2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

AdvDiffuser: Generating Adversarial Safety-Critical Driving Scenarios via Guided Diffusion

Yuting Xie[1], Xianda Guo[2], Cong Wang[3], Kunhua Liu[4] and Long Chen[3†]

***Abstract*— Safety-critical scenarios are infrequent in natural driving environments but hold significant importance for the training and testing of autonomous driving systems. The prevailing approach involves generating safety-critical scenarios automatically in simulation by introducing adversarial adjustments to natural environments. These adjustments are often tailored to specific tested systems, thereby disregarding their transferability across different systems. In this paper, we propose AdvDiffuser, an adversarial framework for generating safety-critical driving scenarios through guided diffusion. By incorporating a diffusion model to capture plausible collective behaviors of background vehicles and a lightweight guide model to effectively handle adversarial scenarios, AdvDiffuser facilitates transferability. Experimental results on the nuScenes dataset demonstrate that AdvDiffuser, trained on offline driving logs, can be applied to various tested systems with minimal warm-up episode data and outperform other existing methods in terms of realism, diversity, and adversarial performance.**

I. Introduction

Safety evaluation of autonomous vehicles (AV) requires scalable long-tail driving scenarios [1]–[3]. However, these kinds of scenarios are rare in the real world, which poses a data-rarity problem. A prevailing alternative is to generate safety-critical scenarios in simulation. Rather than manually designing scenarios from scratch [4], recent works seek to autonomously generate challenging scenarios via perturbing existing scenarios [5]–[8]. Generally, these works modify maneuvers of a single or a small group of background vehicles (BV) via adversarial reinforcement learning [8] or optimization searching on a scenario parameterization space [5]–[7]. Since the specific tested AV system is in the loop, these studies lack investigation into the transferability across diverse types of target AVs, which leaves the generated scenarios less flexible.

Recently, diffusion models have made significant progress in vision and language tasks [9], [10], showcasing their potential in few-shot or zero-shot learning as a highly promising generative model. And several studies have emerged that employ diffusion models to address sequence decision problems [11]–[15], wherein guidance from an auxiliary reward function is injected into the sampling process to produce class-conditional outcomes. These works demonstrate the powerful generative capabilities of diffusion models in handling out-of-distribution data without retraining.

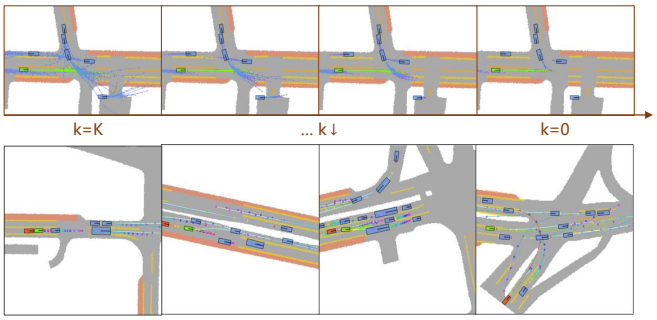

Fig. 1. AdvDiffuser generates safety-critical scenarios for testing AV systems. (Top) Diffusion-based traffic simulation involves noise undergoing k rounds of reverse diffusion process. (Bottom) The diffusion model, coupled with a reward guide, generates adversarial background trajectories. AVs are depicted in green, while vehicles executing attack behaviors are in orange.

Inspired by this, we present a novel framework for generating safety-critical driving scenarios called AdvDiffuser, which utilizes a guided diffusion model to generate adversarial trajectories. AdvDiffuser targets the planner component of AV systems, which includes prediction, planning and control modules, by planning behaviors of BVs online to intentionally disrupt targeted planners and induce collisions. As depicted in Fig 1, AdvDiffuser learns adversarial collective behaviors for BVs from offline driving logs and undergoes an interactive 'warm-up' process for online scenario generation, allowing it to adapt to various target planners effectively. The core idea of AdvDiffuser involves decoupling realism and adversariness within a diffusion model and an auxiliary collision reward model to generate plausible multi-agent trajectories while promoting the creation of adversarial trajectories. The advantage of this decoupling lies in the fact that only a small scale of parameters in the reward model needs adjustment when adapting to new targets, enabling online adversarial training for continuous improvement of autonomy.

We summarize the main contributions of this paper as follows. (1) By incorporating guided sampling into the driving simulation, we present a novel framework for generating safety-critical scenarios. (2) A multi-vehicle traffic simulation method based on a diffusion model is proposed, which

This work was supported by the National Natural Science Foundation of China (Grant No.62373356), and the Science & Technology Support Project for Young People in Colleges of Shandong Province (Grant No. 2023KJ120).

† Corresponding author: L. Chen.

[1]Y. Xie is with the School of Computer Science and Engineering, Sun Yat-Sen University xieyt8@mail2.sysu.edu.cn

[2]X. Guo is with the School of Computer Science, Wuhan University xianda_guo@163.com

[3]C. Wang and L. Chen are with the State Key Laboratory for Management and Control of Complex Systems at the Institute of Automation, Chinese Academy of Sciences wangcong2024@ia.ac.cn long.chen@ia.ac.cn

[4]K. Liu is with the School of Mechanical and Automotive Engineering, Qingdao University of Technology liukh@qut.edu.cn

effectively generates diverse and realistic background traffic flows. (3) Experimental results confirm the transferability advantages of our approach, highlighting its practical benefits in autonomous driving testing.

II. Related Work

A. Testing Autonomous Driving in Simulation

Conventional self-driving vehicle simulation testing relies on log replay, lacking reactivity and causing domain shift issues [16]–[18]. Efforts to address this include integrating reactivity into driving logs for closed-loop safety evaluation [7], [8], [19], [20]. Early heuristic simulators [21]–[23] have been replaced by neural traffic models like SimNet [24] and TrafficGen [25], which mimic human driving behaviors but lack controllability. Mixsim [5] offers controllability with constrained diversity, while Strive [6] employs a VAE-based network for plausible traffic modeling, albeit compromising realism. In contrast, our method models traffic flow as a diffusion process, using VAE's latent codes to balance realism and diversity, thus enabling controllability through an auxiliary reward function.

B. Safety-critical Driving Scenario Generation

Simulating failure scenarios is crucial for comprehensive risk assessment in autonomous systems, particularly due to autonomous systems' superior performance in controlled environments [2]. Manual creation of challenging scenarios faces scalability issues and may lead to unrealistic situations [20], [26], [27]. Recent studies focus on parameterization spaces to identify adversarial parameters using optimization-based methods [7], [28]–[33]. Methods like AdvSim directly perturb trajectory space while maintaining physical feasibility [7], while others optimize parameters within a latent space [5], [6], [30]. Nevertheless, these methods require iterative re-planning, leading to efficiency issues.

Adversarial policy models, such as NADE and D2RL, train vehicles to execute adversarial maneuvers, offering flexibility but introducing complexity and requiring a substantial number of interactions [8], [34], [35]. Our approach utilizes lightweight RL to guide sampling towards safety-critical variations efficiently, leveraging a diffusion formulation for seamless transfer.

C. Diffusion Models for Sequence Decision Problems

Conditional diffusion models can typically be categorized into two types: classifier-guidance and classifier-free [36]. The former improves sampling quality in a specific domain by utilizing gradients from a pre-trained classifier, while the latter directly incorporates class-related context into the noise model, eliminating the need for such a pre-trained classifier. These models excel in vision and language tasks and are now applied in sequence decision problems. Diffuser [11] employs a diffusion model to strategize robot behaviors, which is trained through a diffusion process over random noisy trajectories. It follows the classifier-guidance manner by incorporating gradients from a separately pre-trained reward model. Gu et al. [12] integrate maps and nearby agent states, directly into the diffusion model, to tackle the complex task of predicting multi-pedestrian trajectories. Combining classifier-guidance and classifier-free, M-Diffuser develops a constrained sampling framework for controllable trajectory generation in multi-agent scenarios using learnable differentiable cost functions. Instead of learned cost functions, Zhong et al. [14] use analytical loss functions derived from Signal Timing Logic (STL) rules to control trajectories.

Similar to [13], [14], our approach is built upon guided diffusion formulation, together with incorporating contextual information in a classifier-free manner. But unlike previous attempts that merely touched upon the controllability of trajectory sampling, we are among the first to incorporate conditional diffusion into safety-critical driving scenarios generation tasks. Furthermore, we investigate the generalization potential of diffusion models in this field, an aspect overlooked by most previous works. In this context, our work bears the closest resemblance to MetaDiffusion [15], which leverages a conditioned diffusion model to plan robot behaviors, facilitating generalization across tasks that were previously unseen. Differently, our work focuses on the challenge of generalizing across unknown adversarial targets within an adversarial task setting.

III. AdvDiffuser

A. Problem Formulation

Our goal is to create realistic challenging testing scenarios that induce failures in tested autonomous driving systems. In particular, a driving scenario $\mathcal{S}$ consists of a high-definition map M containing semantic information for drivable areas and lanes, agent states $x_{0:T}$(or trajectory $\boldsymbol{\tau}$), agent actions $u_{0:T}$. We denote x_t (resp., u_t) the joint states of all agents at timestep t. Similarly, we denote m_t as the collection of all agents's surrounding maps and $Past$ as historical states. Further, we introduce a subscript i to indicate the ith vehicle and employ a superscript '+' for tested autonomous vehicles to distinct them from background vehicles. Specifically, we parameterize agent states by their 2D position, heading, and velocity, while actions by steering angle and acceleration.

The objective of autonomous driving systems is to optimize actions through a cost function $\mathcal{C}$ to ensure comfortable and safe maneuvering. In contrast, our objective is to intentionally disrupt their comfort and security, maximizing this cost. As our method generates trajectories directly, which can be converted to actions through a dynamic model, we describe this objective using trajectory representation.

$$\boldsymbol{\tau}^* = \max_{\boldsymbol{\tau}} C(\boldsymbol{\tau}^+, \boldsymbol{\tau}, M, Past) \tag{1}$$

B. Diffusion Model for Multi-agent Trajectories

We employ a diffusion model to generate plausible collective behaviors of background vehicles. The diffusion model conceptualizes data generation as an iterative denoising process, which is the inverse of a forward diffusion process with state transitions following the properties of a Markov chain. As the number of iterations increases, the model eventually converges to a stationary distribution. Therefore,

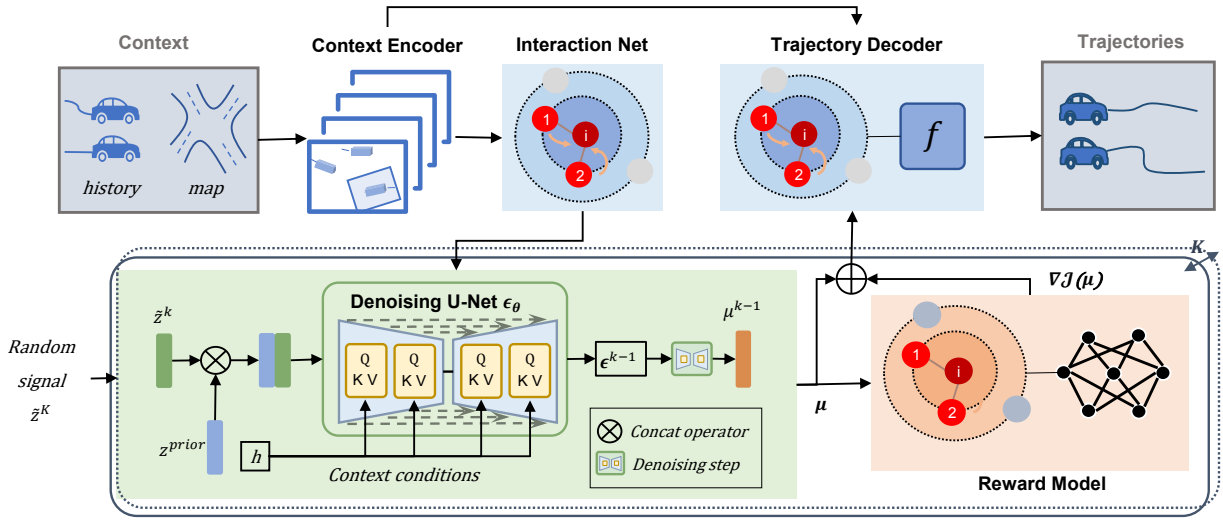

Fig. 2. Overview of our AdvDiffuser framework. Given a context comprising historical vehicle trajectories and the map, AdvDiffuser aims to generate adversarial trajectories for background vehicles. The Interaction Net is a Graph Neural Network (GNN) architecture that encodes inter-vehicle interactions, providing a latent code z_{prior} as a condition for the denoising process. The dynamics model f in Trajectory Decoder transforms vehicle actions into trajectories.

by introducing a learnable noise prediction model, we can restore random signals within specified domains through the iterative denoising process. The diffusion process is implemented within the latent space in our multi-agent trajectory generation model, rather than directly on vehicle trajectories. To ensure the physical plausibility of the generated trajectories, we further integrate it with an additional autoregressive trajectory decoder.

Architecture. The process of sampling multiple vehicle trajectories involves a context encoder, a trajectory decoder, and a reverse diffusion process. The context encoder $E_\theta(m, past)$ takes surrounding maps and historical trajectories as input, generating trajectory context $(h, \mathbf{z})$. As illustrated Fig. 2, a simple Multi-Layer Perceptron (MLP) encodes vehicle state embedding h_i, and an interaction net with two fully connected graph layers aggregates vehicle states, capturing interactions and producing latent codes $\mathbf{z_i}$ for each vehicle.

The reverse diffusion process models a conditional probability distribution $p_\theta(\tilde{\mathbf{z}}_i^{k-1}|\tilde{\mathbf{z}}_i^k, \mathbf{z_i}, \mathbf{h_i})$ to align with the posterior distribution $q(\tilde{\mathbf{z}}_i^k|\boldsymbol{\tau}, h)$. Following [37], we formulate the denoising process with a noise function $\boldsymbol{\epsilon}^k$ as:

$$\boldsymbol{\mu}^{k-1} = \sqrt{\alpha^{k-1}}\left(\frac{\tilde{\mathbf{z}}_\mathbf{i}^\mathbf{k} - \sqrt{\mathbf{1}-\alpha^\mathbf{k}}\boldsymbol{\epsilon}^\mathbf{k}}{\sqrt{\alpha^k}}\right) + \sqrt{1-\alpha^{k-1}}\boldsymbol{\epsilon}^k \tag{2a}$$

$$q(\tilde{\mathbf{z}}_\mathbf{i}^{k-1}|\tilde{\mathbf{z}}_\mathbf{i}^k; \boldsymbol{\epsilon}^k) = \mathcal{N}(\boldsymbol{\mu}^{k-1}, \frac{1-\alpha^{k-1}}{1-\alpha^k}\beta^k) \tag{2b}$$

where k denotes the kth round of diffusion, $\alpha^k = \prod_{s=1}^{k}(1-\beta^s)$ and β^k are fixed variance schedulers that controls scale of noise. By parameterizing the noise function through a trainable noise prediction network ε_θ, we obtained the conditional distribution as:

$$p_\theta(\tilde{\mathbf{z}}_i^{k-1}|\tilde{\mathbf{z}}_i^k, \mathbf{z}_i, h_i) = q(\tilde{\mathbf{z}}_i^{k-1}|\tilde{\mathbf{z}}_i^k; \varepsilon_\theta(\tilde{\mathbf{z}}_i^k, k, \mathbf{z}_i, h_i)). \tag{3}$$

A U-Net architecture is employed in the noise prediction network as illustrated in Fig. 2. To incorporate contextual information, $\mathbf{z}_i$ is concatrated with $\tilde{\mathbf{z}}_\mathbf{i}^k$ and integrated into ε_θ. Additionally, an adaptive group normalization (AdaGN) structure [38] is introduced to the U-Net architecture for taking h_i as another condition.

$$\mathrm{AdaGN}(\mathrm{h_i}, \mathrm{k}, \mathrm{f}) = \mathrm{h_s}(\mathrm{k_s}\mathrm{GroupNorm}(\mathrm{f}) + \mathrm{k_b}) \tag{4}$$

where f is the normalized feature maps of UNet, and h_s donates the affine projection of h_i, while $(k_s, k_b) = \mathrm{MLP}(\varphi(k))$ represents the output of a MLP on sinusoidal encoding of k.

Taking in the denoised latent $\tilde{\mathbf{z}}^0$, the trajectory decoder $D_\theta(\tilde{\mathbf{z}}^0, h)$ generates future trajectories $\boldsymbol{\tau}$ in an autoregressive manner while also incorporating an interaction net to ensure realistic vehicle interactions. The vehicle's acceleration and angle are determined through an iterative process, while the trajectory is updated through a simplistic bicycle model. The state embedding h_i is updated using a recurrent neural network (RNN) as the memory unit. This iterative approach continues until a smooth and coherent path prediction is generated.

Training. The training of the denoiser ε_θ necessitates an auxiliary posterior distribution network $q_\theta(\mathbf{z}|\boldsymbol{\tau}, \mathbf{h})$, which takes future trajectories and current states' context embeddings as inputs to generate training samples $\tilde{\mathbf{z}}_i^0$. The posterior network utilizes a network structure similar to that of the context encoder.

We initially train the denoiser and other modules separately, followed by joint training of all modules. To collaboratively optimize the trajectory decoder, context encoder, and posterior net, we employ the modified Evidence Lower Bound (ELBO) loss function commonly utilized in VAEs [39].

$$\mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}_q(\log p(\boldsymbol{\tau}|\mathbf{z})) - KL(q(\mathbf{z}|\boldsymbol{\tau}, h)|p(\mathbf{z}|h)) \tag{5}$$

Training of the denoising net ε_θ is done by optimizing L_2 noise loss. Notably, $\tilde{\mathbf{z}}^0$ is produced from the posterior distribution network.

$$\mathcal{L}_{\text{noise}} = \sum_{k=1}^{K} \mathbb{E}_{\tilde{\mathbf{z}}^0, \boldsymbol{\epsilon}^k} \left[||\varepsilon_\theta(\tilde{\mathbf{z}}^k, k, \mathbf{z}, h) - \boldsymbol{\epsilon}^k||_2^2 \right] \tag{6}$$

The final joint training of all modules incorporates collision penalties for traffic flow, in addition to the aforementioned two losses.

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{ELBO}} + \mathcal{L}_{\text{noise}} + \mathcal{P}_{coll} \tag{7}$$

C. Reward Model for Guided Sampling

Appealing to [11], reinforcement learning problems can be modeled as a generative process using guided sampling. In this part, we extend the diffusion model presented in Sec. III-B by incorporating a guidance function to generate adversarial vehicle behaviors. Specifically, a binary classifier that identifies adversarial samples is incorporated into the original denoising process transition in Eq.(2b).

$$\begin{aligned} &p_\theta(\tilde{\mathbf{z}}^{k-1}|\tilde{\mathbf{z}}^k, O_{1:T}; \boldsymbol{\epsilon}^k) \approx N(\boldsymbol{\mu}^{k-1} + \Sigma g, \Sigma^{k-1}) \\ &with \quad \Sigma^{k-1} = \frac{1-\alpha^{k-1}}{1-\alpha^k}\beta^k \end{aligned} \tag{8}$$

where O_t donates a binary indicator of whether the latent code leads to an optimal trajectory at time step t and

$$\begin{aligned} g &= \nabla_{\tilde{\mathbf{z}}} \log p(\mathcal{O}_{1:T}|\tilde{\mathbf{z}})|_{\tilde{\mathbf{z}}=\boldsymbol{\mu}} = \sum_{t=0}^{T} \nabla R(\boldsymbol{\mu}, h, \mathbf{z}; \boldsymbol{\mu}) \\ &= \nabla \mathcal{J}(\boldsymbol{\mu}). \end{aligned} \tag{9}$$

where $R(\circ)$ is the reward function associated with the adversarial objective and $\mathcal{J}(\circ)$ computes expectation of accumulated rewards. Thus, adversarial samples can be generated by perturbing the predicted mean during denoising using the gradient of the reward function. Analogous to value-based reinforcement learning, a trainable reward model is formulated to assess the accumulated future rewards $\mathcal{J}(\boldsymbol{\mu})$.

Architecture. We adopted the classic DQN network architecture to construct the reward model. Adhering to commonly used notation in reinforcement learning for clarity, we regard vehicle context $(h, \mathbf{z})$ as states and predicted mean $\boldsymbol{\mu}$ as actions here.

$$\begin{aligned} \mathcal{J}(\boldsymbol{\mu}) &= Q_\theta(s, a) \\ with \quad s &= (h, \mathbf{z}) \; and \; a = \boldsymbol{\mu} \end{aligned} \tag{10}$$

where s represents states, a donates actions, and $Q(\circ)$ is the action-value function. Moreover, for efficient information exchange among vehicles, we integrated the interaction net into the state encoding layer of DQN. Unlike the interaction net employed in trajectory generation, nodes representing target and background vehicles are labeled distinctly.

Training. We initially train the reward model on data from driving logs by randomly selecting target vehicles within the original scenarios and replaying actions for these target vehicles. Additionally, we generate trajectories of other vehicles by feeding a random z into the trajectory decoder, thereby collecting a series of experiences (s_t, a_t, r_t, s_{t+1}) for learning the Q function. Similar to DQN, we use mean squared error here.

$$\mathcal{L}_{\text{Q}} = \mathbb{E}_{s_t, a_t} \left[||R_t + \gamma \cdot \max_a Q_\theta(s_{t+1}, a) - Q_\theta(s_t, a_t)||_2^2 \right] \tag{11}$$

where γ is the discount factor. The reward r_t at timestep t consists of adversarial rewards and penalties on background collisions.

$$R_t = \mathcal{R}_{adv} - \mathcal{P}_{coll} \tag{12}$$

where $\mathcal{R}_{adv}$ represents collision rewards incurred by target AVs in generated scenarios, while $\mathcal{P}_{coll}$ denotes penalties for BVs collisions (excluding collisions involving a BV and an AV). Collisions encompass both inter-vehicle collisions and environmental collisions (such as leaving drivable areas), employing a differentiable collision detection method based on distance measurement following [40].

D. Generating Safety-critical Scenarios

Initially, historical trajectories of all vehicles and their surrounding maps are encoded as contextual embeddings. Conditioned in this context, the diffusion model iteratively denoises random noise while simultaneously evaluating the adversarial reward for each outcome. The denoising results are then refined through gradient guidance of the reward to obtain a latent code capable of generating adversarial background traffic flow after K rounds of denoising. Finally, the trajectory decoder restores the final trajectories from latent space. The interaction between background vehicles and the target vehicle continues until the scenario ends, either through collision or reaching the terminal.

IV. Experiments

We next showcase the capabilities of AdvDiffuser in generating worst scenarios. In Sec. IV-A, we assess its ability to generate diverse and realistic scenarios. Sec. IV-C analyzes how AdvDiffuser affects tested planners' security and comfort. Sec. IV-C discusses transferability among different tested planners and the effectiveness of the few-shot learning.

Dataset. We evaluate our methods on the nuScenes [41] dataset, which consists of 1000 driving scenes, each spanning 20 seconds duration at 2Hz. Following the split and set guidelines in the nuScenes prediction challenge, we split driving logs into 8-second segments, using past trajectories from a 2-second duration to predict future ones for the following 6 seconds. Scenarios are evaluated within these time windows.

Baselines. We compare our model with other state-of-the-art methods for generating adversarial driving scenarios

- *Replay*: rolls out vehicle trajectories from driving logs that are unresponsive to AV actions.
- *AdvSim* [7]: employs a surrogate AV to iteratively determine optimal adversarial actions, initialized by the output of SimNet.
- *Strive* [6]: conducts optimization searches over the latent space of TrafficSim.

TABLE I

EVALUATION OF REAL TRAFFIC SIMULATION. OUR DIFFUSION MODEL WITHOUT A GUIDE GENERATES DIVERSE AND REALISTIC TRAFFIC FLOW. THE OBSERVED ELEVATED COLLISION RATE WITH THE ENVIRONMENT CAN BE ATTRIBUTED TO THE INCLUSION OF ROADSIDE PARKED VEHICLES IN THE ORIGINAL DATASET.

Algorithms	**Diversity**		**Distribution Realism (JSD)**			**Common Sense**	
	minSFDE (m)	FDD (m)	Vel (e^{-2})	Acc (e^{-2})	TTC (e^{-2})	Env Coll (%)	Veh Coll (%)
AdvSim	5.49	3.12	0.89	2.21	0.95	13.06	0.87
Strive	2.70	14.48	0.96	2.33	1.85	12.02	0.16
Ours (traj2traj)	931	**3549**	21.27	14.82	16.63	22.82	0.69
Ours	**2.21**	14.66	**0.81**	**1.68**	**0.57**	**11.89**	**0.02**

- *Adv-RL* [35]: an adversarial BV planner with an actor architecture aligned with SimNet.

Metrics. We highlight realistic safety-critical scenarios and propose a suite of metrics to assess the quality of generated scenarios, as well as evaluate the performance of tested planners on our adversarial scenarios.

- *Diversity:* metrics include Final Displacement Diversity (FDD) and Minimum Scenarios Final Distance Error (minSFDE), as per [5], [40], measuring trajectory diversity.
- *Distribution realism:* Jensen-Shannon divergence (JSD) [42] quantifies distribution gaps in vehicle velocity, acceleration, and time-to-collision (TTC) histograms.
- *Common sense:* collision rate (CR) computes the percentage of background vehicles involved in collisions, assessing scenario realism.
- *Adversarial:* metrics evaluate collision severity based on targeted vehicle collision rates and velocities, as well as assessing maneuver comfort via acceleration and jerk.

Implementation Details. AdvDiffuser is implemented using PyTorch and trained on 4 GeForce RTX 3090 GPUs. The diffusion model is trained for 200 epochs using the Adam optimizer with a learning rate of $5 * 10^{-4}$. For the reward model, we employ a learning rate of 10^{-3} and conduct 100 epochs for pretraining. The batch size is set to be 8.

A. Simulating Real Traffic

To evaluate the performance of AdvDiffuser in simulating real traffic, we assess the diversity and plausibility of generated trajectories. Our model is compared with the generative baselines, *AdvSim* and *Strive* for conventional traffic flow generation. To demonstrate the feasibility of diffusion in the latent space, we also compare its performance with that of directly applying diffusion on trajectories, marked as "traj2traj". For sampling models, ten samples are generated for each initial scene. To ensure comparability, all models utilize an identical GNN-based context encoder.

Tab. I shows the quantitative results. In contrast to the individual control exerted by *AdvSim*, the collective control of multiple agents produces motion distributions that closely resemble real-world patterns, characterized by reduced JSD values and minimal inter-vehicle collision rates. Notably, diffusion on trajectories shows higher FDD values indicating greater diversity of generated scenarios but also exhibits the largest MinSFDE value suggesting significant deviation from real-world scenarios. This observation suggests that the direct diffusion along trajectories may pose challenges and hinder the coverage of original real-world scenarios. Our method integrates the reverse diffusion process over the latent space, yielding superior performance with a slight improvement over *Strive*. This improvement can be credited to the diffusion process, gathering samples that closely mimic the real-world posterior distribution and enriching diversity.

B. Generating Accident-Prone Scenarios

We show that AdvDiffuser creates challenging scenarios leading to collisions and discomfort for the tested planner, along with more realistic background traffic flows. Since rule-based planners remain prevalent in practical AV systems, we use a simple lane-graph-based planner [43] as the attacked planner. *AdvSim* and *Strive* are both optimized for 20 rounds and use the rule-based planner as their surrogate model. *Adv-RL* and the reward model of AdvDiffser adopt the same network structure.

Tab. II presents the results of the quantitative evaluation. Compared to real-world driving scenarios, *Replay* generates slightly more challenging situations, lacking responsiveness that actively avoids AV encounters. *Replay* can serve as a bottom line when evaluating the adversarial performance. Search-based methods, *AdvSim* and *Strive*, have been observed to increase collision rates and worsen accident severity in AVs, causing discomfort from sudden acceleration and jerkiness. However, the deviation in distribution and collision rates of BVs worsens compared to free-flow traffic simulations, attributed to their inclination towards adversarial aspects rather than rational behavior during the optimization process. Notably, *AdvSim* outperforms *Strive* in adversarial scenarios but has worse trajectory rationality, possibly due to its direct search on actions, which is more flexible. While *Strive*'s exploration of a condensed latent space enhances the plausibility. It is worth noting that both methods require significantly more time than other methods due to the planner-on-loop searching process, preventing real-time testing. By incorporating a guided sampling mechanism, AdvDiffuser effectively trains collective adversarial attacks on a comparative scale of adversarial parameters to *Adv-RL*. Nevertheless, AdvDiffuser outperforms the individual control one, *Adv-RL*, across nearly all metrics. Overall, AdvDiffuser demonstrates comparative adversarial performance, with collision rates surpassed only by *AdvSim*, while

TABLE II

EVALUATION OF GENERATED SAFETY-CRITICAL SCENARIOS COMPARED WITH EXISTING ALGORITHMS. THE ATTACKED PLANNER IS A RULE-BASED ONE. ADVDIFFUSER ACHIEVES COMPARABLE ADVERSARIAL PERFORMANCE WHILE EXHIBITING ADVANTAGES IN TERMS OF SCENARIO RATIONALITY AND EFFICIENCY, DEMONSTRATING A MORE COMPREHENSIVE AND BALANCED PERFORMANCE.

Algorithms	**Collision**		**AV Comfortable**		**BV Plausibility**			**Real Time**
	CR (%)	Coll Vel (m/s)	Acc (m/s^2)	Jerk (m/s^3)	JSD (e^{2})	Env Coll (%)	Veh Coll (%)	$> 10Hz$
Replay	8.45	7.73	4.13	12.98	0.14	16.86	0.0	-
AdvSim	**18.54**	4.72	**4.28**	**13.66**	3.48	16.83	0.94	✗
Strive	10.33	5.88	4.27	13.72	2.64	16.04	0.43	✗
Adv-RL	10.56	5.46	3.71	10.91	7.48	21.71	5.21	✓
Ours	11.03	**7.26**	3.89	11.97	**2.49**	**14.91**	**0.27**	✓

TABLE III

TRANSFERABILITY OF SAFETY-CRITICAL SCENARIO GENERATION METHODS ACROSS DIVERSE TARGET PLANNERS. ADVDIFFUSER MAINTAINS STABLE EVALUATION OUTCOMES WHILE ENSURING A REASONABLE BACKGROUND TRAFFIC FLOW.

Algorithms		**Rule-based**			**SimNet**			**TrafficSim**		
		SimNet	*TrafficSim*	Ours (w/o guide)	*Rule-based*	*TrafficSim*	Ours(w/o guide)	*Rule-based*	*SimNet*	Ours (w/o guide)
AdvSim	AV CR (%)	20.89	20.89	20.66	**16.20** (**13**% ↓)	31.69	23.00	**15.96** (**14**% ↓)	33.33	22.54
	BV CR (%)	17.77	17.77	17.77	17.73	16.16	17.73	17.73	16.82	17.93
Strive	AV CR (%)	18.54	16.43	18.77	**9.56** (**7**% ↓)	15.49	18.08	**10.03** (**3**% ↓)	19.48	19.95
	BV CR (%)	16.35	16.39	16.44	16.46	16.51	16.43	16.45	16.54	16.37
Adv-RL	AV CR (%)	22.30	20.66	21.13	**10.74** (**4**% ↑)	23.24	23.71	**9.62** (**9**% ↓)	27.93	26.06
	BV CR (%)	26.91	26.9	26.92	29.31	29.31	29.31	30.75	30.7	30.75
Ours	AV CR (%)	19.01	17.14	20.19	**11.03** (≈)	15.49	16.67	**11.50** (**4**% ↑)	17.37	17.84
	BV CR (%)	14.78	14.79	14.73	14.63	14.81	15.25	14.71	14.98	14.94

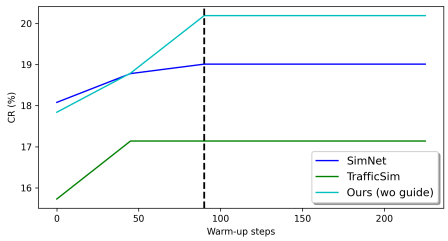

Fig. 3. Evaluation of few-shot learning. AdvDiffuser exhibits increased collision rates when encountering novel targeted planners following a few initial interactive steps.

outperforming in accident severity, scenario plausibility, and efficiency compared to alternatives, highlighting its well-rounded and balanced performance for superior outcomes. These observations are quantitatively supported by Fig. 1.

C. Transferability Analysis

We assess the transferability of adversarial methods across different planners, as shown in Tab. III. In addition to the rule-based planner, we evaluate various traffic simulation planners as discussed in Sec. IV-A. Initially, we examine the performance of AdvDiffuser in a few-shot learning. As depicted in Fig. 3, even with a limited number of warm-up steps, we consistently improve adversarial effectiveness, resulting in increased collision rates for all new targets within 90 steps.

Moreover, we thoroughly assess the transferability of AdvDiffuser and other SoTAs. Notably, *Adv-RL* and *SimNet* consistently demonstrate superior attack outcomes against *AdvSim*, while *Strive* and AdvDiffuser show optimal attacks against our fundamental diffusion model. The strong correlation between an attacker's optimal outcomes and a structurally similar planner supports the intuition that better accuracy in predicting the targeted vehicle leads to more successful attacks. We highlight the significance of columns related to *rule-based*, as *rule-based* shows a low structural correlation with other neural planners and demonstrated the highest resilience against attacks from all these neural adversarial models with the lowest collision rate. This unique attribute makes it a relatively ideal test target for assessing transferability across different source planners. While other adversarial techniques exhibit a slight degradation in performance compared to results trained on *rule-based*. AdvDiffuser stands out by maintaining consistent and stable evaluation outcomes while ensuring a reasonable background traffic flow.

V. DISCUSSION

The absence of a universally acknowledged benchmark for safety-critical driving scenarios poses significant concerns regarding the practical utility of simulation-generated adversarial scenarios on real-world autonomous driving. The validity of using such scenarios as evaluation criteria warrants further investigation, particularly regarding their relevance to actual driving conditions. The pivotal question revolves around discerning the extent to which unrealistic or highly improbable hazardous situations contribute to the enhancement of autonomous driving system safety. Redirecting attention towards assessing the likelihood of a simulated scenario manifesting in real-world settings may offer more pragmatic insights for understanding and refining the efficacy of autonomous driving systems.

REFERENCES

[1] Y. Kang, H. Yin, and C. Berger, "Test your self-driving algorithm: An overview of publicly available driving datasets and virtual testing environments," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 171–185, 2019.

[2] W. Ding *et al.*, "A survey on safety-critical driving scenario generation—a methodological perspective," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[3] L. Chen *et al.*, "Milestones in autonomous driving and intelligent vehicles: Survey of surveys," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1046–1056, 2022.

[4] C. Zhang *et al.*, "Rethinking closed-loop training for autonomous driving," in *European Conference on Computer Vision*. Springer, 2022, pp. 264–282.

[5] S. Suo *et al.*, "Mixsim: A hierarchical framework for mixed reality traffic simulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, Conference Proceedings, pp. 9622–9631.

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

[7] J. Wang *et al.*, "Advsim: Generating safety-critical scenarios for self-driving vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, Conference Proceedings, pp. 9909–9918.

[8] S. Feng *et al.*, "Dense reinforcement learning for safety validation of autonomous vehicles," *Nature*, vol. 615, no. 7953, pp. 620–627, 2023. [Online]. Available: https://doi.org/10.1038/s41586-023-05732-2

[9] C. Saharia *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 479–36 494, 2022.

[10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.

[11] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *International Conference on Machine Learning*, 2022, Conference Proceedings.

[12] T. Gu *et al.*, "Stochastic trajectory prediction via motion indeterminacy diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, Conference Proceedings, pp. 17 113–17 122.

[13] C. Jiang *et al.*, "Motiondiffuser: Controllable multi-agent motion prediction using diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, Conference Proceedings, pp. 9644–9653.

[14] Z. Zhong *et al.*, "Guided conditional diffusion for controllable traffic simulation," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, Conference Proceedings, pp. 3560–3566.

[15] F. Ni *et al.*, "Metadiffuser: Diffusion model as conditional planner for offline meta-rl," in *International Conference on Machine Learning*, 2023, Conference Proceedings.

[16] W. Li *et al.*, "Aads: Augmented autonomous driving simulation using data-driven algorithms," *Science robotics*, vol. 4, no. 28, p. eaaw0863, 2019.

[17] A. Kar *et al.*, "Meta-sim: Learning to generate synthetic datasets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4551–4560.

[18] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9329–9338.

[19] X. Yan *et al.*, "Learning naturalistic driving environment with statistical realism," *Nature Communications*, vol. 14, no. 1, p. 2037, 2023. [Online]. Available: https://doi.org/10.1038/s41467-023-37677-5

[20] C. Zhang *et al.*, "Rethinking closed-loop training for autonomous driving," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*. Springer, 2022, Conference Proceedings, pp. 264–282.

[21] R. F. Benekohal and J. Treiterer, "Carsim: Car-following model for simulation of traffic in normal and stop-and-go conditions," *Transportation Research Record*, vol. 1194, pp. 99–111, 1988.

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

[23] P. A. Lopez *et al.*, "Microscopic traffic simulation using sumo," in *2018 21st international conference on intelligent transportation systems (ITSC)*. IEEE, 2018, pp. 2575–2582.

[24] L. Bergamini *et al.*, "Simnet: Learning reactive self-driving simulations from real-world observations," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, Conference Proceedings, pp. 5119–5125.

[25] L. Feng, Q. Li, Z. Peng, S. Tan, and B. Zhou, "Trafficgen: Learning to generate diverse and realistic traffic scenarios," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3567–3575.

[26] J. M. Scanlon *et al.*, "Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain," *Accident Analysis & Prevention*, vol. 163, p. 106454, 2021.

[27] Q. Li *et al.*, "Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 3, pp. 3461–3475, 2022.

[28] Q. Zhang, S. Hu, J. Sun, Q. A. Chen, and Z. M. Mao, "On adversarial robustness of trajectory prediction for autonomous vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, Conference Proceedings, pp. 15 159–15 168.

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

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

[31] N. Hanselmann, K. Renz, K. Chitta, A. Bhattacharyya, and A. Geiger, "King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients," in *European Conference on Computer Vision*. Springer, Conference Proceedings, pp. 335–352.

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

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

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

[35] S. Feng, X. Yan, H. Sun, Y. Feng, and H. X. Liu, "Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment," *Nature Communications*, vol. 12, no. 1, p. 748, 2021. [Online]. Available: https://doi.org/10.1038/s41467-021-21007-8

[36] L. Yang *et al.*, "Diffusion models: A comprehensive survey of methods and applications," *ACM Computing Surveys*, 2022.

[37] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2020.

[38] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[39] S. R. Bowman *et al.*, "Generating sentences from a continuous space," *arXiv preprint arXiv:1511.06349*, 2016.

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

[41] H. Caesar *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 621–11 631.

[42] M. Igl *et al.*, "Symphony: Learning realistic and diverse agents for autonomous driving simulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2445–2451.

[43] M. Montemerlo *et al.*, "Junior: The stanford entry in the urban challenge," *Journal of field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.