sure whether AVs have enough safety and robustness in distinct scenarios. For example, when one AV is driving on the road, a kid suddenly runs into the drive lane chasing a ball. This emergency case leaves the AV a very short time to react, and even a subtle misbehave could cause vital damage. This kind of situation is named *safety-critical* scenarios and is usually extremely rare in the normal driving case, as shown in Figure 1.

To efficiently evaluate the safety of AVs, more and more people from the government, industry, and academia start to focus on the generation of safety-critical scenarios. National Highway Traffic Safety Administration (NHTSA), a government agency of the United States Department of Transportation, summarized driving system testable cases [6] and pre-crash scenarios [7] as well as published a review of simulation frameworks and standards related to driving scenarios [8]. Waymo, one of the leading companies in autonomous driving, released a safety report to illustrate how they reconstruct fatal crashes in simulations from collected data [9]. Meanwhile, academic research aims to generate safety-critical scenarios [10], [11] surges by developing methods in Deep Generative Models (DGMs) [12] and adversarial attack techniques [13]. With the upcoming massive deployment of self-driving cars, an overview that systematically summarizes existing works in safety-critical scenario generation is urgently demanded.

The **objective of this survey** is to thoroughly review the literature on safety-critical scenario generation from a methodological perspective and provide a panorama of the approaches developed so far. The key contributions are as follows:

- We built a taxonomy to categorize existing algorithms of safety-critical scenarios generation according to the information and general framework they leverage.
- We summarized the tools used for scenario generation, including autonomous driving simulators and open-sourced packages for scenario design, as well as commonly used scenario datasets.
- We identify five challenges of safety-critical scenario generation and corresponding future directions to further push the frontier.

Existing surveys. There have existed surveys [3], [4], [14] summarizing the scenario-based method for autonomous vehicle evaluation. Among these previous works, [4] gives a clear and inspiring definition and categorization of traffic scenarios. Their framework divide the scenario generation

Wenhao Ding, Mansur Arief, Haohong Lin, and Ding Zhao were with the Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA. e-mail: {wenhaod, marief, haohongl}@andrew.cmu.edu, dingzhao@cmu.edu.

Chejian Xu and Bo Li were with the Computer Science Department, University of Illinois Urbana-Champaign, Urbana, IL, USA. e-mail: chejian2@illinois.edu, lbo@illinois.edu.

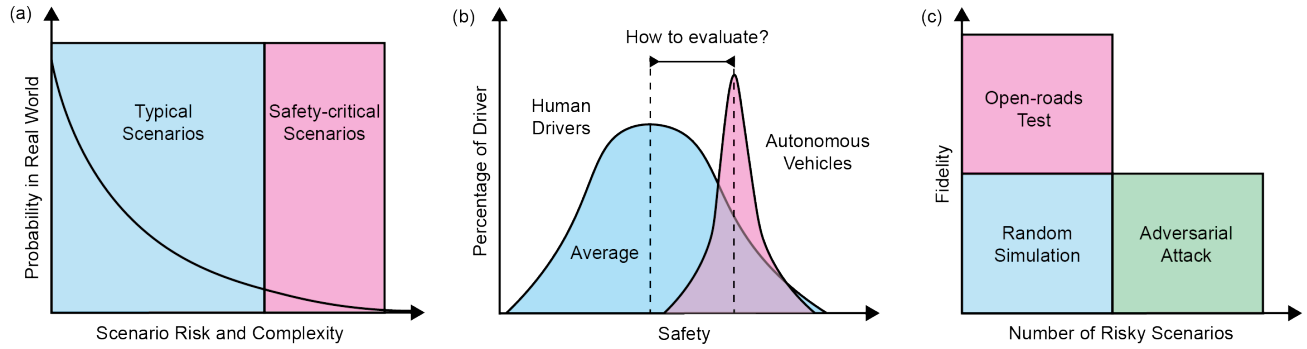


Fig. 1. The overview of autonomous vehicle evaluation. (a) Most of the scenarios that happen in the real world are typical scenarios; safety-critical scenarios are extremely rare. (b) AVs are supposed to have higher average safety than human drivers, but the gap is not easy to be evaluated and measured. (c) Comparison between different generation methods. Most existing methods cannot satisfy the fidelity and safety-critic metrics simultaneously.

problem into two processes: (1) scenario collection, where the scenarios either come from expert abstract knowledge or real-world collected dataset; (2) scenario selection, where the scenario is selected by sampling or feedback optimization. A recent survey [3] about scenario-based testing algorithms for autonomous driving systems divides a scenario into five layers (road level, traffic infrastructure, manipulation of previous two layers, object, environment) and discusses the parameters searching in each layer separately. In [14], the authors propose another perspective by categorizing traffic scenarios into three layers: functional, logical, and concrete. This is a hierarchical structure since the abstracting level decreases from left to right, and the number of scenarios increases at the same time. These categorizations provide inspiring yet different perspectives on scenario-based testing. These works provide a great overview of the entire framework of the evaluation of autonomous driving systems.We want to emphasize that our work differs from them in that we focus on the algorithms of *safety-critical* scenario generation - the core component in evaluating safety and robustness. Our taxonomy is built according to the information that the generation algorithms leverage, focusing on how these generation methods deal with the structure of traffic participants in a scenario and how the autonomous vehicle interacts with surrounding objects. Moreover, we identify five challenges that limit the current generation method, which we hope would inspire future directions.

Organization of this survey. Section II gives an overview of safety-critical scenarios, including the definition, representation, and metrics used for selecting the scenarios. Then, we divide the generation methods into three types: *Data-driven Generation* (Section III) purely samples collect dataset and uses density estimation models for a generation; *Adversarial Generation* (Section IV) considers the ego vehicle during the generation and builds an adversarial learning framework. *Knowledge-based Generation* (Section V) either uses pre-defined rules by experts or integrates external knowledge during the generation. We also summarize the traffic simulators, datasets, and open-sourced packages that can accelerate the development of driving scenarios in Section VI. In Section VII, we list five challenges of safety-critical scenario generation and discuss the potential research directions lighted up by these challenges. Section VIII closes this survey with general conclusions and lessons learned from the current stage.

II. Overview of Safety-critical Scenario

A. Definitions

In this survey, we define a scenario with static and dynamic contents in it and also consider the behavior of dynamic objects. Formally, we have the following definition:

Definition 2.1 (Driving Scenario): The driving scenario is defined by a combination of three sets: $x \in \mathcal{X} = \{\mathcal{S}, \mathcal{I}, \mathcal{B}\}$. $\mathcal{S}$ *represents the static environment, including road shape, traffic sign, traffic light, etc.* $\mathcal{I}$ represents the initial condition and properties of dynamic objects. $\mathcal{B}$ represents the sequential behaviors of dynamic objects.

Separating the static and dynamic contents benefits the representation and generation of scenarios and has shown great success in [91]. The road geometry and static traffic objects (e.g., traffic signs and traffic lights) determine the background of the scenario and the long-term goal of the testing. The dynamic objects that participate in the traffic enable complex interaction with the AV and influence its short-term decisions of it. Note that in this definition, we do not have any specific requirements for the AV. Since modern AV algorithms are usually divided into several important components, we consider the following definition of an AV system:

Definition 2.2 (AV System): An AV system is denoted as F*, which consists of three modules:* $F = F_{per} \circ F_{pla} \circ F_{con}$. F_{per} *denotes the perception module that takes sensor data in and outputs locations of surrounding objects.* F_{pla} *denotes the planning module that provides a feasible trajectory using the prediction of potential obstacles.* F_{con} *denotes the control module that executes control signals to follow given waypoints.*

With the above two definitions, we then define the objective of safety-critical scenario generation:

Definition 2.3 (Safety-critical Scenario Generation): Assume the distribution of scenario x *is parametrized by* θ*, the safety-critical scenarios can be generated by*

$$\hat{\theta} = \arg\max \mathbb{E}_{x \sim p_\theta(x)} \left[g(F, x) \right] \tag{1}$$

where $g(\cdot, \cdot)$ *is the metric function that measures the risk level and safety-related properties (e.g., collision rate and distance driven out of road).* F *is the AV system that operates in scenario* x *in and outputs a sequence of control signals.*

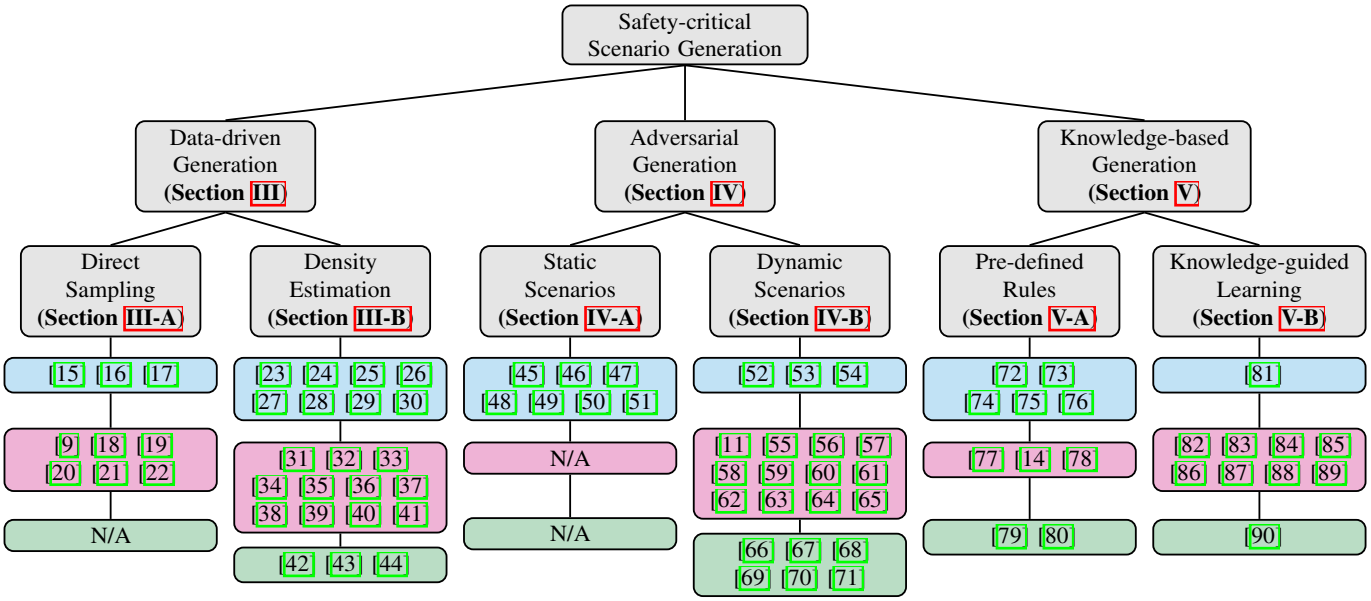


Fig. 2. Taxonomy of safety-critical scenarios generation methods. The colors of boxes denote the modules of the AV system that the generation algorithms target on Perception, Planning, Control.

This optimization problem is not easy to solve mainly because of two aspects: a) the representation of the distribution $p_\theta(x)$, and b) the selection of the metric g. We will discuss how existing methods deal with these two problems in the following two subsections, respectively.

B. Representation of Scenario

We have multiple choices to represent the scenario, and generally, the selection depends on the module to be evaluated. If the target module is a perception module, we use high-dimensional sensing data, such as images and LiDAR point cloud. Directly generating high-fidelity sensing data is quite difficult, and there are an increasing number of works focused on this area [92]. An alternating method is leveraging differential renderer [93], [94] and LiDAR simulator [46], [81] to generate high-dimensional data with ray casting algorithms. If we want to evaluate motion planning or control modules, we can turn to lower dimensions. We can use either trajectories or policy models of dynamic objects. Using a trajectory is less flexible than using a policy model, but the scenarios with trajectory representation are more controllable.

C. Metrics for Generation

The most important factor of solving the optimization problem Definition 2.3 is the metric of risk g. A proper risk metric makes the generated scenarios useful for discovering failure cases of AV systems, while an improper risk metric only provides useless scenarios that either are too trivial for AV systems or too rare to happen in the real world. The level of risk is mainly reflected by the interaction between the autonomous vehicle and participants in the scenario, which can be naturally described by the distance – a small distance means the risk of collision is high. This intuition can be described by Time-to-Collision (TTC) [95]:

$$TTC_F(t) = \frac{X_L(t) - X_F(t) - l_L}{\dot{X}_F(t) - \dot{X}_L(t)}, \ \forall \ \dot{X}_F(t) > \dot{X}_L(t), \quad (2)$$

where X denotes the position, $\dot{X}$ denotes the derivative of X with respect to time or the speed, and l_L denotes the leading vehicle's length; L and F as subscripts refer to leading and following vehicles in a car-following process. Following this time-based metric, there are numerous variants of TTC such as Time Exposed TTC (TET) [96] and Modified TTC (MTTC) [97].

Another two main types of metrics are distance-based and deceleration-based metrics. The first one uses the distance available to avoid a collision, e.g., Proportion of Stopping Distance (PSD) [98]. The second one defines dangerous situations using the rate deceleration during an emergency, e.g., Deceleration Rate to Avoid the Crash (DRAC) [99]. Please check [100] for more metrics belonging to these two types. In addition, we can also use the posthoc analysis of unsafe behavior to indicate the risk, for example, collision rate, average distance driven out of the road, and the frequency of running stop signs and red lights [73], [74].

However, measuring only the risk is not enough for scenario generation. We need to ensure that the scenarios are realistic and able to happen in the physical world. We also want to discover as many as possible scenarios rather than a single one. In fact, these kinds of constraints make the generation of scenarios a much harder task. We will discuss this topic more in Section VII.

Finally, we highlight the need to balance the effort in modeling the input scenario with the effort in utilizing the scenario to compute the final metrics. This is of particular importance since the uncertainty in the final metrics is due to both modeling uncertainty and sampling uncertainty. While the

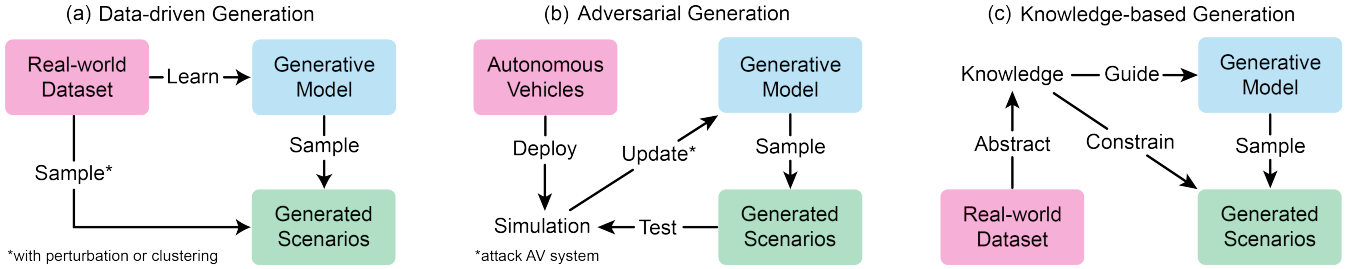


Fig. 3. Illustration of three types of generation methods. (a) Data-driven methods only use the collected data to sample directly or via generative models. (b) Adversarial methods use the feedback from the autonomous vehicle that is deployed in the simulation. (c) Knowledge-based methods leverage the information mainly from external knowledge as constraints or guidance to the generation.

former can be driven down by designing as accurate a model as possible, the latter can only be reduced by running longer, and a larger number of simulation runs [101]. A good review of this input-induced uncertainty and ways to deal with it in modeling can be found in [102]–[104].

III. Data-driven Generation

In this section, we consider the algorithms that only leverage information from the collected datasets. For example, we control several vehicles to pass an intersection by making them follow the trajectories recorded in the same intersection in the real world. These methods are mainly divided into two parts. The first part directly samples from the dataset $x \sim \mathcal{D}$ to reproduce the real-world log, which usually suffers from the problem of rareness. The second part is using density estimation models (e.g. DGMs) $p_\theta(x)$ parameterized by θ to learn the distribution of scenarios, which enables the generation of unseen scenarios. Usually, the learning objective of these models is maximizing the log-likelihood

$$\hat{\theta} = \arg\max \sum_x \log p_\theta(x), \tag{3}$$

and the sampling process is conducted by $x \sim p_\theta(x)$ from random noises.

A. Direct Sampling

An intuitive way of generation is directly sampling from a collected dataset which reproduces the scenario from the road test log. According to different algorithms used before and during the sampling, we summarize the following three groups.

1) Data Replay: Most autonomous driving companies maintain scenario bases to store the scenarios they recognized as important [105]. During this process, one crucial step is the tools that can automatically convert the scenario from log to virtual simulations [15]. In addition, efficiently selecting the critical scenario from a huge number of scenarios is another problem. [19] extracts useful scenarios according to the cooperative actions for cooperative maneuver planning evaluation. In [18], the authors developed a method to select a testing ground to accelerate the performance estimation of AVs performance on public streets, where the main contribution is describing the risk intensities of the traffic system in an area of interest with Non-Homogeneous Poisson Process model [106].

2) Clustering: To better categorize the collected scenario, [20] and [21] propose to use unsupervised clustering methods to put similar scenarios into groups, which helps improve the efficiency of testing AVs on a specific type of scenario. However, clustering the entire scenario might be inefficient and inaccurate since the scenarios are usually complex and composed of finite building blocks. The concept of *Traffic Primitive* is proposed in [22] to represent those blocks. They use the Hierarchical Dirichlet Process Hidden Markov Model (HDP-HMM) [107], a non-parametric Bayesian learning method, to unsupervised extract primitives from scenarios to better cluster similar scenarios.

3) Random Perturbation: The main obstacle to direct conversion is that the diversity of scenarios is limited. Therefore, recent works from some leading companies have started to use random perturbation to augment the number of scenarios. Baidu creates a physical-based LiDAR model [16] to transfer a dense point cloud to match the line-style output of the LiDAR sensor. By randomly placing the Computer-Aided Design model of vehicles and pedestrians, their algorithms can generate a huge number of scenarios. Similarly, Uber builds a more precise LiDAR model [17] using neural networks (NNs) to mimic the reflection details of the real-world sensors. Besides the high-dimensional representation, Waymo tries to reconstruct more fatal crashes from collected data by randomly perturbing important parameters [9].

B. Density Estimation Methods

Consider the driving scenarios following a distribution, and then we can use collected data to learn a density model to approximate this distribution. We divide this type of algorithm into three categories according to the density model they use.

1) Bayesian Networks: Bayesian Networks is a probabilistic graphical model that uses nodes to represent objects and edges to represent the relation between nodes. This structured model can naturally describe the objects in the scenario. [38] uses Dynamic Bayesian Network [108] to model the complex traffic scenarios, and [39] uses factor graphs to model driving behaviors between multiple vehicles. In [41], Importance Sampling (IS) is used to sample driving scenarios represented by Bayesian Networks. In [40], the authors try to find as many risky scenarios as possible and cluster them. They first uniformly sample from the clusters and then use IS to sample the specific scenarios represented by factor

graphs. Besides using Dynamic Bayesian Network or factor graphs to model the structures of traffic participants, Gaussian Process (GP) is a powerful non-parametric method to model the distribution of sequential scenarios [34]. Following this direction, [35] combines GP and Dirichlet Process to build a model with an infinite number of clusters to discover traffic primitives [22], which can be combined to create new scenarios. [27] generates point cloud sequential datasets via minimizing the gap between real-world LiDAR and simulation data. Similarly, Meta-sim [28] and Meta-Sim2 [29] try to minimize the sim-to-real gap to reconstruct traffic scenarios for automatic labeling. Their main contribution is that they use a scene graph to represent the scenario, which is a hierarchical structure that makes the generation more efficient. In [30], the scene graph is also used to generate image scenarios.

2) Deep Learning Models: Deep learning models are also introduced into the generation in [36] and [37]. [36] inputs the current state of the AV and a high-definition map to a Long Short Term Memory (LSTM) [109] module to sequentially generate the trajectory of surrounding vehicles and pedestrians. They train their model with normal traffic data since their target is to generate naturalistic scenarios. [37] proposes a quite complex system to generate scenarios in a simulator, which uses Convolution Neural Network (CNN) [110] as a selector to generate agents surrounding the AV. In TrafficSim [111], both Gated Recurrent Unit (GRU) and CNN are used to learn the behaviors of multi-agents from real-world data. This method can generate realistic multi-agent traffic scenarios.

3) Deep Generative Models (DGMs): Recently, DGMs have shown great success in generating image and voice data. There are five types of modern generative models: Generative Adversarial Nets (GAN) [112], Variational Auto-encoder (VAE) [113], Autoregressive Models [114], [115], Flow-based model [116], [117], Diffusion model [118], [119]. The readers can find more details about DGMs in this survey [12]. An auto-encoder structure is designed in [120] to separately generate vehicle initial positions and vehicle trajectories. With the power of VAE, [31] learns a latent space of encounter trajectories and generates unseen scenarios by sampling from the latent space. However, with less understanding of the latent code, the generation is not controllable. In [33], the authors propose CMTS, which combines normal and collision trajectories to generate safety-critical scenarios by doing interpolation in the latent space. As for the usage of GAN, [32] introduces recurrent models to generate realistic scenarios of highway lane changes. They use real-world data in the discriminator to help the improvement of the generator. The advantage of DGMs is that they can learn a low-dimensional latent space of high-dimensional and structured data using NNs. Therefore, we can easily generate high-dimensional sensing scenarios. SurfelGAN is proposed in [23] to directly generate point cloud data to represent scenarios from the view of the AV. [24] can add new vehicles to collected driving videos to generate realistic video scenarios, where they also consider the motion planning of vehicles. [25] generates traffic videos with multi-object scene synthesis using a GAN framework. To make the video realistic, they integrate physical conditions into the generation. A data-driven scenario simulator is designed in [26] to generate both LiDAR and trajectory data to augment the diversity of driving scenarios.

4) Imitation learning: Another way of mimicking the dataset is using imitation learning (IL) methods [121], which takes the observation as input and directly outputs the behavior to control agents. The training of IL is exactly the same as supervised learning. After training, the IL model [42]–[44] can reproduce the same behavior as real-world agents when the model encounters the same observation. However, when the observation is not covered by the dataset, the behavior of the model could be unreasonable and unpredictable.

IV. Adversarial Generation

In this section, we consider a more efficient way for generation, which actively creates risky scenarios by attacking the AV system. For example, we control a pedestrian to cross the road and intentionally make it collide with the AV. Although the AV may avoid the collision in most cases, we can still obtain safety-critical scenarios where accidents happen. This framework, named adversarial generation, consists of two components, one is the generator, and the other is the victim model, i.e., the AV. Then the targeted generation process can be formulated as

$$\hat{\theta} = \arg\min \mathbb{E}_{p_\theta(x)}[Q(x, \pi)] \tag{4}$$

where $Q(\cdot, \pi)$ is a quantitative function to indicate the performance of the policy π taken by the AV. We notice that the adversarial generation will mainly focus on a specific small set of scenarios therefore it would be good to consider the diversity by adding constraint or entropy of the distribution $\mathbb{H}(x)$. Since we consider the influence of AV, this type is also named Vehicle-in-the-loop testing in previous work [122]. Since the autonomous driving system consists of multiple modules, we divide these methods according to the type of victim models. When the model is used for single frame inputs, e.g., object detection and segmentation, we only need to generate static scenarios. When the victim model requires a sequential testing case, we generate dynamic scenarios which contain the motion of all objects.

A. Static Scenarios Generation

Under the adversarial framework, directly generating high-dimensional data could be challenging. Thus most methods generate the poses of objects and then use renders to get the final output. [45] learns the poses of vehicles and uses a differentiable render to get the first-point-view images to attack object detection algorithms. [49] and [50] extend the original Domain Randomization methods [123] to the adversarial generation. [49] proposes Structured Domain Randomization, which uses a Bayesian Network to actively generate the poses of vehicles. [50] proposes Adversarial Domain Randomization, which targets the scenario of parking lots.

All of the above methods focus on image generation, but some works generate point cloud scenarios. [46] puts an adversarial object on the top of a vehicle and optimizes the shape of the object to make the vehicle disappear in LiDAR detection algorithms. Similarly, [47] shares the same idea but uses both LiDAR and image information. After the optimization

of the shape of the object, [48] use 3D printing to build the object in the real world. The experiment shows that the object on the road is indeed ignored by the detection algorithms. In addition to using renders, [51] tries to directly add new points to the existing point cloud to attack segmentation algorithms.

B. Dynamic Scenario Generation

When the target is to evaluate planning and control modules, the scenarios are required to be dynamic and sequential. For these kinds of algorithms, we further divide current works into two types according to the flexibility of the scenario.

1) Initial Condition: The first type is controlling the initial conditions of the scenario (e.g., initial velocity and spawn position) or providing the entire trajectory at the beginning. The advantage is the low dimension of search space and the little computational resource required. In [11] and [55], the authors generate the initial poses of a cyclist attacking the AV. [55] approximates the parameters with a Gaussian distribution, which limits the diversity of generated scenario. As a remedy, [55] use normalizing flow models to learn a multi-modal distribution. It also uses the distribution of real-world data as a constraint to improve fidelity. Parallel work [56] generates a set of possible driving paths and identifies all the possible safe driving trajectories that can be taken starting at different times. Similarly, [57] and [124] directly optimize existing trajectories to perturb the driving paths of surrounding vehicles. They use Bayesian Optimization [125] for the optimization, and the scenario is represented with a point cloud. To generate real-world traffic scenarios, [65] optimizes the adversarial trajectory in the latent space of a VAE model. Besides controlling the dynamic objects, some works also focus on searching the weather parameters (e.g., sun and rain) to create different scenarios. [52] searches the type of weather using REINFORCE [126] algorithm. [53] uses Generative Adversarial Imitation Learning [43] to generate parameters of weather and uses the cross-entropy method for efficient scenario searching.

2) Adversarial Policy: The second type is building a policy model to sequentially control the dynamic objects, which contains the most number of existing works. This type is usually formulated as a Reinforcement Learning (RL) problem [127], where the AV belongs to the environment and the generator is the agent we can control. Intuitively, we have much more flexibility under this setting, but the complexity also increases. Since the AV and objects in scenarios interact step-wise, this problem can be formulated in an RL framework, and there are lots of works using RL methods. [59] and [60] use Deep Q-Network to generate discrete adversarial traffic scenarios. [70] uses Advantage Actor-Critic (A2C) [128] to control one surrounding vehicle in the car following scenarios. [68] uses Deep Deterministic Policy Gradient (DDPG) [129] to generate adversarial policy to control surrounding agents to generate lane-changing scenarios. [69] uses Multi-agent DDPG [130] to control two surrounding vehicles (which are called Non-player Characters) to attack the ego vehicle. This method also sets auxiliary goals for Non-player characters to avoid generating unrealistic scenarios. [58] proposes CriSGen that uses constraint-based optimization and [54] uses Bayesian Optimization.

A series of works on generating risky and adversarial scenarios in the context of IS also appear in the literature. [131] uses heuristic approaches to generate dangerous lane change scenarios. [132] constructs IS distribution to sample dangerous AV lane change scenarios. [133] extends IS approach to dynamic systems to sample dangerous AV car-following scenarios. [134] uses piece-wise models to design a more expressive IS distribution. [135] uses Gaussian Mixture Model (GMM) for IS distribution and further analyzes the efficiency of GMM-based IS distribution for random forest and NN classifiers [136]. ReLU-activated NNs are considered in [137] to estimate the dangerous set and compute an IS estimator for a risk upper bound for Gaussian case, with a more general case presented in [67]. The Adaptive IS approach is used to construct adversarial environments to accelerate policy evaluation [138]. Finally, there is a series of works named Adaptive Stress Testing (AST) that explores different ways to generate stress testing scenarios. [71] uses Monte Carlo tree search (MCTS) to search action of testing scenario, but this method does not target autonomous driving systems. [61] generates a scenario controlling a pedestrian to cross the road. [62] improves the last paper by using LSTM to generate initial conditions and actions in each step. Instead of defining heuristic reward functions, [63] leverage the Go-Explore framework to find failure cases. [64] extends previous works to high-fidelity simulation and changes the learning algorithm to Proximal Policy Optimization (PPO) [139].

V. Knowledge-based Generation

In Section III and Section IV, we discussed methods that purely use data or interact with the AV to generate scenarios. However, scenarios are constructed in the physical world, therefore, need to satisfy traffic rules and physical laws. The samples in the density we estimate or the adversarial examples we generate could easily violate these constraints. In addition, domain knowledge also improves the efficiency of the generation. After traffic accidents happen, we humans analyze the scenario and find the reasons that cause the accidents. Finding the underlying causality, e.g., the view of the sensor is blocked, is important to efficiently generate safety-critical scenarios.

Therefore, in this section, we consider methods that incorporate external domain knowledge into the generation process. We will first explore rule-based methods that artificially design the structure and parameters of scenarios. Then, we turn to the learning-based methods that use explicit knowledge to guide the generation. Assume we can obtain certain domain knowledge $c \in \mathcal{C}$ from experts, then we can augment the learning with

$$\hat{\theta} = \arg\max \mathbb{E}_{x \sim p_\theta(x|c)} \left[g(F, x) \right] \tag{5}$$

where the scenarios are sampled from a conditional distribution $p_\theta(x|c)$. In addition, we can also use $\mathcal{C}$ as constraints to manipulate existing scenarios:

$$\tilde{x} = \arg\max g(F, x) \quad s.t.\ \mathcal{C}(x) \geq 0 \tag{6}$$

where $\mathcal{C}(x) \geq 0$ means the constraint is satisfied.

A. Pre-defined Rules

Driving scenarios are very common in daily life thus we humans can easily design scenarios with pre-defined rules and specific conditions to trigger events. [79] uses prior knowledge to design random risk scenarios with equations. The authors also conduct interventional experiments by training RL agents on different scenarios to get comparable results. [14] focuses on functional and logical scenarios, which can be represented by natural language from human experts. [78] views ontologies as knowledge-based systems in the field of AV and proposes a generation of traffic scenes in natural language as a basis for scenario creation. In [76], the authors combine formal specification [140] of scenarios and safety properties to generate test cases from formal simulation.

There are also a large number of works that build the entire platform with pre-defined scenarios implemented. We categorize them under the knowledge-based generation method and will also discuss more details about these platforms in Section VI. A 2D platform named SMARTS is developed in [80] containing multiple diverse behavior models using both rule-based and learning-based models. [74] proposes MetaDrive, a 3D simulator that supports different road shapes defined by users or directly imported from existing real-world datasets (e.g. Argoverse [141]). [73] is a competition built on top of CARLA [142] and [143], which consists of a large number of pre-defined scenarios. In [144], the authors build a rule-based scenario case zoo in CARLA [142], which shares some similar scenarios with [143]. To manage complex traffic scenarios with hundreds of objects, SUMMIT [72] is specifically designed for generating massive mixed traffic with an autopilot algorithm. To explore the causality between vehicles in the scenario, CausalCity [77] is developed for evaluating causal discovery algorithms. It builds an agency mechanism to define high-level behaviors.

B. Knowledge-guided Learning

1) Knowledge as Condition: Combining learning methods and domain knowledge is a popular trend in the machine learning area. One of the biggest obstacles is how to define the representation space of knowledge. This representation should be easily integrated into NNs and still has interpretability. In [90], the authors assume that diversity and high skill are important for scenarios. They use RL methods to generate diverse scenarios and show improvement of the AV trained in their scenarios. [81] represent the explicit knowledge (e.g., the vehicles should not have overlap, the orientation of vehicles should follow the direction of the lane) as first-order-logic [145], which can be embedded into a tree structure. Then they search in the latent space of a VAE model and apply the knowledge to the tree encoder to constraint the searching process. They evaluate their scenarios on point cloud segmentation methods and show that their scenario can cause failure to some models that are not robust. In [82] and [146], the authors explore the fundamental reason for the safety-critical scenario, which can be represented by causality. This work assumes there is a causal graph that can represent the relationships between objects in a scenario. Then, they propose an autoregressive generative model that can use this graph to increase the efficiency of the generation.

2) Constraint Optimization: Another way of using explicit knowledge is resorting to the constraint optimization framework. We know that safety-critical scenarios are extremely rare in the real-world log, and random augmentation could be inefficient to generate safety-critical scenarios. One of the heuristic metrics of risk is the drivable area for the autonomous vehicle. [85] and [86] minimize the drivable area by controlling the surrounding vehicles with an evolutionary method and constraint optimization methods, respectively. To explore more different kinds of scenarios, [87] generates the motion of other traffic participants with a backtracking search. To make the scenarios diverse, [88] build a pipeline that introduces the road topology from OpenStreetMap [147]. Using the safety-critical scenarios from [88], [89] designs a comprehensive open-source toolbox to train and evaluate RL motion planners for AVs with customized configuration from users. To obtain a robust trajectory prediction model, [83] and [84] generate adversarial trajectory by perturbing existing trajectory with feasible constraints.

VI. Dataset and Tools for Scenario Generation

In this section, we introduce the tools that are useful for safety-critical scenario generation. We first discuss the scenario datasets, then turn to the traffic simulators. Finally, we review existing platforms that support the function of scenario generation.

A. Scenario Dataset

For modern machine learning methods, datasets are crucial and necessary. Specifically, for the scenario generation task, there are also many published datasets by companies and academic organizations. In Table I, we summarize and compare available scenario datasets in various aspects that we are especially interested in.

1) Fidelity: In Table I, we include datasets that are either collected from onboard sensors on public roads or synthetic virtual worlds simulated by traffic simulators. Collecting real-world data is very time-consuming, which requires humans to operate vehicles or drones to record data in the real world in a variety of environments. However, these data are more representative of real-world data distribution. Models developed with these realistic data can be directly applied to the real world. Synthetic datasets, on the other hand, are simple to collect. But they heavily rely on the authenticity of traffic simulators, which usually fail to accurately imitate and render real-world data.

2) Collect View: In addition to different levels of reality, we also present datasets with different views. For example, HighD [158], InD [167], and RounD [168] datasets collect data in bird's-eye view (BEV), which is recorded from drone cameras. KITTI [149] and Argoverse [174] datasets collect data in first-person view (FPV), which is captured by cameras in front of a car. The BEV data happens in a fixed region, therefore it is more useful to analyze the behavior of objects in a fixed background. In contrast, FPV data is more suitable to train AV algorithms that take egocentric information.

TABLE I
COMPARISON OF SCENARIO DATASETS. WE LIST EXISTING DATASETS AND DIFFERENT ASPECTS THAT WE ARE INTERESTED IN. ✓/× IN THE WEATHER COLUMN MEANS THE DATASET CONTAINS/DOESN'T CONTAIN DATA UNDER VARIOUS WEATHER CONDITIONS. BEV: BIRD'S-EYE VIEW, FPV: FIRST-PERSON VIEW. D: DAYTIME, N: NIGHTTIME. H: HIGHWAY, I: INTERSECTION, RA: ROUNDABOUT, C: CAMPUS, U: URBAN, S: SUBURBAN, R: RURAL.

Dataset	Year	Real	View	Data Sensor				Annotation			Traffic Condition			
				Image	LiDAR	RADAR	Traj.	3D	2D	Lane	Weather	Time	Region	Jam
CamVid [148]	2009	✓	FPV	RGB	×	×	×	×	✓	✓	×	D	U	×
KITTI [149]	2013	✓	FPV	RGB/Stereo	✓	×	✓	✓	✓	✓	×	D	U/R/H	×
Cyclists [150]	2016	✓	FPV	RGB	×	×	×	×	✓	×	×	D	U	×
Cityscapes [151]	2016	✓	FPV	RGB/Stereo	×	×	✓	✓	✓	×	×	D	U	✓
SYNTHIA [152]	2016	×	FPV	RGB	×	×	×	✓	✓	×	✓	D/N	U	✓
Campus [153]	2016	✓	BEV	RGB	×	×	✓	×	✓	✓	×	D	C	×
RobotCar [154]	2016	✓	FPV	RGB	✓	×	✓	✓	✓	×	✓	D/N	U	×
Mapillary [155]	2017	✓	FPV	RGB	×	×	×	×	✓	✓	✓	D/N	U	×
P.F.B. [156]	2017	×	FPV	RGB	×	×	✓	✓	✓	×	✓	D/N	U	×
BDD100K [157]	2018	✓	FPV	RGB	×	×	✓	×	✓	✓	✓	D	U/H	×
HighD [158]	2018	✓	BEV	RGB	×	×	✓	×	✓	×	×	D	H	✓
Udacity [159]	2018	✓	FPV	RGB	×	×	×	×	✓	×	×	D	U	×
KAIST [160]	2018	✓	FPV	RGB/Stereo	✓	×	✓	×	✓	×	×	D/N	U	×
Argoverse [161]	2019	✓	FPV	RGB/Stereo	✓	×	✓	✓	×	✓	✓	D/N	U	×
TRAF [162]	2019	✓	FPV	RGB	×	×	✓	×	✓	×	✓	D/N	U	×
ApolloScape [163]	2019	✓	FPV	RGB/Stereo	✓	×	✓	✓	✓	✓	✓	D	U	✓
ACFR [164]	2019	✓	BEV	RGB	×	×	✓	×	✓	×	×	D	RA	×
H3D [165]	2019	✓	FPV	RGB	✓	×	✓	✓	×	×	×	D	U	✓
INTERACTION [166]	2019	✓	BEV	RGB	×	×	✓	×	✓	✓	×	D	I/RA	×
InD [167]	2020	✓	BEV	RGB	×	×	✓	×	✓	×	×	D	I	×
RounD [168]	2020	✓	BEV	RGB	×	×	✓	×	✓	×	×	D	RA	×
nuScenes [169]	2020	✓	FPV	RGB	✓	✓	✓	✓	✓	×	✓	D/N	U	×
Lyft Level 5 [170]	2020	✓	BEV	RGB	✓	✓	✓	×	✓	✓	×	D	S	×
Waymo Open [171]	2020	✓	FPV	RGB	✓	×	✓	✓	✓	✓	✓	D/N	U/S	×
A*3D [172]	2020	✓	FPV	RGB	✓	×	×	✓	✓	×	✓	D/N	U	✓
RobotCar Radar [173]	2020	✓	FPV	RGB	✓	✓	✓	×	×	×	✓	D/N	U	×
Argoverse 2 [174]	2021	✓	FPV	RGB/Stereo	✓	×	✓	✓	×	✓	✓	D/N	U	×
PandaSet [175]	2021	✓	FPV	RGB	✓	×	✓	✓	×	×	×	D/N	U	×
ONCE [176]	2021	✓	FPV	RGB/Stereo	✓	×	✓	✓	✓	×	✓	D/N	U	×

3) Data Sensor: For each dataset, we examine whether it has the following data types: RGB image, stereo image, LiDAR data, Radar, and trajectory. All datasets included in the table have RGB images since it is a common data type in traffic scenarios. RGB images are usually collected by cameras mounted on vehicles or drones, and they can be utilized for a variety of computer vision tasks such as object detection and image segmentation. Stereo images are captured by stereo cameras and LiDAR data is collected by LiDAR sensors. Both of them provide 3D information that is particularly useful in 3D tasks such as 3D object detection. RADAR is also a common sensor that returns similar 3D information as LiDAR but at a cheaper price. It can work in harsher conditions (e.g., rain and storm) due to the longer wavelength than LiDAR's lights. The trajectory data is either recorded by sensors such as GPS or converted from object tracking. This type of data is frequently used in trajectory prediction tasks for planning and control purposes.

4) Annotation Type: The availability of various types of annotations is crucial to each dataset, as it determines which tasks the dataset may be used for. We explore three forms of data annotations in Table I: 3D object annotations, 2D object annotations, and lane annotations. 3D annotations can be divided into two categories: 3D bounding box annotations and 3D point cloud annotations. A 3D bounding box annotation describes a cube that exactly holds one specific object. 3D point cloud annotations assign point-wise labels to each point in the point cloud, indicating the point's category. Many tasks in autonomous driving, such as 3D object detection and 3D segmentation, require 3D annotations. Similarly, 2D annotations include 2D bounding box annotations and pixel-wise 2D semantic annotations. Lane annotations describe different types of lanes in the data, as well as the boundaries of drivable regions. This lane information can be used to integrate map and traffic rule information into the algorithm of AVs, enabling more efficient decision-making functions.

5) Traffic Condition: Datasets with a limited level of diversity in conditions and situations are skewed to redundant and highly safe scenarios, which leads to the long tail problem [177], [178]. In order to evaluate the performance of AVs in various scenarios, a dataset must include data under a variety of settings. We mainly consider the following four key aspects: weather, time, region, and traffic density. The weather conditions change the entire world as well as the style of the data collected by sensors. For example, the nuScenes dataset [169] includes data from sunny, rainy, and cloudy conditions. As a result, the

TABLE II
COMPARISON OF TRAFFIC SIMULATION

Simulator	Year	Open Source	Realistic Perception	Customized Scenario	Back-end	Map Source		API Support		
						Real World	Human Design	Python	C++	ROS
TORCE [179]	2000	✓	✓	×	None	×	✓	×	✓	×
Webots [180]	2004	✓	✓	✓	ODE	✓	✓	✓	✓	✓
CarRacing [181]	2016	✓	×	×	None	×	✓	✓	×	×
CARLA [142]	2017	✓	✓	✓	UE4	×	✓	✓	✓	✓
SimMobilityST [182]	2017	✓	×	✓	None	×	✓	✓	×	×
GTA-V [156]	2017	×	✓	✓	RAGE	×	×	×	×	×
highway-env [183]	2018	✓	×	✓	None	×	✓	✓	×	×
Deepdrive [184]	2018	✓	✓	✓	UE4	×	✓	✓	✓	×
esmini [185]	2018	✓	✓	✓	Unity	×	✓	✓	✓	×
AutonoViSim [186]	2018	×	✓	✓	PhysX	×	✓	×	×	×
AirSim [187]	2018	✓	✓	✓	UE4	×	✓	✓	✓	✓
SUMO [188]	2018	✓	×	✓	None	✓	✓	✓	✓	×
Apollo [189]	2018	✓	×	✓	Unity	×	✓	✓	✓	×
Sim4CV [190]	2018	✓	✓	✓	UE4	×	✓	✓	✓	×
SUMMIT [72]	2020	✓	✓	×	UE4	✓	✓	✓	×	✓
MultiCarRacing [191]	2020	✓	×	×	None	×	✓	✓	×	×
SMARTS [80]	2020	✓	×	✓	None	×	✓	✓	×	×
LGSVL [192]	2020	✓	✓	✓	Unity	✓	✓	✓	×	✓
CausalCity [77]	2021	✓	✓	✓	UE4	×	✓	✓	×	×
MetaDrive [74]	2021	✓	✓	✓	Panda3D	✓	✓	✓	×	×
L2R [193]	2021	✓	✓	✓	UE4	✓	✓	✓	×	×
AutoDRIVE [194]	2021	✓	✓	✓	Unity	×	✓	✓	✓	✓

images have different appearances depending on the weather. The time condition specifies whether the data was obtained during the day or at night. The main distinction between these two scenarios is the lighting. The region denotes the location from which the data is gathered. For example, the INTERACTION dataset [166] is collected at intersections and roundabouts, whereas the KITTI dataset [149] is collected in urban, rural, and highway settings. Finally, traffic density considers the number of objects in traffic scenarios. Higher density indicates a traffic jam or congestion, which requires traffic participants to pay more attention to surrounding objects and take action more carefully.

B. Traffic Simulation Platforms

In Table II, we summarize existing traffic simulators and compare them in different aspects that we are particularly interested in.

1) Open Source: Open-source simulators are easily customized, which enables users to design and evaluate various safety-critical scenarios easily, as these scenarios are typically rare and sophisticated and require a high level of customization. Therefore, whether the simulator is open source becomes one of our primary considerations. Most of the simulation platforms we list in Table II are open source. Simulators such as AirSim [187] and SMARTS [80], for example, release their source code to the public, making it easier for users to modify and enrich the testing environment.

2) Realistic Perception: The realism and fidelity in the virtual simulation have a significant impact on AV algorithms. High-fidelity, photo-realistic simulators provide data that is similar to the physical world, allowing for a more accurate assessment of AV performance in the real world. For example, SUMMIT [72] is able to simulate 3D towns with a large number of vehicles, pedestrians, and buildings. Besides, weather conditions can be further simulated, which include controlling the strength of precipitation, cloudiness, fog density, etc. Simple traffic simulators, on the other hand, are incapable of supporting such detailed simulations. However, they are usually light-weighted and easy to use (especially for RL algorithms), where algorithms can be tested quickly without complicated configurations. For example, Highway-env [183] is a 2D simulator that can be installed using only one command and provide preliminary experimental results in a short amount of time. We provide screenshots of six typical simulators in Figure 4.

3) Customized Scenario: The freedom to customize scenarios is of great importance since we mainly focus on the generation and evaluation of safety-critical scenarios. The customization of scenarios usually involves the modification of vehicles' positions, speeds, and behaviors. For example, CARLA [142] offers a systematical way for users to define a scenario, through which users can specify the number of vehicles in a town as well as their own behaviors. Even weather and the surrounding environment can be arbitrarily adjusted to create variations in the visual appearance of the scene. These functions provide higher flexibility for users and allow for more comprehensive testing of the reliability of AVs.

4) Back-end Engine: The simulator engines have a direct impact on the fidelity of the simulated vehicle dynamics and the rendered 3D environment. Most simulation platforms are

built upon Unreal Engine 4 (UE4) [1] or Unity [2], which are popular and professional game engines. Other less popular engines, such as Panda3D [195], are also considered. Some non-realistic environments, such as Highway-env [183], do not even need back-end engines due to the low requirement of rendering. A simple graphical user interface (GUI) is sufficient for such type of platform.

5) Map Source: Maps are critical components in AV testing systems. The maps in these traffic simulators are either created by humans or based on real-world data. For example, Learn-to-Race (L2R) [193] platform has three racetracks in its racing simulator, all of which are based on real-world racetracks. Human-designed maps can be further classified into two types: rule-based maps and procedurally generated maps. Highway-env [183] incorporates 11 rule-based maps, including highways, parking lots, roundabouts, etc., all of which are manually written by humans. MetaDrive [74] maintains several basic roadblocks and generates numerous maps in a procedural manner by randomly selecting one roadblock at one time.

6) API Support: API support for specific programming languages is crucial in large-scale automated evaluation since it allows users to run batches of scenarios. Popular programming languages like Python and C++ are supported by most simulation platforms. Some simulators like CARLA [142] and LGSVL [192] also provide support for Robot Operating System (ROS) [3], allowing users to integrate other open-source modules developed by the ROS community.

C. Scenario Design Platform

There are several user-friendly platforms that support scenario design, which already implements a lot of rule-based scenarios that are normal or safety-critical. We list some popular platforms in this section.

CARLA Scenario Runner [143] provides traffic scenario definitions and an execution engine for CARLA. Scenarios can be defined through a Python interface that allows users to easily describe sophisticated and synchronized maneuvers that involve multiple entities like vehicles, pedestrians, and other traffic participants. It also supports the OpenSCENARIO [196] standard file format for scenario descriptions, making it simple and efficient to incorporate a variety of existing traffic scenarios from the community.

SCENIC [75] defines a language for scenario specification and generation. It describes distributions over scenes and the behaviors of their agents over time. One advantage of SCENIC over other scenario languages is that it combines the concise, readable syntax for spatiotemporal relationships with the ability to impose hard and soft constraints over the scenario.

SafeBench [197] is an open-source platform focusing on systematically evaluating the safety and robustness of autonomous driving algorithms based on diverse testing scenarios and comprehensive evaluation metrics. The platform integrates eight types of safety-critical scenarios and incorporates four generation algorithms. Users can also design their own traffic scenarios and scenario generation algorithms following the instructions. SafeBench also provides several RL-based autonomous driving algorithms with pre-trained RL model weights. Users can easily test and improve the generated scenarios based on feedback from diverse autonomous driving algorithms.

DI-Drive Casezoo [144] consists of a set of scenarios used to train and evaluate diving policy in a simulator. Similar to CARLA Scenario Runner [143], DI-drive Casezoo has a routing scenario and multiple single scenarios that can be triggered along the route. There are 18 route scenarios and 8 types of single scenarios that can be triggered depending on the route definition. Route scenario is defined in an XML file with its corresponding scenarios. Trigger locations along the route are defined in JSON files. A single scenario is defined in a Python file, describing the behaviors of traffic participants.

SUMO NETEDIT [198] is a graphical scenario editor which can be used to create traffic networks from scratch and to modify all aspects of existing networks, including basic network elements (junctions, edges, and lanes), advanced network elements (e.g., traffic lights), and additional infrastructure (e.g., bus stops). This tool is specifically designed for SUMO [188], which mainly generates large-scale traffic conditions without high-fidelity rendering.

SMARTS Scenario Studio [199] is a scenario design tool in the SMARTS [80] platform that supports flexible and expressive scenario specification. Scenario definitions are written in the Domain Specific Language, which describes the traffic environment, such as traffic vehicles, routes, and agent missions. Scenario Studio also supports configuration files from SUMO's NETEDIT [198]. Maps edited by NETEDIT [198] can be easily included and reused in Scenario Studio, which enriches the training and testing environments in the SMARTS platform.

CommonRoad [89] is a simulator and an open-source toolbox to train and evaluate RL-based motion planners for AVs. Scenario configurations are written in XML files. Users can read, modify, visualize, and store their own traffic scenarios using the Python API provided by CommonRoad. In addition, CommonRoad also supports more scenario specifications, such as Lanelet2 [200] and OpenSCENARIO [196].

VII. Challenges and Potential Solutions

In previous sections, we discussed the safety-critical scenario generation methods from three different views: data-driven, adversarial, and knowledge-based. We notice that generating safety-critical scenarios could be a hard problem due to the many constraints and required properties. In this section, we identify five main challenges that cover the difficulty of the generating process, as shown in Figure 5. Digging into the details of these challenges also helps us discover potential directions that can improve existing algorithms. The five challenges are as the following:

- **Fidelity.** Our ultimate goal is to develop safe AVs that can run in the real world. Therefore, it is useless to make AVs pass difficult but unrealistic scenarios. We need to ensure that generated scenarios have the chance to happen in real traffic situations.

[1] https://www.unrealengine.com
[2] https://www.unity.com
[3] https://www.ros.org

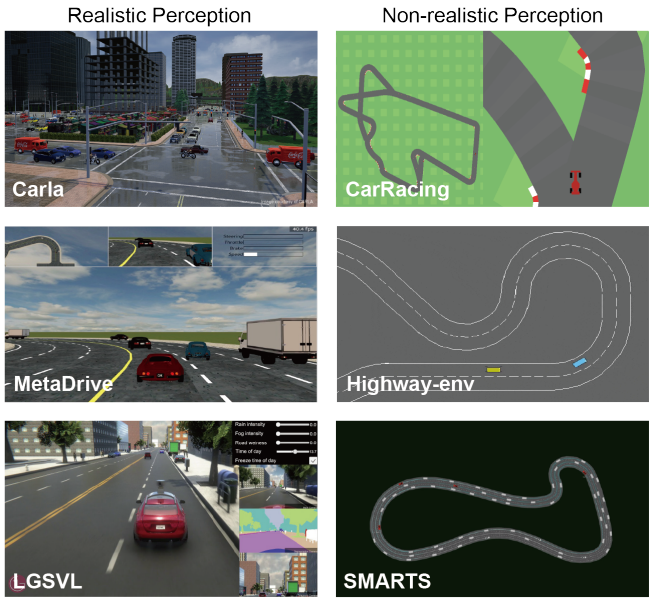


Fig. 4. Screenshots of six typical simulators.

- **Efficiency.** Safety-critical scenarios are extremely rare in the real world. The generation needs to consider the efficiency and increase the density of the scenarios we are interested in.
- **Diversity.** Safety-critical scenarios are also diverse. The generation algorithm should be able to discover and generate as many different safety-critical scenarios as possible.
- **Transferability.** Scenarios are dynamic due to the interaction between the AVs and their surrounding objects. The scenarios we generate should be variable for different AVs rather than targeting one specific AV.
- **Controllability.** In most times, we want to reproduce or repeat specific scenarios rather than random ones. The generative model should be able to follow instructions or conditions to generate corresponding scenarios.

We will discuss more details from the above five perspectives in the following sections, and we will show that the combination of the previous three types of generation methods could be very promising ways to solve those challenges.

A. Fidelity

The generation algorithm can create infinite scenarios, but not all of them are able to happen in the real world. Particularly, under the adversarial generation framework, the searched scenarios are likely to violate the basic traffic rules. The intuitive way to avoid this problem is adding constraints during the generation, but sometimes the constraints are not easy to define. Another promising direction is combining real-world data and adversarial generation, where the real-world data can be used as a prior distribution or constraints. Metrics such as Kullback–Leibler divergence [201] and Wasserstein distance [202] can be used to minimize the gap between the generated and real-world scenarios.

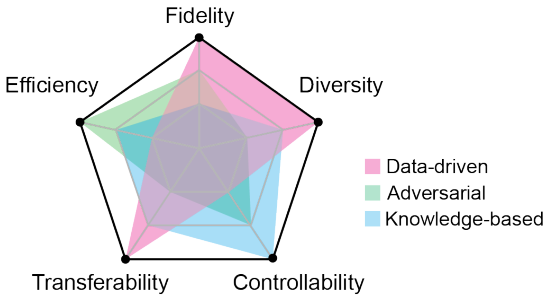


Fig. 5. Comparison between three types of generation under the five challenges.

The fidelity of the scenario is also reflected by the high-dimensional sensing data, which usually requires powerful DGMs to generate. The state-of-the-art methods mainly focus on the generation of static images such as faces. Recently, Neural Radiance Fields (NeRF) [203] is popular for visual scene generation. This method uses NNs to learn the ray-casting functions and then output different views of a scene. Extending this method to large-scale traffic scenario generation is also an interesting direction, which has been explored in Block-NeRF [204]. Block-NeRF builds a large-scale traffic scene from pure image data.

B. Efficiency

Due to the black-box property of most autonomous systems, it is inefficient to generate adversarial examples without accessing the inner information of the systems. In the adversarial attack area, methods with surrogate models [205] or gradient estimation methods [206] are utilized to tackle this problem. They either learn a differentiable surrogate model to imitate the original autonomous systems or query the system to estimate the approximate direction gradient.

It is also noticed that uniform sampling from collected data is quite inefficient because of the rareness of safety-critical scenarios. Therefore, previous methods propose to use IS, which focuses on the region of the distribution that we are interested in. However, it is difficult to extend IS methods to high-dimensional cases. In addition, even for the adversarial generation methods, the black-box property of victim AVs is still the biggest obstacle. Without access to the internal information of failure, the generation algorithms are unable to update their scenarios efficiently.

one potential solution to this problem is leveraging the symbolic reasoning [207] and causal discovery [208] techniques. Instead of performing the optimization only in large numerical space, using symbolic representation helps the generative model to reason about the elements that make the scenario safety-critical. Causal discovery methods can uncover the underlying causality behind safety-critical scenarios, finding the mechanisms that cause the risk.

C. Diversity

Safety-critical scenarios are rare but also diverse. Most of the current generative methods focus on searching for the best scenario that satisfies the requirements but ignores diversity.

To comprehensively evaluate the performance of AV, we need a large range of scenarios. It is easy to fall into the risk of over-fitting if the testing scenarios are very similar. To increase the diversity of the scenario, there are generally two directions. One is from the optimization perspective, where sampled-based methods, such as the evolutionary method or Bayesian Optimization can be used to get feasible solutions from multiple modes. The other direction is applying regularization to the density estimation model or building multi-modal distribution (e.g., Gaussian Mixture Model) to represent the scenario.

D. Transferability

In the safety and robustness areas, the adversarial attack is considered a common way to generate risky examples. The transferability, which means the generated samples are also applicable to other algorithms, is a crucial factor in evaluating the generation method. For example, the pixel-level adversarial attack only works for the target victim. It is believed that the attack should happen at group level [209] or semantic level [210] to achieve better transferability. One example in the safety-critical scenario is controlling one surrounding vehicle (SV) to hit the AV step by step. After the policy of the SV is trained, we change the target AV. The new AV shows a very different behavior and follows a route that the SV has never seen before. It is most likely that this scenario is not risky for the new AV. In this example, we should let the SV learn at a higher level, where it contains the semantic meaning of risk. The SV can suddenly show up behind another vehicle, which leaves the AV a very short time to react. Essentially, we need to build a hierarchical scenario where the high level makes a plan for the risky scenario, and the low level executes that plan with control commands.

E. Controllability

Controllability is useful in two cases. One is that we want to repeat one specific scenario with several parameters fixed, and the other is generating different scenarios with similar settings. For example, we want to test the performance in a highway environment, and then we want to generate vehicles that approach the AV from different directions. Conditional generative models [211], [212] are widely used to generate controllable samples, which learn a joint distribution of the condition and the data. Sometimes, the condition could be simple as numerical values or as complex as natural languages. The challenge is that current NN-based models have poor generalization, therefore, fail when the given unseen conditions during the generating. One promising direction could be to increase the generalizability under such a zero-shot setting.

F. Extension to Other Applications

Autonomous driving is essentially an application of mobile robots. To make intelligent hardware widely used in the real world, other types of applications should also be evaluated on safety-critical scenarios – for example, household robots and manipulation tasks. However, the generation of indoor scenarios could be much more difficult. The traffic scenario basically consists of dynamic objects on a 2D surface, but the indoor scenarios contain a large number of objects interacting in a 3D space. The relations between these objects are also diverse and complex. The direction of how to extend scenario generation methods to these applications is still challenging.

VIII. Conclusion and Discussion

In this survey, we review existing safety-critical scenario generation methods and categorize them into three types. We also review the simulation platforms and datasets that can be used for scenario generation. Most importantly, we identify five challenges for this topic and point out potential directions to tackle these challenges. In the end, we summarize the important message that the readers can take away from this survey, including why is safety-critical scenario generation important, how to select generation algorithms from so many existing methods, and what are future directions to improve existing algorithms.

A. Importance of This Topic

Most existing driving systems are still evaluated on naturalistic scenarios collected from daily life or human-designed scenarios. Even if they can autonomously drive more than thousands of miles without disengagement, we are still not sure about their safety and robustness, e.g., whether the AV can keep safe when the surrounding vehicles behave aggressively or whether the AV can successfully stop when a pedestrian suddenly runs out from behind an object. The generation methods accelerate the development and evaluation of driving systems by creating diverse and realistic safety-critical scenarios.

B. How to Select Generation Algorithms

Although we have categorized existing methods with a taxonomy, it is still not clear how to select a specific algorithm according to different situations. We summarized three combinations to help make choices in general cases.

- **Data-driven generation + Adversarial generation**. If the goal is to broadly evaluate your system under diverse scenarios to discover the weakness, combining multi-modal density models and adversarial training is preferred. Randomly sample from the generative models and search the safety-critical scenarios with adversarial generation.
- **Data-driven generation + Knowledge-based generation**. When there are specific requirements that can be converted into constraints, using constraint optimization to manipulate existing scenarios (e.g., trajectories) is preferred. In that case, the scenarios will concentrate on one single cluster with low diversity, which is helpful if the target is to test the driving system on specific scenarios.
- **Adversarial generation + Knowledge-based generation** If we already have rules that can design safety-critical scenarios but also want to increase the diversity of generated scenarios by automatically learning the parameters, combining the adversarial generation and rule-based method is proffered.

C. What are Future Directions

Properly integrate inductive bias. We want to emphasize that driving scenarios are created by the relations of objects and physical laws rather than being designed by the human mind. Purely relying on NNs or optimization methods is not the ultimate solution for generating realistic and critical scenarios. Instead, the model should leverage as much external knowledge and rules as possible to make the generated scenarios interpretable and satisfy the goal. For instance, the representation of the scenario is crucial and usually determines the quality of the generation. A representation that naturally embeds rules and laws within a structure could be easily optimized by considering the complex relationship between objects. In addition, correctly injecting the distribution of real-world data is important to ensure the reality of generated scenarios. Using Offline RL [213] and imitation learning could be a potential direction to achieve this goal.

Use scenarios to increase robustness and safety. Another aspect that is worth investigating is how to effectively use the generated scenarios to improve robustness and safety. The most intuitive way is training the autonomous system against generated safety-critical scenarios under the adversarial training framework. However, the adversarial scenarios usually represent the worst cases, therefore, learning to a robust yet conservative system. It is not easy to select the difficulty and category of scenarios due to the problems of imbalanced data and over-fitting. These problems bridge the topic discussed in this survey to other areas such as robust optimization [214] and distributional robust optimization [215], which have broad literature to be explored.

Use scenarios to improve generalization. Besides increasing the robustness, the generated scenarios could also be used to improve the generalization of AVs. For instance, gradually training AVs with increasing risk levels under curriculum learning [216] framework may help systems easily generalize to more types of safety-critical scenarios. One recent survey [217] that investigates the generalization problem in RL emphasizes the importance of environment generation in increasing the similarity between training and testing domains. This direction extends the scenario generation from safety to broader views that require goal-conditioned environment generation.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the support from the National Science Foundation (under grants CNS:CAREER-2047454).

REFERENCES

[1] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, "Deep face recognition: A survey," in *2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*. IEEE, 2018, pp. 471–478.

[2] Z. Saquib, N. Salam, R. P. Nair, N. Pandey, and A. Joshi, "A survey on automatic speaker recognition systems," *Signal Processing and Multimedia*, pp. 134–145, 2010.

[3] Z. Zhong, Y. Tang, Y. Zhou, V. d. O. Neves, Y. Liu, and B. Ray, "A survey on scenario-based testing for automated driving systems in high-fidelity simulation," *arXiv preprint arXiv:2112.00964*, 2021.

[4] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, "Survey on scenario-based safety assessment of automated vehicles," *IEEE access*, vol. 8, pp. 87 456–87 477, 2020.

[5] "California Department of Motor Vehicle Disengagement Report," https://www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/disengagement-reports/, 2022, [Online].

[6] E. Thorn, S. C. Kimmel, M. Chaka, B. A. Hamilton *et al.*, "A framework for automated driving system testable cases and scenarios," United States. Department of Transportation. National Highway Traffic Safety Administration, Tech. Rep., 2018.

[7] W. G. Najm, J. D. Smith, M. Yanagisawa *et al.*, "Pre-crash scenario typology for crash avoidance research," United States. National Highway Traffic Safety Administration, Tech. Rep., 2007.

[8] S. C. Schnelle, M. K. Salaani, S. J. Rao, F. S. Barickman, D. Elsasser *et al.*, "Review of simulation frameworks and standards related to driving scenarios," United States. Department of Transportation. National Highway Traffic Safety Administration, Tech. Rep., 2019.

[9] J. M. Scanlon, K. D. Kusano, T. Daniel, C. Alderson, A. Ogle, and T. Victor, "Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain," 2021.

[10] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1625–1634.

[11] W. Ding, B. Chen, M. Xu, and D. Zhao, "Learning to collide: An adaptive safety-critical scenarios generating method," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2243–2250.

[12] G. Harshvardhan, M. K. Gourisaria, M. Pandey, and S. S. Rautaray, "A comprehensive survey and analysis of generative models in machine learning," *Computer Science Review*, vol. 38, p. 100285, 2020.

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[14] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for development, test and validation of automated vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1821–1827.

[15] R. van der Made, M. Tideman, U. Lages, R. Katz, and M. Spencer, "Automated generation of virtual driving scenarios from test drive data," in *24th International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration*, no. 15-0268, 2015.

[16] J. Fang, D. Zhou, F. Yan, T. Zhao, F. Zhang, Y. Ma, L. Wang, and R. Yang, "Augmented lidar simulator for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1931–1938, 2020.

[17] S. Manivasagam, S. Wang, K. Wong, W. Zeng, M. Sazanovich, S. Tan, B. Yang, W.-C. Ma, and R. Urtasun, "Lidarsim: Realistic lidar simulation by leveraging the real world," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 167–11 176.

[18] M. Arief, P. Glynn, and D. Zhao, "An accelerated approach to safely and efficiently test pre-production autonomous vehicles on public streets," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2006–2011.

[19] C. Knies and F. Diermeyer, "Data-driven test scenario generation for cooperative maneuver planning on highways," *Applied Sciences*, vol. 10, no. 22, p. 8154, 2020.

[20] F. Kruber, J. Wurst, and M. Botsch, "An unsupervised random forest clustering technique for automatic traffic scenario categorization," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2811–2818.

[21] F. Kruber, J. Wurst, E. S. Morales, S. Chakraborty, and M. Botsch, "Unsupervised and supervised learning with the random forest algorithm for traffic scenario clustering and classification," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 2463–2470.

[22] W. Wang and D. Zhao, "Extracting traffic primitives directly from naturalistically logged data for self-driving applications," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1223–1229, 2018.

[23] Z. Yang, Y. Chai, D. Anguelov, Y. Zhou, P. Sun, D. Erhan, S. Rafferty, and H. Kretzschmar, "Surfelgan: Synthesizing realistic sensor data for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 118–11 127.

[24] Y. Chen, F. Rong, S. Duggal, S. Wang, X. Yan, S. Manivasagam, S. Xue, E. Yumer, and R. Urtasun, "Geosim: Realistic video simulation via geometry-aware composition for self-driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7230–7240.

[25] S. Ehrhardt, O. Groth, A. Monszpart, M. Engelcke, I. Posner, N. Mitra, and A. Vedaldi, "Relate: Physically plausible multi-object scene syn-

thesis using structured latent spaces," *arXiv preprint arXiv:2007.01272*, 2020.

[26] W. Li, C. Pan, R. Zhang, J. Ren, Y. Ma, J. Fang, F. Yan, Q. Geng, X. Huang, H. Gong *et al.*, "Aads: Augmented autonomous driving simulation using data-driven algorithms," *Science robotics*, vol. 4, no. 28, 2019.

[27] A. Xiao, J. Huang, D. Guan, F. Zhan, and S. Lu, "Synlidar: Learning from synthetic lidar sequential point cloud for semantic segmentation," *arXiv preprint arXiv:2107.05399*, 2021.

[28] A. Kar, A. Prakash, M.-Y. Liu, E. Cameracci, J. Yuan, M. Rusiniak, D. Acuna, A. Torralba, and S. Fidler, "Meta-sim: Learning to generate synthetic datasets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4551–4560.

[29] J. Devaranjan, A. Kar, and S. Fidler, "Meta-sim2: Unsupervised learning of scene structure for synthetic data generation," in *European Conference on Computer Vision*. Springer, 2020, pp. 715–733.

[30] A. Savkin, R. Ellouze, N. Navab, and F. Tombari, "Unsupervised traffic scene generation with synthetic 3d scene graphs," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1229–1235.

[31] W. Ding, W. Wang, and D. Zhao, "A new multi-vehicle trajectory generator to simulate vehicle-to-vehicle encounters," *arXiv preprint arXiv:1809.05680*, 2018.

[32] M. Håkansson and J. Wall, "Driving scenario generation using generative adversarial networks," *Master Thesis*, 2021.

[33] W. Ding, M. Xu, and D. Zhao, "Cmts: A conditional multiple trajectory synthesizer for generating safety-critical driving scenarios," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4314–4321.

[34] Z. Huang, M. Arief, H. Lam, and D. Zhao, "Synthesis of different autonomous vehicles test approaches," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2000–2005.

[35] Y. Guo, V. V. Kalidindi, M. Arief, W. Wang, J. Zhu, H. Peng, and D. Zhao, "Modeling multi-vehicle interaction scenarios using gaussian random field," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3974–3980.

[36] S. Tan, K. Wong, S. Wang, S. Manivasagam, M. Ren, and R. Urtasun, "Scenegen: Learning to generate realistic traffic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 892–901.

[37] M. Wen, J. Park, and K. Cho, "A scenario generation pipeline for autonomous vehicle simulators," *Human-centric Computing and Information Sciences*, vol. 10, pp. 1–15, 2020.

[38] T. A. Wheeler, M. J. Kochenderfer, and P. Robbel, "Initial scene configurations for highway traffic propagation," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 279–284.

[39] T. A. Wheeler and M. J. Kochenderfer, "Factor graph scene distributions for automotive safety analysis," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 1035–1040.

[40] ——, "Critical factor graph situation clusters for accelerated automotive safety validation," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 2133–2139.

[41] B. Wulfe, S. Chintakindi, S.-C. T. Choi, R. Hartong-Redden, A. Kodali, and M. J. Kochenderfer, "Real-time prediction of intermediate-horizon automotive collision risk," *arXiv preprint arXiv:1802.01532*, 2018.

[42] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[43] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, pp. 4565–4573, 2016.

[44] J. Chen, B. Yuan, and M. Tomizuka, "Deep imitation learning for autonomous driving in generic urban scenarios with enhanced safety," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2884–2890.

[45] L. Jain, V. Chandrasekaran, U. Jang, W. Wu, A. Lee, A. Yan, S. Chen, S. Jha, and S. A. Seshia, "Analyzing and improving neural networks by generating semantic counterexamples through differentiable rendering," *arXiv preprint arXiv:1910.00727*, 2019.

[46] J. Tu, M. Ren, S. Manivasagam, M. Liang, B. Yang, R. Du, F. Cheng, and R. Urtasun, "Physically realizable adversarial examples for lidar object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 716–13 725.

[47] M. Abdelfattah, K. Yuan, Z. J. Wang, and R. Ward, "Towards universal physical attacks on cascaded camera-lidar 3d object detection models," *arXiv preprint arXiv:2101.10747*, 2021.

[48] Y. Cao, C. Xiao, D. Yang, J. Fang, R. Yang, M. Liu, and B. Li, "Adversarial objects against lidar-based autonomous driving systems," *arXiv preprint arXiv:1907.05418*, 2019.

[49] A. Prakash, S. Boochoon, M. Brophy, D. Acuna, E. Cameracci, G. State, O. Shapira, and S. Birchfield, "Structured domain randomization: Bridging the reality gap by context-aware synthetic data," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7249–7255.

[50] R. Khirodkar, D. Yoo, and K. M. Kitani, "Vadra: Visual adversarial domain randomization and augmentation," *arXiv preprint arXiv:1812.00491*, 2018.

[51] Y. Zhu, C. Miao, F. Hajiaghajani, M. Huai, L. Su, and C. Qiao, "Adversarial attacks against lidar semantic segmentation in autonomous driving," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 329–342.

[52] N. Ruiz, S. Schulter, and M. Chandraker, "Learning to simulate," *arXiv preprint arXiv:1810.02513*, 2018.

[53] M. O'Kelly, A. Sinha, H. Namkoong, J. Duchi, and R. Tedrake, "Scalable end-to-end autonomous vehicle testing via rare-event simulation," *arXiv preprint arXiv:1811.00145*, 2018.

[54] Y. Abeysirigoonawardena, F. Shkurti, and G. Dudek, "Generating adversarial driving scenarios in high-fidelity simulators," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8271–8277.

[55] W. Ding, B. Chen, B. Li, K. J. Eun, and D. Zhao, "Multimodal safety-critical scenarios generation for decision-making algorithms evaluation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1551–1558, 2021.

[56] Z. Ghodsi, S. K. S. Hari, I. Frosio, T. Tsai, A. Troccoli, S. W. Keckler, S. Garg, and A. Anandkumar, "Generating and characterizing scenarios for safety testing of autonomous vehicles," *arXiv preprint arXiv:2103.07403*, 2021.

[57] J. Wang, A. Pun, J. Tu, S. Manivasagam, A. Sadat, S. Casas, M. Ren, and R. Urtasun, "Advsim: Generating safety-critical scenarios for self-driving vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9909–9918.

[58] A. Nonnengart, M. Klusch, and C. Müller, "Crisgen: Constraint-based generation of critical scenarios for autonomous vehicles," in *International Symposium on Formal Methods*. Springer, 2019, pp. 233–248.

[59] S. Feng, X. Yan, H. Sun, Y. Feng, and H. X. Liu, "Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment," *Nature communications*, vol. 12, no. 1, pp. 1–14, 2021.

[60] H. Sun, S. Feng, X. Yan, and H. X. Liu, "Corner case generation and analysis for safety assessment of autonomous vehicles," *arXiv preprint arXiv:2102.03483*, 2021.

[61] M. Koren, S. Alsaif, R. Lee, and M. J. Kochenderfer, "Adaptive stress testing for autonomous vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1–7.

[62] M. Koren and M. J. Kochenderfer, "Efficient autonomy validation in simulation with adaptive stress testing," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 4178–4183.

[63] ——, "Adaptive stress testing without domain heuristics using go-explore," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–6.

[64] M. Koren, A. Nassar, and M. J. Kochenderfer, "Finding failures in high-fidelity simulation using adaptive stress testing and the backward algorithm," *arXiv preprint arXiv:2107.12940*, 2021.

[65] D. Rempe, J. Philion, L. J. Guibas, S. Fidler, and O. Litany, "Generating useful accident-prone driving scenarios via a learned traffic prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 305–17 315.

[66] M. Arief, Z. Huang, G. K. S. Kumar, Y. Bai, S. He, W. Ding, H. Lam, and D. Zhao, "Deep probabilistic accelerated evaluation: A robust certifiable rare-event simulation methodology for black-box safety-critical systems," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 595–603.

[67] M. Arief, Y. Bai, W. Ding, S. He, Z. Huang, H. Lam, and D. Zhao, "Certifiable deep importance sampling for rare-event simulation of black-box systems," *arXiv preprint arXiv:2111.02204*, 2021.

[68] B. Chen, X. Chen, Q. Wu, and L. Li, "Adversarial evaluation of autonomous vehicles in lane-change scenarios," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[69] A. Wachi, "Failure-scenario maker for rule-based agent using multi-agent adversarial reinforcement learning and its application to autonomous driving," *arXiv preprint arXiv:1903.10654*, 2019.

[70] S. Kuutti, S. Fallah, and R. Bowden, "Training adversarial agents to exploit weaknesses in deep control policies," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 108–114.

[71] R. Lee, M. J. Kochenderfer, O. J. Mengshoel, G. P. Brat, and M. P. Owen, "Adaptive stress testing of airborne collision avoidance systems," in *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*. IEEE, 2015, pp. 6C2–1.

[72] P. Cai, Y. Lee, Y. Luo, and D. Hsu, "Summit: A simulator for urban driving in massive mixed traffic," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4023–4029.

[73] G. Ros, V. Koltun, F. Codevilla, and A. Lopez, "Carla autonomous driving challenge 2019," [EB/OL], https://carlachallenge.org/.

[74] Q. Li, Z. Peng, Z. Xue, Q. Zhang, and B. Zhou, "Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning," *arXiv preprint arXiv:2109.12674*, 2021.

[75] D. J. Fremont, E. Kim, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, "Scenic: A language for scenario specification and data generation," *Machine Learning*, pp. 1–45, 2022.

[76] D. J. Fremont, E. Kim, Y. V. Pant, S. A. Seshia, A. Acharya, X. Bruso, P. Wells, S. Lemke, Q. Lu, and S. Mehta, "Formal scenario-based testing of autonomous vehicles: From simulation to the real world," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–8.

[77] D. McDuff, Y. Song, J. Lee, V. Vineet, S. Vemprala, N. Gyde, H. Salman, S. Ma, K. Sohn, and A. Kapoor, "Causalcity: Complex simulations with agency for causal discovery and reasoning," *arXiv preprint arXiv:2106.13364*, 2021.

[78] G. Bagschik, T. Menzel, and M. Maurer, "Ontology based scene creation for the development of automated vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1813–1820.

[79] A. Rana and A. Malhi, "Building safer autonomous agents by leveraging risky driving behavior knowledge," *arXiv preprint arXiv:2103.10245*, 2021.

[80] M. Zhou, J. Luo, J. Villella, Y. Yang, D. Rusu, J. Miao, W. Zhang, M. Alban, I. Fadakar, Z. Chen *et al.*, "Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving," *arXiv preprint arXiv:2010.09776*, 2020.

[81] W. Ding, B. Li, K. J. Eun, and D. Zhao, "Semantically controllable scene generation with guidance of explicit knowledge," *arXiv preprint arXiv:2106.04066*, 2021.

[82] W. Ding, H. Lin, B. Li, and D. Zhao, "Causalaf: Causal autoregressive flow for goal-directed safety-critical scenes generation," *arXiv preprint arXiv:2110.13939*, 2021.

[83] Q. Zhang, S. Hu, J. Sun, Q. A. Chen, and Z. M. Mao, "On adversarial robustness of trajectory prediction for autonomous vehicles," *arXiv preprint arXiv:2201.05057*, 2022.

[84] Y. Cao, D. Xu, X. Weng, Z. Mao, A. Anandkumar, C. Xiao, and M. Pavone, "Robust trajectory prediction against adversarial attacks," *arXiv preprint arXiv:2208.00094*, 2022.

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

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

[87] E. Rocklage, H. Kraft, A. Karatas, and J. Seewig, "Automated scenario generation for regression testing of autonomous vehicles," in *2017 ieee 20th international conference on intelligent transportation systems (itsc)*. IEEE, 2017, pp. 476–483.

[88] M. Klischat, E. I. Liu, F. Holtke, and M. Althoff, "Scenario factory: Creating safety-critical traffic scenarios for automated vehicles," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–7.

[89] X. Wang, H. Krasowski, and M. Althoff, "Commonroad-rl: a configurable reinforcement learning environment for motion planning of autonomous vehicles," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 466–472.

[90] S. Shiroshita, S. Maruyama, D. Nishiyama, M. Y. Castro, K. Hamzaoui, G. Rosman, J. DeCastro, K.-H. Lee, and A. Gaidon, "Behaviorally diverse traffic simulation via reinforcement learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2103–2110.

[91] O. E. L. Team, A. Stooke, A. Mahajan, C. Barros, C. Deck, J. Bauer, J. Sygnowski, M. Trebacz, M. Jaderberg, M. Mathieu *et al.*, "Open-ended learning leads to generally capable agents," *arXiv preprint arXiv:2107.12808*, 2021.

[92] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8110–8119.

[93] H. Kato, Y. Ushiku, and T. Harada, "Neural 3d mesh renderer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3907–3916.

[94] S. Liu, T. Li, W. Chen, and H. Li, "Soft rasterizer: A differentiable renderer for image-based 3d reasoning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7708–7717.

[95] J. C. Hayward, "Near miss determination through use of a scale of danger," 1972.

[96] M. M. Minderhoud and P. H. Bovy, "Extended time-to-collision measures for road traffic safety assessment," *Accident Analysis & Prevention*, vol. 33, no. 1, pp. 89–97, 2001.

[97] K. Ozbay, H. Yang, B. Bartin, and S. Mudigonda, "Derivation and validation of new simulation-based surrogate safety measure," *Transportation research record*, vol. 2083, no. 1, pp. 105–113, 2008.

[98] B. L. Allen, B. T. Shin, and P. J. Cooper, "Analysis of traffic conflicts and collisions," Tech. Rep., 1978.

[99] S. Almqvist, C. Hyden, and R. Risser, "Use of speed limiters in cars for increased safety and a better environment," *Transportation Research Record*, no. 1318, 1991.

[100] S. S. Mahmud, L. Ferreira, M. S. Hoque, and A. Tavassoli, "Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs," *IATSS research*, vol. 41, no. 4, pp. 153–163, 2017.

[101] Z. Huang, M. Arief, H. Lam, and D. Zhao, "Evaluation uncertainty in data-driven self-driving testing," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 1902–1907.

[102] S. G. Henderson, "Input model uncertainty: Why do we care and what should we do about it?" in *Winter Simulation Conference*, vol. 1, 2003, pp. 90–100.

[103] R. R. Barton, B. L. Nelson, and W. Xie, "A framework for input uncertainty analysis," in *Proceedings of the 2010 Winter Simulation Conference*. IEEE, 2010, pp. 1189–1198.

[104] S. Shafaei, S. Kugele, M. H. Osman, and A. Knoll, "Uncertainty in machine learning: A safety perspective on autonomous driving," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2018, pp. 458–464.

[105] N. Webb, D. Smith, C. Ludwick, T. Victor, Q. Hommes, F. Favaro, G. Ivanov, and T. Daniel, "Waymo's safety methodologies and safety readiness determinations," *arXiv preprint arXiv:2011.00054*, 2020.

[106] H. Pham and X. Zhang, "Nhpp software reliability and cost models with testing coverage," *European Journal of Operational Research*, vol. 145, no. 2, pp. 443–454, 2003.

[107] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "An hdp-hmm for systems with state persistence," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 312–319.

[108] Z. Ghahramani, "Learning dynamic bayesian networks," *International School on Neural Networks, Initiated by IIASS and EMFCSC*, pp. 168–197, 1997.

[109] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[110] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[111] S. Suo, S. Regalado, S. Casas, and R. Urtasun, "Trafficsim: Learning to simulate realistic multi-agent behaviors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 400–10 409.

[112] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[113] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[114] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *International Conference on Machine Learning*. PMLR, 2016, pp. 1747–1756.

[115] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[116] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.

[117] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," *arXiv preprint arXiv:1806.07366*, 2018.

[118] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *arXiv preprint arXiv:2006.11239*, 2020.

[119] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[120] L. Feng, Q. Li, Z. Peng, S. Tan, and B. Zhou, "Trafficgen: Learning to generate diverse and realistic traffic scenarios," *arXiv preprint arXiv:2210.06609*, 2022.

[121] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[122] D. Fernández Llorca and E. Gómez, "Trustworthy autonomous vehicles," Joint Research Centre (Seville site), Tech. Rep., 2021.

[123] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[124] N. Hanselmann, K. Renz, K. Chitta, A. Bhattacharyya, and A. Geiger, "King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients," *arXiv preprint arXiv:2204.13683*, 2022.

[125] P. I. Frazier, "A tutorial on bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.

[126] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3, pp. 229–256, 1992.

[127] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[128] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1928–1937.

[129] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[130] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *arXiv preprint arXiv:1706.02275*, 2017.

[131] D. Zhao, H. Peng, H. Lam, S. Bao, K. Nobukawa, D. J. LeBlanc, and C. S. Pan, "Accelerated evaluation of automated vehicles in lane change scenarios," in *ASME 2015 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2015, pp. V001T17A002—-V001T17A002.

[132] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa, and C. S. Pan, "Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques," *IEEE Transactions on Intelligent Transportation Systems*.

[133] D. Zhao, X. Huang, H. Peng, H. Lam, and D. J. LeBlanc, "Accelerated evaluation of automated vehicles in car-following maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 733–744, 2018.

[134] Z. Huang, D. Zhao, H. Lam, D. J. LeBlanc, and H. Peng, "Evaluation of automated vehicles in the frontal cut-in scenario — an enhanced approach using piecewise mixture models," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 197–202.

[135] Z. Huang, Y. Guo, M. Arief, H. Lam, and D. Zhao, "A versatile approach to evaluating and testing automated vehicles based on kernel methods," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 4796–4802.

[136] Z. Huang, H. Lam, and D. Zhao, "Rare-event simulation without structural information: a learning-based approach," in *2018 Winter Simulation Conference (WSC)*. IEEE, 2018, pp. 1826–1837.

[137] M. Arief, Z. Huang, G. K. S. Kumar, Y. Bai, S. He, W. Ding, H. Lam, and D. Zhao, "Deep probabilistic accelerated evaluation: A certifiable rare-event simulation methodology for black-box autonomy," *arXiv preprint arXiv:2006.15722*, 2020.

[138] M. Xu, P. Huang, F. Li, J. Zhu, X. Qi, K. Oguchi, Z. Huang, H. Lam, and D. Zhao, "Accelerated policy evaluation: Learning adversarial environments with adaptive importance sampling," *arXiv preprint arXiv:2106.10566*, 2021.

[139] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[140] A. v. Lamsweerde, "Formal specification: a roadmap," in *Proceedings of the Conference on the Future of Software Engineering*, 2000, pp. 147–159.

[141] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8748–8757.

[142] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[143] S. R. Contributors, "Carla Scenario Runner," https://github.com/carla-simulator/scenario_runner, 2019.

[144] D. drive Contributors, "DI-drive: OpenDILab decision intelligence platform for autonomous driving simulation," https://github.com/opendilab/DI-drive, 2021.

[145] R. M. Smullyan, *First-order logic*. Courier Corporation, 1995.

[146] W. Ding, H. Lin, B. Li, and D. Zhao, "Generalizing goal-conditioned reinforcement learning with variational causal reasoning," *arXiv preprint arXiv:2207.09081*, 2022.

[147] J. Bennett, *OpenStreetMap*. Packt Publishing Ltd, 2010.

[148] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[149] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[150] X. Li, F. Flohr, Y. Yang, H. Xiong, M. Braun, S. Pan, K. Li, and D. M. Gavrila, "A new benchmark for vision-based cyclist detection," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2016, pp. 1028–1033.

[151] 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*, 2016, pp. 3213–3223.

[152] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3234–3243.

[153] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *European conference on computer vision*. Springer, 2016, pp. 549–565.

[154] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017. [Online]. Available: http://dx.doi.org/10.1177/0278364916679498

[155] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4990–4999.

[156] S. R. Richter, Z. Hayder, and V. Koltun, "Playing for benchmarks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2213–2222.

[157] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636–2645.

[158] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2118–2125.

[159] "Udacity Dataset," https://github.com/udacity/self-driving-car/tree/master/datasets, 2018.

[160] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon, "Kaist multi-spectral day/night data set for autonomous and assisted driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 934–948, 2018.

[161] M.-F. Chang, J. W. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[162] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, "Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8483–8492.

[163] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2702–2719, 2019.

[164] A. Zyner, S. Worrall, and E. M. Nebot, "Acfr five roundabouts dataset: Naturalistic driving at unsignalized intersections," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 4, pp. 8–18, 2019.

[165] A. Patil, S. Malla, H. Gang, and Y.-T. Chen, "The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9552–9557.

[166] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kümmerle, H. Königshof, C. Stiller, A. de La Fortelle, and M. Tomizuka, "INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps," *arXiv:1910.03088 [cs, eess]*, Sep. 2019.

[167] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1929–1934.

[168] R. Krajewski, T. Moers, J. Bock, L. Vater, and L. Eckstein, "The round dataset: A drone dataset of road user trajectories at roundabouts in germany," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–6.

[169] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[170] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, L. Chen, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska, "One thousand and one hours: Self-driving motion prediction dataset," *arXiv preprint arXiv:2006.14480*, 2020.

[171] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2446–2454.

[172] Q.-H. Pham, P. Sevestre, R. S. Pahwa, H. Zhan, C. H. Pang, Y. Chen, A. Mustafa, V. Chandrasekhar, and J. Lin, "A* 3d dataset: Towards autonomous driving in challenging environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2267–2273.

[173] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, "The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Paris, 2020. [Online]. Available: https://arxiv.org/abs/1909.01300

[174] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," 2021.

[175] P. Xiao, Z. Shao, S. Hao, Z. Zhang, X. Chai, J. Jiao, Z. Li, J. Wu, K. Sun, K. Jiang *et al.*, "Pandaset: Advanced sensor suite dataset for autonomous driving," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 3095–3101.

[176] J. Mao, M. Niu, C. Jiang, H. Liang, J. Chen, X. Liang, Y. Li, C. Ye, W. Zhang, Z. Li *et al.*, "One million scenes for autonomous driving: Once dataset," *arXiv preprint arXiv:2106.11037*, 2021.

[177] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9268–9277.

[178] H. B. Lee, H. Lee, D. Na, S. Kim, M. Park, E. Yang, and S. J. Hwang, "Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks," *arXiv preprint arXiv:1905.12917*, 2019.

[179] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner, "Torcs, the open racing car simulator," *Software available at http://torcs. sourceforge. net*, vol. 4, no. 6, p. 2, 2000.

[180] O. Michel, "Webots: Professional mobile robot simulation," *Journal of Advanced Robotics Systems*, vol. 1, no. 1, pp. 39–42, 2004. [Online]. Available: http://www.ars-journal.com/International-Journal-of-Advanced-Robotic-Systems/Volume-1/39-42.pdf

[181] "Openai gym car racing," 2016. [Online]. Available: https://gym.openai.com/envs/CarRacing-v0/

[182] C. L. Azevedo, N. M. Deshmukh, B. Marimuthu, S. Oh, K. Marczuk, H. Soh, K. Basak, T. Toledo, L.-S. Peh, and M. E. Ben-Akiva, "Simmobility short-term: An integrated microscopic mobility simulator," *Transportation Research Record*, vol. 2622, no. 1, pp. 13–23, 2017.

[183] E. Leurent, "An environment for autonomous driving decision-making," https://github.com/eleurent/highway-env, 2018.

[184] "Deepdrive Simulation," https://deepdrive.io/, 2018.

[185] "Environment Simulator Minimalistic (esmini)," https://github.com/esmini/esmini, 2018.

[186] A. Best, S. Narang, L. Pasqualin, D. Barber, and D. Manocha, "Autonovisim: Autonomous vehicle simulation platform with weather, sensing, and traffic control," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1048–1056.

[187] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and service robotics*. Springer, 2018, pp. 621–635.

[188] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2575–2582.

[189] "Apollo Simulation," http://apollo.auto/platform/simulation.html, 2018.

[190] M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem, "Sim4cv: A photo-realistic simulator for computer vision applications," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 902–919, 2018.

[191] W. Schwarting, T. Seyde, I. Gilitschenski, L. Liebenwein, R. Sander, S. Karaman, and D. Rus, "Deep latent competition: Learning to race using visual control policies in latent space," in *Conference on Robot Learning*, 2020.

[192] 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)*. IEEE, 2020, pp. 1–6.

[193] J. Herman, J. Francis, S. Ganju, B. Chen, A. Koul, A. Gupta, A. Skabelkin, I. Zhukov, M. Kumskoy, and E. Nyberg, "Learn-to-race: A multimodal control environment for autonomous racing," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9793–9802.

[194] T. V. Samak, C. V. Samak, and M. Xie, "Autodrive simulator: A simulator for scaled autonomous vehicle research and education," *arXiv preprint arXiv:2103.10030*, 2021.

[195] M. Goslin and M. R. Mine, "The panda3d graphics engine," *Computer*, vol. 37, no. 10, pp. 112–114, 2004.

[196] J.-M. Jullien, C. Martel, L. Vignollet, and M. Wentland, "Openscenario: a flexible integrated environment to develop educational activities based on pedagogical scenarios," in *2009 Ninth IEEE International Conference on Advanced Learning Technologies*. IEEE, 2009, pp. 509–513.

[197] C. Xu, W. Ding, W. Lyu, Z. Liu, S. Wang, Y. He, H. Hu, D. Zhao, and B. Li, "Safebench: A benchmarking platform for safety evaluation of autonomous vehicles," in *Advances in Neural Information Processing Systems*, 2022.

[198] "SUMO NETEDIT," https://sumo.dlr.de/docs/Netedit/index.html, 2018.

[199] "SMARTS Scenario Studio," https://github.com/huawei-noah/SMARTS/tree/master/smarts/sstudio, 2020.

[200] F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr, "Lanelet2: A high-definition map framework for the future of automated driving," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1672–1679.

[201] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[202] L. V. Kantorovich, "Mathematical methods of organizing and planning production," *Management science*, vol. 6, no. 4, pp. 366–422, 1960.

[203] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European conference on computer vision*. Springer, 2020, pp. 405–421.

[204] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis," *arXiv preprint arXiv:2202.05263*, 2022.

[205] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017, pp. 506–519.

[206] C. Guo, J. Gardner, Y. You, A. G. Wilson, and K. Weinberger, "Simple black-box adversarial attacks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2484–2493.

[207] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," *arXiv preprint arXiv:1904.12584*, 2019.
[208] P. Spirtes and K. Zhang, "Causal discovery and inference: concepts and recent methodological advances," in *Applied informatics*, vol. 3, no. 1. SpringerOpen, 2016, pp. 1–28.
[209] K. Xu, S. Liu, P. Zhao, P.-Y. Chen, H. Zhang, Q. Fan, D. Erdogmus, Y. Wang, and X. Lin, "Structured adversarial attack: Towards general implementation and better interpretability," *arXiv preprint arXiv:1808.01664*, 2018.
[210] C. Xiao, J.-Y. Zhu, B. Li, W. He, M. Liu, and D. Song, "Spatially transformed adversarial examples," *arXiv preprint arXiv:1801.02612*, 2018.
[211] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.
[212] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, pp. 3483–3491, 2015.
[213] R. F. Prudencio, M. R. Maximo, and E. L. Colombini, "A survey on offline reinforcement learning: Taxonomy, review, and open problems," *arXiv preprint arXiv:2203.01387*, 2022.
[214] H.-G. Beyer and B. Sendhoff, "Robust optimization–a comprehensive survey," *Computer methods in applied mechanics and engineering*, vol. 196, no. 33-34, pp. 3190–3218, 2007.
[215] H. Rahimian and S. Mehrotra, "Distributionally robust optimization: A review," *arXiv preprint arXiv:1908.05659*, 2019.
[216] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe, "Curriculum learning: A survey," *arXiv preprint arXiv:2101.10382*, 2021.
[217] R. Kirk, A. Zhang, E. Grefenstette, and T. Rocktäschel, "A survey of generalisation in deep reinforcement learning," *arXiv preprint arXiv:2111.09794*, 2021.

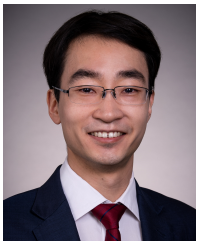
Wenhao Ding is a Ph.D. candidate in Mechanical Engineering Department at Carnegie Mellon University. He is also a master's student in Machine Learning Department at Carnegie Mellon University. He received a B.S. degree in Electronic Engineering from Tsinghua University. His research interests lie in safety-critical scenario generation, robust and generalizable decision-making, and causal reinforcement learning. He is the recipient of the 2022 Qualcomm Innovation Fellowship.

Chejian Xu is a Ph.D. student in Computer Science at the University of Illinois at Urbana-Champaign, Urbana, U.S. He received a B.S. degree in Computer Science from Zhejiang University, Zhejiang, China in 2021. His research interests lie in adversarial machine learning, safety-critical scenario generation, and reinforcement learning.

Mansur Arief is a Ph.D. student in Mechanical Engineering Department at Carnegie Mellon University. He received his bachelor's degree from the Industrial and Systems Engineering Department at ITS and a master's degree from the Industrial and Operations Engineering Department at the University of Michigan. His research focuses on the safety evaluation of intelligent systems and robotics.

Haohong Lin Haohong Lin is a Ph.D. student at Carnegie Mellon University, advised by Professor Ding Zhao. Motivated by the passion for connecting generalizable ML to real-world applications, his research interests focus on controllable content generation, causation-aided deep generative models, and autonomous driving. Prior to joining CMU, he received his bachelor's degree from Zhejiang University.

Bo Li is an assistant professor in the Department of Computer Science at the University of Illinois at Urbana–Champaign. She is the recipient of the MIT Technology Review TR-35 Award, Alfred P. Sloan Research Fellowship, NSF CAREER Award, Dean's Award for Excellence in Research, Intel Rising Star award, Symantec Research Labs Fellowship, Rising Star Award, Research Awards from Tech companies such as Amazon, Facebook, Intel, and IBM, and best paper awards at several top machine learning and security conferences. Her research focuses on both theoretical and practical aspects of trustworthy machine learning, security, machine learning, privacy, and game theory.



Ding Zhao is an assistant professor in the Department of Mechanical Engineering at Carnegie Mellon University. His research focuses on both the theoretical and practical aspects of trustworthy AI with applications on autonomous vehicles, smart manufacturing, intelligent transportation, assistant robots, healthcare diagnosis, and cybersecurity. He is the recipient of the MIT Technology Review TR-35 China Award, National Science Foundation CAREER Award, Carnegie-Bosch Research Award, Ford University Collaboration Award, Struminger Teaching Award, and industrial fellowship awards from Adobe, Bosch, and Toyota.