STD: Generating Realistic and Diverse Safety-Critical Scenarios for Autonomous Vehicles

Anonymous Author(s)

Abstract

Autonomous driving technology has experienced significant advancements over the past decade. Developing autonomous vehicles requires extensive testing with realistic and challenging safety-critical scenarios, such as collisions, to evaluate and improve the performance of driving planners. However, due to the long-tail effect, sparsity, and rare occurrence of such scenarios in real-world datasets, obtaining sufficient data remains challenging. To address this, we propose Scenario Transformer Diffusion (STD), a methodology for the automated generation of diverse and realistic safety-critical scenarios alongside expert trajectories. To ensure realism, we designed a diffusion-based traffic generation model that enhances agent interactions, captures temporal dynamics, and reduces reliance on original trajectories. To reduce the sparsity of safety-critical scenarios, we introduced a safety-critical scenario generation module that incorporates adversarial, stability, and realism factors to guide adversarial agents toward collision scenarios. Additionally, our expert trajectory optimization module identifies solutions within these generated scenarios, providing valuable data for enhancing data-driven planners. Extensive experiments demonstrate that STD advances the state-of-the-art in generating stable, realistic, and diverse safety-critical scenarios. Furthermore, STD not only effectively tests planner performance but also enhances the training data distribution through its expert-generated trajectories, ultimately contributing to improved planner robustness and safety in real-world applications. Anonymous Website Page: Click Here.

CCS Concepts

• **Information systems** → **Spatial-temporal systems**; • **Applied computing** → *Transportation*; • **Computing methodologies** → Multi-agent planning.

Keywords

Autonomous Vehicle, Scenario Generation, Diffusion Model

1 Introduction

In recent years, with the rapid development of autonomous driving technology, data-driven algorithms have been extensively employed in the domains of autonomous driving planning and control [42, 47, 56]. These algorithms, leveraging large amounts of real-world driving data, are capable of learning to handle complex and dynamic traffic scenarios, demonstrating remarkable adaptability. Compared to traditional rule-based algorithms, data-driven models can flexibly respond to diverse driving situations, establishing themselves as the foundational technology in contemporary autonomous driving systems [9].

The safety of autonomous driving is a critical factor in assessing the technological maturity of this field [7, 18]. In real-world scenarios, autonomous driving systems must handle a wide range of complex situations, particularly in sudden and extreme conditions, to ensure vehicle safety. A major challenge in achieving this level of safety is the scarcity of low-frequency, high-risk safety-critical scenarios, which occur roughly once every 6.1 million miles of driving [11, 60]. For training purposes, autonomous driving systems may lack sufficient training data to handle safety-critical scenarios, which can lead to incorrect decisions in these situations and ultimately cause the system to fail. For testing, both data-driven and traditional rule-based autonomous driving planners require evaluation within safety-critical scenarios to verify their robustness [48, 49]. However, this type of data exhibits a "long-tail effect" [35, 63]—while large volumes of data have been collected to cover typical everyday driving situations, safety-critical scenarios remain exceedingly rare. Therefore, obtaining a diverse collection of low-frequency, high-risk safety-critical scenarios is essential.

We believe that an ideal set of safety-critical scenarios should satisfy two key criteria: (1) the scenarios should be comprehensive, realistic, and diverse, as previously mentioned, to effectively overcome the long-tail effect, and (2) they should effectively evaluate the actual performance of planners in low-frequency, high-risk conditions. End-to-end testing is essential, as it provides a direct assessment of the quality and effectiveness of the scenarios, aligning with the practical requirements and expectations of users [5, 45].

Currently, there are two primary methods used to generate these scenarios: sampling from real-world data and scenario generation in simulated environments [15].

The first method involves sampling safety-critical scenarios from real-world driving data. [2, 26–28, 51] ensure high realism in the generated scenarios. However, safety-critical scenarios are extremely rare in everyday driving data and do not cover all potential extreme situations, making this method inefficient and limited.

A natural alternative method is to automatically generate safety-critical scenarios in simulated environments, enabling the efficient generation of diverse scenarios. Although substantial progress has been made in recent studies, there are still three major limitations. These limitations impact the realism and diversity of the generated scenarios and hinder their effectiveness in the actual testing and performance improvement of autonomous driving planners.

L1: Neglecting dynamic interactions in complex traffic environments results in unrealistic and impractical scenario generation. For instance, methods such as [13, 14, 21, 54, 57] optimize pre-selected adversary trajectories to collide with the ego agent in environments with a limited number of agents. However, in complex traffic settings, adversarial agents may collide prematurely with other non-target agents, making it difficult to generate the intended adversarial scenarios.

L2: Dependence on original trajectories limits scenario diversity. For instance, methods like [39, 58] impose adversarial constraints on original trajectories to trigger collisions and generate target scenarios. While these methods ensure realism, they exhibit certain limitations when it comes to generating more diverse safety-critical scenarios within the same environment. This falls short of the

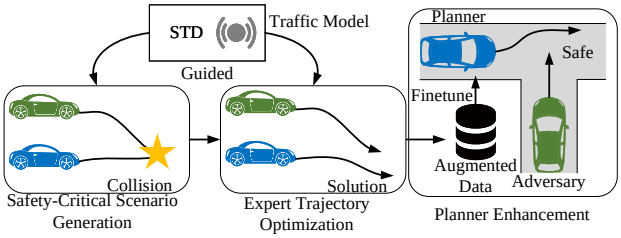


Figure 1: Framework Overview

previously mentioned criteria of generating both realistic and diverse scenarios. Furthermore, these methods require the presence of original trajectories to function effectively.

L3: Lack of practical application of generated scenarios for testing or improving planner performance. For example, [6, 14, 22, 25] do not adequately evaluate the actual impact of these scenarios on planner performance, a factor of primary concern to users. Moreover, these studies do not confirm their effectiveness in improving planner performance, limiting the practical value of these methods for optimizing real-world autonomous driving systems.

To address these three limitations, we propose Scenario Transformer Diffusion (STD), a query-centric multi-agent traffic generation model for safety-critical scenarios. STD integrates adversarial guidance and data-driven generation, improving dynamic interactions, enhancing scenario realism and diversity, and boosting the performance of autonomous driving planners. Below, we detail how STD addresses these three limitations.

M1: Improvements in dynamic interactions within complex traffic environments. We designed the Social Transformer and Decoder components to enhance agent interaction and temporal comprehension. For guidance, our safety-critical scenario generation module includes collision guidance, reducing unintended collisions with non-target agents. Both measures improve the generation of safety-critical scenarios in complex traffic environments.

M2: Overcoming dependence on original trajectories for scenario generation. STD generates future agent trajectories using environmental conditions (initial states, attributes, and maps) without relying on original trajectories. This allows us to generate more diverse safety-critical scenarios, improving coverage of sparse cases.

M3: Practical application of safety-critical scenario generation for enhancing and testing planners. STD not only generates safety-critical scenarios but also provides expert trajectories to handle these scenarios. By fine-tuning planners with these expert trajectories, we effectively enhance planners' performance. Additionally, we test the performance of existing planners in safety-critical scenarios for a more comprehensive evaluation.

Through these measures, STD effectively overcomes existing limitations, achieving notable advancements in generating safety-critical scenarios and optimizing planner performance. In summary, our contributions are as follows:

- We propose Scenario Transformer Diffusion (STD), a query-centric multi-agent traffic generation model designed specifically to generate safety-critical scenarios.
- We propose a safety-critical scenario generation module and an expert trajectory optimization module to enhance the STD's generation process. The first module focuses on generating realistic and feasible safety-critical scenarios, while the second provides expert solutions tailored to these scenarios, effectively integrating scenario generation with adaptive response strategies.
- We leverage these expert solutions to fine-tune autonomous driving planners, significantly enhancing their performance and robustness in handling safety-critical scenarios.
- We evaluate and demonstrate the performance of the generated safety-critical scenarios and their corresponding expert trajectories, conducting a clustering analysis on the types of collisions within these scenarios. Additionally, we experimentally demonstrate the effectiveness of the generated safety-critical scenarios in testing and enhancing the performance of autonomous driving planners.

2 Preliminaries

We consider an interactive traffic environment where the ego vehicle and surrounding vehicles are represented as $\mathcal{V} = \{V_1, \cdots, V_N\}$, driving on a complex multi-lane road. The ego agent can sense the surrounding agents' states (e.g., position, heading, speed) and make decisions at each time step t over the target time horizon T.

Action. The actions of the agents set $\mathcal{V}$ at time t are represented by $m^t = \left[m_1^t, \cdots, m_N^t\right]$, where the action $m = [a, \omega]^T$ includes acceleration a and angular velocity ω.

State. At time step t, the state of the agent set $\mathcal{V}$ is $s^t = \left[s_1^t, \cdots, s_N^t\right]$, where each state $s = [x, y, \theta, v]^T$ includes 2D coordinates, heading, and speed. The next state s^{t+1} is computed from the current state s^t and action m^t using the dynamics model f, as $s^{t+1} = f(s^t, m^t)$.

Attribute. Each agent $\mathcal{V}$ also has attributes $Q = [l, w]$, where l and w represent the length and width, respectively.

Vector Map. The local vector map of the agent set $\mathcal{V}$ at time step t is represented as $\mathcal{I}^t \in \mathbb{R}^{N \times Z \times P \times R}$, where Z is the number of polylines in the map, P is the number of points on each polyline, and R is the number of attributes per point (position and angle).

Decision-making Context. The decision-making context of the agents is represented as $C = \left\{\mathcal{S}^{t-T_h:t}, Q, \mathcal{I}^t\right\}$, including the historical states $\mathcal{S}^{t-T_h:t} = \left\{s^{t-T_h}, \cdots, s^t\right\}$ over the past T_h time steps, their attributes Q, and the local vector map $\mathcal{I}^t$.

Time Step. We discretize the continuous time horizon into time steps, with a total length of $T_h + T$, where T_h is the historical length and T is the length to be generated. The time interval between consecutive steps is Δt.

Trajectory. The generated trajectory $\tau = [\tau_a, \tau_s]^T$ consists of the action sequence $\tau_a = \left[m^0, \cdots, m^{T-1}\right]^T$ and the state sequence $\tau_s = \left[s^1, \cdots, s^T\right]^T$. STD generates only the action trajectory τ_a, and to ensure physical feasibility, τ_s is derived from τ_a: $\tau_s = f(s^0, \tau_a)$.

3 Methodology

3.1 Framework Overview

Figure 1 illustrates the overall framework of the proposed STD, comprising four main components: traffic model, safety-critical scenario generation module, expert trajectory optimization module, and planner enhancement. STD generates safety-critical scenarios along with expert trajectories that address these scenarios, thus providing more challenging training data for autonomous driving planners. Additionally, the generated safety-critical scenarios are

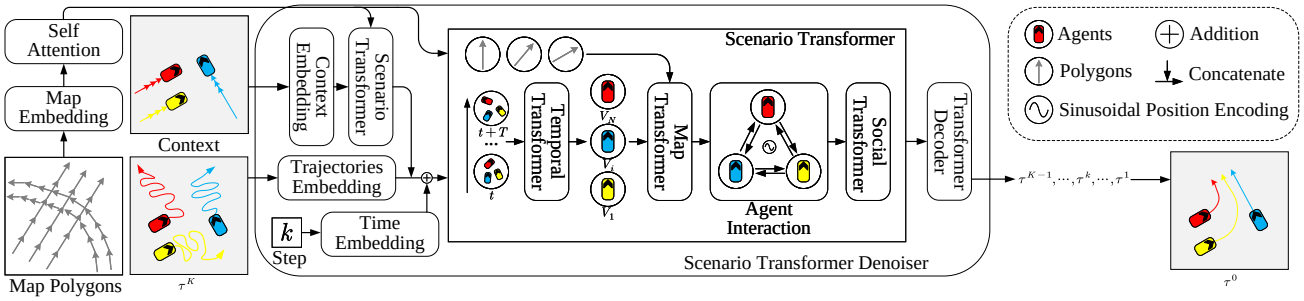


Figure 2: Detailed Architecture of STD Denoiser

used to evaluate the performance of autonomous driving planners under extreme conditions.

Traffic Model is designed to generate stable, realistic, and diverse traffic scenarios. Its inputs consist of trajectory noise and initial conditions, including agents' states, attributes, and map. Based on these conditions, the model incrementally denoises the trajectory noise through diffusion to produce high quality traffic scenarios.

Safety-Critical Scenario Generation Module guides the adversarial agent to minimize its distance from the ego agent at each diffusion denoising step to create collision scenarios. In this module, we designed a hybrid guidance function composed of adversarial, environmental, collision, and initialization guidance, achieving highly effective results in the generated scenarios. Specifically, adversarial guidance increases the likelihood of collisions between the adversarial and ego agents, while other guidance ensure that the generated scenarios are both usable and realistic.

Expert Trajectory Optimization Module assists traffic model in identifying expert solutions, or expert trajectories, within generated safety-critical scenarios. This module also employs a hybrid guidance function, which includes original, environmental, collision, and initialization guidance, to regenerate the ego agent's trajectory as a stable, feasible, and realistic expert trajectory.

Planner Enhancement. We design a workflow to evaluate the real performance of the generated safety-critical scenarios and expert trajectories. Specifically, we use expert trajectories as training data to fine-tune pre-trained planners, thereby effectively enhancing the fine-tune planners' ability to handle safety-critical scenarios.

3.2 Scenario Transformer Diffusion Model

In our approach, trajectories are generated through an iterative denoising process, which is learned by reversing a predefined diffusion process. As stated in Section 2, the trajectory input to the model is $\tau = [\tau^a, \tau^s]^T$. Starting with a clean trajectory sampled from the data distribution, $\tau^0 \sim q(\tau^0)$, the forward noising process introduces Gaussian noise at each step k, generating a sequence of trajectories with progressively increasing noise $(\tau^0, \tau^1, \cdots, \tau^K)$:

$$q(\tau^{1:K}|\tau^0) := \prod_{k=1}^{K} q(\tau^k|\tau^{k-1}) \tag{1}$$

$$q(\tau^k|\tau^{k-1}) := \mathcal{N}(\tau^k; \sqrt{1-\beta_k}\tau^{k-1}, \beta_k I) \tag{2}$$

where β_k is the variance at each step, and with sufficiently large K, we approximate $q(\tau^K) \approx \mathcal{N}(\tau^K; 0, I)$.

STD learns this reverse process, allowing noisy samples to be denoised back into reasonable trajectories. Each step of this reverse process is conditioned on the agent's decision-making context C:

$$p_\theta(\tau^{0:K}|C) := p_\theta(\tau^K) \prod_{k=1}^{K} p_\theta(\tau^{k-1}|\tau^k, C) \tag{3}$$

$$p_\theta(\tau^{k-1}|\tau^k, C) := \mathcal{N}(\tau^{k-1}; \mu_\theta(\tau^k, k, C), \Sigma_\theta(\tau^k, k, C)) \tag{4}$$

where θ are the model parameters. According to [23, 40], the Gaussian transition variance is fixed as $\Sigma_\theta(\tau^k, k, C) = \Sigma_k = \sigma_k^2 I = \beta_k I$.

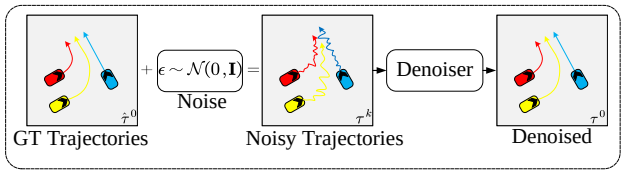
Denoiser Architecture. The architecture of STD is shown in Figure 2. Initially, the agent's decision-making context is embedded and mapped into d dimensions, processed through L_c layers of the Scenario Transformer, resulting in the transformed features $F'_C \in \mathbb{R}^{T_h \times d}$. Similarly, map polygons are embedded into d dimensions, denoted as $F_M \in \mathbb{R}^{L \times d}$, with a multi-head self-attention mechanism extracting features among the polygons, enabling the agent to effectively leverage localized vectorized maps. The output feature of this process is $F'_M \in \mathbb{R}^d$.

The noisy trajectory $F_\tau \in \mathbb{R}^{T \times d}$ is concatenated with the agent's context and further combined with the encoded denoising step features $F_t \in \mathbb{R}^{(T_h+T) \times d}$, yielding the intermediate feature $F_A \in \mathbb{R}^{(T_h+T) \times d}$. This feature is processed through L_{enc} layers of stacked Scenario Transformers, resulting in $F'''_A \in \mathbb{R}^{(T_h+T) \times d}$.

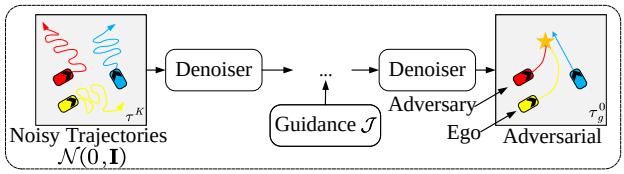
Finally, the denoised trajectory τ^{k-1} is obtained through a standard Transformer decoder and a MLP. The output feature $F_{out} \in \mathbb{R}^{T \times d}$ from the decoder is passed through the MLP to generate the final denoised trajectory (detailed in Appendix A.1).

Scenario Transformer. Figure 2 also shows the structure of the Scenario Transformer, which is composed of three parts: Temporal, Map, and Social Transformer. Our network architecture is entirely Transformer-based, with two key modifications: (1) Unlike [43, 62] that rely on a global coordinate system centered on a focal agent, we use a query-centric approach. This models agent relationships symmetrically, decoupling from any global system and reducing redundant computation, enabling parallel multi-agent traffic generation. (2) Inspired by [30, 44], we use a position encoder to apply sinusoidal encoding to relative agent relationships, ensuring consistent position similarity.

Temporal Transformer analyzes encoded trajectories to effectively capture and understand dynamic interactions and relationships among agents over time. This process provides crucial temporal information for scenario generation, enabling the system to accurately model and respond to changes in complex environments. Specifically, the self-attention mechanism of the Temporal



(a) Training



(b) Guided Sampling

Figure 3: Training and Sampling

Transformer can be represented as:

$$F'_A[t] = \text{MHA}\left(\begin{array}{l} \text{Q} : [F_A[t] + \text{PE}] \\ \text{K, V} : \{[F_A[\iota] + \text{PE}]\}_{\iota \in \Omega(t)} \end{array} \right) \tag{5}$$

where $t \in \{1, \cdots, T_h + T\}$, $\Omega(t)$ is the set of token that need processing at time step t. MHA$(\cdot)$ denotes the multi-head attention mechanism. PE represents the sinusoidal encoding. $F'_A[t] \in \mathbb{R}^d$ is the output of the Temporal Transformer.

Map Transformer employs a cross multi-head attention mechanism, using the central agent's features as queries and local vectorized map features as keys and values. Map Transformer aims to enable agents to capture surrounding map environments to ensure driving within feasible areas. Specifically, the cross-attention mechanism of the Map Transformer can be represented as:

$$F''_A[i] = \text{MHA}\left(\begin{array}{l} \text{Q} : F'_A[i] \\ \text{K, V} : \{[F'_M[j] + \text{PE}]\}_{j \in \Omega(i)} \end{array} \right) \tag{6}$$

where $i \in \{1, \cdots, N\}$, $\Omega(i)$ is the set of token that need processing at agent V_i. $F''_A[i] \in \mathbb{R}^d$ is the output of the Map Transformer.

Social Transformer overcomes the limitations of agent-centric coordinate systems [41] by employing a query-centric self-attention mechanism to capture relative agent relationships. This leverages the symmetry and repeatability of interactions, reducing redundant calculations and enhancing performance [64, 65]. Additionally, a position encoder sinusoidally encodes these relationships [30, 44], ensuring consistent relative position similarity and avoiding unnecessary dimensional expansion.

To explore relative relationships, we transform agents' coordinates into the query agent's system using a transformation matrix:

$$R[i, j] = \left[R^{pos}[i, j], R^{\theta}[i, j], R^{v}[i, j] \right] \tag{7}$$

where $R^{pos}[i, j]$, $R^{\theta}[i, j]$, and $R^{v}[i, j]$ represent the 2D relative position, relative heading angle, and relative velocity between agent V_i and agent V_j, respectively, with details provided in the Appendix A.2.

The cross-attention mechanism of the Social Transformer can be represented as:

$$F'''_A[i] = \text{MHA}\left(\begin{array}{l} \text{Q} : F''_A[i] \\ \text{K, V} : \{[F''_M[j] + \text{PE}(R[i, j])]\}_{j \in \Omega(i)} \end{array} \right) \tag{8}$$

where PE$(\cdot)$ represents sinusoidal encoding using the relative position inside parentheses to ensure consistency in relative position similarity [30]. $F'''_A[i] \in \mathbb{R}^d$ is the output of the Social Transformer.

Training. STD directly predicts the mean μ of τ^0, instead of τ^{k-1}. We uniformly sample the denoising step $k = \mathcal{U}(1, K)$. We directly add the noise loaded at step τ^k to the τ^0 using the formula $\tau^k = \sqrt{\bar{a}_k}\tau^0 + \sqrt{1 - \bar{a}_k}\epsilon$, where $\epsilon \sim \mathcal{N}(0, \text{I})$ and $\bar{a}_k = \prod_{l=0}^{k}(1 - \beta_l)$. We denote the model's direct output as $\hat{\tau}^0 = D_\theta(\tau^k, k, C)$, as shown in Figure 3a. The supervised training loss is given by:

$$\mathcal{L}_{loss} = \mathbb{E}_{\epsilon, k, \tau^0, C} \left\| \tau^0 - \hat{\tau}^0 \right\|^2 \tag{9}$$

3.3 Safety-Critical Scenario Generation

Guided Sampling. Safety-critical scenario generation typically falls into two categories: one involves generating trajectories using traffic priors and then guiding them with a guidance function [39]; the other directly uses trajectories from datasets, guided by a guidance function [13, 14]. STD is distinct in that it incorporates a guidance function into each denoising step, ensuring both the stability and authenticity of the generated safety-critical scenarios. Additionally, through multiple sampling, the inherent randomness of noise allows STD to generate a diverse set of safety-critical scenarios. A schematic of the guided sampling is shown in Figure 3b.

Inspired by [12, 59, 61], we reformulate the reverse process as:

$$p_\theta\left(\tau_g^{k-1} | \tau_g^k, C\right) := \mathcal{N}\left(\tau_g^{k-1}; \mu_\theta + \Sigma_\theta \nabla \mathcal{J}(\mu_\theta), \Sigma_\theta\right) \tag{10}$$

where $\mathcal{J}$ is referred to as the guidance function. τ_g^k denotes the denoised trajectory generated after guided sampling at step k.

Safety-Critical Scenario Generation Module. Our goal is to create situations where an adversarial agent collides with the ego agent. Our criterion for selecting adversaries is defined as:

$$(i^*, t^*) = \arg\min_{i,t} \|P_i^t - P_1^t\| \tag{11}$$

where the ego agent is denoted as V_1. P_i^t represents the global coordinates of agent V_i at time step t, and t^* is the time step when the adversarial agent V_{i^*} is closest.

In the safety-critical scenario generation module, we designed an innovative hybrid guidance function composed of adversarial, environmental, collision and initialization guidance. Thus, our optimization objective is:

$$\min_{\mu} \left(\mathcal{J}_{adv} + \mathcal{J}_{env} + \mathcal{J}_{coll} + \mathcal{J}_{init}\right) \tag{12}$$

where each guidance function has a corresponding weight.

Adversarial Guidance aims to make the selected adversary more likely to collide with the ego agent by minimizing the distance at each time step t in two-dimensional position. Specifically, the adversarial guidance is defined as:

$$\mathcal{J}_{adv} = \sum\nolimits_t \xi^t \|P_{i^*}^t - P_0^t\|^2 \tag{13}$$

where ξ^t is defined as the softmin of the 2D distance between V_{i^*} and V_1 at time step t.

Environmental Guidance penalizes agents for driving in non-drivable areas. It detects this by checking the overlap between

gridded non-drivable map layers and agent boundaries [46]. Specifically, the environmental loss can be represented as:

$$\mathcal{L}_{env}(i) = \begin{cases} 1 - d/r_i & \text{if overlap} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where d is the distance between the collision point c and the agent center, and r_i is half of the diagonal of agent V_i's bounding box.

The environmental guidance is the sum of the environmental losses of all agents under time decay:

$$\mathcal{J}_{env} = \frac{1}{N} \sum_i \sum_t \gamma^t \mathcal{L}_{env}(i) \tag{15}$$

where γ is a decay factor.

Collision Guidance is to ensure that only the adversary and the ego agent collide in the generated safety-critical scenarios, avoiding collisions between other agents. Specifically, we use pairwise collision loss and efficient differentiable relaxation to simplify optimization. We approximate each agent with two circles and calculate the L2 distance between the nearest centers of each pair of agents. It can be represented as:

$$\mathcal{L}_{coll}(i,j,t) = \begin{cases} 1 - \frac{\|P_i^t - P_j^t\|}{r_i + r_j} & \|P_i^t - P_j^t\| \leqslant r_i + r_j \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

where r_i represents the radius of agent V_i's approximating circle.

The collision guidance is the sum of collision losses of all agents under time decay:

$$\mathcal{J}_{coll} = \frac{1}{N^2} \sum_{(i,j), i \neq j \neq 1} \sum_t \gamma^t \mathcal{L}_{coll}(i,j,t) \tag{17}$$

Initialization Guidance aims to minimize the deviation between the guided trajectory and the original trajectory. This setup ensures that the naturalness and rationality of the original trajectory are preserved to the greatest extent during the generation of safety-critical scenario. It is represented as:

$$\mathcal{J}_{init} = \frac{1}{N} \sum_i \sum_t \|P_i^t - P_{i_{init}}^t\|^2 \tag{18}$$

where $P_{i_{init}}^t$ is the global coordinate of agent V_i at the initial state.

3.4 Expert Trajectory Optimization

The expert trajectory optimization module aims to generate optimal expert trajectories for the ego agent in collision scenarios, enhancing its performance. In contrast, the safety-critical scenario generation module modifies adversarial agents' behavior to force collisions with the ego agent. The key distinction is that the former focuses on optimizing the ego agent's trajectory, while the latter targets the adversarial agents' trajectories. Specifically, the expert trajectory optimization guides the STD to generate ego agent trajectories, while other agents follow their original paths from the safety-critical scenarios. This approach ensures that the ego agent's trajectory is optimized for handling such scenarios, improving its ability to manage potential collisions.

In the expert trajectory optimization module, we propose a hybrid guidance function that combines original, collision, environmental and initialization guidance. The optimization objective is:

$$\min_{\mu} \left(\mathcal{J}_{ori} + \mathcal{J}_{ego} + \mathcal{J}_{env} + \mathcal{J}_{init} \right) \tag{19}$$

The definition of $\mathcal{J}_{ego}$ is similar to $\mathcal{J}_{coll}$, but $\mathcal{J}_{ego}$ calculates only the collision loss between the ego agent and other agents, and it carries a greater weight (detailed in Appendix A.3). $\mathcal{J}_{env}$ and $\mathcal{J}_{init}$ are similar to their counterparts in Section 3.3 but are only directed towards the ego agent, while other agents are guided by $\mathcal{J}_{ori}$.

Original Guidance directs non-ego agents to follow the trajectories from the original safety-critical scenarios. It is defined as:

$$\mathcal{J}_{ori} = \frac{1}{N} \sum_{i, i \neq 1} \sum_t \gamma^t \|P_i^t - P_{i_{adv}}^t\|^2 \tag{20}$$

where $P_{i_{adv}}^t$ is the global coordinate of V_i in safety-critical scenario.

3.5 Planner Enhancement

The ultimate goal of generating safety-critical scenarios is to enhance the planner's ability to respond in such scenarios[17]. To achieve this, we employed the following strategies: (1) Using STD to generate a large number of safety-critical scenarios. (2) Generating expert solutions to address these safety-critical scenarios. (3) Utilizing these solutions to fine-tune the original planner, thereby enhancing its safety and robustness in safety-critical scenarios.

To demonstrate the effectiveness of the STD-generated safety-critical scenarios and the performance of the expert solution trajectories, we designed a simple yet effective planner. Specifically, we utilize existing trajectory prediction algorithms (e.g., [19, 29, 53]) to generate future trajectories for the ego agent, and then employ a dynamics model [36] to translate these trajectories into actions to drive the motion of the ego agent. Through this approach, we are able to assess the planner's ability and effectiveness in handling safety-critical scenarios after fine-tuning.

4 Experiments

4.1 Experimental Settings

Datasets. nuScenes [4] is a widely used autonomous driving dataset with 5.5 hours of annotated trajectories in diverse scenes and dense traffic. It contains 1,000 20-second traffic segments annotated at 2 Hz, which we interpolated to 10 Hz.

Simulation Environment. Our simulation environment is initialized using agent positions and historical states from real driving data. The simulation runs at 10 Hz, with a 10-second timestep. To assess scenario diversity and long-term performance, the simulation duration is set to 20 s. All experiments are conducted in a closed-loop environment with a decision frequency of 2 Hz.

Baselines. Previous works focus on different aspects, making direct comparisons challenging. For example, [1, 7] focus on specific scenarios, while [38, 50] attack the entire autonomous driving stack, rather than just the planning module. To comprehensively evaluate our approach, we compared it with state-of-the-art safety-critical scenario generation methods, including CTG++(Adv) [61] and STRIVE [39]. Additionally, to assess diversity and long-term performance, we compared our approach with SimNet [3], TrafficSim [46], and BITS [55].

4.2 Evaluation of Safety-Critical Scenario

Metrics. In the evaluation of safety-critical scenario generation and expert trajectory optimization, we primarily focus on the **effectiveness**, **realism**, and **stability** of the scenario.

Table 1: Evaluation of Safety-Critical Scenario Generation

Method	GCR ↑	RB ↓	AOR ↓	OFR ↓	OCR ↓
STRIVE	43.00	0.088	**0.00**	9.28	8.79
CTG++(Adv)	71.00	0.190	1.31	6.73	5.92
STD	**73.00**	**0.060**	0.01	**5.18**	**4.65**

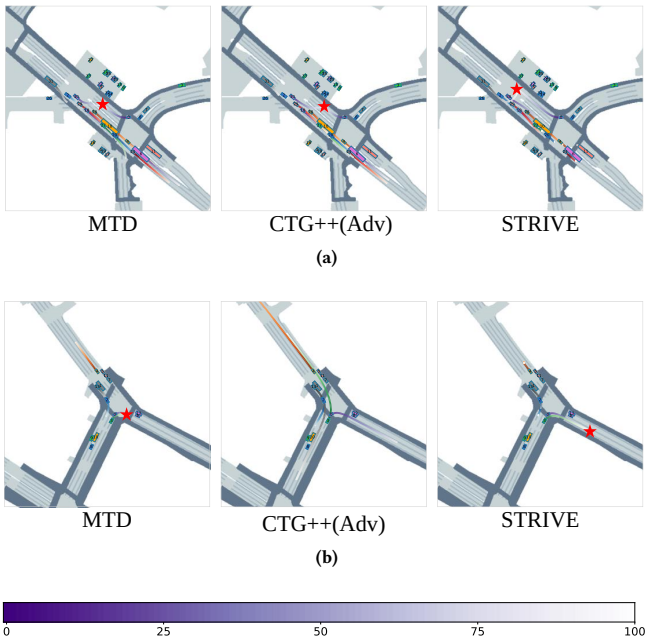


Figure 4: Comparison of Generated Safety-Critical Scenarios

Effectiveness is assessed through the generated collision rate and the expert resolution rate. During the safety-critical scenario generation phase, effectiveness is measured by the generated collision rate, while in the expert trajectory optimization phase, it is evaluated by the expert resolution rate.

• **Generated Collision Rate (GCR)** is determined by calculating the proportion of scenarios where the ego agent collides with the adversary agent, obtained by dividing the number of collision scenarios by the total number of scenarios.

• **Expert Resolution Rate (ERR)** measures the expert's ability to find collision-avoidance trajectories in the generated safety-critical scenarios, calculated by dividing the number of successful expert solutions by the number of collision scenarios.

Realism is assessed by computing the Wasserstein distance between the normalized histograms of driving features from the generated and real-world motions [55].

• **Realism Bias (RB)** is defined as the average Wasserstein distance of the distributions for longitudinal acceleration magnitude, lateral acceleration magnitude, and jerk.

Stability is assessed by reporting the failure rate, collision rate, and off-road rate. A critical failure is defined as an agent either colliding with another agent or driving off the road for more than 1 second.

• **Failure Rate (FR)** is the average proportion of agents experiencing critical failures in the scenario.

• **Collision Rate (CR)** is the average proportion of agents colliding with another agent.

Table 2: Evaluation of Expert Trajectories Generation

Method	ERR ↑	RB ↓	EOR ↓	OFR ↓	OCR ↓
STRIVE	48.84	0.075	**0.65**	11.34	11.65
CTG++(Adv)	59.15	0.234	6.54	15.26	14.08
STD	**75.34**	**0.060**	6.14	**8.29**	**8.17**

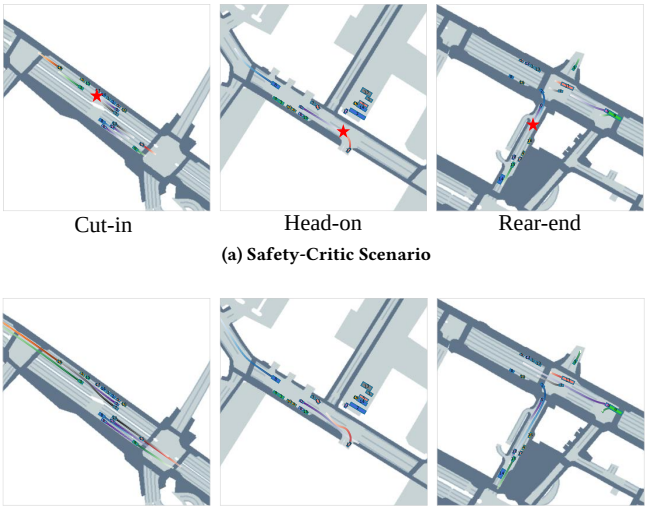


Figure 5: Expert Trajectories in Safety-Critical Scenarios

• **Off-road Rate (OR)** represents the proportion of timesteps an agent spends outside the drivable area, averaged across all agents. Additionally, in metrics such as AOR, EOR, OFR, and OCR, the letters A, O, and E denote adversarial agent, other agent, and ego agent, respectively.

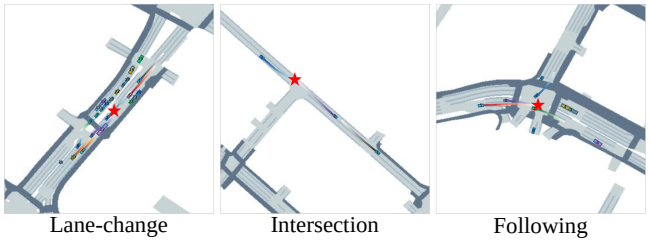
Table 1 presents the quantitative results for safety-critical scenario generation. We evaluated various methods using metrics such as GCR, RB, Adv OR (AOR), Other FR (OFR), and Other CR (OCR). Except for Adv OR, which is only 0.01% lower than STRIVE, STD outperforms all baseline methods across GCR, RB, OFR, and OCR, demonstrating its ability to generate stable and realistic safety-critical scenarios. This superior performance is attributed to STD being a generative model trained on real-world datasets, ensuring that output trajectories conform to real-world distributions.

We demonstrate the advantages of STD through two examples. In Figure 4a, both CTG++(Adv)'s Agent 29 and STRIVE's Agent 30 approach the ego agent, compromising scenario realism. In contrast, STD avoids this issue. In Figure 4b, due to STRIVE and CTG++(Adv) optimizing all potential adversaries, STRIVE takes longer to generate safety-critical scenarios, and CTG++(Adv) even fails to find a reasonable solution. STD successfully avoids these issues. In Figure 4c, the trajectory color indicates the corresponding timesteps. Unless stated otherwise, all trajectories in this paper follow this mapping.

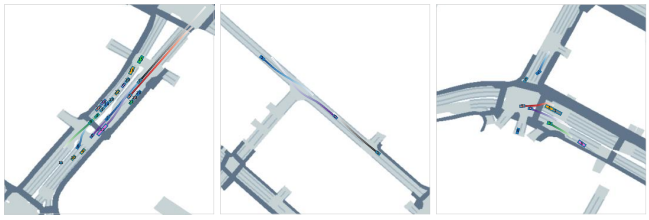
Table 2 presents quantitative metrics for expert solutions. STD outperforms baseline methods in ERR, RB, OFR, and OCR. Its performance on Ego OR (EOR) is better than CTG++(Adv) but slightly below STRIVE. High-quality expert trajectories provide comprehensive training data for planners, mitigating the data sparsity issue in safety-critical scenarios. Based on the statistics from [37], we selected the three most common cases in daily traffic and present

Table 3: Planner Performance in Reg. and Adv. Scenarios

Planner	Scenario	PCR ↓	OR ↓	AV ↑	Jerk ↓
Pre-Trained	Regular	20.00	3.54	4.77	1.05
	Adversarial	44.00	2.34	4.75	1.04
Fine-Tuned	Regular	19.00	9.70	6.04	0.65
	Adversarial	39.00	8.85	6.20	0.62



(a) Pre-Trained Planner



(b) Fine-Tuned Planner

Figure 6: Different Planners in Safety-Critical Scenarios

them in Figure 5. The results demonstrate that STD successfully generates expert solutions for all three safety-critical scenarios.

4.3 Evaluating Generated Scenario with Planner

Metrics. To further illustrate the effectiveness of STD, we evaluated the performance of AutoBot [19] in both adversarial scenarios generated by STD and regular scenarios.

• **Planner's collision rate (PCR)** is defined as the number of collision scenarios divided by the total number of scenarios.

We measured effectiveness by recording the planner's collision rate, as shown in the "Pre-Trained" row of Table 3. In the adversarial scenarios, the Pre-Trained model exhibited a 24.00% increase in PCR, indicating that the scenarios generated by STD effectively test the safety performance of the planner in extreme environments. Existing studies primarily focus on evaluating offline metrics, whereas the actual impact of generated scenarios on planner performance is of primary concern to users. Our experimental results more accurately demonstrate how these scenarios affect planner performance, addressing a critical aspect not thoroughly explored in numerous studies, including [6, 14, 24, 61].

4.4 Evaluation of Planner Enhancement

In this subsection, we analyze the effectiveness of STD's expert trajectories in enhancing the data distribution.

Metrics. We measure the planner's performance by evaluating the PCR, OR, average velocity, and jerk.

• **Average Velocity (AV)** is used to assess the efficiency of the autonomous vehicle [31, 52]. A higher AV indicates better efficiency.

Table 4: Evaluation of Div. and Long-Term Performance

Method	FR ↓	CR ↓	OR ↓	Cov ↑	TD ↑	RB ↓
SimNet	24.58	15.80	3.05	395.2	0.00	0.098
TrafficSim	27.08	20.39	3.42	861.0	4.28	0.119
BITS	14.82	8.62	**0.95**	1078.7	4.66	0.120
CTG++	14.35	9.17	1.45	**1518.9**	**7.13**	**0.069**
STRIVE	15.27	10.55	1.20	937.4	5.97	0.096
STD	**11.21**	**6.16**	1.71	1089.8	6.54	0.086
Dataset	17.09	15.95	0.50	539.3	0.00	0.000

• **Jerk**, defined as the rate of change of acceleration, measures driving comfort due to its significant impact on passenger experience [66]. Higher jerk values indicate greater passenger discomfort. We assess the planner's comfort by calculating the average jerk.

We employed expert trajectories generated in safety-critical scenarios, performed data cleaning, and fine-tuned the planner. The fine-tuning results, presented in the "Fine-Tuned" row of Table 3, show a slight increase in OR compared to the pre-trained planner, likely due to the incorporation of numerous safety-critical trajectories, resulting in a more aggressive driving style. The fine-tuned model achieved lower PCR, higher AV, and reduced jerk in both regular and adversarial scenarios. These outcomes demonstrate that the generated expert trajectories enhance the safety, efficiency, and comfort of autonomous driving planners, thereby validating STD's effectiveness in augmenting the data distribution.

Figure 6 displays the visualization of the pre-trained and fine-tuned planners in safety-critical scenarios. These comparisons demonstrate that the fine-tuned planner significantly improves its ability to handle complex traffic situations. This further validates the effectiveness of STD in augmenting the data distribution.

4.5 Analyzing Safety-Critical Scenario

In this subsection, we analyzed the safety-critical scenarios generated by STD. We first excluded scenarios where STD failed to generate collisions. Inspired by [8], we applied K-Means [34] clustering with $k = 7$ to categorize collision types and manually assigned semantic labels based on [37]. Examples of each collision type are shown in Figure 7. The green bars represent the proportion of each collision type among all successfully generated safety-critical scenarios, while the orange bars indicate the proportion for which an expert solution was found to avoid the collision.

4.6 Diversity and Long-Term Performance

Metrics. **Diversity** is evaluated by coverage and trajectory diversity.

• **Coverage (Cov)** assesses the extent to which agents cover the scene. In our experiments, we adhere to the experimental configuration of [55] to calculate coverage density.

• **Trajectory Diversity (TD)** is measured by the Wasserstein distance between the density distributions of trajectories from different trials. Based on previous studies [61], we define TD as follows:

$$\text{Diversity} = 2/[n\,(n-1)] \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \text{Wass}\left(\rho_i, \rho_j\right) \qquad (21)$$

where ρ_i is the density distribution of the i-th trial.

Table 4 presents the quantitative results of long-term closed-loop simulations for generating regular scenarios. To account for environmental differences, we re-evaluated all baseline methods

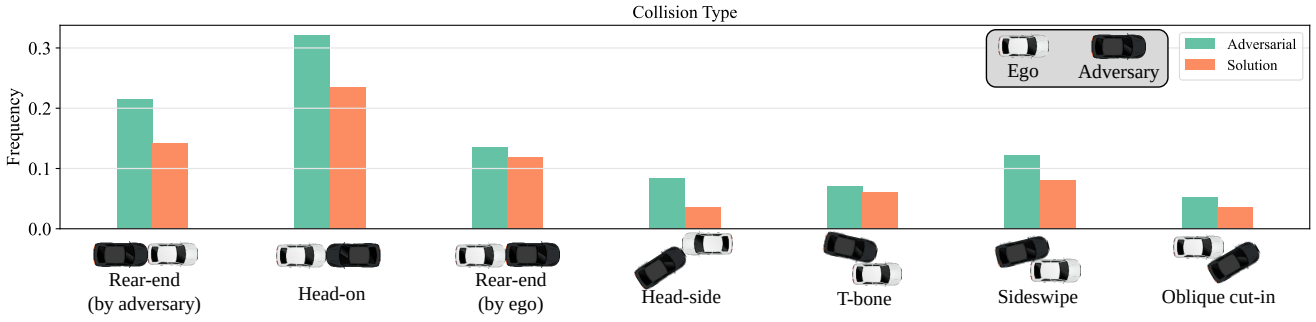


Figure 7: Frequency and Solution Rate of Different Collision Types in Safety-Critical Scenarios

within a consistent setting. Additionally, we report results on a real-world dataset (Dataset), which exhibits noise as indicated by a non-zero failure rate, primarily due to agent collisions from inaccurate bounding box annotations. Notably, STD achieved a lower failure rate and generated more diverse scenarios compared to the real-world dataset. In our comparisons, STD outperformed SimNet and TrafficSim across all metrics. Although CTG++ slightly surpassed STD in diversity and realism, it fell short in terms of stability. BITS showed slight advantages in OR and Cov metrics, while STRIVE performed marginally better in OR but lagged behind STD in other metrics. Importantly, STD achieved the best results across all baseline methods in the FR and CR metrics, highlighting its superior stability, particularly in preventing agent collisions during long-term scenario generation.

4.7 Ablation Study

Impact of Hybrid Guidance. We investigated the contribution of each component in the hybrid guidance function of the safety-critical scenario generation module by evaluating the following variations: Specifically, we considered the following variations:

- STD-w/o-I. Removal of initialization guidance.
- STD-w/o-A. Replacement of adversarial guidance with a variant similar to STRIVE.
- STD-w/o-C. Removal of collision guidance.
- STD-w/o-E. Removal of environmental guidance.

As shown in Table 5, the complete hybrid guidance achieves the best performance across multiple metrics, particularly in maintaining realism and generating effective safety-critical scenarios with a well-balanced performance.

Impact of Components. We further assessed the role of specific components within STD, namely the Social Transformer and Decoder, by evaluating the following variations:

- STD-w/o-D: Removal of the Decoder component.
- STD-w/o-S: Removal of the Social Transformer component.

The results in Table 6 indicate that the Social Transformer contributes to enhancing stability and realism, while the Decoder plays a role in promoting scene diversity. Overall, the complete STD model achieves balanced performance, capable of generating scenarios that are stable, realistic, and diverse.

5 Related Work

Methods for obtaining safety-critical scenarios include real-world data collection and simulation-based generation. The real-world data collection approach samples and extracts safety-critical scenarios from daily driving data. Works like [2, 26–28, 51] use historical analysis and clustering to identify sparse, low-frequency scenarios. However, safety-critical scenarios are rare in daily driving data, making extraction difficult and time-consuming. Moreover, the limited diversity of real-world scenes means sampled data may not cover all extreme cases, leaving gaps in autonomous system training [10]. Another approach is to generate safety-critical scenarios in a simulated environment. Using tools like CARLA [16], manually designing autonomous vehicle paths to generate safety-critical scenarios is time-consuming and labor-intensive. This is especially true for large numbers of complex scenarios, making it difficult to meet the growing demand for diversity [20]. Approaches such as initializing opponents' states [13, 14], adversarial optimization in Diffusion models [54], kinematic gradients [21], and LLM-based generation [57] have limitations. They often overlook dynamic factors in multi-vehicle interactions and rely heavily on original trajectories for diversity. Additionally, they lack end-to-end testing on planners [6, 14, 22, 25].

Table 5: Ablation of Hybrid Guidance

Method	GCR ↓	RB ↓	AOR ↓	OFR ↓	OCR ↓
STD	**73.00**	0.060	**0.01**	**5.18**	**4.65**
STD-w/o-I	72.00	0.080	0.02	5.67	5.20
STD-w/o-A	70.00	0.071	0.04	6.30	5.70
STD-w/o-C	72.00	**0.051**	0.07	7.74	7.27
STD-w/o-E	72.00	0.077	1.82	9.16	5.83

Table 6: Ablation of STD Components

Method	FR ↓	CR ↓	OR ↓	Cov ↑	TD ↑	RB ↓
STD	**11.21**	**6.16**	1.71	1089.8	6.54	**0.086**
STD-w/o-D	12.44	7.90	**1.65**	1021.2	6.04	0.093
STD-w/o-S	14.61	9.64	1.77	**1639.6**	**7.03**	0.103

6 Conclusion

In this work, we propose Scenario Transformer Diffusion (STD), a novel method for generating safety-critical scenarios. STD is developed based on a query-centric multi-agent traffic generation model, capable of generating stable, realistic, and diverse safety-critical scenarios, along with expert trajectories. Extensive experimental results demonstrate that STD surpasses current state-of-the-art methods in terms of stability, realism, and diversity. Additionally, STD not only effectively tests the performance of autonomous driving planners but also enhances the data distribution through its generated expert trajectories.

References

[1] Yasasa Abeysirigoonawardena, Florian Shkurti, and Gregory Dudek. 2019. Generating adversarial driving scenarios in high-fidelity simulators. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 8271–8277.

[2] Mansur Arief, Peter Glynn, and Ding Zhao. 2018. 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, 2006–2011.

[3] Luca Bergamini, Yawei Ye, Oliver Scheel, Long Chen, Chih Hu, Luca Del Pero, Błażej Osiński, Hugo Grimmett, and Peter Ondruska. 2021. Simnet: Learning reactive self-driving simulations from real-world observations. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5119–5125.

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11621–11631.

[5] Alessandro Calò, Paolo Arcaini, Shaukat Ali, Florian Hauer, and Fuyuki Ishikawa. 2020. Generating avoidable collision scenarios for testing autonomous driving systems. In *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*. IEEE, 375–386.

[6] Wei-Jer Chang, Francesco Pittaluga, Masayoshi Tomizuka, Wei Zhan, and Manmohan Chandraker. 2023. Controllable Safety-Critical Closed-loop Traffic Simulation via Guided Diffusion. arXiv:2401.00391 [cs.RO]

[7] Baiming Chen, Xiang Chen, Qiong Wu, and Liang Li. 2021. Adversarial evaluation of autonomous vehicles in lane-change scenarios. *IEEE transactions on intelligent transportation systems* 23, 8 (2021), 10333–10342.

[8] Lu Chen, Yunjun Gao, Xinhan Li, Christian S Jensen, and Gang Chen. 2017. Efficient Metric Indexing for Similarity Search and Similarity Joins. *IEEE Transactions on Knowledge and Data Engineering* 29, 3 (2017), 556–571.

[9] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. 2024. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[10] Lu Chen, Qilu Zhong, Xiaokui Xiao, Yunjun Gao, Pengfei Jin, and Christian S Jensen. 2018. Price-and-time-aware dynamic ridesharing. In *2018 IEEE 34th international conference on data engineering (ICDE)*. IEEE, 1061–1072.

[11] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9268–9277.

[12] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* 34 (2021), 8780–8794.

[13] Wenhao Ding, Baiming Chen, Bo Li, Kim Ji Eun, and Ding Zhao. 2021. Multimodal safety-critical scenarios generation for decision-making algorithms evaluation. *IEEE Robotics and Automation Letters* 6, 2 (2021), 1551–1558.

[14] Wenhao Ding, Baiming Chen, Minjun Xu, and Ding Zhao. 2020. Learning to collide: An adaptive safety-critical scenarios generating method. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2243–2250.

[15] Wenhao Ding, Chejian Xu, Mansur Arief, Haohong Lin, Bo Li, and Ding Zhao. 2023. A survey on safety-critical driving scenario generation—A methodological perspective. *IEEE Transactions on Intelligent Transportation Systems* 24, 7 (2023), 6971–6988.

[16] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*. PMLR, 1–16.

[17] Jiawei Fu, Xiaotong Zhang, Zhiqiang Jian, Shitao Chen, Jingmin Xin, and Nanning Zheng. 2023. Efficient safety-enhanced velocity planning for autonomous driving with chance constraints. *IEEE Robotics and Automation Letters* 8, 6 (2023), 3358–3365.

[18] Zahra Ghodsi, Siva Kumar Sastry Hari, Iuri Frosio, Timothy Tsai, Alejandro Troccoli, Stephen W Keckler, Siddharth Garg, and Anima Anandkumar. 2021. Generating and characterizing scenarios for safety testing of autonomous vehicles. In *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 157–164.

[19] Roger Girgis, Florian Golemo, Felipe Codevilla, Martin Weiss, Jim Aldon D'Souza, Samira Ebrahimi Kahou, Felix Heide, and Christopher Pal. 2022. Latent Variable Sequential Set Transformers for Joint Multi-Agent Motion Prediction. In *International Conference on Learning Representations*.

[20] Yanchen Guan, Haicheng Liao, Zhenning Li, Jia Hu, Runze Yuan, Yunjian Li, Guohui Zhang, and Chengzhong Xu. 2024. World models for autonomous driving: An initial survey. *IEEE Transactions on Intelligent Vehicles* (2024).

[21] Niklas Hanselmann, Katrin Renz, Kashyap Chitta, Apratim Bhattacharyya, and Andreas Geiger. 2022. King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients. In *European Conference on Computer Vision*. Springer, 335–352.

[22] Kunkun Hao, Wen Cui, Yonggang Luo, Lecheng Xie, Yuqiao Bai, Jucheng Yang, Songyang Yan, Yuxi Pan, and Zijiang Yang. 2023. Adversarial safety-critical scenario generation using naturalistic human driving priors. *IEEE Transactions on Intelligent Vehicles* (2023).

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[24] Zhiyu Huang, Zixu Zhang, Ameya Vaidya, Yuxiao Chen, Chen Lv, and Jaime Fernández Fisac. 2024. Versatile Scene-Consistent Traffic Scenario Generation as Optimization with Diffusion. *arXiv preprint arXiv:2404.02524* (2024).

[25] Moritz Klischat, Edmond Irani Liu, Fabian Holtke, and Matthias Althoff. 2020. Scenario factory: Creating safety-critical traffic scenarios for automated vehicles. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 1–7.

[26] Christian Knies and Frank Diermeyer. 2020. Data-driven test scenario generation for cooperative maneuver planning on highways. *Applied Sciences* 10, 22 (2020), 8154.

[27] Friedrich Kruber, Jonas Wurst, and Michael Botsch. 2018. An unsupervised random forest clustering technique for automatic traffic scenario categorization. In *2018 21st International conference on intelligent transportation systems (ITSC)*. IEEE, 2811–2818.

[28] Friedrich Kruber, Jonas Wurst, Eduardo Sánchez Morales, Samarjit Chakraborty, and Michael Botsch. 2019. Unsupervised and supervised learning with the random forest algorithm for traffic scenario clustering and classification. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2463–2470.

[29] Shuncheng Liu, Xu Chen, Ziniu Wu, Liwei Deng, Han Su, and Kai Zheng. 2022. HeGA: heterogeneous graph aggregation network for trajectory prediction in high-density traffic. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 1319–1328.

[30] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. 2022. DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR. In *International Conference on Learning Representations*.

[31] Shuncheng Liu, Han Su, Yan Zhao, Kai Zeng, and Kai Zheng. 2021. Lane change scheduling for autonomous vehicle: A prediction-and-search framework. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3343–3353.

[32] I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[33] KM Lynch. 2017. *Modern Robotics*. Cambridge University Press.

[34] J Macqueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*.

[35] Osama Makansi, Özgün Cicek, Yassine Marrakchi, and Thomas Brox. 2021. On exposing the challenging long tail in future prediction of traffic actors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13147–13157.

[36] Hormoz Marzbani, Hamid Khayyam, Đai Võ Quoc, and Reza N Jazar. 2019. Autonomous vehicles: Autodriver algorithm and vehicle dynamics. *IEEE Transactions on Vehicular Technology* 68, 4 (2019), 3201–3211.

[37] Rebecca J Mitchell, MR Bambach, and Barbara Toson. 2015. Injury risk for matched front and rear seat car passengers by injury severity and crash type: An exploratory study. *Accident Analysis & Prevention* 82 (2015), 171–179.

[38] Matthew O'Kelly, Aman Sinha, Hongseok Namkoong, Russ Tedrake, and John C Duchi. 2018. Scalable end-to-end autonomous vehicle testing via rare-event simulation. *Advances in neural information processing systems* 31 (2018).

[39] Davis Rempe, Jonah Philion, Leonidas J Guibas, Sanja Fidler, and Or Litany. 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.

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

[41] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. 2020. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer, 683–700.

[42] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. 2018. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems* 1, 1 (2018), 187–210.

[43] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. 2022. Motion transformer with global intention localization and local movement refinement. *Advances in Neural Information Processing Systems* 35 (2022), 6531–6543.

[44] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. 2024. Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[45] Jian Sun, He Zhang, Huajun Zhou, Rongjie Yu, and Ye Tian. 2021. Scenario-based test automation for highly automated vehicles: A review and paving the way for systematic safety assurance. *IEEE transactions on intelligent transportation systems* 23, 9 (2021), 14088–14103.

[46] Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. 2021. Trafficsim: Learning to simulate realistic multi-agent behaviors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10400–10409.

[47] Siyu Teng, Xuemin Hu, Peng Deng, Bai Li, Yuchen Li, Yunfeng Ai, Dongsheng Yang, Lingxi Li, Zhe Xuanyuan, Fenghua Zhu, et al. 2023. Motion planning for autonomous driving: The state of the art and future perspectives. *IEEE Transactions on Intelligent Vehicles* 8, 6 (2023), 3692–3711.

[48] Haoxiang Tian, Guoquan Wu, Jiren Yan, Yan Jiang, Jun Wei, Wei Chen, Shuo Li, and Dan Ye. 2022. Generating critical test scenarios for autonomous driving systems via influential behavior patterns. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*. 1–12.

[49] Cumhur Erkan Tuncali and Georgios Fainekos. 2019. Rapidly-exploring random trees for testing automated vehicles. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 661–666.

[50] Jingkang Wang, Ava Pun, James Tu, Sivabalan Manivasagam, Abbas Sadat, Sergio Casas, Mengye Ren, and Raquel Urtasun. 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.

[51] Wenshuo Wang and Ding Zhao. 2018. Extracting traffic primitives directly from naturalistically logged data for self-driving applications. *IEEE Robotics and Automation Letters* 3, 2 (2018), 1223–1229.

[52] Yuyang Xia, Shuncheng Liu, Xu Chen, Zhi Xu, Kai Zheng, and Han Su. 2022. Rise: A velocity control framework with minimal impacts based on reinforcement learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2210–2219.

[53] Yuyang Xia, Shuncheng Liu, Quanlin Yu, Liwei Deng, You Zhang, Han Su, and Kai Zheng. 2024. Parameterized Decision-Making with Multi-Modality Perception for Autonomous Driving. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 4463–4476.

[54] Chejian Xu, Ding Zhao, Alberto Sangiovanni-Vincentelli, and Bo Li. 2023. Diffscene: Diffusion-based safety-critical scenario generation for autonomous vehicles. In *The Second Workshop on New Frontiers in Adversarial Machine Learning*.

[55] Danfei Xu, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. 2023. Bits: Bi-level imitation for traffic simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2929–2936.

[56] Kecheng Xu, Xiangquan Xiao, Jinghao Miao, and Qi Luo. 2020. Data driven prediction architecture for autonomous driving and its application on apollo platform. In *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 175–181.

[57] Jiawei Zhang, Chejian Xu, and Bo Li. 2024. ChatScene: Knowledge-Enabled Safety-Critical Scenario Generation for Autonomous Vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15459–15469.

[58] Linrui Zhang, Zhenghao Peng, Quanyi Li, and Bolei Zhou. 2023. Cat: Closed-loop adversarial training for safe end-to-end driving. In *Conference on Robot Learning*. PMLR, 2357–2372.

[59] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3836–3847.

[60] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. 2023. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 9 (2023), 10795–10816.

[61] Ziyuan Zhong, Davis Rempe, Yuxiao Chen, Boris Ivanovic, Yulong Cao, Danfei Xu, Marco Pavone, and Baishakhi Ray. 2023. Language-guided traffic simulation via scene-level diffusion. In *Conference on Robot Learning*. PMLR, 144–177.

[62] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. 2023. Guided conditional diffusion for controllable traffic simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3560–3566.

[63] Weitao Zhou, Zhong Cao, Yunkang Xu, Nanshan Deng, Xiaoyu Liu, Kun Jiang, and Diange Yang. 2022. Long-tail prediction uncertainty aware trajectory planning for self-driving vehicles. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 1275–1282.

[64] Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. 2023. Query-centric trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17863–17873.

[65] Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. 2022. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8823–8833.

[66] Meixin Zhu, Yinhai Wang, Ziyuan Pu, Jingyun Hu, Xuesong Wang, and Ruimin Ke. 2020. Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving. *Transportation Research Part C: Emerging Technologies* 117 (2020), 102662.

A Details of Methodology

A.1 Details of Denoiser Architecture

Map Feature Encoding. The map feature encoding uses a multi-head self-attention mechanism to process map polygons. Each map polygon $F_M[i]$ is embedded into a d-dimensional feature space, and its interactions with other map polygons are captured using attention over the set $\Omega(i)$ of neighboring map features. The encoding can be expressed as:

$$F'_M[i] = \mathrm{MHA}\left(\begin{array}{l}\mathrm{Q}: [F_M[i] + \mathrm{PE}] \\ \mathrm{K,V}: \{[F_M[j] + \mathrm{PE}]\}_{j\in\Omega(i)}\end{array}\right) \tag{22}$$

where $i \in \{1, 2, \cdots, L\}$.

Feature Concatenation. The noisy trajectory feature $F_\tau \in \mathbb{R}^{T\times d}$ is concatenated with the decision-making context features $F'_C \in \mathbb{R}^{T_h\times d}$, and combined with the denoising features $F_t \in \mathbb{R}^{(T_h+T)\times d}$, resulting in an intermediate feature $F_A \in \mathbb{R}^{(T_h+T)\times d}$:

$$F_A = \mathrm{Concat}(F_\tau, F'_C) + F_t \tag{23}$$

Transformer Decoder. The intermediate feature F_A is processed by stacked layers of the Scenario Transformer to yield F'''_A. The final denoised trajectory τ^{k-1} is generated through a standard Transformer decoder followed by a MLP. The decoder applies multi-head cross-attention between the current feature and the context, expressed as:

$$F_{out}[t] = \mathrm{MHA}\left(\begin{array}{l}\mathrm{Q}: [F'''_A[t] + \mathrm{PE}] \\ \mathrm{K,V}: \{[F'''_A[\iota] + \mathrm{PE}]\}_{\iota\in\Omega(t)}\end{array}\right) \tag{24}$$

where $t \in \{T_h + 1, \cdots, T_h + T\}$.

Final Output. The output feature $F_{out} \in \mathbb{R}^{T\times d}$ from the decoder is passed through a multi-layer perceptron (MLP), yielding the final denoised trajectory τ^{k-1}:

$$\tau^{k-1} = f(s^0, \mathrm{MLP}(F_{out})) \tag{25}$$

where s^0 is the initial state of the agent and f represents the denoising function applied to the output of the MLP.

A.2 Relative Relationship Transformation

To explore the relative relationships between agents, we transform the coordinates of each agent into the query agent's coordinate system using the following transformation matrix:

$$R^{pos}[i,j] = (P[j] - P[i])\begin{bmatrix}\cos H[i] & -\sin H[i] \\ \sin H[i] & \cos H[i]\end{bmatrix} \tag{26}$$

$$R^{\theta}[i,j] = H[j] - H[i] \tag{27}$$

$$R^{v}[i,j] = (v[j] - v[i])\begin{bmatrix}\cos H[i] & -\sin H[i] \\ \sin H[i] & \cos H[i]\end{bmatrix} \tag{28}$$

where $R^{pos}[i,j]$, $R^{\theta}[i,j]$, and $R^{v}[i,j]$ represent the 2D relative position, relative heading angle, and relative velocity between agent V_i and agent V_j, respectively. Here, $P[i]$, $H[i]$, and $v[i]$ denote the 2D coordinates, heading angle, and velocity of agent V_i in its own coordinate system.

A.3 Detail of Expert Trajectory Optimization

In the expert trajectory optimization module, the objective of collision guidance is to ensure that the ego agent does not collide with any other agents. Specifically, this can be expressed as:

$$\mathcal{J}_{coll} = \frac{1}{N^2} \sum_{(i,j),i=1,i\neq j} \sum_t \gamma^t \mathcal{L}_{coll}(i,j,t) \tag{29}$$

B Experiment Details

B.1 Implementation Details

The STD model is trained to generate 5.2 seconds of future motion based on 3 seconds of historical motion (both sampled at 10 Hz, with the historical trajectory including the current state). The specific parameters are as follows: the timestep for generating future motion is $T = 52$, the timestep for historical motion is $T_h = 31$, and the diffusion steps are $K = 100$. Regarding the details of the STD structure, the number of stacked layers of the Scenario Transformer mentioned in Section 3.2 is $L_c = L_{enc} = 2$, and the feature dimension is $d = 128$. The local vector map covers a 50-meter radius centered on each agent, with $Z = 15$ polylines and $P = 80$ points per polyline, and the attribute dimension is $R = 3$ (including 2D coordinates and polyline orientation). The dynamics model f adopts the unicycle model [33]. During training, we consider the $N = 20$ nearest agents to the sampled agent, while during inference, the number of nearby agents is not limited. We use the AdamW [32] optimizer for training with a learning rate of 0.0001.

B.2 Visualization Results

We recommend visiting the website to view more visualization results. Anonymous Website Page: Click Here.