Supplementary Material: SAFE-SIM: Safety-Critical Closed-Loop Traffic Simulation with Diffusion-Controllable Adversaries

Wei-Jer Chang[1], Francesco Pittaluga[2], Masayoshi Tomizuka[1], Wei Zhan[1], and Manmohan Chandraker[2,3]

[1] UC Berkeley [2] NEC Labs America [3] UC San Diego

Compared to previous works, our methodology enables controllable adversaries through multiple controllable factors to generate closed-loop safety-critical simulations. This allows for the generation of a broad range of safety-critical behaviors across diverse scenarios.

A Qualitative Results

For insights into closed-loop simulation outcomes, we invite readers to view the supplementary videos.

We present two sets of qualitative results. The first set, illustrated in Figure A1, displays a variety of safety-critical simulations where altering the trajectory proposals modifies the collision types. The second set, depicted in Figure A2, showcases simulations that demonstrate different collision scenarios achieved by adjusting the Time-To-Collision (TTC) to influence the safety-criticality of the situation. Unlike the STRIVE method, which tends to generate scenarios with limited variability, our approach utilizes multiple control mechanisms (such as varying trajectory proposals and safety-criticality levels) to create a broader spectrum of safety-critical conditions. This flexibility is particularly beneficial for testing and evaluating autonomous driving algorithms under various challenging conditions.

B Details on Partial Diffusion

B.1 Methodology for Generating Trajectory Proposals

To generate trajectory proposals for the partial diffusion process, which aims to create potential collision scenarios, we present a straightforward method based on lane relationships. In addition to selecting different lane relationships to represent various types of collisions, we further refine our control over these scenarios by introducing two primary variations: 1) the relative distance to the conflict point and 2) the normal offsets of the lane, as illustrated in Figure A3:

1. **Relative Distance to the Conflict Point:** This adjustment allows for the precise management of how vehicles navigate interactions, such as passing or yielding, by selecting specific accelerations that achieve the desired distance to the conflict point.

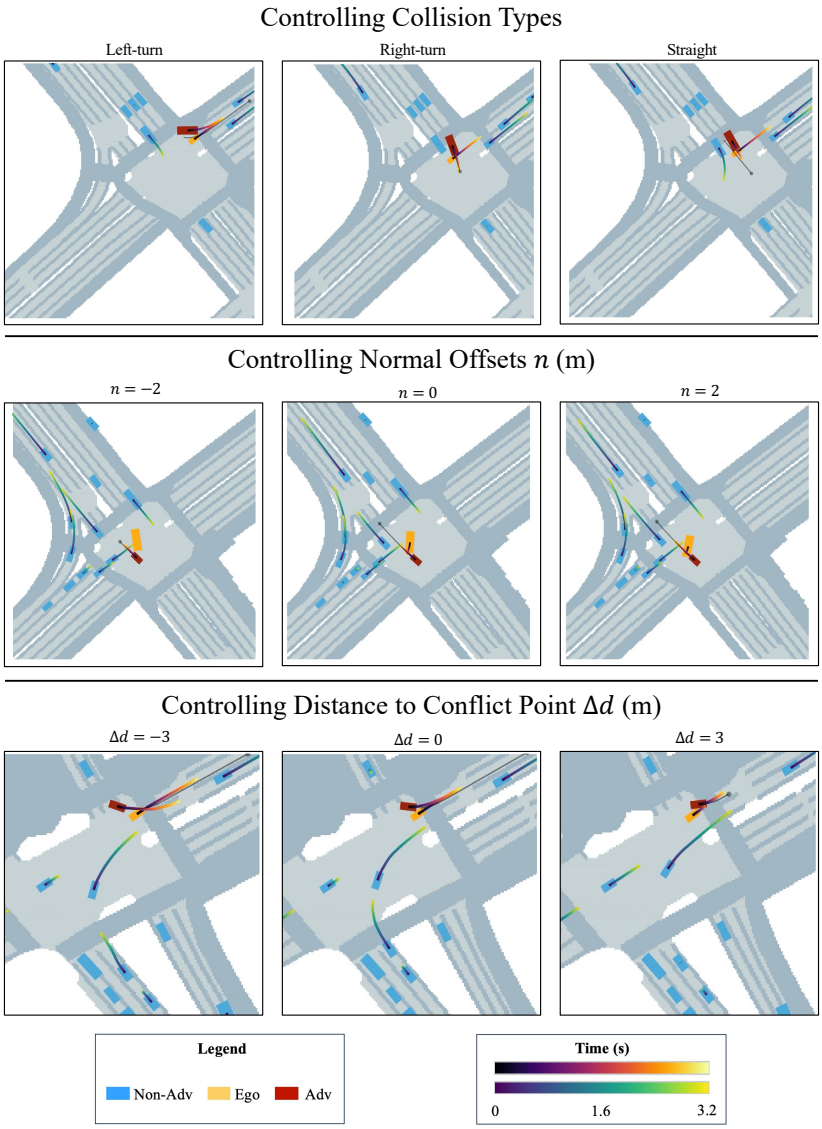

Fig. A1: Illustration of Diverse Collision Scenarios via Partial Diffusion. This figure showcases example simulations that highlight how varying trajectory proposals can influence the occurrence and type of collisions. The black line represents the trajectory proposals for the adversarial vehicle.

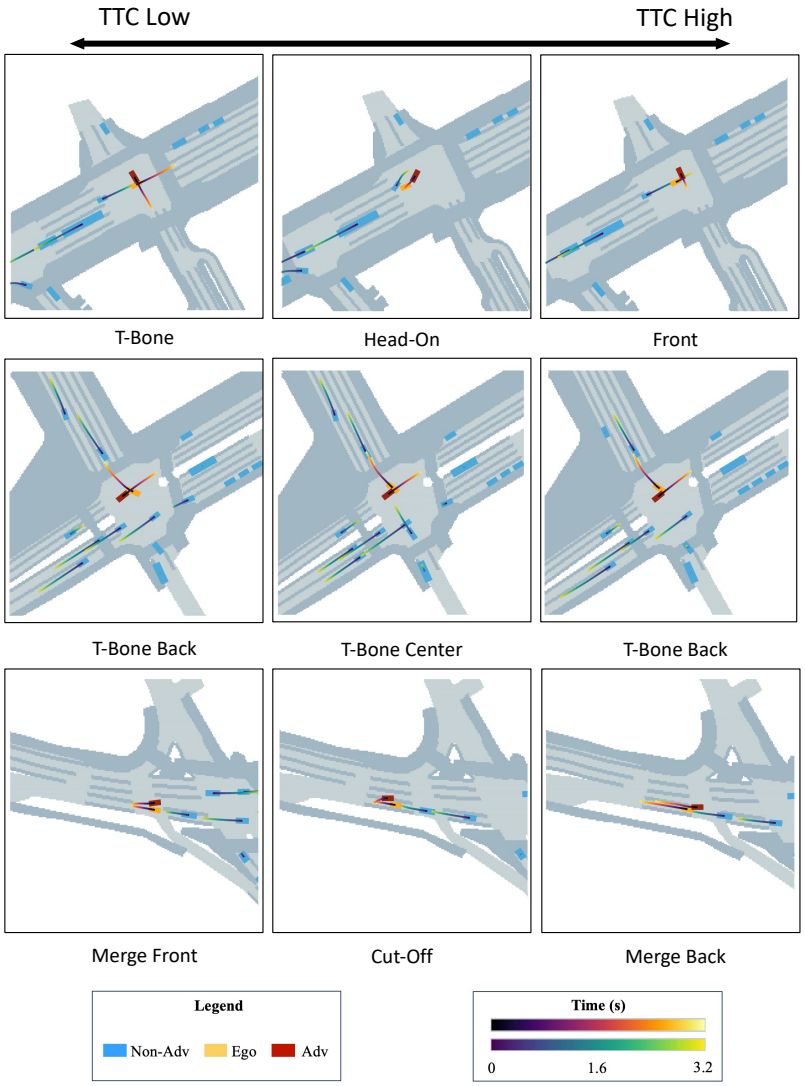

Fig. A2: Impact of Time-To-Collision (TTC) Control on Collision Scenarios. This figure demonstrates example simulations where adjusting the TTC parameter influences the dynamics and outcomes of collision scenarios, showcasing the method's versatility in testing autonomous driving algorithms under different conditions.

2. **Lane's Normal Offsets:** Modifying these offsets enables the generation of trajectories that accurately reflect the spatial dynamics of vehicle positioning within lanes.

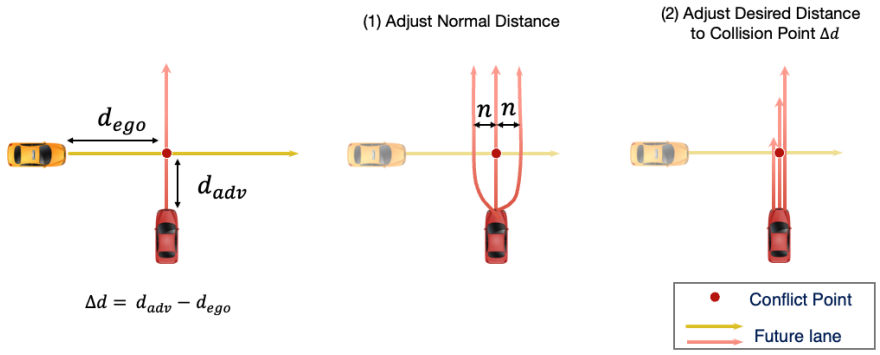

Fig. A3: Methods for generating different trajectory proposals.

In addition, we can also generate proposals based on different lanes to have different relationships.

Note that is essential to generate trajectory proposals within a closed-loop simulation, updated at every planning cycle. Since the diffusion model outputs action sequences, after generating the initial proposals, we employ inverse dynamics to calculate the corresponding turning rates.

B.2 Ablation study of Partial Diffusion

We measure the Mean Squared Error (MSE) to quantify the difference between initial trajectory proposals and the outcomes from the partial diffusion model, focusing on the first second of the trajectory in each planning iteration. Table A1 reveals that the trajectory MSE varies with the diffusion ratio. This ratio is adjustable, enabling the calibration of the model to align with user needs and maintain a balance between the original proposals and the diffusion model's output. A partial diffusion ratio of $\gamma = 0.0$ corresponds to the highest collision rate, suggesting that our initial trajectory proposals effectively signal potential collisions. After the diffusion model's denoising step, the lateral acceleration diminishes significantly, leading to more realistic trajectory generations. This underscores the importance of our proposed partial diffusion process, highlighting its effectiveness in balancing the alignment between trajectory proposals and the model's output, which represents the underlying data distribution.

C Metrics Definitions

This section outlines the definitions of the metrics used in our evaluations, averaged across all scenarios, except for the realism metric.

Partial Diffusion ratio γ	**Coll Rate** (%) ↑	**Adv Offroad** (%) ↓	**Traj MSE** (m^2) ↓	**Adv max lateral acc** (m/s^3) ↓
0.0	**26.8**	**7.5**	**3.09**	17.5
0.2	14.6	12.5	20.2	2.3
0.4	17.1	10.0	19.2	2.3
0.6	9.8	12.5	16.6	2.0
0.8	12.2	10.0	22.2	2.44
1.0	14.6	**7.5**	35.6	**1.75**
w/o Partial Diffusion	7.4	10.0	33.4	2.22

Table A1: Ablation study on the Partial Diffusion ratio.

C.1 Traffic Simulation Metrics

Off-road. This metric measures the percentage of agents that go off-road in a given scenario. An agent is considered off-road if its centroid moves into a non-drivable area.

Collision. This metric represents the percentage of agents involved in collisions with other agents during the simulation.

Realism. Adopting the approach from [39], realism is quantified using the Wasserstein distance. This metric compares the normalized histograms of the driving profiles, focusing on the mean values of three key properties: longitudinal acceleration, lateral acceleration, and jerk. A lower value indicates a higher degree of realism.

C.2 Adversarial Behavior and Collision Metrics

Collision Relative Speed. Collision Relative Speed is defined as the ego planner's speed minus the adversarial vehicle's speed at the collision timestep.

To control the relative speed, we introduce the relative speed cost function:

$$J_v = \sum_{t=1}^{T} |v_t^1 - v_t^a - v_{\text{diff}}| \cdot \mathbf{1}\{d(t) < d_{\text{col}}\}, \tag{A1}$$

where v_{diff} is the desired speed difference between the ego and the adversarial vehicles, influencing the relative speed at the point of collision. The function $\mathbf{1}\{d(t) < d_{\text{col}}\}$ is an indicator function that applies the cost only when the distance $d(t)$ between the ego and adversarial vehicle is less than a specified threshold d_{col}.

Time-to-Collision Cost. The Time to Collision (TTC) cost [22] assesses collision risk based on the relative speed and orientation between agents. For two agents located at positions (x_i, y_i) and (x_j, y_j) with respective velocities (v_{x_i}, v_{y_i}) and (v_{x_j}, v_{y_j}), we define their relative position and velocity. The relative position is given by $dx = x_i - x_j$ and $dy = y_i - y_j$, representing the positional differences along the x and y axes. Similarly, the relative velocity is calculated as $dv_x = v_{x_i} - v_{x_j}$ and $dv_y = v_{y_i} - v_{y_j}$, which are the differences in their velocities along the x and y axes. The TTC is computed under a constant velocity assumption, solving a quadratic equation to find the time of collision t_{col}, with a collision considered when relative distance is minimal.

The real part of the solution provides the time to the point of closest approach, $\tilde{t}_{\text{col}}$, calculated as:

$$\tilde{t}_{\text{col}} = \begin{cases} -\frac{dv_x dx + dv_y dy}{\tilde{dv}^2} & \text{if } \tilde{t}_{\text{col}} \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{A2}$$

and the distance at that time, $\tilde{d}_{\text{col}}$, is given by:

$$\tilde{d}^2_{\text{col}} = \begin{cases} \frac{(dv_x dy - dv_y dx)^2}{\tilde{dv}^2} & \text{if } \tilde{t}_{\text{col}} \geq 0, \\ dx^2 + dy^2 & \text{otherwise.} \end{cases} \tag{A3}$$

We define the TTC cost J_{ttc} as:

$$J_{\text{ttc}} = \sum_{t=1}^{T} -\exp\left(-\frac{\tilde{t}^2_{\text{col}(t)}}{2\lambda_t} - \frac{\tilde{d}^2_{\text{col}(t)}}{2\lambda_d} \right), \tag{A4}$$

where λ_t and λ_d are the time and distance bandwidth parameters. This cost is evaluated over a time horizon T, with a higher cost for scenarios having low time to collision and proximity. For further details on the derivation of this cost function, we direct readers to [22].

In our evaluations, we focus on the average TTC cost of 0.5 seconds preceding a collision. This metric effectively captures the criticality of the safety scenarios, reflecting the potential risk of imminent collisions.

Time-to-Collision. Additionally, we compute the average Time-to-Collision (TTC) for each timestep within the crucial 0.5-second window before collisions occur in our scenarios. It's important to note that this TTC is not the actual time until a collision, but rather a theoretical estimate based on the constant velocity model assumption for each timestep.

D Implementation Details

In this section, we discuss the implementation details of our diffusion model and the experimental settings.

D.1 Diffusion Model Training and Parameterization

The training objective is to minimize the expected difference between the true initial trajectory and the one estimated by the model, formalized by the loss function [21] [39]:

$$\mathcal{L} = \mathbb{E}_{\epsilon,k,\tau_0,c}\left[\|\tau_0 - \hat{\tau}_0\|^2\right] \tag{A5}$$

where τ_0 and $\mathbf{c}$ are sampled from the training dataset, $k \sim \mathcal{U}\{1, 2, \ldots, K\}$ is the timestep index sampled uniformly at random, and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is Gaussian noise used to perturb τ_0 to produce the noised trajectory τ_k.

In each denoising step, our model predicts the mean of the next denoised action trajectory Eq. (3). Instead of predicting the noise ϵ that is used to corrupt the trajectory [10], we directly output the denoised clean trajectory $\hat{\tau}_0$ [21] [39]. The predicted mean based on $\hat{\tau}_0$ and τ_k:

$$\tau_{k-1} = \mu_\theta(\tau_k, \hat{\tau}_0) = \frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{1-\bar{\alpha}_k}\hat{\tau}_0 + \frac{\sqrt{\alpha_k}(1-\bar{\alpha}_{k-1})}{1-\bar{\alpha}_k}\tau_k \tag{A6}$$

where β_k represents the variance from the noise schedule in the diffusion process, α_k is defined as $\alpha_k := 1 - \beta_k$, indicating the incremental noise reduction at each step, and $\bar{\alpha}_k$ is the cumulative product of α_j up to step k, mathematically expressed as $\bar{\alpha}_k = \prod_{j=0}^{k} \alpha_j$.

D.2 Diffusion Process Details

For the diffusion process, we utilize a cosine variance schedule as described in [18], with the number of diffusion steps set to $K = 100$. The variance scheduler parameters are configured with a lower bound β_1 of 0.0001 and an upper bound β_K of 0.05. The diffusion model takes in a 1-second history and is trained to predict the next 3.2 seconds with a step time $dt = 0.1$. Our model was trained on four NVIDIA RTX A6000 GPUs for 70000 iterations using the Adam optimizer, with a learning rate set to 1×10^{-5}. The diffusion model's implementation is based on methodologies from open-source repositories [10, 18], and the simulation framework is developed based on [17, 35].

D.3 Guidance details

To simultaneously incorporate multiple guidance functions in our model, we assign weights to balance their contributions. In our experiments, particularly with non-adversarial agents, we implement a combination of route guidance (J_{route}) and Gaussian collision guidance (J_{gc}) across $M = 20$ examples. Notably, we apply a filtration process exclusively to J_{gc}, aiming to prevent imminent collisions For adversarial agents, we maintain the same weighting across all guidance functions, but uniquely control the weighting for J_{ttc} to achieve controllable behavior. In this setting, we select the sample that yields the highest adversarial cost (J_{adv}), ensuring effective and targeted adversarial scenarios.

Collision Guidance: The collision guidance is based on different agent interactions. Following the methodology of [39], we extend the denoising process of all agents within a scene into the batch dimension. During inference, to generate M samples, we proceed under the assumption that each sample corresponds to the same m-th example of the scene. For the ego vehicle, the future state predictions are derived from a diffusion model identical to the one used for other agents. The collision distance for the ego vehicle is then computed considering these predictions and their interactions with other agents within the scene.

D.4 Selecting Adversarial Agents

To effectively select adversarial agents for safety-critical simulation, we developed two strategies: dynamically selecting adversarial agents or selecting interacting agents. Inspired by [26], we proposed to dynamically adjusting the weighting coefficient ρ^i of J_{adv} during the guided diffusion process, encouraging a collision by minimizing the positional distance between controlled agents and the tested ego car:

$$\rho_{i,t} = \frac{\exp(-d^{i,1}(t))}{\sum_j \exp(-d^{j,1}(t))} \tag{A7}$$

where $d^{i,1}(t)$ represents the euclidean distance between agent i and the ego vehicle at time t. Intuitively, the $\rho^{i,t}$ coefficients, defined by the softmax operation, identify a candidate agent to collide with the ego vehicle. The agent with the highest $\rho^{i,t}$ value is considered the most likely "adversary" based on proximity, and this formulation prioritizes causing a collision with this adversary. This approach weights the adversarial loss J_{adv} to highlight key interactions, preventing the unrealistic of all agents acting adversarially towards the ego vehicle.

An alternate strategy selects interacting agents as adversaries based on their lane positions relative to the ego. Agents within a certain lane proximity to the ego are randomly chosen. In this scenario, the selected ith agent is treated as $\rho^{i,t} = 1$, with all others set to zero, for the duration of the simulation.

E Experimental Settings.

We dynamically select adversarial agents as described in Eq. (A7), based on the criteria outlined in Tab. 2. In contrast, Tables 3, 4, and 5 use preselected and fixed adversarial agents. Additionally, Tables 3 and 4 focus on intersection scenarios where interactions are more involved. The selected scenarios will be available at our webpage.

F Additional Experiments

F.1 Controllability: Controlling Relative Speed.

In our safety-critical simulation framework, we examine the effects of manipulating the desired relative speed between the ego vehicle and the adversarial agent.

Rel Speed Control (m/s)	**Ego-Adv Rel Speed** (m/s)	**Coll Rate** (%) ↑	**Realism** ↓
-2.0	0.90	0.29	0.83
0.0	1.26	0.38	0.89
2.0	1.94	0.44	0.88

Table A2: Controlling relative collision speed. This table illustrates the ability of our framework to modulate the relative speed between ego and adversarial agents, influencing collision rates while maintaining realism.

Method	**Collision** (%) ↑	**Other Offroad** (%) ↓	**Other Collision** (%) ↓	**Adv Offroad** (%) ↓	**Collision Rel Speed** (m/s) ↓	**Realism** ↓
Ours	**43.2**	**1.9**	1.90	**11.4**	**-0.12**	0.38
Our (-J_{route})	38.6	5.6	2.91	15.9	1.07	**0.29**
Ours (-J_{col})	25.0	4.9	**1.41**	**11.4**	0.94	0.33

Table A3: Ablation Study for J_{reg}.

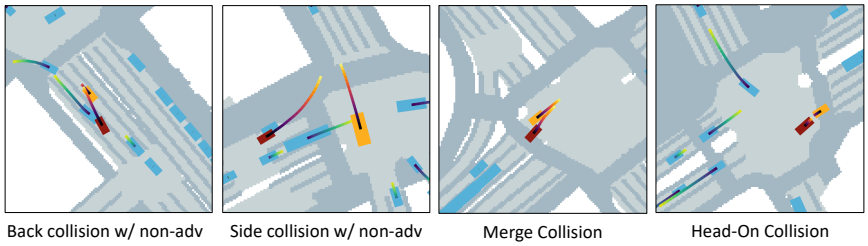

Fig. A4: Qualitative Samples of Safe-Sim Limitation and Failure Cases. In certain scenarios, the adversarial agent collides with non-adversarial agents before challenging the ego agent. Additionally, the adversarial agent may cause at-fault collisions.

As shown in Tab. A2, our proposed relative speed control results in a notable impact on both the actual ego-adversary relative speed and the collision rate. For instance, setting a lower desired relative speed target (-2.0 m/s) generally results in a decreased ego-adversary relative speed, and vice versa for a higher target (2.0 m/s). However, these adjustments do not directly translate to matching values in the simulations due to the nature of closed-loop interactions. The planner's reactive behavior to the adversarial agent's actions contributes to this discrepancy, as it may take evasive maneuvers or adjust its speed, potentially avoiding collisions altogether. Moreover, the realism metric across different relative speed settings remains relatively consistent, suggesting that the adjustments do not compromise the realism of the driving scenarios.

F.2 Ablation Study for J_{reg}.

We provide ablation study for the regularization term J_{reg}. Note, for Tab. 5 in the main paper, adversarial agents were selected before simulation based on their lane proximity to the ego. For Tab. A3, adversarial agents were selected dynamically during simulation via Eq. (A7).

F.3 Qualitative Analysis of Safe-Sim's Limitations

In Fig. A4, we present qualitative examples highlighting areas where Safe-Sim can be improved, including collisions with non-adv agents and at-fault collisions.