DiffScene: Diffusion-Based Safety-Critical Scenario Generation for Autonomous Vehicles

Chejian Xu[1], **Aleksandr Petiushko**[2], **Ding Zhao**[3], **Bo Li**[1]
[1]University of Illinois at Urbana-Champaign
[2]Gatik AI
[3]Carnegie Mellon University
chejian2@illinois.edu, alex.petiushko@gatik.ai, dingzhao@cmu.edu, lbo@illinois.edu

Abstract

The field of Autonomous Driving (AD) has witnessed significant progress in recent years. Among the various challenges faced, the safety evaluation of autonomous vehicles (AVs) stands out as a critical concern. Traditional evaluation methods are costly and inefficient, often requiring extensive driving miles in order to encounter rare safety-critical scenarios, which are distributed along the long tail of the complex real-world driving landscape. In this paper, we propose a unified framework, Diffusion-Based Safety-Critical Scenario Generation (DiffScene), to generate high-quality safety-critical scenarios, which are realistic and safety-critical for efficient AV evaluation. In particular, we propose a diffusion-based generation framework, leveraging its power of approximating the distribution of low-density spaces. We design several adversarial optimization objectives to guide the diffusion generation under predefined adversarial budgets. These objectives, such as *safety-based objective*, *functionality-based objective*, and *constraint-based objective*, ensure the generation of safety-critical scenarios while adhering to specific traffic constraints. Extensive experimentation has been conducted to validate the efficacy of our approach. Compared with 6 SOTA baselines, DiffScene generates scenarios that are (1) more safety-critical under different metrics, (2) more realistic under 5 distance functions, and (3) more transferable to different AV algorithms. In addition, we demonstrate that training AV algorithms with scenarios generated by DiffScene leads to significantly higher performance under safety-critical metrics. These findings highlight the potential of DiffScene in addressing the challenges of AV safety evaluation and enhancement, paving the way for safer AV development.

1 Introduction

Innovations driven by recent progress in machine learning (ML) have demonstrated human-competitive performance in various fields (Silver et al. 2018; He et al. 2015; Agostinelli et al. 2019). However, the safety evaluation of these ML models is still challenging, especially in real-world safety-critical applications such as Autonomous Driving (AD).

To evaluate the safety and robustness of AV systems, the prevailing approaches deploy them in the real world and test them with various traffic scenarios. AV companies also reconstruct safety-critical scenarios collected during their on-road testing in the simulators (Webb et al. 2020) to test. Deviation theories such as importance sampling and cross-entropy have been introduced to measure the risk of AVs (Zhao 2016; O'Kelly et al. 2018; Bucklew and Bucklew 2004). However, due to the high dimensionality, complexity, and rareness of safety-critical driving scenarios in the real world, it is very challenging and inefficient to test AV safety (CDMV 2022; Arief et al. 2020).

Recently, with the successes of deep generative models, a promising way is to directly generate such safety-critical scenarios rather than sampling from real-world data (Yang et al. 2020; Chen et al. 2021; Ehrhardt et al. 2020). The advantages include improved evaluation efficiency and scenario diversity (Ding, Xu, and Zhao 2020). For example, RELATE (Ehrhardt et al. 2020) use a GAN framework to generate realistic traffic videos with multi-object scene synthesis. STRIVE (Rempe et al. 2022) generates adversarial trajectory by optimizing the latent space of a VAE model. However, most generation methods focus on only modeling the existing data distribution or applying scenario-specific rules. They fail to generate *controllable* rare events such as safety-critical scenarios *efficiently*.

In this work, to solve these challenges, we propose a diffusion-enabled generation framework DiffScene, which is able to generate safety-critical scenarios effectively while preserving its realism, satisfying real-world physical constraints, and can be used to further evaluate and improve the safety and robustness of various AV algorithms. Specifically, we first leverage the diffusion model to capture the low-density spaces in the distribution to generate realistic safety-critical scenarios efficiently. Then we propose a guided adversarial optimization process to modify the generation results. During each diffusion step, we optimize and constrain the generated scenarios using 3 different objectives: *safety-based objective*, *functionality-based objective*, and *constraint-based objective*. Extensive experiments on different scenario settings and AV algorithms show that DiffScene is able to generate scenarios that are more safety-critical, realistic, and transferable than baselines. We also demonstrate that our scenarios achieve higher downstream utility: training AV algorithms with the generated scenarios leads to significantly higher performance in terms of the safety-critical metrics compared to baselines.

Our contributions are summarized as follows: 1) We propose DiffScene, a unified safety-critical scenario gen-

eration framework that leverages diffusion models to generate realistic safety-critical traffic scenarios by introducing diverse safety-critical objectives. 2) We propose three different safety-critical objectives, focusing on safety, functionality, and (safe) constraints, respectively, to ensure the effectiveness and naturalness of the generated scenarios. 3) We conduct extensive experiments using Carla under different traffic settings (e.g., different routes and maps) with 3 different reinforcement learning-based (RL) AV algorithms. We show that our scenarios achieve higher risk scores (i.e., more safety-critical) in terms of 3 safety-critical metrics and smaller distances to benign data distributions (i.e., more realistic) in terms of 5 distance functions compared to existing safety-critical scenario generation algorithms. 4) We also provide comprehensive evaluations under diverse settings to show that existing RL-based AV algorithms are vulnerable to `DiffScene` scenarios. AV algorithms trained with `DiffScene` scenarios achieve significantly higher performance in terms of the safety-critical metrics, demonstrating the potential utilities of `DiffScene`.

2 Related Work

Deep generative models. Various generative models have been proposed to advance ML development. VAEs (Kingma and Welling 2013) maximize the variational lower bound of training samples, while GANs (Goodfellow et al. 2020) use a generator-discriminator framework to enhance data quality. Recently, diffusion models (Sohl-Dickstein et al. 2015; Song and Ermon 2019; Ho, Jain, and Abbeel 2020) have achieved state-of-the-art performance on generation tasks by defining a Markov chain that adds Gaussian noise to data and then learns to reverse the process. Improvements like DDPM (Ho, Jain, and Abbeel 2020) improve sample quality, while LDMs (Rombach et al. 2022) reduce training and inference costs by operating in the latent space. However, generating structured data, such as dynamic trajectories, remains challenging, especially when requiring safety-critical objectives. This paper addresses these challenges by designing trajectory representations and leveraging diffusion models to generate realistic, safety-critical scenarios guided by adversarial optimization.

Safety-critical scenario generation. Scenario generation algorithms generally fall into three categories: *data-driven*, *adversary-based*, and *knowledge-based*. Data-driven approaches (Scanlon et al. 2021; Knies and Diermeyer 2020; Ding, Wang, and Zhao 2018; Ding, Xu, and Zhao 2020; Suo et al. 2023) rely on real-world data but struggle with the imbalance between safe and risky scenarios. Adversary-based methods (Ding et al. 2021a; Zhang et al. 2022; Feng et al. 2021; Wang et al. 2021; Cao et al. 2022) focus on exploiting AV weaknesses but often produce less realistic and diverse scenarios. Knowledge-based generation (Zhong et al. 2022; Ding et al. 2021b; Wang, Krasowski, and Althoff 2021; Bagschik, Menzel, and Maurer 2018; Zhong et al. 2023; Tan et al. 2023; Cao et al. 2023) incorporates safety constraints or traffic rules, though formalizing these rules remains difficult. Recent work on diffusion models for traffic scenarios (Zhong et al. 2022, 2023) has largely focused on static, 2D scenarios without dynamic AV interaction. In contrast, our approach generates dynamic, 3D safety-critical scenarios with diverse safety objectives, ensuring naturalness and flexibility. Our method is applicable beyond traffic simulation, as demonstrated by its extension to aircraft scenarios in NASA's GUAM simulator (Cook and Gregory 2021; Simmons, Buning, and Murphy 2021; Acheson, Gregory, and Cook 2021). Further details are provided in App. Section 11.

3 `DiffScene`

3.1 Problem Statement

Formally, we define a traffic scenario as $z \in \mathcal{Z} := \{\mathcal{U}, \mathcal{I}, \mathcal{A}\}$. $\mathcal{U}$ represents the participating agents. $\mathcal{I}$ denotes the initial condition and properties of each agent. $\mathcal{A}$ represents the full sequential actions. Each action sequence $\boldsymbol{a} \in \mathcal{A}$ is defined for certain agent $u \in \mathcal{U}$ as

$$\boldsymbol{a}(u) := [a_0, a_1, \cdots, a_T], \tag{1}$$

where a_t is the action taken at timestep t, and T is the maximum horizon length. Consider a model M maps the initial condition $\mathcal{I}$ to an initial system state $\boldsymbol{s}_0$ and derives the whole sequence of system states based on action sequences $\mathcal{A}$:

$$\boldsymbol{s}_t = M(\boldsymbol{s}_0, \mathcal{A}, t) \tag{2}$$

Similarly, we define the state sequence for each agent as

$$\boldsymbol{s}(u) := [s_0, s_1, \cdots, s_T], \tag{3}$$

where s_t is the state of agent u at timestep t. The trajectory of u consists of its state and action sequences:

$$\boldsymbol{\tau}_u := \{\boldsymbol{s}(u), \boldsymbol{a}(u)\}. \tag{4}$$

In a safety-critical scenario, we follow previous work (Ding et al. 2020; Wang et al. 2021) and consider the participating agents $\mathcal{U} := \{u_{ego}\} \cup \mathcal{U}_{sv}$, where u_{ego} is the ego vehicle controlled by certain AV algorithm f , and $\mathcal{U}_{sv} = \{u^0_{sv}, u^1_{sv}, \cdots\}$ denotes safety-critical surrounding vehicles (SVs) controlled by an adversary. $\mathcal{R}_{adv}(\boldsymbol{\tau}_{sv}, f)$ is an adversarial risk function measuring the risk of the current scenario, e.g., collision rate, where the ego vehicle is controlled by f and the safety-critical SVs takes trajectories $\boldsymbol{\tau}_{sv} = \{\boldsymbol{\tau}^0_{sv}, \boldsymbol{\tau}^1_{sv}, \cdots\}$. $\mathcal{C}(\boldsymbol{\tau}_{sv})$ is a cost function over the SV trajectories evaluating the naturalness (cost) of the safety-critical trajectories. Given the AV algorithm f, the goal of the safety-critical scenario generator is to create safety-critical trajectories $\boldsymbol{\tau}_{sv}$ for the safety-critical SVs such that the risk of the scenario is maximized while the generated safety-critical trajectories maintain a low naturalness cost:

$$\arg\max_{\boldsymbol{\tau}_{sv}} \mathcal{R}_{adv}(\boldsymbol{\tau}_{sv}, f), \text{ s.t. } \mathcal{C}(\boldsymbol{\tau}_{sv}) < c, \tag{5}$$

where c is a threshold for the naturalness cost budget.

Due to the high dimensionality and rareness of the safety-critical scenarios, we consider a diffusion-based, adversarially guided generation framework to sample and optimize realistic safety-critical traffic scenarios. Specifically, we first leverage a goal-agnostic diffusion model trained on large-scale benign driving data to generate realistic benign traffic scenarios with low naturalness cost $\mathcal{C}(\boldsymbol{\tau}_{sv})$. Then we optimize the generated scenario based on different adversarial objectives at each diffusion step to maximize the risk $\mathcal{R}_{adv}(\boldsymbol{\tau}_{sv}, f)$ and maintain low cost. The detailed pipeline of our method is shown in Figure 1.

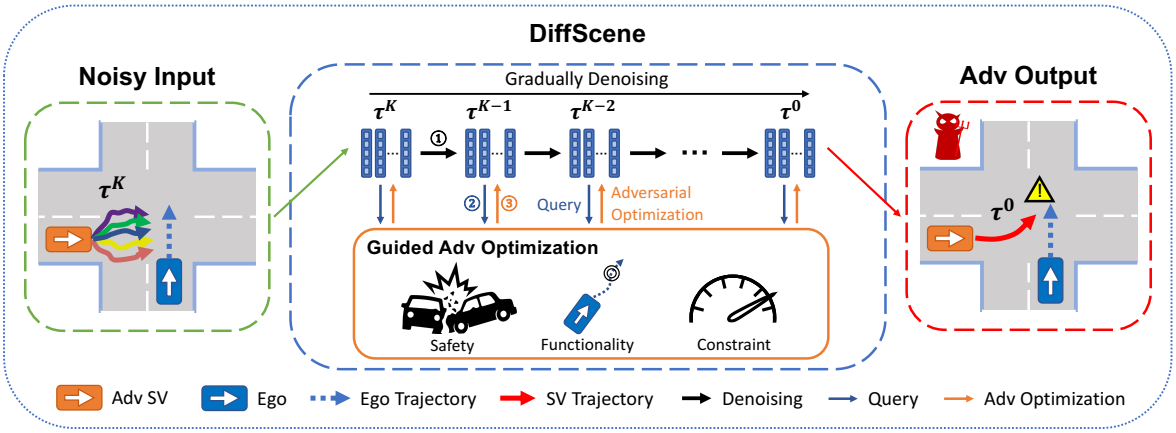

Figure 1: Overview of DiffScene. Given an initial noisy trajectory $\boldsymbol{\tau}^K$, we iteratively perform denoising steps and adversarial optimization steps to obtain the final adversarial SV trajectory $\boldsymbol{\tau}^0$. In each iteration, we first perform a denoising step to calculate the denoised trajectory following Equation (7). Then we perform multiple adversarial optimization steps using different adversarial objectives. The final output maximizes the risk of the generated safety-critical scenarios and maintains a low naturalness cost.

3.2 Diffusion-based Scenario Generation

Diffusion models (Ho, Jain, and Abbeel 2020; Nichol and Dhariwal 2021) approximate the data distribution by a Markov chain starting from a Gaussian distribution. The model learns to reverse a forward diffusion process and generate data by incrementally denoising the sequence from Gaussian noise. We leverage the reverse diffusion process to generate traffic scenarios with high naturalness, since the model is trained to approximate *natural* traffic distributions.

Trajectory representation A trajectory $\boldsymbol{\tau}$ is composed of a state sequence $\boldsymbol{s}$ and an action sequence $\boldsymbol{a}$. We formulate each trajectory as a matrix:

$$\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} s_0 & s_1 & \dots & s_T \\ a_0 & a_1 & \dots & a_T \end{bmatrix}, \tag{6}$$

where each column consists of a state-action pair at a certain timestep along the horizon of the trajectory.

Trajectory generation with diffusion models We first use a diffusion model to generate the benign trajectory $\boldsymbol{\tau}$ for the SV. The generation process is an iterative denoising procedure starting from the initial data distribution $p_\theta(\boldsymbol{\tau}^K) \approx \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, where K is the total number of diffusion steps. Each denoising transition $\boldsymbol{\tau}^k \to \boldsymbol{\tau}^{k-1}$ from step k to step $k-1$ is parameterized by the diffusion model:

$$p_\theta(\boldsymbol{\tau}^{k-1}|\boldsymbol{\tau}^k) = \mathcal{N}(\boldsymbol{\tau}^{k-1}; \mu_\theta(\boldsymbol{\tau}^k, k), \Sigma_\theta(\boldsymbol{\tau}^k, k)), \tag{7}$$

where θ denotes the parameters of the diffusion model. The covariances in the reverse diffusion process are often fixed and depend on the diffusion step: $\boldsymbol{\Sigma}_\theta(\boldsymbol{\tau}^k, k) = \boldsymbol{\Sigma}^k$, where we adopt a cosine schedule following previous work (Nichol and Dhariwal 2021; Janner et al. 2022). The distribution of the final generated clean data (i.e., $k = 0$) is represented as

$$p_\theta(\boldsymbol{\tau}^0) = p_\theta(\boldsymbol{\tau}^K) \prod_{k=1}^{K} p_\theta(\boldsymbol{\tau}^{k-1}|\boldsymbol{\tau}^k). \tag{8}$$

To train the diffusion model, we adopt a forward diffusion process starting from the clean trajectory $\boldsymbol{\tau}^0$. We gradually add Gaussian noise to the original trajectory until step K where $\boldsymbol{\tau}^K$ is approximately Gaussian. The forward diffusion process from step $k-1$ to step k is defined as

$$q(\boldsymbol{\tau}^k|\boldsymbol{\tau}^{k-1}) = \mathcal{N}(\boldsymbol{\tau}^k; \sqrt{1-\beta_k}\boldsymbol{\tau}^{k-1}, \beta_k \boldsymbol{I}) \tag{9}$$

where $\beta_1, \beta_2, \cdots \beta_K$ are fixed noise added to the trajectory data at each forward diffusion step. This forward process q contains no trainable parameters, which allows us to construct noisy trajectories from original data. At each training iteration, we train the diffusion model to approximate and reconstruct the natural clean data $\boldsymbol{\tau}^0$ through the denoising process. We use a simplified objective to train the diffusion model (Ho, Jain, and Abbeel 2020), given by

$$\mathcal{L}(\theta) = \mathbb{E}_{\epsilon, k, \boldsymbol{\tau}^0}[\|\boldsymbol{\tau}^0 - \hat{\boldsymbol{\tau}}\|^2] \tag{10}$$

where ϵ is the noise added to the clean trajectory and $\hat{\boldsymbol{\tau}} = \mu_\theta(\boldsymbol{\tau}^k, k)$ is the reconstructed trajectory.

3.3 Guided Adversarial Optimization

The diffusion model is trained to generate realistic trajectories for SV. To ensure the generated trajectories achieve high risk while maintaining low naturalness cost, we introduce an efficient adversarial optimization process with different optimization objectives. We define an objective function $\mathcal{J}(\boldsymbol{\tau})$ to characterize the risk and the naturalness of a generated trajectory. At each reverse diffusion step k, we modify the denoising process by adding the gradient of $\mathcal{J}$ as guidance:

$$p_\theta(\boldsymbol{\tau}^{k-1}|\boldsymbol{\tau}^k) \approx \mathcal{N}(\boldsymbol{\tau}^{k-1}; \mu + \Sigma g, \Sigma), \tag{11}$$

where $g = \nabla \mathcal{J}(\boldsymbol{\tau})$ specifies the optimization direction. By iteratively optimizing the trajectory towards the desired direction provided by $\mathcal{J}$, the diffusion model will finally generate an SV trajectory satisfying the optimization goals.

This adversarial optimization process enables flexible control over the generated scenarios. We introduce the following three types of objectives: *safety-based objective* $\mathcal{J}_{safe}(\boldsymbol{\tau})$ provides a safety-critical guarantee for the generated scenarios, *functionality-based objective* $\mathcal{J}_{fun}(\boldsymbol{\tau})$ focuses on interfering the regular operations of ego vehicles, and *constraint-based objective* $\mathcal{J}_{con}(\boldsymbol{\tau})$ controls the generated scenarios to satisfy specific rules or constraints. The final safety-critical objective $\mathcal{J}(\boldsymbol{\tau})$ is a combination of the three objectives mentioned above:

$$\mathcal{J}(\boldsymbol{\tau}) = \omega_s \mathcal{J}_{safe}(\boldsymbol{\tau}) + \omega_f \mathcal{J}_{fun}(\boldsymbol{\tau}) + \omega_c \mathcal{J}_{con}(\boldsymbol{\tau}) \quad (12)$$

where ω_s, ω_f, and ω_c are three hyper-parameters controlling the weights of three different objectives.

Safety-based objective targets on the safety of the ego vehicle, which tries to maximize the driving risk of the ego vehicle. Specifically, we define safety-based objective as

$$\mathcal{J}_{safe}(\boldsymbol{\tau}) = -D(\boldsymbol{\tau}) + \lambda \mathbb{1}_{collision}(\boldsymbol{\tau}), \quad (13)$$

where $D(\boldsymbol{\tau})$ represents the minimal distance between the ego vehicle and the safety-critical SV in a scenario where SV follows trajectory $\boldsymbol{\tau}$, $\mathbb{1}_{collision}(\boldsymbol{\tau})$ is an indicator function to represent if the trajectory will cause collision between the ego vehicle and the safety-critical SV, and λ is a hyper-parameter. This safety-based objective encourages the SV to stay close to the ego vehicle so that the probability of collisions will increase.

Functionality-based objective targets on the functional ability of the ego vehicle to finish a given driving task. Specifically, in each testing scenario, the ego vehicle is expected to follow and complete a specific pre-defined route and reach the destination. The functionality-based objective controls a safety-critical SV to prevent the ego vehicle from completing its driving task. For example, the SV can stop the ego vehicle by trying to block the road. We define functionality-based objective as

$$\mathcal{J}_{fun}(\boldsymbol{\tau}) = r(\boldsymbol{\tau}), \quad (14)$$

where $r(\boldsymbol{\tau})$ denotes the percentage of the route not completed by the ego vehicle in a scenario with safety-critical SV following trajectory $\boldsymbol{\tau}$.

Constraint-based objective targets on the desired rules and constraints applied on the safety-critical SV in order to keep it realistic. In a real-world scenario, trajectories must adhere to specific traffic regulations and physical limitations. Depending on the specific requirements, a range of objectives can be incorporated into the framework. For example, considering the comfort perspective, the ego vehicle should avoid jerks and nudges and maintain a low acceleration: $\mathcal{J}_a(\boldsymbol{\tau}) = -\sum_{t=0}^{T} a_t$. From a traffic rule perspective, the ego vehicle should not break traffic rules, avoiding violations like running red lights or ignoring stop signs: $\mathcal{J}_r(\boldsymbol{\tau}) = n(\boldsymbol{\tau})$, where $n(\boldsymbol{\tau})$ counts the number of traffic rule violations during the scenario. In terms of naturalness, the SV should drive at a speed that aligns with typical driving behavior: $\mathcal{J}_v(\boldsymbol{\tau}) = \sum_{t=0}^{T} -|v_t - v^*|$, where v^* is the common driving speed of a vehicle and v_t is the speed of the SV at t. The constraint-based objective offers flexibility by allowing a weighted combination of these various objectives. To illustrate our framework, we have chosen to focus on the naturalness constraint in our experiments, as represented by $\mathcal{J}_{con}(\boldsymbol{\tau}) = \mathcal{J}_v(\boldsymbol{\tau})$. Exploring other combinations is left for future research.

The detailed process of `DiffScene` is shown in Algorithm 1 in Section 9. We first use the benign driving data to train a diffusion model μ_θ approximating the real trajectory distribution and a separate model $\mathcal{J}_\phi$ predicting the safety-critical objective $\mathcal{J}(\boldsymbol{\tau})$. At each reverse diffusion step, the diffusion model first predicts the denoised clean trajectory $\hat{\boldsymbol{\tau}}$ following Equation (7). Then we perform a multi-step optimization using the gradient of safety-critical objective $\mathcal{J}_\phi(\boldsymbol{\tau})$. The multi-step optimization process provides flexible control over the trade-off between the goal of being safety-critical and staying close to the real data distribution. At the end of each denoising step, we calibrate the generated trajectory using the ground truth initial system state calculated by model M. We align the initial state s_0 in the generated SV trajectory with the real initial SV state to make sure every trajectory starts from the same true state. After the initial state calibration, the generated trajectory is then used as the noisy input for the next denoising step until we get the final safety-critical trajectory $\boldsymbol{\tau}_{sv} = \boldsymbol{\tau}^0$. Different from CTG (Zhong et al. 2022) and Diffuser (Janner et al. 2022), Algorithm 1 generates the whole safety-critical trajectory using only one reverse diffusion process. Since the reverse process is time-consuming, our `DiffScene` is much more efficient and enables real-time scenario generation in practice.

4 Experiments

In this section, we conduct comprehensive experiments to evaluate our `DiffScene` in diverse settings. We first apply `DiffScene` to 3 top pre-crash traffic scenarios defined by the National Highway Traffic Safety Administration (NHTSA) (Najm et al. 2007) to generate different safety-critical scenarios. We evaluate `DiffScene` using various metrics measuring the effectiveness and naturalness of the generated scenarios. Then we investigate the downstream utility of our generated scenarios in terms of improving the safety and robustness of AV algorithms after finetuning with them. Finally, we conduct several ablation studies exploring the transferability of the generated scenarios and the safety-critical trade-offs in the adversarial optimization process.

We find that: 1) `DiffScene` is much more effective in terms of generating safety-critical scenarios compared with baselines. Scenarios generated by `DiffScene` achieve higher scores on safety-critical metrics and better performance on constraint satisfaction compared to scenarios generated by existing generation algorithms. 2) `DiffScene` achieves lower naturalness cost. The generated scenarios are more similar to benign scenarios in terms of both trajectory similarity and action similarity. 3) `DiffScene` demonstrates better downstream utility. AV algorithms fine-tuned with our safety-critical scenarios achieve lower risk scores than those fine-tuned on scenarios generated by baselines. 4) The transferability of `DiffScene` is higher than existing scenario generation algorithms. `DiffScene` scenarios are able to cause higher risks across different AV algorithms.

5) There is a trade-off for the generated scenarios in terms of their safety-critical and naturalness properties, balanced by the number of adversarial optimization steps during each denoising step in DiffScene.

4.1 Experimental Design and Setting

Scenario settings We select the 3 most representative scenario settings of pre-crash traffic summarized by NHTSA (Najm et al. 2007), representing the most challenging scenarios in the real world: Crossing Negotiation (S1), Red-light Running (S2), and Red-light Running (S3). We also detail the scenario setting in Section 6. Besides these scenarios considering one SV, we also construct a multi-agent scenario setting and show the details in Section 6.

Simulation platform We use Carla (Dosovitskiy et al. 2017; Xu et al. 2022) as our simulator, which provides realistic simulations of traffic scenarios. We consider 10 different routes in each scenario setting and use 10 different seeds to generate different testing scenarios in each route, obtaining 100 testing scenarios in total for each scenario generation algorithm. To illustrate the adaptability of our framework for deployment in other environments, we have integrated our framework into the GUAM simulator (Cook and Gregory 2021; Simmons, Buning, and Murphy 2021; Acheson, Gregory, and Cook 2021), a specialized platform designed for aircraft simulation. Examples of this implementation are provided in Appendix Section 11.

Baselines We consider six state-of-the-art scenario generation baselines. **Adversarial RL (AR)** uses an RL-based safety vehicle (*SAC*) to generate safety-critical scenarios. **Carla Scenario Generator (CS)** (Dosovitskiy et al. 2017) employs rule-based methods and grid search to find optimal scenarios in three settings. **Learning-to-collide (LC)** (Ding et al. 2020) generates scenarios by sampling from a Bayesian network modeling traffic participant relationships. **AdvSim (AS)** (Wang et al. 2021) uses Bayesian optimization to manipulate the safety vehicle's trajectory, represented by a kinematic bicycle model (Polack et al. 2017), to attack the ego vehicle. **Adversarial Trajectory Optimization (AT)** (Zhang et al. 2022) enhances scenario optimization with knowledge-based constraints, applying PSO-based optimization. **STRIVE (ST)** (Rempe et al. 2022) learns a traffic model and performs adversarial optimization. We adapt it to SafeBench (Xu et al. 2022) with official parameters.

AV algorithms and models To evaluate the effectiveness and transferability of scenario generation algorithms, we test generated scenarios against various AV algorithms, focusing on RL-based methods due to their minimal domain knowledge requirements (Sallab et al. 2017; Chen, Yuan, and Tomizuka 2019; Kiran et al. 2021). Specifically, we control the ego vehicle with three representative deep RL algorithms: *SAC*, *PPO* (Schulman et al. 2017), and *TD3* (Fujimoto, Hoof, and Meger 2018). We train models using these algorithms in benign scenarios and then evaluate them in safety-critical scenarios.

To assess transferability, we also conduct black-box attacks. We train a surrogate SAC model with the same configuration but different initialization. Scenarios generated against the surrogate model are then tested on the three target models. Additional implementation and training details are provided in Section 7.

Data collection To train the diffusion model μ_θ, we first construct a benign trajectory dataset in Carla by training several RL models from scratch in benign scenarios, collecting a total of 6,995 trajectories. Once trained, the diffusion model can generate trajectories across all scenario settings.

For training the safety-critical objective model $\mathcal{J}_\phi$, we generate 5,000 trajectories per scenario setting using the trained diffusion model, calculating $\mathcal{J}(\boldsymbol{\tau})$ as the ground truth. Each scenario setting uses 4,000 trajectories for training and 1,000 for testing. We train three separate $\mathcal{J}_\phi$ models for three different scenario settings.

4.2 Evaluation Metrics

We evaluate the generated scenarios on three levels: safety, functionality, and constraint adherence. Naturalness is assessed by comparing the similarity between generated and benign scenarios. Detailss are provided in Section 8.

Effectiveness A good safety-critical scenario should expose weaknesses in AV algorithms, interfere with normal operation, and satisfy physical constraints. We use three metrics: **Collision Rate (CR)** measures the average collision rate: $\mathbb{E}\boldsymbol{\tau} \sim \mathcal{P}[\mathbb{1}collision(\boldsymbol{\tau})]$. **Incomplete Route (IR)** evaluates the percentage of the route not completed by the ego vehicle: $\mathbb{E}\boldsymbol{\tau} \sim \mathcal{P}[r(\boldsymbol{\tau})]$. **Speed Satisfaction (SS)** measures adherence to normal driving speed: $\mathbb{E}\boldsymbol{\tau} \sim \mathcal{P}[\mathbb{E}t[\mathbb{1}(|vt - v^*| < \delta_v)]]$.

Naturalness Generated scenarios should be realistic and natural. We measure similarity to benign scenarios using five metrics: **Trajectory Similarity** assesses the path similarity using *Symmetric Segment-Path Distance (SSPD)*, *Fréchet Distance (Fréchet)*, and *Dynamic Time Warping (DTW)*. **Action Similarity** evaluates behavioral similarity based on acceleration distribution using *Wasserstein Distance (WD)* and *Kullback–Leibler Divergence (KL)*.

4.3 Effectiveness of DiffScene

We generate safety-critical scenarios in 3 different scenario settings and evaluate them using 3 different AV algorithms based on 3 metrics. For *Speed Satisfaction*, we set the speed threshold $\delta_v = 1$. Quantitative results are shown in Section 4.2, and qualitative comparisons are shown in Section 10.2.

The evaluation results can be analyzed from different perspectives. From the scenario generation algorithm perspective, we observe that DiffScene achieves the best scores among all the methods, demonstrating its advantage of creating more safety-critical scenarios while satisfying rules and constraints. From the scenario setting perspective, *Red-light Running (S2)* is the most safety-critical scenario setting, with the highest collision rate of 87% achieved by DiffScene. The Right-turn (S3) is the safest scenario setting, where DiffScene achieves 79% collision rate. From the collision rate perspective, we notice that DiffScene achieves

Scenario	Metric	Benign	AR	CS	LC	AS	AT	ST	DiffScene
S1	Collision Rate	0.00 ± 0.00	0.19 ± 0.03	0.60 ± 0.14	0.58 ± 0.52	0.57 ± 0.33	0.62 ± 0.49	0.72 ± 0.11	$\mathbf{0.85} \pm 0.08$
	Incomplete Route	0.00 ± 0.00	0.14 ± 0.10	0.27 ± 0.05	0.27 ± 0.24	0.26 ± 0.16	0.28 ± 0.22	0.32 ± 0.04	$\mathbf{0.39} \pm 0.05$
	Speed Satisfaction	-	0.09 ± 0.01	0.26 ± 0.01	0.33 ± 0.02	0.14 ± 0.01	0.30 ± 0.04	0.31 ± 0.05	$\mathbf{0.43} \pm 0.01$
S2	Collision Rate	0.00 ± 0.00	0.38 ± 0.09	0.63 ± 0.15	0.71 ± 0.43	0.57 ± 0.14	0.71 ± 0.50	0.73 ± 0.08	$\mathbf{0.87} \pm 0.10$
	Incomplete Route	0.00 ± 0.00	0.18 ± 0.03	0.29 ± 0.06	0.33 ± 0.20	0.25 ± 0.07	0.33 ± 0.23	0.34 ± 0.04	$\mathbf{0.40} \pm 0.05$
	Speed Satisfaction	-	0.12 ± 0.00	0.26 ± 0.01	0.27 ± 0.02	0.24 ± 0.01	0.30 ± 0.05	0.32 ± 0.06	$\mathbf{0.47} \pm 0.01$
S3	Collision Rate	0.00 ± 0.00	0.34 ± 0.22	0.68 ± 0.16	0.59 ± 0.27	0.29 ± 0.30	0.59 ± 0.50	0.53 ± 0.40	$\mathbf{0.79} \pm 0.15$
	Incomplete Route	0.00 ± 0.00	0.13 ± 0.09	0.22 ± 0.04	0.21 ± 0.10	0.09 ± 0.09	0.19 ± 0.16	0.23 ± 0.18	$\mathbf{0.27} \pm 0.08$
	Speed Satisfaction	-	0.08 ± 0.00	0.19 ± 0.01	0.21 ± 0.01	0.20 ± 0.02	0.34 ± 0.00	0.34 ± 0.01	$\mathbf{0.38} \pm 0.00$

Table 1: **Effectiveness evaluation.** We report *Collision Rate (CR)*, *Incomplete Route (IR)*, and *Speed Satisfaction (SS)* to measure the effectiveness of the generated safety-critical scenarios in terms of safety-level, functionality-level, and constraint-level in 3 different scenario settings. We show the averaged score and standard deviation of the results on 3 different AV algorithms. We also provide the benign performance of the AV algorithms as a reference. (All scores are the higher the better).

over 75% average collision rate in all the 3 scenario settings, showing that existing RL-based AV algorithms are vulnerable to DiffScene scenarios. Finally, from the speed satisfaction perspective, we find that the generated scenarios are hard to achieve higher scores. This is due to the physical constraints of the vehicles: the limited acceleration. It will always take some time to increase the speed from 0 to v^* even with the highest acceleration.

4.4 Naturalness of DiffScene

To evaluate naturalness, we calculate different kinds of similarity scores between the generated scenarios and benign scenarios. Specifically, we first use the surrogate SAC model to control the SV in the 3 different scenario settings and collect the output trajectories from the simulation results as benign trajectories since the SAC model is trained on our normal traffic data and represents the benign driving behavior. Then we calculate the similarities between these benign trajectories and the generated trajectories.

Trajectory similarity Since trajectory similarity metrics are strongly affected by the length of the traveled path, we preprocess the generated trajectories by cutting the end of the paths to so that they are longer than the benign path by a maximum of δ_τ, where δ_τ is a length threshold. To accurately eliminate the effect of length on the similarity results, we set $\delta_\tau = 0.5$ when calculating trajectory similarity. The results are shown in Section 4.4, where we only report the scores for AR, LC, AT, ST, and DiffScene since the paths generated by CS and AS are fixed straight lines.

We note that our method achieves the lowest scores among the baselines, which shows that the DiffScene trajectories are the closest to the benign ones. Among the 3 scenario settings, DiffScene has the lowest similarity score in S2, which again demonstrates that S2 is more safety-critical: easier to achieve high collision rate with a low cost. ST has the highest similarity scores since it has weak restrictions on the trajectory similarity of the generated scenarios to the benign scenarios. We provide additional qualitative results for ST in Figure 7, which shows the large deviation introduced by ST. In the table, we omit the scores greater than 100 caused by ST.

Action similarity Action similarity metrics evaluate the distance between the acceleration distribution of the generated scenarios and the benign ones, which are barely affected by the path length. Therefore, we calculate the action similarity without limiting the length threshold in Section 4.4.

As shown in Section 4.4, the action similarity scores of the scenarios generated by DiffScene are almost the lowest, meaning that the action distribution of the SV is more similar to the benign distribution. Since the KL divergence can be very large when the two distributions are extremely different, we omit the scores greater than 10.

4.5 Downstream Utility of DiffScene

We evaluate the downstream utility of the generated safety-critical scenarios by measuring the safety improvements of AV algorithms after being finetuned on these scenarios. We use the *Crossing Negotiation (S1)* scenario setting as an example. For the scenarios generated by each generation algorithm, we use 80% of them as the training set. The remaining 20% scenarios from all algorithms together form a standard test set. We finetune the target SAC model in the different training sets using 3 different random seeds, each for 500 episodes, and report the averaged testing result on the standard test set. Section 4.4 shows the results of *Collision Rate* and *Incomplete Route* scores of the ego vehicle after fine-tuning. We also show the performance of these fine-tuned ego vehicles in benign scenarios together with the variance in Section 6 in the Appendix. When calculating average *Collision Rate*, we omit the results with *Incomplete Route* scores higher than 0.7 to be more fair. We also show the performance of the target SAC model on the standard testing dataset before finetuning it as a reference.

According to Section 4.4, SAC finetuned on the DiffScene scenarios achieves the lowest *collision rate* and *incomplete route*, which also means that the DiffScene is more useful in terms of improving the robustness of the AV algorithms. Results from Section 6 also show that most adversarial-trained agents achieve 0 collision rate, which means the adversarial-trained agents can achieve better performance while still preserving the benign utility. Among the baselines, LC is the most helpful algorithm in

S.	M.	AR	LC	AT	ST	DiffScene
S1	SSPD	1.07	0.36	0.35	94.78	**0.19**
	Fréchet	6.51	1.45	1.12	>100	**1.04**
	DTW	69.10	57.80	21.16	>100	**12.96**
S2	SSPD	0.54	0.48	0.29	>100	**0.17**
	Fréchet	3.38	1.64	1.11	>100	**1.04**
	DTW	34.74	81.85	18.62	>100	**11.96**
S3	SSPD	0.38	0.33	0.40	>100	**0.25**
	Fréchet	2.80	2.40	2.14	>100	**1.99**
	DTW	30.65	65.55	35.44	>100	**24.58**

(a) Trajectory similarity evaluation

S.	M.	AR	CS	LC	AS	AT	ST	DiffScene
S1	WD	1.74	0.53	0.62	0.47	0.96	0.95	**0.37**
	KL	>10	>10	>10	>10	2.17	1.77	**1.43**
S2	WD	1.78	0.55	0.56	0.59	0.92	0.88	**0.38**
	KL	>10	>10	>10	>10	2.39	2.17	**1.34**
S3	WD	1.24	0.59	0.63	0.59	1.03	1.07	**0.48**
	KL	>10	>10	>10	>10	**0.99**	1.38	1.41

(b) Action similarity evaluation

Table 2: **Naturalness evaluation.** For trajectory similarity evaluation, we report the *SSPD*, *Fréchet*, and *DTW* to measure the similarity between the SV paths in the generated and real collected scenarios. For action similarity evaluation, we report the *WD* and *KL* scores to measure the similarity between the behaviors of the SV in the generated and real collected scenarios. We evaluate the scenarios on 3 different target AD algorithms and report the averaged scores. (S: Scenario Setting, M: Metric. All scores are the lower the better).

terms of reducing the collision rate, while AT is the most helpful algorithm to improve route completion. However, they are still not as effective as DiffScene.

4.6 Ablation Studies

In this section, we show the ablation study results focusing on the transferability of the generated scenarios and the safety-critical trade-offs inside DiffScene.

Transferability In our experiments, we perform a transferability-based black-box attack, where we generate and optimize safety-critical scenarios against a surrogate SAC model and evaluate the generated scenarios using 3 different RL-based AV algorithms. We show the standard deviation of the testing results on 3 different algorithms in Section 4.2. We also show the heatmap of *collision rate* for each AV algorithm achieved by each generation algorithm in 3 different scenario settings in Figure 3 in Section 10.3.

The numbers in Section 4.2 show that in many cases, DiffScene has the lowest standard deviation across 3 different algorithms, meaning that the scenarios generated by DiffScene can be easily transferred to other AV algorithms. Baselines with low standard deviations usually suffer from limited effectiveness, e.g., AR and CS. The detailed results in Figure 3 also verify our conclusions. In the heatmap,

Metric	SAC	AR	CS	LC	AS	AT	ST	DiffScene
CR	0.90	0.82	0.37	0.32	0.75	0.33	0.76	**0.26**
IR	0.36	0.27	0.15	0.32	0.30	0.13	0.29	**0.11**

Table 3: **Downstream utility evaluation.** We report *Collision Rate (CR)* and *Incomplete Route (IR)* after training with generated scenarios to measure the downstream utility of corresponding generation algorithms. We finetune the target SAC model in the generated *S1* scenarios using 3 different random seeds and show the averaged testing results. (All scores are the lower the better).

our DiffScene shows little difference across 3 different AV algorithms. CS, LC, AT, and ST have low collision rate under PPO, while AS has low collision rate under TD3. In practice, safety-critical scenarios with higher transferability can be used to detect vulnerabilities of other AV algorithms and help to improve their robustness, which is more useful in real-world applications.

Impact of the number of adversarial optimization steps To explore the effects and trade-offs of the adversarial optimization steps during the diffusion process, we generate the safety-critical scenarios with different numbers of adversarial optimization steps N, and evaluate the *collision rate* and trajectory similarities of the generated scenarios. We start from $N = 0$, where we do not apply any adversarial optimization, and gradually increase to 30. The figures are shown in Figures 4 and 5 in Section 10.4.

We find that as N increases, the collision rate will also increase, meaning that the adversarial optimization steps do help to generate more safety-critical scenarios. However, when applying a larger N, SSPD and Fréchet will also be larger, showing that more adversarial optimization steps will lead to more naturalness cost. From this result, we can clearly see a trade-off between the effectiveness and naturalness of the generated scenarios. We can easily control and balance them by choosing a proper number of guided adversarial optimization steps in DiffScene.

5 Conclusion

In this paper, we propose DiffScene, a diffusion-based, safety-critical guided generation framework to generate realistic and safety-critical scenarios. Extensive experiments in Carla show that our framework is able to generate safety-critical scenarios against different AV algorithms under various settings. We show that our generated scenarios are more effective, natural, and transferable, and have higher downstream utilities. We also show that current RL-based AV algorithms are vulnerable to the generated safety-critical scenarios. In the meantime, we need to control DiffScene to make sure that the generated safety-critical scenarios are not used for adversarial purposes (see Section 12 for more discussion). We hope this study will shed light on future research on identifying weaknesses in existing AVs, thus facilitating more efficient and effective AV development.

Acknowledgments

This work is partially supported by the National Science Foundation under grant No. 2046726, NSF AI Institute ACTION No. IIS-2229876, DARPA GARD, the National Aeronautics and Space Administration (NASA) under grant No. 80NSSC20M0229, ARL Grant W911NF-23-2-0137, the Alfred P. Sloan Fellowship, the Meta research award, the AI Safety Fund, and the eBay research award.

References

Acheson, M. J.; Gregory, I. M.; and Cook, J. 2021. Examination of unified control incorporating generalized control allocation. In *AIAA Scitech 2021 Forum*, 0999.

Agostinelli, F.; McAleer, S.; Shmakov, A.; and Baldi, P. 2019. Solving the Rubik's cube with deep reinforcement learning and search. *Nature Machine Intelligence*, 1(8): 356–363.

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

Bagschik, G.; Menzel, T.; and Maurer, M. 2018. Ontology based scene creation for the development of automated vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, 1813–1820. IEEE.

Bucklew, J. A.; and Bucklew, J. 2004. *Introduction to rare event simulation*, volume 5. Springer.

Cao, Y.; Ivanovic, B.; Xiao, C.; and Pavone, M. 2023. Reinforcement Learning with Human Feedback for Realistic Traffic Simulation. *arXiv preprint arXiv:2309.00709*.

Cao, Y.; Xiao, C.; Anandkumar, A.; Xu, D.; and Pavone, M. 2022. Advdo: Realistic adversarial attacks for trajectory prediction. In *European Conference on Computer Vision*, 36–52. Springer.

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

Chen, J.; Li, S. E.; and Tomizuka, M. 2021. Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*.

Chen, J.; Yuan, B.; and Tomizuka, M. 2019. Model-free deep reinforcement learning for urban autonomous driving. In *2019 IEEE intelligent transportation systems conference (ITSC)*, 2765–2771. IEEE.

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

Cook, J. W.; and Gregory, I. M. 2021. A Robust Uniform Control Approach for VTOL Aircraft. In *2021 Autonomous VTOL Technical Meeting and Electric VTOL Symposium*.

Ding, W.; Chen, B.; Li, B.; Eun, K. J.; and Zhao, D. 2021a. Multimodal safety-critical scenarios generation for decision-making algorithms evaluation. *IEEE Robotics and Automation Letters*, 6(2): 1551–1558.

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

Ding, W.; Li, B.; Eun, K. J.; and Zhao, D. 2021b. Semantically Controllable Scene Generation with Guidance of Explicit Knowledge. *arXiv preprint arXiv:2106.04066*.

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

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

Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, 1–16.

Ehrhardt, S.; Groth, O.; Monszpart, A.; Engelcke, M.; Posner, I.; Mitra, N.; and Vedaldi, A. 2020. RELATE: Physically plausible multi-object scene synthesis using structured latent spaces. *Advances in Neural Information Processing Systems*, 33: 11202–11213.

Feng, S.; Yan, X.; Sun, H.; Feng, Y.; and Liu, H. X. 2021. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature communications*, 12(1): 1–14.

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, 1587–1596. PMLR.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Janner, M.; Du, Y.; Tenenbaum, J.; and Levine, S. 2022. Planning with Diffusion for Flexible Behavior Synthesis. In *International Conference on Machine Learning*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kiran, B. R.; Sobh, I.; Talpaert, V.; Mannion, P.; Al Sallab, A. A.; Yogamani, S.; and Pérez, P. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*.

Knies, C.; and Diermeyer, F. 2020. Data-Driven Test Scenario Generation for Cooperative Maneuver Planning on Highways. *Applied Sciences*, 10(22): 8154.

Najm, W. G.; Smith, J. D.; Yanagisawa, M.; et al. 2007. Pre-crash scenario typology for crash avoidance research. Technical report, United States. National Highway Traffic Safety Administration.

Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, 8162–8171. PMLR.

O'Kelly, M.; Sinha, A.; Namkoong, H.; Tedrake, R.; and Duchi, J. C. 2018. Scalable end-to-end autonomous vehicle testing via rare-event simulation. *Advances in neural information processing systems*, 31.

Polack, P.; Altché, F.; d'Andréa Novel, B.; and de La Fortelle, A. 2017. The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In *2017 IEEE intelligent vehicles symposium (IV)*, 812–818. IEEE.

Rempe, D.; Philion, J.; Guibas, L. J.; Fidler, S.; and Litany, O. 2022. Generating Useful Accident-Prone Driving Scenarios via a Learned Traffic Prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17305–17315.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695.

Sallab, A. E.; Abdou, M.; Perot, E.; and Yogamani, S. 2017. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19): 70–76.

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

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

Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; Lillicrap, T.; Simonyan, K.; and Hassabis, D. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419): 1140–1144.

Simmons, B. M.; Buning, P. G.; and Murphy, P. C. 2021. Full-envelope aero-propulsive model identification for lift+ cruise aircraft using computational experiments. In *AIAA Aviation 2021 Forum*, 3170.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2256–2265. PMLR.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32.

Suo, S.; Wong, K.; Xu, J.; Tu, J.; Cui, A.; Casas, S.; and Urtasun, R. 2023. MixSim: A Hierarchical Framework for Mixed Reality Traffic Simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9622–9631.

Tan, S.; Ivanovic, B.; Weng, X.; Pavone, M.; and Kraehenbuehl, P. 2023. Language conditioned traffic generation. *arXiv preprint arXiv:2307.07947*.

Wang, J.; Pun, A.; Tu, J.; Manivasagam, S.; Sadat, A.; Casas, S.; Ren, M.; and Urtasun, R. 2021. AdvSim: Generating Safety-Critical Scenarios for Self-Driving Vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9909–9918.

Wang, X.; Krasowski, H.; and Althoff, M. 2021. CommonRoad-RL: a configurable reinforcement learning environment for motion planning of autonomous vehicles. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 466–472. IEEE.

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

Xu, C.; Ding, W.; Lyu, W.; Liu, Z.; Wang, S.; He, Y.; Hu, H.; Zhao, D.; and Li, B. 2022. SafeBench: A Benchmarking Platform for Safety Evaluation of Autonomous Vehicles. In *Advances in Neural Information Processing Systems*.

Yang, Z.; Chai, Y.; Anguelov, D.; Zhou, Y.; Sun, P.; Erhan, D.; Rafferty, S.; and Kretzschmar, H. 2020. Surfelgan: Synthesizing realistic sensor data for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11118–11127.

Zhang, Q.; Hu, S.; Sun, J.; Chen, Q. A.; and Mao, Z. M. 2022. On Adversarial Robustness of Trajectory Prediction for Autonomous Vehicles. *arXiv preprint arXiv:2201.05057*.

Zhao, D. 2016. *Accelerated Evaluation of Automated Vehicles.* Ph.D. thesis.

Zhong, Z.; Rempe, D.; Chen, Y.; Ivanovic, B.; Cao, Y.; Xu, D.; Pavone, M.; and Ray, B. 2023. Language-Guided Traffic Simulation via Scene-Level Diffusion. *arXiv preprint arXiv:2306.06344*.

Zhong, Z.; Rempe, D.; Xu, D.; Chen, Y.; Veer, S.; Che, T.; Ray, B.; and Pavone, M. 2022. Guided Conditional Diffusion for Controllable Traffic Simulation. *arXiv preprint arXiv:2210.17366*.