2025 IEEE 21st International Conference on Automation Science and Engineering (CASE)
August 17-21, 2025. Los Angeles, CA, US

2025 IEEE 21st International Conference on Automation Science and Engineering (CASE) | 979-8-3315-2246-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/CASE58245.2025.11163744

SceneDM: Consistent Diffusion Models for Coherent Multi-agent Trajectory Generation

Zhiming Guo[1], Xing Gao[2,✉], Jianlan Zhou[1], Xinyu Cai[2], Xuemeng Yang[2], Licheng Wen[2], and Xiao Sun[2]

***Abstract*— Realistic multi-agent motion simulations are essential for the advancement of self-driving algorithms. However, the majority of existing works tend to overlook the kinematic realism of the simulated motions. In this paper, we present SceneDM, a novel consistent diffusion model designed to jointly generate consistent and realistic motions for all types of agents within a traffic scene. To employ temporal dependencies and improve the kinematic realism of the generated motions, we introduce an innovative constructive noise pattern alongside smoothing regularization techniques integrated into the framework of the diffusion model. Moreover, the inference procedure of this model is tailored to effectively ensure local temporal consistency. Furthermore, a scene-level scoring function is incorporated to evaluate the safety and road adherence of the generated agents' motions, helping to filter out unrealistic simulations. Through empirical validation in the Waymo Sim Agents task, we substantiate the effectiveness of SceneDM in improving the smoothness and realism of generated agent trajectories. The project webpage is available at https://alperen-hub.github.io/SceneDM.**

I. INTRODUCTION

Traffic simulations complement real-world logged traffic scenarios, providing an economical and safe way to evaluate autonomous driving systems before their deployment in the real world. However, the generation of such scenes is non-trivial, because of (i) diverse agent types, including vehicles, pedestrians, bicycles, etc., and their complex interactions; (ii) the multi-mode nature of the generated scenes.

Several rule-based strategies [1], [2] provide some intuitive solutions but struggle to provide complex traffic scenes. Alternatively, recent works resort to deep models to handle the complexity of traffic scenes. For example, some work [3]–[5] exploit motion prediction methods to obtain future trajectories of agents. On the other hand, generative models are exploited, including Generative Adversarial Nets (GAN) based methods [6]–[9] and Variational Auto-encoder (VAE) ones [10], [11]. Besides, a variety of diffusion-based methods have been proposed recently, such as MID [12], MotionDiffuser [13], and CTG [14].

The realism of generated scenarios for simulation is crucial, encompassing both the realistic interactions between agents and the surrounding environment and the realism of agent motion. Achieving these two facets depends on the model's ability to realize spatial and temporal consistency. Specifically, temporal consistency is manifested in the continuity and smooth nature of generated trajectories, whereas spatial consistency is evidenced by agents' collision avoidance. Recently, several studies like CTG++ [15] have leveraged controllable generative models to enhance the spatial consistency of generated traffic scenes. Nonetheless, temporal consistency has been largely overlooked, potentially resulting in unrealistic motions of agents.

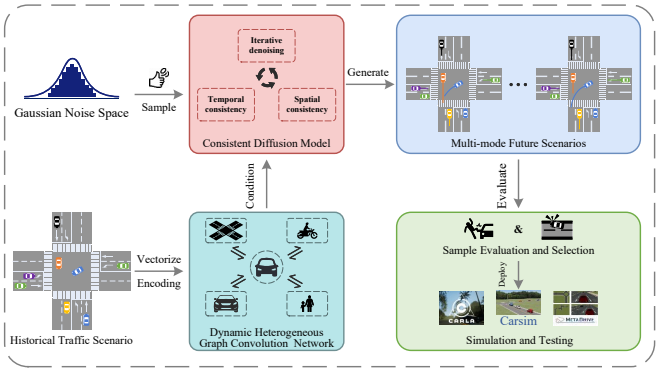


Fig. 1: High-level demonstration of the proposed SceneDM.

To tackle these problems, we propose a novel diffusion framework, termed SceneDM, for scene-level multi-agent trajectory generation, as illustrated in Fig. 1. SceneDM generates realistic and consistent future trajectories for multiple types of agents based on historical trajectories and map information.

In particular, in the context of generating locally smooth agent trajectories akin to those observed in the real world, we introduce a novel coherent diffusion model. Building on the observation that noise significantly influences the pattern of generated images in the image domain [16], we propose a constructive noise pattern within the diffusion model. Intuitively, in a smooth motion trajectory, the states of an agent at adjacent time steps exhibit similarity. To achieve this, we propose a noise pattern construction strategy to ensure the similarity of Gaussian noise samples at adjacent time steps, thereby maintaining the similarity of the states. Correspondingly, we introduce a smooth regularization to further enforce the model to produce similar states in the adjacent time at the training phase. Furthermore, we develop a temporal-consistent guidance sampling strategy to generate coherent trajectories once the diffusion model is trained with the proposed method. In the design of the denoising network, we employ alternative temporal and spatial attention modules to capture the spatial and temporal dependencies of future

[1]Huazhong University of Science and Technology, Wuhan, China. [2]Shanghai Artificial Intelligence Laboratory, Shanghai, China. This work was performed during Zhiming's internship at Shanghai Artificial Intelligence Laboratory. ✉Corresponding author, email: gxyssy@163.com.

This work was supported in part by the National Natural Science Foundation of China under Grant 62401367 and in part by Shanghai Artificial Intelligence Laboratory.

agent motions.

On the other hand, generative models are confronted with the issue of unreliability. Due to factors such as model convergence and generalization, sampled trajectories may exhibit outliers like going off-road or driving in reverse. These generated outliers, if utilized for simulating self-driving algorithms, will exert adverse effects on the algorithm's performance. Consequently, we introduce a scene-level scoring function to evaluate the compliance of generated scenarios with traffic regulations, focusing on collision avoidance and road adherence. This scoring function serves to filter out implausible simulations, thereby ensuring the practicality of the generated samples. The main contributions of SceneDM are summarized as follows:

- We propose a novel consistent diffusion model for the joint generation of trajectories for multiple agents within a traffic scene. In particular, the proposed diffusion strategy enables the model to improve the temporal consistency of generated trajectories, leading to a significant improvement in their local smoothness.
- We customize a temporal-consistent guidance sampling procedure to enhance the kinematic realism of agent trajectories during inference. Besides, we introduce a scene-level scoring function to select traffic scenarios that comply with traffic regulations, enhancing both realism and practicality in simulations.
- We validate the proposed framework in the large-scale Waymo Open Sim Agents task that consists of real-world traffic scenarios with complex interactions. The proposed SceneDM achieves excellent performance, particularly in terms of kinematic realism.

II. RELATED WORK

We provide an overview of a series of traffic simulation methods, including heuristic strategies and deep learning-based methods, as listed in Table I.

Heuristic methods. Early traffic simulation platforms like Carla [1] commonly employ hand-crafted rules [17] or heuristic strategies [1] to control the motion of agents. However, they are insufficient to emulate the complexity and realism of real-world traffic scenarios.

Motion prediction induced methods. Some recent works [3], [18], [19] exploit the results of motion prediction tasks to generate multi-mode traffic scenes. For instance, SimNet [20] models the vehicle's driving process as a Markov process and implements state distribution and transition functions with deep neural networks. BITS [21] proposes a bi-level hierarchy model to imitate driving behaviors from real-world data. TrafficSim [22] formulates a joint actor policy with an implicit latent variable model and employs GRU and CNN to learn multi-agent behaviors from real-world data. Besides, Joint-Multipath++ [23] and Trafficgen [3], derived from the motion forecasting model Multipath++ [5], utilize multi-context gating blocks to handle various interactions in observed data. Furthermore, built upon motion prediction model MTR [24], [25], MTR_E [19] introduces a collision-mitigation policy, while CAD [26] groups the agents in a scene and produces trajectories of each group with different models.

TABLE I: Comparison of traffic simulation methods.

Simulation Platform/Algorithm	Multi-type Agent	Multi-mode Sample	Temporal Consistency	Spatial Consistency	Scenario Filtering
Carsim	✗	✗	✔	✔	✗
Carla	✔	✗	✔	✔	✗
TrafficSim	✗	✔	✗	✔	✗
TrafficGen	✗	✔	✗	✔	✗
MTR_E	✔	✔	✗	✔	✗
MTG	✗	✔	✗	✔	✗
MID	✗	✔	✗	✗	✗
CTG	✗	✔	✗	✗	✔
MotionDiffuser	✔	✔	✗	✔	✗
Proposed	✔	✔	✔	✔	✔

Generative models. Another category of approaches utilizes generative models to learn the probability distribution of trajectory data and generate new trajectory samples. Some methods [8], [9] utilize Generative Adversarial Networks (GANs) to generate diverse trajectories for traffic agents. However, GANs may suffer from mode collapse and will produce unrealistic scenarios [27]. In addition, MTG [10] and CVAE-H [11] employ VAE to extract representations of historical trajectories of agents and generate future trajectories. In a more relevant work, MID [12] presents a diffusion-based framework to formulate pedestrian trajectory prediction. Besides, Scene Diffusion [28] utilizes latent diffusion in an end-to-end differentiable architecture to generate arrangements of discrete bounding boxes for agents. CTG [14], [15] develops a conditional diffusion model for controlled vehicle trajectory generation, ensuring that the generated trajectories have desired properties, such as speed limits. However, these diffusion-based models consider a specific agent type and overlook the kinematic realism of generated motions.

In contrast with these methods, we propose a consistent diffusion framework to generate trajectories with scene consistency for various types of agents in the scene. In particular, we design a novel diffusion strategy to address the temporal consistency of generated trajectories and improve the kinematic realism.

III. METHOD

A. Notions and Preliminaries

SceneDM aims to generate future trajectories for N agents in a given scenario simultaneously, leveraging both the map information and their historical trajectory data. In this paper, the current time is denoted as $t = 0$. The future trajectory of an agent is represented as $\boldsymbol{s} = \left\{\boldsymbol{y^t} \in \mathbb{R}^H | t = 1, 2, \cdots, T\right\}$, where $\boldsymbol{y^t}$ is an H-dimensional vector including 3-D coordinates and heading and T represents the length of the generated future trajectory.

Diffusion models consist of a diffusion process that gradually transforms a data distribution into unstructured noise and a reverse process to recover the data distribution [30], [31]. In this paper, we employ subscripts to indicate the step in the diffusion process and reverse process, such as original data $\boldsymbol{s_0}$ and latent variable $\boldsymbol{s_k}$. During the forward diffusion process, Gaussian noise is gradually added to the original data $\boldsymbol{s}_0$ to obtain latent variable $\boldsymbol{s}_1, \cdots, \boldsymbol{s}_K$,

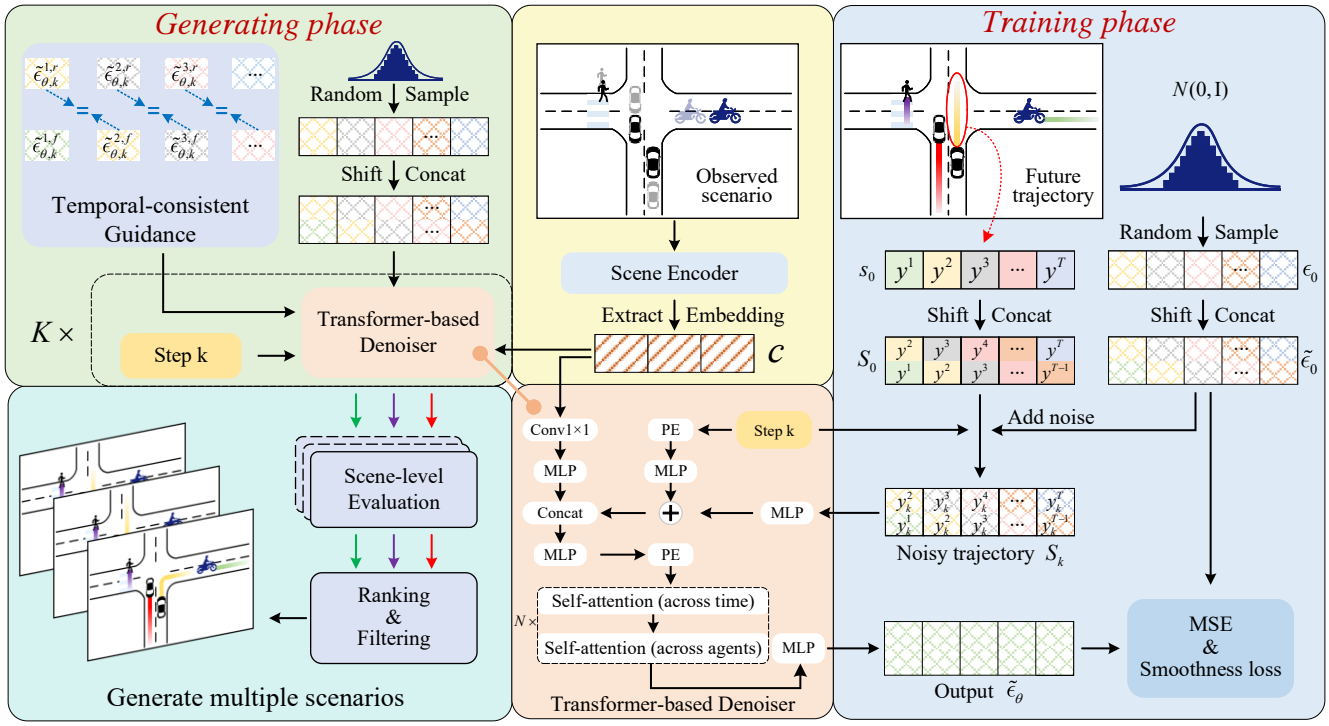


Fig. 2: The SceneDM framework comprises a scene encoder and a Transformer-based denoising network, depicted in the yellow and orange blocks, respectively. The denoising network conditions on the agent embedding from the scene encoder to eliminate noise from the noisy trajectory. During the training of the proposed consistent diffusion model, a new constructive noise pattern is demonstrated in the blue block. To further enhance the smoothness of generated trajectories, a smoothness regularization is introduced. During the generating phase, we customize the DDIM [29] algorithm with temporal-consistent guidance for ensuring temporal consistency, as illustrated in the green block. Finally, a scene-level scoring module helps to filter out unrealistic generations, enhancing the practicality of the generated samples in the simulation.

where K denotes the maximum number of diffusion steps. According to DDPM [31], the forward diffusion process is parameterized as a Markov chain, with the final variable,

$$\boldsymbol{s}_K = \sqrt{\bar{\alpha}_K}\boldsymbol{s}_0 + \sqrt{(1-\bar{\alpha}_K)}\boldsymbol{\epsilon}, \tag{1}$$

where $\bar{\alpha}_K$ is a positive constant representing the noise level and $\boldsymbol{\epsilon}$ denotes the noise sampled from the standard Gaussian distribution $\mathcal{N}(0, \mathrm{I})$. When K is sufficiently large, $\boldsymbol{s}_K$ converges to the Gaussian distribution.

Diffusion models provide parameterized Gaussian transitions to model the reverse process. Given diffusion step $1, \cdots, K$ and the condition $\boldsymbol{c}$, diffusion models formulate the reverse process as follows:

$$p_\theta\left(\boldsymbol{s}_{0:K} \mid \boldsymbol{c}\right) = p\left(\boldsymbol{s}_K \mid \boldsymbol{c}\right) \prod_{k=1}^{K} p_\theta\left(\boldsymbol{s}_{k-1} \mid \boldsymbol{s}_k, \boldsymbol{c}, k\right), \tag{2}$$

$$p_\theta\left(\boldsymbol{s}_{k-1} \mid \boldsymbol{s}_k, \boldsymbol{c}, k\right) = \mathcal{N}\left(\boldsymbol{\mu}_\theta\left(\boldsymbol{s}_k, \boldsymbol{c}, k\right), \boldsymbol{\Sigma}_\theta\left(\boldsymbol{s}_k, \boldsymbol{c}, k\right)\right), \tag{3}$$

where $p\left(\boldsymbol{s}_K \mid \boldsymbol{c}\right) = p\left(\boldsymbol{s}_K\right) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and the θ indicates the parameters of the entire framework.

Diffusion models [31] are optimized to approximate $p_\theta\left(\boldsymbol{s}_{k-1} \mid \boldsymbol{s}_k, \boldsymbol{c}, k\right)$ or equivalently predict the added noise $\boldsymbol{\epsilon}$ in the diffusion process, in accordance with the objective function

$$L_{mse} = \mathbb{E}_{\boldsymbol{\epsilon}, \boldsymbol{s}_0}\left\|\epsilon_\theta(\boldsymbol{s}_k, k, \boldsymbol{c}) - \boldsymbol{\epsilon}\right\|, \tag{4}$$

with $\epsilon_\theta(\boldsymbol{s}_k, k, \boldsymbol{c})$ representing the predicted noise.

B. Framework

The framework of SceneDM consists of a scene encoder to learn vectorized representations of dynamic scenarios and a designed decoder for the reverse diffusion process, as illustrated in Fig. 2. The scenario encoder encodes scene elements such as road networks and agent historical trajectories into a set of latent embeddings. The embeddings of agents are then fed into the Transformer-based decoder as the condition in the reverse process.

We adopt the heterogeneous graph convolutional recurrent network, proposed in [32], as the scene encoder. Following [32], each scenario is modeled as a dynamic heterogeneous graph where different types of nodes, including lane nodes and agent nodes, represent distinct scene elements and dynamic edges capture the evolution of various interactions. Specifically, lane-lane edges are connected following the road topology, while agent-agent and agent-lane edges are dynamically linked according to their distance from agents' positions at specific time. Categorical embeddings are further applied to agent nodes to distinguish various types of agents. Road information and diverse agent features are fused with the heterogeneous graph convolution jointly yet differently. The produced d-dimensional embedding for each agent en-

codes its motion characteristics and context information. Embeddings of all the N agents are indicated with tensor $\boldsymbol{c} \in \mathbb{R}^{N\times d}$.

In the design of the decoder, we adopt an attention mechanism to handle multi-agent interactions and temporal dependencies of trajectories. The temporal attention enables the model to learn continuous trajectories over time. Meanwhile, the spatial attention permits the model to capture agent-agent interactions and generate consistent trajectories, *i.e.,* without collision. Specifically, we take the alternating temporal attention layer and spatial attention layers as a basic module, similarly to [33]. Multiple such modules are stacked to process the complex interactions in the denoising process.

Specifically, as illustrated in the orange block in Fig. 2, the condition $\boldsymbol{c}$, which encapsulates distinct motion characteristics and individual context information of all agents, expands an additional dimension for temporal features through 1D convolution, *i.e.,* $\boldsymbol{c} \in \mathbb{R}^{N\times T\times d}$. Meanwhile, for each diffusion step k, we encode the diffusion step and noisy variable $\boldsymbol{s_k}$ through multi-layer perceptrons (MLPs). These embeddings are concatenated with the condition $\boldsymbol{c}$, resulting in a fused feature ($\mathbb{R}^{N\times T\times 2d}$). To highlight the positional relationships of the sequence data, we further impose positional encoding on the fused feature. This feature is then fed into the transformer that is composed of attention layers across time and agent alternatively. Finally, an MLP based on the features of the transformer produces the noise to be removed.

C. Consistent Diffusion Model

In real-world scenarios, agents exhibit smooth and continuous motion patterns as they navigate through their environment. Therefore, local smoothness is an important factor in evaluating the reality of generated trajectories. To address it, we propose a novel consistent diffusion model to generate smooth and realistic trajectories by imposing the same noise on the overlapping parts of adjacent elements.

Diffusion process. As shown in Figure 2, we firstly augment the trajectory sequence $\boldsymbol{s_0} = [\boldsymbol{y^1}, \boldsymbol{y^2}, \cdots, \boldsymbol{y^T}]$ to make adjacent elements have overlapping parts. Specifically, for each state $\boldsymbol{y^t}$ in the trajectory $\boldsymbol{s_0} = [\boldsymbol{y^1}, \boldsymbol{y^2}, \cdots, \boldsymbol{y^T}]$, we concatenate it with the state $\boldsymbol{y^{t+1}}$ at the subsequent moment. By concatenating the states of adjacent frames, we create an augmented variable $\tilde{\boldsymbol{y}}^{\boldsymbol{t}} \in \mathbb{R}^{2H}$ that incorporates information from both $\boldsymbol{y^t}$ and $\boldsymbol{y^{t+1}}$, providing an informative input for subsequent processing. Most importantly, $\tilde{\boldsymbol{y}}^{\boldsymbol{t}}$ overlaps partly with both $\tilde{\boldsymbol{y}}^{\boldsymbol{t-1}}$ and $\tilde{\boldsymbol{y}}^{\boldsymbol{t+1}}$. Correspondingly, we obtain an augmented sequence of trajectory states, denoted as $\boldsymbol{S_0} = [\tilde{\boldsymbol{y}}^{\boldsymbol{1}}, \tilde{\boldsymbol{y}}^{\boldsymbol{2}}, \cdots, \tilde{\boldsymbol{y}}^{\boldsymbol{T-1}}]$. We then gradually add Gaussian noise to $\boldsymbol{S_0}$ to obtain $\boldsymbol{S_k}$. To maintain the consistency of the state information at the same timestamp within the noisy trajectory $\boldsymbol{S_k}$, we introduce **a constructive noise pattern** by applying the same augmentation approach to the sampled noise. Specifically, we first sample noise $\boldsymbol{\epsilon_0} = [\boldsymbol{\epsilon_0^1}, \boldsymbol{\epsilon_0^2}, \cdots, \boldsymbol{\epsilon_0^T}]^T \in \mathbb{R}^{T\times H}$ from $\mathcal{N}(0, \mathrm{I})$. Then $\boldsymbol{\epsilon_0}$ is shifted and concatenated to obtain the augmented noise sequence $\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{0}} \in \mathbb{R}^{T-1\times 2H}$ for $\boldsymbol{S_0}$. In other words,

$$\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{0}}^{\boldsymbol{t}} = \mathrm{Concat}(\boldsymbol{\epsilon_0^t}, \boldsymbol{\epsilon_0^{t+1}}), t = 1, 2, \ldots, T-1, \quad (5)$$

$$\boldsymbol{S_K} = \sqrt{\bar{\alpha}_K}\boldsymbol{S_0} + \sqrt{(1-\bar{\alpha}_K)}\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{0}}. \quad (6)$$

Reverse process. Together with the condition $\boldsymbol{c}$ from encoder, $\boldsymbol{S_k}$ is passed through the Transformer denoiser to predict the noise $\tilde{\epsilon}_\theta(\boldsymbol{S_k}, k, \boldsymbol{c}) \in \mathbb{R}^{T-1\times 2H}$ to be removed. Similar to DDPM [31], the model is optimized as:

$$L_{mse} = \mathbb{E}_{\boldsymbol{\epsilon}, \boldsymbol{S}_0} \left\| \tilde{\boldsymbol{\epsilon}}_{\boldsymbol{0}} - \tilde{\epsilon}_\theta(\boldsymbol{S_k}, k, \boldsymbol{c}) \right\|, \quad k = 1, 2, \ldots, K. \quad (7)$$

Besides, we introduce a regularization to further improve smoothness. The smoothness is calculated as the difference between adjacent motion states of the original sequence data, and equivalently the difference between the front and rear halves of the element of the augmented sequence. For example, with the motion state indicating the linear velocity of the agent, the difference reflects the linear acceleration. The smoothness loss term regularizes the difference of the generated samples to approximate that of the ground truth observed in the real-world. Mathematically,

$$L_{smooth} = \mathbb{E}_{\boldsymbol{\epsilon}, \boldsymbol{S}_0, k, t} \left\| (\boldsymbol{\epsilon_0^{t+1}} - \boldsymbol{\epsilon_0^t}) - (\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta,k}}^{\boldsymbol{t,r}} - \tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta,k}}^{\boldsymbol{t,f}}) \right\|, \quad (8)$$

where we use $\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta,k}}^{\boldsymbol{t,f}}$ and $\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta,k}}^{\boldsymbol{t,r}}$ to indicate the first and rear part of each predicted noise $\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta,k}}^{\boldsymbol{t}}$, respectively.

Combining the aforementioned two losses, we define a novel hybrid optimization objective, $L_{hybrid} = L_{mse} + \lambda L_{smooth}$. By introducing the smoothness regularization term, the model is encouraged to generate trajectories that exhibit realistic and smooth motion patterns. Both parameters of the scene encoder and the Transformer-based denoiser are trained simultaneously. The hyperparameter λ is used to adjust the balance between these two losses, with $\lambda = 1$ adopted in the experiments.

D. Temporal-consistent Guidance Sampling

We present here the temporal-consistent guidance sampling strategy for the consistent diffusion model, generalized from DDIM. During the generating phase, the same constructive noise pattern as the training phase is adopted. Specifically, we first sample noise sequence from the standard Gaussian distribution $\mathcal{N}(\mathbf{0}, I)$ and then perform shifting and concatenation operations to the sampled noise sequence to obtain the initial noisy trajectory sequence $\boldsymbol{S_K} \in \mathbb{R}^{T-1\times 2H}$. At each step of the denoising process, the information corresponding to the same state of the original sequence data is forced to be consistent, like $\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta,k}}^{\boldsymbol{t-1,r}}$ and $\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta,k}}^{\boldsymbol{t,f}}$, through averaging them:

$$\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta,k}}^{\boldsymbol{t-1,r}}, \tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta,k}}^{\boldsymbol{t,f}} \leftarrow \mathrm{Mean}(\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta,k}}^{\boldsymbol{t-1,r}}, \tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta,k}}^{\boldsymbol{t,f}}), \quad t = 2, 3, \ldots, T. \quad (9)$$

During the sampling process, SceneDM refines and generates the trajectories by iterative computing transitions from $k = K$ to $k = 0$ as follows:

$$\boldsymbol{S_{k-1}} = \sqrt{\frac{\bar{\alpha}_{k-1}}{\bar{\alpha}_k}}\boldsymbol{S_k} + \sqrt{1-\bar{\alpha}_{k-1}} \cdot \tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta,k}} - \sqrt{\frac{\bar{\alpha}_{k-1}(1-\bar{\alpha}_k)}{\bar{\alpha}_k}} \cdot \tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta,k}}, \quad (10)$$

TABLE II: Results on the waymo sim agents benchmark v0. Higher values indicate better performance, with the top two results indicated in **bold** and underlined.

	Meta	Kinematic Metric					Interactive Metric				Map Metric		
	Realism	Overall	Linear Speed	Linear Acc.	Angel Speed	Angel Acc.	Overall	Dist To Object	Coll.	Time To Coll.	Overall	Dist To Roadedge	Offroad
SceneDMF	0.506	**0.424**	0.432	**0.277**	0.523	0.467	0.526	0.368	**0.462**	0.813	0.606	0.621	0.598
SceneDM	0.500	**0.424**	0.432	0.276	0.523	0.467	0.517	0.365	0.447	0.810	0.596	0.619	0.584
MVTE [34]	**0.517**	0.420	0.443	0.222	**0.535**	**0.481**	**0.529**	**0.382**	0.451	**0.832**	**0.649**	**0.664**	**0.641**
MVTA [34]	0.509	0.418	0.437	0.220	0.533	**0.481**	0.519	0.373	0.436	0.830	0.637	0.655	0.629
MTR_E [19]	0.491	0.418	0.428	0.235	0.534	0.475	0.491	0.346	0.409	0.798	0.607	0.654	0.584
Joint-Multipath++ [23]	0.489	0.407	0.432	0.230	0.515	0.452	0.499	0.344	0.420	0.813	0.602	0.639	0.583
Wayformer [35]	0.472	0.361	0.408	0.127	0.473	0.437	0.494	0.358	0.403	0.810	0.608	0.645	0.589
MTR+++ [19]	0.470	0.360	0.412	0.107	0.484	0.437	0.493	0.346	0.414	0.797	0.603	0.655	0.577
CAD [26]	0.432	0.336	0.346	0.253	0.433	0.311	0.436	0.330	0.311	0.789	0.572	0.638	0.540
QCNeXt [18]	0.392	0.311	**0.477**	0.242	0.325	0.199	0.445	0.376	0.324	0.757	0.443	0.610	0.360
Constant Velocity	0.238	0.047	0.074	0.058	0.019	0.035	0.337	0.208	0.202	0.737	0.368	0.454	0.325

where $\bar{\alpha}_k$ represent the noise levels at diffusion step k. By iteratively applying the transitions, the sampling process gradually removes the noise and generates plausible future trajectories.

E. Scene-level Scoring Module

Generative models may produce unrealistic scenes or traffic-rule violation ones, such as agents colliding or going out of the road boundary. Such data may adversely affect the subsequent simulation task for autonomous driving. To address this issue, we propose a scoring module to assess the generated scenes at a scene-level. It consists of safety verification and road-adherence measurements. Specifically, for each generated candidate trajectory $\boldsymbol{s_i}$, we compute its overlap with the trajectories of other agents in the scene in parallel and denote the number of collisions as $r_1(s_i)$. Similarly, the road-adherence is measured by $r_2(s_i)$, which represents the number of times the agent goes out of the road boundary. The final trajectory scoring function for $\boldsymbol{s_i}$ is obtained by:

$$F(s_i) = r_1(s_i) + r_2(s_i). \tag{11}$$

By calculating the average $F(\boldsymbol{s_i})$ of all the N agents, we obtain the scenario score. Generated scenarios are ranked and selected accordingly.

IV. Experiment and Results

A. Dataset and Metrics

We use the Waymo Open Motion Dataset in our experiment. In each scenario, there are a maximum of 128 agents, consisting of three types: vehicles, bicycles, and pedestrians. The dataset provides the historical trajectory information of these agents for a duration of 1.1 seconds, including 3D coordinates, heading, vehicle velocity, agent type, *etc*. The sim agents task requires simulating 32 future trajectories for each agent in the scene, generating their motions for the upcoming 8-second duration, including the centroid coordinates and heading of each agent. Unlike motion prediction task that measures displacement errors, this benchmark evaluates the distribution similarity between the generated trajectories and real-world data through kinematic, interactive, and map-based metrics. These metrics collectively form the meta-metric realism, as elaborated in [36].

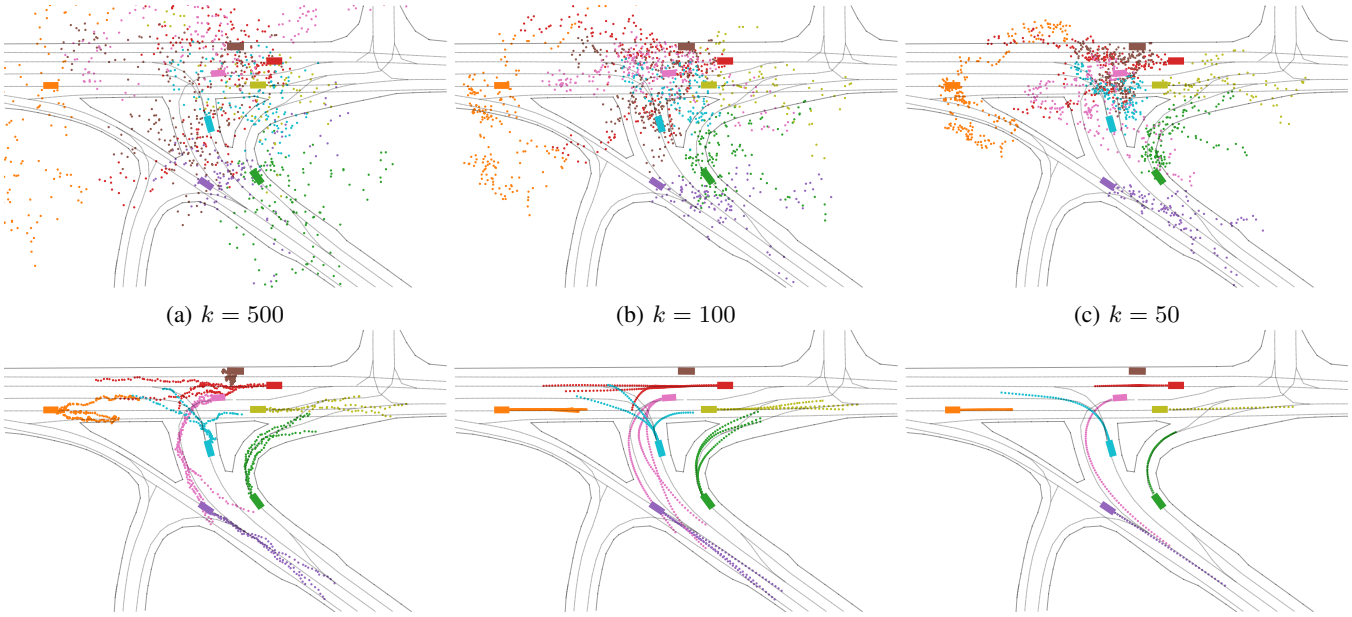
B. Implementation Details

We utilize a scene-centric coordinate system where all agents within the scene share the same coordinate system. We take the location of the autonomous vehicle at $t = 0$ as the origin, and adopt its current driving direction as x-axis. Instead of directly generating 3D coordinates of agents, we choose to generate agents' velocities and integrate them to generate trajectories. Within the decoder, we perform six iterations of the attention mechanism, alternating between the time dimension and the agent dimension. The embedding dimension in the decoder is set to 512. During the training phase, the initial learning rate is set to 0.0001 and decreases with the step decay learning rate scheme. Finally, we denote the variants of the proposed method, with and without the Scene-level Scoring Module (introduced in Sec. III E), as SceneDMF and SceneDM, respectively.

C. Results and Analysis

Quantitative results. We quantitatively compare our method with a wide range of methods. As shown in Table II, the proposed models reach a realism meta-metric of 0.506, and achieve the highest scores in several metrics, such as the overall kinematic metric. Notably, with velocity as the motion state, the proposed method realizes smooth states of generated sequence data with the best linear acceleration metric. As shown in Table II and III, SceneDMF further improves the performance by filtering scenarios with the scoring module, especially in terms of the collision metric. However, we also observe that SceneDM(F) has potential for further enhancement, such as improving road network embedding to better interact with maps.

Qualitative results. In Figure 3, we illustrate the dynamic generation process of a scenario. The process commences with the sampling of Gaussian noise when $k = 500$. During the denoising process, SceneDM progressively eliminates the noise and reduces the trajectory uncertainty. It ultimately converges when $k = 0$, yielding trajectories that adhere to the distribution of real-world data. To provide a concise representation, we randomly select three distinct trajectories from the set of 32 generated future trajectories for display. As demonstrated in Figure 3, 1) when the agent goes on the straight lane, SceneDM captures diverse speed modes and is



(a) $k = 500$ (b) $k = 100$ (c) $k = 50$

(d) $k = 10$ (e) $k = 0$ (f) Ground truth

Fig. 3: The qualitative results of the generation process are demonstrated for three samples. Initially, SceneDM samples noise from a standard Gaussian distribution at $k = 500$ and progressively eliminates trajectory noise until convergence at $k = 0$.

TABLE III: Ablation results on 1000 scenarios randomly sampled from the validation set of the waymo motion dataset.

Variants	Agent Int.	Sequence Augm.	Noise const.	Smooth loss	Const. guidance	Comp. filter	Kinematic Metric	Interactive Metric	Map Metric	Realism
Baseline							0.4108	0.5052	0.5899	0.4891
-	✓						0.4159	0.5161	0.5964	0.4965
-	✓	✓					0.3904	0.5047	0.5408	0.4682
-	✓	✓	✓	✓			0.4209	0.5181	0.5863	0.4963
SceneDM	✓	✓	✓	✓	✓		0.4243	0.5174	0.5894	0.4982
SceneDMF	✓	✓	✓	✓	✓	✓	0.4245	0.5257	0.5959	0.5030

capable of lane changing; 2) On an intersection, SceneDM generates multi-mode trajectories, including going straight, turning left, and turning right, *etc.* 3) For turning trajectories, SceneDM covers diverse turning radii, consistent with those in the real world. These observations demonstrate that SceneDM effectively models the multi-mode characteristics of the agents and produces smooth and realistic motions. More qualitative results, like interactions between multiple types of agents, are further provided on the Project Webpage.

D. Ablation Studies

We further study the different module proposed in this paper. We randomly sampled 1,000 scenarios from the validation set to conduct ablation experiments. As shown in Table III, the baseline model, which is composed of just temporal attention layers and ablates the proposed consistent training and sampling, performs strongly but degrades significantly compared with the proposed SceneDM. With agent-wise self-attention layers introduced into the denoiser architecture, the performance is enhanced, especially in terms of the interactive metric. Merely augmenting sequence data does not improve performance. However, when combined with the proposed constructive noise pattern and smooth regularization, performance is significantly improved. For instance, the kinematic metric is enhanced from 0.3904 to 0.4209, and further to 0.4243 with the temporal-consistent guidance sampling. These demonstrate that maintaining temporal consistency within the noisy trajectory enhances the model's ability to learn short-term temporal dependencies and generate smooth trajectories. Finally, with the scene-level scoring module, SceneDMF achieves increment in terms of both interactive and map-based metrics, indicating the improvement on the traffic rule compliance.

V. Conclusion

In this paper, we propose a diffusion based multi-agent trajectory generation framework, called SceneDM, to jointly produce coherent motions of various types of agents in a traffic scene. In particular, we design an effective **consistent diffusion model** through introducing a novel constructive noise pattern coupled with smooth regularization techniques into DDPM. Furthermore, we customize the fast sampling algorithm DDIM with the proposed temporal-consistent guidance. These designs permit the model to effectively capture the temporal consistency of generated trajectories, resulting in **smooth and realistic motion patterns**. Furthermore,

we propose a plug-and-play scene-level evaluation module to enhance spatial consistency and road adherence of the generated scenarios. SceneDM achieves excellent performance, especially in terms of kinematic realism, in the challenging waymo sim agents task. In the future, it is worth exploring controllable generation, for example generating safety-critical traffic scenarios.

REFERENCES

[1] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on Robot Learning*, pages 1–16, 2017.

[2] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using SUMO. In *21st international conference on intelligent transportation systems (ITSC)*, pages 2575–2582, 2018.

[3] Lan Feng, Quanyi Li, Zhenghao Peng, Shuhan Tan, and Bolei Zhou. Trafficgen: Learning to generate diverse and realistic traffic scenarios. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3567–3575, 2023.

[4] Shuhan Tan, Kelvin Wong, Shenlong Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. SceneGen: Learning to generate realistic traffic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 892–901, 2021.

[5] Balakrishnan Varadarajan, Ahmed Hefny, Avikalp Srivastava, Khaled S Refaat, Nigamaa Nayakanti, Andre Cornman, Kan Chen, Bertrand Douillard, Chi Pang Lam, Dragomir Anguelov, et al. MultiPath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7814–7821, 2022.

[6] Martin Håkansson and Joel Wall. Driving scenario generation using generative adversarial networks. 2021.

[7] Raunak Bhattacharyya, Blake Wulfe, Derek J Phillips, Alex Kuefler, Jeremy Morton, Ransalu Senanayake, and Mykel J Kochenderfer. Modeling human driving behavior through generative adversarial imitation learning. *IEEE Transactions on Intelligent Transportation Systems*, 24(3):2874–2887, 2022.

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

[9] Zhao-Heng Yin, Lingfeng Sun, Liting Sun, Masayoshi Tomizuka, and Wei Zhan. Diverse critical interaction generation for planning and planner evaluation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7036–7043, 2021.

[10] Wenhao Ding, Wenshuo Wang, and Ding Zhao. Multi-vehicle trajectories generation for vehicle-to-vehicle encounters. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[11] Geunseob Oh and Huei Peng. CVAE-H: Conditionalizing variational autoencoders via hypernetworks and trajectory forecasting for autonomous driving. *arXiv preprint arXiv:2201.09874*, 2022.

[12] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[13] Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, Dragomir Anguelov, et al. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9644–9653, 2023.

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

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

[16] Weifeng Chen, Jie Wu, Pan Xie, Hefeng Wu, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video generation with diffusion models. *arXiv preprint arXiv:2305.13840*, 2023.

[17] Rahim F Benekohal and Joseph Treiterer. CARSIM: Car-following model for simulation of traffic in normal and stop-and-go conditions. *Transportation Research Record*, 1194:99–111, 1988.

[18] Zikang Zhou, Zihao Wen, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. QCNeXt: A next-generation framework for joint multi-agent trajectory prediction. *arXiv preprint arXiv:2306.10508*, 2023.

[19] Cheng Qian, Di Xiu, and Minghao Tian. The 2nd place solution for 2023 waymo open sim agents challenge. *arXiv preprint arXiv:2306.15914*, 2023.

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

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

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

[23] Wenxi Wang and Haotian Zhen. Joint-Multipath++ for simulation agents. *Technical report*, 2023.

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

[25] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. MTR++: Multi-agent motion prediction with symmetric scene modeling and simnet intention querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3955–3971, 2024.

[26] Hsu-kuang Chiu and Stephen F Smith. Collision avoidance detour for multi-agent trajectory forecasting. *arXiv preprint arXiv:2306.11638*, 2023.

[27] Ruochen Jiao, Xiangguo Liu, Bowen Zheng, Dave Liang, and Qi Zhu. TAE: A semi-supervised controllable behavior-aware trajectory generator and predictor. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12534–12541, 2022.

[28] Ethan Pronovost, Kai Wang, and Nick Roy. Generating driving scenes with diffusion. *arXiv preprint arXiv:2305.18452*, 2023.

[29] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021,*, Virtual Event, Austria, May 2021.

[30] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265, 2015.

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

[32] Xing Gao, Xiaogang Jia, Yikang Li, and Hongkai Xiong. Dynamic scenario representation learning for motion forecasting with heterogeneous graph convolutional recurrent networks. *IEEE Robotics and Automation Letters*, 8(5):2946–2953, 2023.

[33] Jiquan Ngiam, Vijay Vasudevan, Benjamin Caine, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene Transformer: A unified architecture for predicting future trajectories of multiple agents. In *International Conference on Learning Representations*, 2021.

[34] Yu Wang, Tiebiao Zhao, and Fan Yi. Multiverse Transformer: 1st place solution for waymo open sim agents challenge 2023. *arXiv preprint arXiv:2306.11868*, 2023.

[35] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2980–2987, 2023.

[36] Nico Montali, John Lambert, Paul Mougin, Alex Kuefler, Nicholas Rhinehart, Michelle Li, Cole Gulino, Tristan Emrich, Zoey Yang, Shimon Whiteson, et al. The waymo open sim agents challenge. In *Advances in Neural Information Processing Systems*, volume 36, pages 59151–59171, 2023.