SceneControl: Diffusion for Controllable Traffic Scene Generation

Jack Lu[3⋆], Kelvin Wong[1,2⋆], Chris Zhang[1,2], Simon Suo[2], and Raquel Urtasun[1,2]

***Abstract*— We consider the task of traffic scene generation. A common approach in the self-driving industry is to use manual creation to generate scenes with specific characteristics and automatic generation to generate canonical scenes at scale. However, manual creation is not scalable, and automatic generation typically use rules-based algorithms that lack realism. In this paper, we propose SceneControl, a framework for *controllable traffic scene generation*. To capture the complexity of real traffic, SceneControl learns an expressive diffusion model from data. Then, using guided sampling, we can flexibly control the sampling process to generate scenes that exhibit desired characteristics. Our experiments show that SceneControl achieves greater realism and controllability than the existing state-of-the-art. We also illustrate how SceneControl can be used as a tool for interactive traffic scene generation.**

I. Introduction

Simulation is an essential tool to safely and scalably develop and deploy self-driving vehicles (SDVs). A core component of simulation is the ability to simulate realistic traffic scenarios. This is typically decomposed into two tasks: (1) specifying the initial placement and attributes (*e.g.*, size, speed, *etc.*) of the actors in a scene; and (2) unrolling a policy to simulate their behaviors. We focus on the first task—*traffic scene generation*. This task is challenging because we need to generate scenes that capture the nuanced interactions between actors in the real world. We also need to do so scalably, across a diversity of road topologies, and realistically, so that the simulation accurately reflects what may occur in the real world; *e.g.*, the actors' initial kinematics should not induce inevitable collisions when unrolling the simulation.

A common solution is to ask human experts to manually create *specific* traffic scenes and use automated algorithms to generate *canonical* traffic scenes at scale. For manual creation, test engineers manually specify each actor's placement and attributes by hand. This granular level of editing allows them to create scenes with specific interactions between actors. However, it is tedious and time-consuming to create realistic scenes in this way, limiting its scalability. For automatic generation, rules-based methods [33], [32], [15], [22] are used to generate scenes at scale with heuristics like "vehicles drive on lane centerlines"; but designing these rules requires significant time and expertise, and their rigidity makes it challenging to generate realistic scenes that cover the diversity of situations that can occur in the real world.

To address the limitations of rules-based methods, recent work [29], [8], [2], [37] use generative modeling to learn to generate traffic scenes instead. By learning the distribution of traffic scenes from data, these methods avoid the need to design complex heuristics, enabling them to better model the diversity and complexity of real traffic. However, these methods still generate scenes that defy common sense; *e.g.*, frequently generating scenes with colliding or off-road actors. Moreover, they focus on generating unconditional scenes and neglect the use case of generating scenes with specific semantic characteristics. This limits the scope of their usefulness to generating canonical traffic scenes.

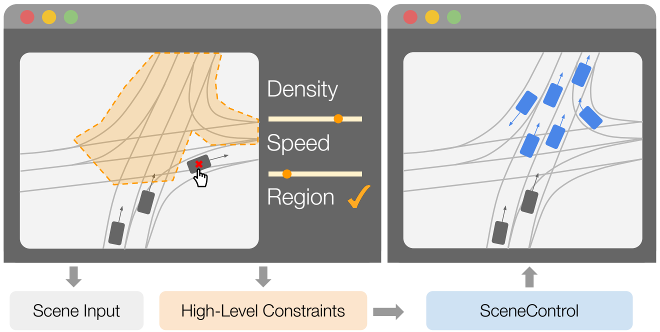

Fig. 1. SceneControl is a framework for controllable traffic scene generation that can generate realistic, constraint-satisfying traffic scenes.

In this paper, we aim to bridge the gap between manual creation of specific scenes and automatic generation of canonical scenes. We want a framework for *controllable traffic scene generation*, where engineers or machines could generate unconditional scenes from scratch or use high-level constraints to create scenes that exhibit specific characteristics; *e.g.* adding dense traffic to an intersection. This would enable us to controllably generate diverse scene variations at scale. At the same time, it would greatly improve the efficiency and realism of human-in-the-loop scene creation.

Towards this goal, we propose SceneControl, a framework for controllable traffic scene generation. SceneControl first learns a diffusion model [26], [11] of traffic scenes. Specifically, SceneControl combines a transformer decoder [31] with a lane graph neural network [18], [4] to efficiently model actor-to-actor/map interactions. Then, at generation time, we represent high-level constraints with guidance functions and use guided sampling [6] to sample realistic, constraint-satisfying traffic scenes from the diffusion model. Notably, this approach allows us to flexibly incorporate any constraints (which may be unknown during training) into the sampling process without re-training. We leverage this controllability for two purposes: (1) to generate scenes that exhibit common sense realism (*e.g.*, collision-free); and (2) to generate scenes that satisfy user-specified high-level constraints (*e.g.*, on the actors' locations, speeds, *etc.*).

We evaluate SceneControl's realism and controllability on

⋆Indicates equal contribution. [1]Waabi, [2]University of Toronto, [3]New York University. Research done at Waabi. Contact: `kwong@waabi.ai`

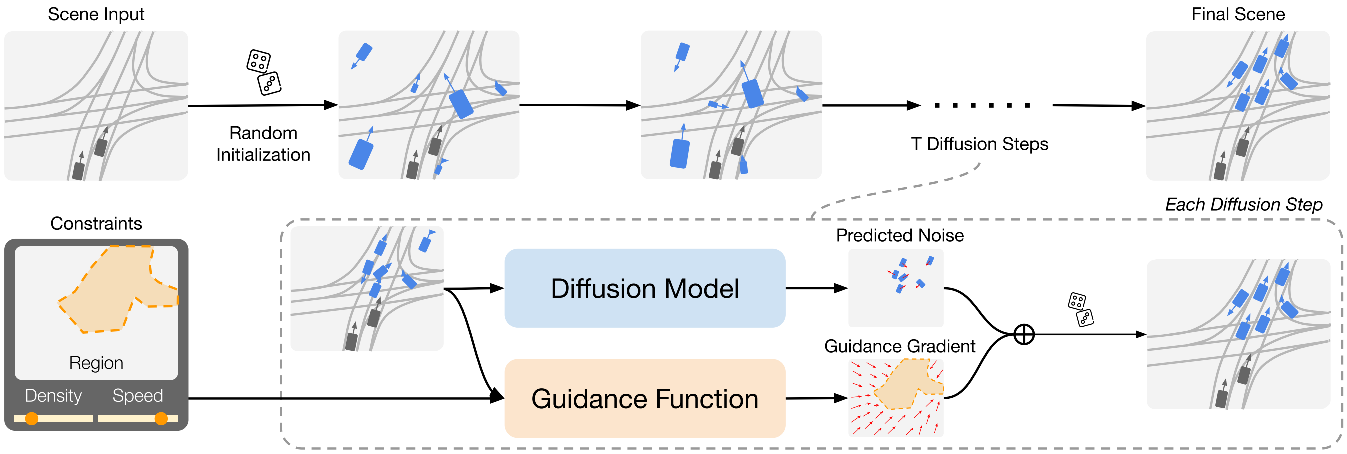

Fig. 2. SceneControl is a framework for controllable traffic scene generation. To generate a traffic scene, we first sample a random noise vector per actor and then iteratively refine them into a realistic traffic scene over T diffusion steps. At each diffusion step, we combine a learned diffusion model with a guidance function to de-noise the input scene into one that is more realistic and better satisfies our desired constraints.

two datasets consisting of urban and high traffic scenes. Our experiments show that SceneControl generates more realistic traffic scenes than prior work. At the same time, SceneControl can generate traffic scenes that better satisfy high-level constraints on location, size, and speed, achieving a greater degree of realism and controllability than prior work evaluated in this setting. Finally, leveraging these properties, we illustrate how SceneControl can be used to build a tool for interactive traffic scene generation (see Fig. 1).

II. Related Work

Traffic scenes for self-driving simulation: A core component of simulation is the ability to simulate realistic traffic scenarios. Traffic scenes specify the initial placement and attributes for each actor in these scenarios; *e.g.*, its pose, bounding box, speed, *etc*. In the self-driving industry, traffic scenes are typically manually created by test engineers using tools like domain specific languages [9], [25], [10] or graphical editors [19], [13], [7]. This low-level approach gives them control to create specific scenes but it is also inefficient and not scalable. In this work, we aim to improve scalability through controllable traffic scene generation, enabling engineers or machines to use high-level constraints to generate realistic scenes that exhibit desired characteristics.

Generative models of traffic scenes: Learning generative models of traffic scenes from data is a promising approach to scalable traffic scene generation. Early work use Bayesian networks to model the distribution of traffic scenes in highway [33], [32] and urban [15] maps. Recent work use deep generative modeling to further improve realism. Scene graph-based models [16], [5] learn generative models over hierarchical scene graph representations of traffic scenes; but its hand-crafted scene grammar limits its expressivity to capture the diversity of real traffic. A more flexible approach uses generative adversarial networks [2] or normalizing flows [37] to model all actors in the scene jointly. However, they have not been evaluated beyond a few qualitative demonstrations [2] or roundabout maps with few actors [37]. Autoregressive models [29], [8] have shown the best sample quality to date. These models factorize the generation process into an iterative one, inserting actors one by one according to a fixed order. Recent work in indoor scene synthesis [21] relaxes the fixed order assumption but have not been shown to work well in more complex traffic scenes.

While promising, existing methods generate scenes that defy common sense; *e.g.*, with collision or off-road actors. Most also focus on generating scenes unconditionally [2], [37], [29], [8]. LCTGen [28] is a concurrent work that uses GPT-4 [20] to generate structured scene descriptions to condition TrafficGen [8]. In contrast, we control our diffusion model via guided sampling, bypassing LCTGen's limited scene grammar and allowing us to use arbitrary constraints, including common sense ones that improve realism.

Diffusion models: A diffusion model [11], [26] models the data-generating process as an iterative de-noising procedure. Starting from random noise, a diffusion model progressively refines the noise into a sample from the data distribution. This iterative refinement reverses a forward diffusion process, where data is iteratively corrupted with noise until it is indistinguishable from noise. An important characteristic of diffusion models is that they admit flexible mechanisms for conditional sampling via *guided sampling* [6], [12]. Guided sampling biases the reverse process to regions of the model distribution with high constraint satisfaction, enabling conditional sampling from an unconditional model. Notably, the same model can be used for various conditional sampling tasks without re-training [1], and the guidance strength can be varied to trade-off realism and constraint satisfaction. Together, these techniques have enabled controllable diffusion models across a range of modalities, including images [6], audio [17], point clouds [38], motion trajectories [14], *etc*.

Our work is most related to Scene Diffusion [24], a concurrent work that uses a latent diffusion model for traffic scene generation. However, they do not consider *controllability*—an important characteristic for practical applications. Our work is also related to [36], [35], [3], which are diffusion models for controllable traffic behavior simulation. Like most

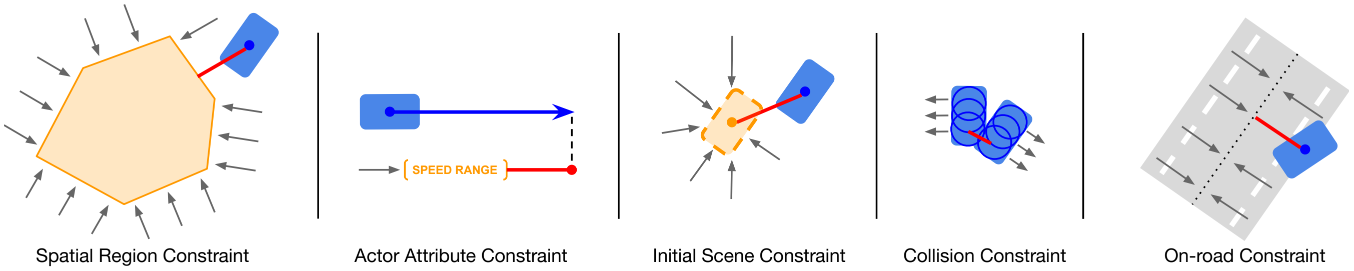

Fig. 3. Illustration of our guidance functions. Blue boxes are the generated actors; orange regions indicate the constraint regions; gray arrows indicate the direction of the guidance function's gradient; and red lines illustrate the value of the guidance function.

work in behavior simulation, they assumes a given initial scene; *e.g.*, from a log, manual creation, *etc.* We close this gap by enabling controllable scene generation. Finally, Scenario Diffusion [23] is a concurrent work that learns a conditional latent diffusion model for controllable scene and behavior simulation. Their approach requires training a new model for each condition whereas we use guided sampling to enable control over arbitrary constraints at inference time.

III. Controllable Traffic Scene Generation

A. Problem Formulation

Notation: We parameterize a traffic scene with n actors by the joint actor states $\mathbf{s}_{1:n} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$ and an HD map $\mathbf{m}$ of the region of interest, which provides contextual information about the surrounding road topology. The number of actors n varies between scenes, and we focus on a setting where n is given; *e.g.*, where users specify the desired density of the scene. Each actor state $\mathbf{s}_i \in \mathbb{R}^6$ is represented by the actor's centroid location $(x_i, y_i) \in \mathbb{R}^2$, bounding box length and width $(l_i, w_i) \in \mathbb{R}^2_+$ heading angle $\theta_i \in [0, 2\pi)$, and speed $s_i \in \mathbb{R}_{\geq 0}$. We represent the HD map as a *lane graph* $G = (V, E)$, where each vertex $u \in V$ is a lane segment and an edge $(u, v) \in E$ indicates that v is a successor, predecessor, or left/right neighbour of u.

Formulation: We formulate the task of *controllable traffic scene generation* in two stages. First, we learn a generative model of traffic scenes $p_\varphi(\mathbf{s}_{1:n}|\mathbf{m})$ to capture the distribution of real world traffic scenes. Then, during inference, we sample from a perturbed distribution [14],

$$\tilde{p}_\varphi(\mathbf{s}_{1:n}|\mathbf{m}) \propto p_\varphi(\mathbf{s}_{1:n}|\mathbf{m}) g(\mathbf{s}_{1:n}, \mathbf{m}) \tag{1}$$

where g is a *guidance function* that encodes the degree to which a scene $\mathbf{s}_{1:n}$ satisfies some high-level constraints. Sampling from this perturbed distribution corresponds to generating scenes that are both realistic under $p_\varphi(\mathbf{s}_{1:n}|\mathbf{m})$ and constraint-satisfying under $g(\mathbf{s}_{1:n}, \mathbf{m})$. By varying the guidance function, we can flexibly encode different constraints into the generation process. For example, using the identity recovers unconditional scene generation whereas using a collision cost encourages collision-free scenes instead. Notably, our formulation decouples realism from controllability, allowing us to re-use the same model with various implementations of g without re-training.

We propose to parameterize $p_\varphi(\mathbf{s}_{1:n}|\mathbf{m})$ with a diffusion model, a family of generative models that achieves state-of-the-art sample quality and admits flexible schemes for sampling from perturbed distributions. In the subsequent sections, we first describe our diffusion model of traffic scenes and then discuss how to sample from $\tilde{p}_\varphi(\mathbf{s}_{1:n}, \mathbf{m})$.

B. A Diffusion Model of Traffic Scenes

A diffusion model is a latent variable model that learns to reverse a forward diffusion process [11], [26]. For notational brevity, let $\mathbf{x}_0 = (\mathbf{s}_1 \cdots \mathbf{s}_n) \in \mathbb{R}^{n \times 6}$ denote n states $\mathbf{s}_{1:n}$.

Forward diffusion process: Starting from data $\mathbf{x}_0 \sim q(\mathbf{x}_0)$, the forward diffusion process $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ gradually corrupts the clean actor states $\mathbf{x}_0$ with Gaussian noise over T steps according to a variance schedule $\beta_1, \ldots, \beta_T$,

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{\beta_t}\mathbf{x}_{t-1}, (1 - \beta_t)\mathbf{I}) \tag{2}$$

This yields a chain of noisy actor states $\mathbf{x}_1, \ldots, \mathbf{x}_T$.

Reverse diffusion process: Given a sufficiently large T and suitable variance schedule, the distribution of $\mathbf{x}_T$ is well-approximated by an isotropic Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$. If we know the reverse distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, we can generate a sample $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ by sampling $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and reversing the forward process. Since the reverse distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is intractable to compute, we learn to approximate it instead. Because we are interested in sampling from the conditional distribution $p(\mathbf{s}_{1:n}|\mathbf{m})$, we learn to reverse the forward process conditional on the HD map $\mathbf{m}$,

$$p_\varphi(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{m}) = \mathcal{N}(\mu_\varphi(\mathbf{x}_t, t, \mathbf{m}), \mathbf{\Sigma}_\varphi(\mathbf{x}_t, \mathbf{m}, t)) \tag{3}$$

where $\mu_\varphi(\mathbf{x}_t, t, \mathbf{m})$ and $\mathbf{\Sigma}_\varphi(\mathbf{x}_t, t, \mathbf{m})$ is the approximate mean and covariance of the reverse distribution at each step t, and φ are learnable parameters. Then, to sample a scene $\mathbf{x}_0 \sim p_\varphi(\mathbf{x}_0|\mathbf{m})$, we reverse the forward process, using $p_\varphi(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{m})$ in place of $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ at each step t.

Following existing work [11], we fix $\mathbf{\Sigma}_\varphi(\mathbf{x}_t, t, \mathbf{m}) = \beta_t\mathbf{I}$ and parameterize the approximate mean as

$$\mu_\varphi(\mathbf{x}_t, t, \mathbf{m}) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}}\epsilon_\varphi(\mathbf{x}_t, t, \mathbf{m})\right) \tag{4}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{j=1}^t \alpha_j$. Intuitively, $\epsilon_\varphi(\mathbf{x}_t, t, \mathbf{m})$ predicts the noise ϵ that corrupts $\mathbf{x}_0$ into $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$. Therefore, learning a diffusion model amounts to learning a noise prediction model $\epsilon_\varphi(\mathbf{x}_t, t, \mathbf{m})$ to de-noise $\mathbf{x}_t$ into a sample of the data $\mathbf{x}_0 \sim p_\varphi(\mathbf{x}_0)$.

		Distribution JSD						Common Sense	
	Model	Nearest Dist.	Lat. Dev.	Ang. Dev.	Length	Width	Speed	Collision %	Off-road %
AV2	ATISS [21]	0.26	0.26	0.75	0.49	0.20	0.24	22.59	49.09
	ATISS++ [21]	0.30	0.22	0.21	0.23	0.15	0.18	1.58	24.84
	SceneGen [29]	0.20	**0.08**	0.38	0.26	0.21	0.06	1.80	1.10
	SceneControl (Ours)	**0.11**	0.16	**0.18**	**0.22**	**0.09**	**0.05**	**1.37**	**0.00**
Highway	ATISS [21]	0.20	0.56	0.75	0.36	0.34	0.13	8.27	35.34
	ATISS++ [21]	0.24	0.37	0.11	0.34	**0.17**	0.12	9.10	3.84
	SceneGen [29]	0.13	0.19	0.29	0.36	0.21	**0.08**	1.06	**0.11**
	SceneControl (Ours)	**0.10**	**0.18**	**0.07**	**0.31**	**0.17**	0.11	**0.32**	1.73

TABLE I: COMPARISON TO THE STATE-OF-THE-ART ON ARGOVERSE2 AND HIGHWAY.

Architecture: We parameterize the noise prediction model $\epsilon_\varphi(\mathbf{x}_t, t, \mathbf{m})$ as a transformer-based architecture with a lane graph GNN [4] to model complex actor-to-actor/map interactions. In contrast to image-based diffusion models, we do not rasterize the traffic scene but instead directly operate over the vector representation of the actor states and the lane graph. Overall, our architecture is lightweight, permutation-equivariant, and handles a variable number of actors. It consists of three components: (1) a set of encoders to featurize the input states and map; (2) a transformer decoder to model interactions; and (3) a decoder to predict the diffusion noise.

Given noisy actor states $\mathbf{x}_t \in \mathbb{R}^{n \times 6}$ and an HD map $\mathbf{m}$, we encode each state vector with a multi-layer perceptron (MLP) and encode lane graph representation of $\mathbf{m}$ using a lane graph GNN. We also embed the diffusion timestep t with sinusoidal positional encoding [31] and an MLP.

$$\mathbf{h}_s^0 = \mathrm{MLP}(\mathbf{x}_t), \mathbf{h}_m = \mathrm{GNN}(\mathbf{m}), \mathbf{h}_t = \mathrm{MLP}(\mathrm{PE}(t)) \quad (5)$$

Next, we use a series of interleaving self-attention and cross-attention layers to fuse the actor features $\mathbf{h}_s^0$ and lane graph features $\mathbf{h}_m$. Here, self-attention uses actor state features $\mathbf{h}_s^k$ as queries, keys, and values, allowing our model to extract actor-to-actor interactions. To condition on $\mathbf{m}$, cross-attention instead uses lane graph features $\mathbf{h}_m$ as the keys and values, allowing our model to capture actor-to-map interactions. After each pair of attention layers, we fuse the diffusion timestep embedding $\mathbf{h}_t$ into the resulting features

$$\mathbf{h}_s^{k+1} = \mathrm{CrossAttn}(\mathrm{SelfAttn}(\mathbf{h}_s^k), \mathbf{h}_m) + \mathbf{h}_t \quad (6)$$

After K blocks of self-attention and cross-attention, we use an MLP to predict the forward diffusion noise

$$\epsilon_\varphi(\mathbf{x}_t, t, \mathbf{m}) = \mathrm{MLP}(\mathbf{h}_s^K) \quad (7)$$

Learning: We learn the noise prediction model $\epsilon_\varphi(\mathbf{x}_t, t, \mathbf{m})$ using noise-matching [11],

$$L(\varphi) = \mathbb{E}_{\mathbf{x}_0, \mathbf{m}, t, \epsilon}\left[\|\epsilon - \epsilon_\varphi(\mathbf{x}_t, t, \mathbf{m})\|^2\right] \quad (8)$$

where $\mathbf{x}_0$ and $\mathbf{m}$ is the joint actor states and HD map for a real traffic scene, $t \sim \mathrm{Uniform}(1, T)$ is a diffusion step, and $\mathbf{x}_t$ is the actor states $\mathbf{x}_0$ corrupted with noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

C. Controllable Scene Generation with Guidance

Having specified our generative model $p_\varphi(\mathbf{s}_{1:n}|\mathbf{m})$, we now discuss how to sample from the perturbed distribution $\tilde{p}_\varphi(\mathbf{s}_{1:n}|\mathbf{m}) \propto p_\varphi(\mathbf{s}_{1:n}|\mathbf{m}) g(\mathbf{s}_{1:n}, \mathbf{m})$ to generate scenes that satisfy high-level constraints. For diffusion models, we use *guided sampling* [6], [12]. Given the number of actors n (specifying the desired scene density), we first sample n random noise vectors, which we denote $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then, at each step t of reverse diffusion, rather than sampling from $p_\varphi(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{m}) = \mathcal{N}(\mu_\varphi(\mathbf{x}_t, t, \mathbf{m}), \beta_t \mathbf{I})$, we sample from

$$\mathcal{N}(\mu_\varphi(\mathbf{x}_t, t, \mathbf{m}) - \gamma_t \beta_t \nabla_{\mathbf{x}_t} g(\mathbf{x}_t, \mathbf{m}), \beta_t \mathbf{I}) \quad (9)$$

where γ_t is a time-varying coefficient that controls the guidance strength. Notably, this approach does not require retraining a new model for each guidance function, allowing us to flexibly incorporate any constraints into scene generation. Fig. 3 illustrates our considered guidance functions.

Spatial region constraints: SceneControl allows us to insert actors into specific regions of interest in a scene; *e.g.*, to manually populate specific areas around the SDV or automatically densify all intersections. We design a guidance function based on the signed distance function of an actor's centroid (x_i, y_i) to the boundary of a 2D polygon c_{region},

$$g_{\text{region}}(\mathbf{s}_i, c_{\text{region}}) = \max\{0, \mathrm{SDF}(c_{\text{region}}, (x_i, y_i))\} \quad (10)$$

Actor attributes constraints: SceneControl allows us to constrain actor attributes such as speed, bounding box size, *etc*. Unlike manually specifying each attribute, which can lead to unrealistic scenes (*e.g.*, a truck with Ferrari speed at a tight turn), SceneControl automatically adapts all attributes when controlling for a subset of them. To this end, we design a guidance function as the distance of an actor's attribute a_i to the boundary of a 1D range $c_{\text{attr}} = (c_{\min}, c_{\max})$,

$$g_{\text{attr}}(\mathbf{s}_i, c_{\text{attr}}) = \max\{0, a_i - c_{\max}, c_{\min} - a_i\} \quad (11)$$

Initial scene constraints: SceneControl allows us to generate traffic scenes from an empty map or from a scene with existing actors $c_{\text{init}} = \{\hat{\mathbf{s}}_i | i \in \mathcal{I}\}$. To this end, we design a guidance function to penalize the difference between the existing actors' sampled *vs.* original states,

$$g_{\text{init}}(\mathbf{s}_i, c_{\text{init}}) = \mathbb{1}[i \in \mathcal{I}] \|\mathbf{s}_i - \hat{\mathbf{s}}_i\| \quad (12)$$

By adjusting the guidance strength, we can interpolate between keeping the initial scene fixed *vs.* allowing for adjustments that improve realism; *e.g.*, moving existing actors closer together when densifying an already dense scene.

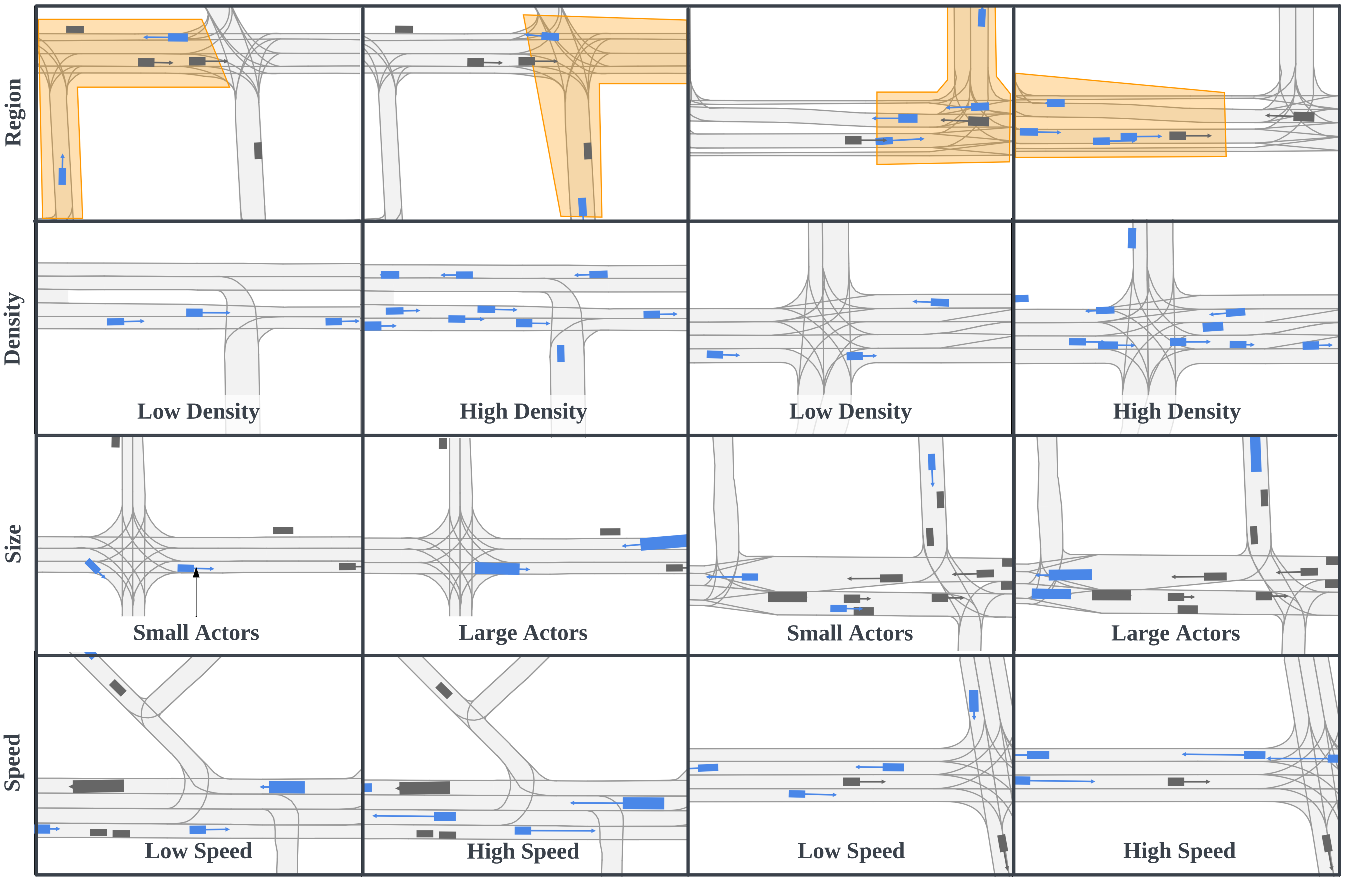

Fig. 4. SceneControl enables generating realistic traffic scenes that satisfy high-level user-specified constraints. By specifying a spatial region constraint (in orange), we can insert actors into specific regions of an existing scene (**Row 1**). By changing the desired number of actors, we can vary the density of the scenes (**Row 2**). In addition, we can also specify constraints on the actors' attributes, allowing us to vary the generated actors' size (**Row 3**) and speed (**Row 4**). Existing actors in the scene are depicted in gray and inserted actors in blue.

Common sense constraints: SceneControl allows us to incorporate common sense priors to improve realism. In particular, collisions are rare and actors typically drive on lanes. Therefore, we devise guidance functions that penalize collisions and off-lane driving. For collision, we use a differentiable relaxation [27]; we approximate each actor's bounding box with five circles and compute the L2 distance $\mathrm{d}((x_i, y_i), (x_j, y_j))$ between the centroids of the closest circles between the pair of actors (with radii r_i and r_j),

$$g_{\text{collision}}(\mathbf{s}_i, \mathbf{s}_j) = \max\left\{0, 1 - \frac{\mathrm{d}((x_i, y_i), (x_j, y_j))}{r_i + r_j}\right\} \quad (13)$$

For off-lane driving, we use the minimum projection distance between an actor's centroid and its closest lane,

$$g_{\text{lane}}(\mathbf{s}_i, \mathbf{m}) = \min_{\text{lane}\in\mathbf{m}} \mathrm{d}_{\text{proj}}((x_i, y_i), \text{lane}) \quad (14)$$

IV. Experiments

A. Experiment Setup

Datasets: We perform experiments on two datasets that cover complex urban traffic and high-speed highway driving. Our first dataset, Argoverse 2 Sensor [34], has 110,071 urban traffic scenes for training and 23,547 for validation. Our second dataset, Highway, has 160,000/40,000 highway scenes in its train/validation splits. Both datasets provide 3D bounding box labels per scene and HD maps.

Baselines: We compare against the state-of-the-art in traffic scene generation. **SceneGen** [29] is an autoregressive model over traffic scenes, where actors are iteratively inserted in a fixed left-to-right, bottom-to-top order. We compare to **ATISS/ATISS++** [21], an order-agnostic autoregressive model for indoor scene generation adapted to our setting. While ATISS follows its original implementation, for fair comparison, we strengthened ATISS++ for our setting.

Metrics: Evaluating the performance of a generative model is an open challenge since no single metric is sufficient for measuring all aspects of realism and controllability necessary for downstream applications [30]. Therefore, we propose to evaluate our models on a suite of metrics instead.

- **Distribution JSD** measures realism by the similarity between the distribution of real and synthetic traffic scenes. We report the Jensen-Shannon divergence between distributions of scene statistics that capture actor attributes, actor-to-actor interactions, and actor-to-map reasoning.
- **Common Sense** metrics measures realism by the frequency to which a model generates infraction-free traffic scenes, namely scenes that are free from collision and off-road actors. We report the percentage of collision actors

	Model	Guide	Region Constraint			Size Constraint			Speed Constraint		
			Suc. %	JSD	Col. %	Suc. %	JSD	Col. %	Suc. %	JSD	Col. %
AV2	ATISS++ [21]		8.34	0.23	4.66	85.50	0.22	20.34	76.52	0.22	18.86
	SceneGen [29]		7.66	0.22	1.54	86.67	0.21	2.53	82.09	0.20	1.98
	SceneControl (Ours)		24.78	**0.17**	1.42	80.33	**0.14**	2.23	81.33	**0.15**	1.55
	SceneControl (Ours)	✓	**77.22**	0.19	**1.35**	**95.62**	**0.14**	**2.17**	**98.87**	**0.14**	**1.37**
Highway	ATISS++ [21]		17.86	0.47	5.91	78.48	0.40	19.80	59.52	0.36	5.91
	SceneGen [29]		27.66	0.22	0.77	75.86	0.20	0.97	64.93	0.21	0.77
	SceneControl (Ours)		35.69	**0.16**	0.79	79.46	**0.15**	0.68	71.56	**0.15**	**0.47**
	SceneControl (Ours)	✓	**86.60**	**0.16**	**0.15**	**94.66**	**0.14**	**0.54**	**90.84**	**0.16**	**0.47**

TABLE II: CONTROLLABLE TRAFFIC SCENE GENERATION ON ARGOVERSE2 AND HIGHWAY. SUC. % DENOTES THE CONSTRAINT SATISFACTION SUCCESS RATE. JSD IS THE AVERAGE DISTRIBUTION JSD. COL. % IS THE COLLISION RATE. GUIDE INDICATES WHETHER GUIDED SAMPLING IS USED.

(**Collision %**) and off-road actors (**Off-road %**).

- **Constraint Satisfaction** measures controllability by the percentage of synthetic traffic scenes that satisfy a user-specified constraint. Since this metric's definition varies by the desired constraints, we defer details to Sec. IV-C, where we describe our controllability experiments.

B. Comparison to the State-of-the-art

Our first experiment benchmarks SceneControl and the baselines in unconditional traffic scene generation. Here, we want to generate realistic scenes from scratch given only an HD map of the region of interest. This setting represents a foundational capability necessary for controllable traffic scene generation and allows us to fairly compare SceneControl against the existing state-of-the-art. For fair comparison, we control for the number of actors each model is requested to generate. For SceneControl, we use guided sampling with common sense constraints to improve realism.

Tab. I summarizes our results on Argoverse2 and Highway. Overall, SceneControl generates the most realistic traffic scenes in both urban and highway settings. In particular, SceneControl achieves the lowest nearest distance JSD and collision rate, suggesting that it models actor-to-actor interactions better than the baselines. SceneControl also achieves competitive results on lateral/angular deviation JSD and off-road rate, highlighting its capacity for actor-to-map reasoning. In contrast, ATISS/ATISS++ struggle since they were designed for simpler indoor scenes with few objects. SceneGen is more competitive but its use of simple parametric distributions limits expressivity. By foregoing such assumptions, SceneControl achieves significant improvements in angular deviation, length, and width JSD.

C. Controllable Traffic Scene Generation

Beyond generating traffic scenes from scratch, we are also interested in generating scenes that satisfy high-level constraints; *e.g.*, where the actor is placed, how fast they drive, how large they are. SceneControl does this effectively, achieving greater controllability and realism than prior work.

Setup: We systematically evaluate controllability across several experiments. To evaluate *spatial region* controllability, for each scene in the validation set, we remove all actors from a random non-empty lane, generate a new scene with a region constraint based on the selected lane polygon, and report the percentage of new actors whose centroid lies within the lane polygon. To evaluate *actor attributes* controllability, we remove all actors with speed (size) within a random range, generate a new scene with speed (size) constraints based on the selected range, and report the percentange of new actors whose speed (size) fall within the selected range.

Guidance		Distribution JSD		Common Sense	
Col.	Lane	Near. Dist.	Lat. Dev.	Col. %	Offroad %
		0.29	0.22	17.20	**0.00**
✓		0.12	0.23	1.44	0.11
	✓	0.30	0.16	16.68	**0.00**
✓	✓	**0.11**	**0.16**	**1.40**	**0.00**

TABLE III: COMMON SENSE GUIDANCE ON ARGOVERSE2.

Results: Tab. II summarizes our results on Argoverse2 and Highway. Across all experiments, SceneControl consistently achieves greater constraint satisfaction and realism than the baselines. Since the baselines are designed for unconditional traffic scene generation, they ignore and rarely satisfy user-specified constraints. In contrast, using guided sampling, SceneControl achieves a high degree of constraint satisfaction without impairing realism.

Interactive scene generation: In Fig. 4, we illustrate how SceneControl can be used as an interactive tool for controllable traffic scene generation. Specifically, SceneControl can insert actors into specific regions of an existing scene, vary the number/density of actors in a scene, and constrain actors' size and speed to specific ranges. We also demonstrate our interactive tool in the attached supplementary video.

Incorporating common sense: Tab. III shows that guided sampling with common sense constraints significantly improves SceneControl's realism across both distribution JSD and common sense metrics.

V. CONCLUSION

In this paper, we proposed SceneControl, a framework for controllable traffic scene generation. SceneControl is an expressive diffusion model of traffic scenes that enables us to generate realistic traffic scenes that satisfy arbitrary constraints. This allows us to flexibly control the generation process to scalably create traffic scenes that exhibit our desired characteristics, opening up exciting opportunities to improve how we design scenarios for training and testing autonomy, making the safety case, and beyond.

References

[1] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. *CoRR*, 2023.

[2] Luca Bergamini, Yawei Ye, Oliver Scheel, Long Chen, Chih Hu, Luca Del Pero, Blazej Osinski, Hugo Grimmett, and Peter Ondruska. Simnet: Learning reactive self-driving simulations from real-world observations. In *ICRA*, 2021.

[3] Wei-Jer Chang, Francesco Pittaluga, Masayoshi Tomizuka, Wei Zhan, and Manmohan Chandraker. Controllable safety-critical closed-loop traffic simulation via guided diffusion, 2023.

[4] Alexander Cui, Sergio Casas, Kelvin Wong, Simon Suo, and Raquel Urtasun. GoRela: Go relative for Viewpoint-Invariant motion forecasting. Nov. 2022.

[5] Jeevan Devaranjan, Amlan Kar, and Sanja Fidler. Meta-Sim2: Unsupervised learning of scene structure for synthetic data generation. In *ECCV*, 2020.

[6] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021.

[7] Alexey Dosovitskiy, Germán Ros, Felipe Codevilla, Antonio M. López, and Vladlen Koltun. CARLA: an open urban driving simulator. In *CoRL*, 2017.

[8] Lan Feng, Quanyi Li, Zhenghao Peng, Shuhan Tan, and Bolei Zhou. TrafficGen: Learning to generate diverse and realistic traffic scenarios. *CoRR*, 2022.

[9] Association for Standardization of Automation and Measuring Systems. ASAM OpenSCENARIO.

[10] Daniel J. Fremont, Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Alberto L. Sangiovanni-Vincentelli, and Sanjit A. Seshia. Scenic: a language for scenario specification and scene generation. In *PLDI*, 2019.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *CoRR*, 2022.

[13] Applied Intuition. Large-scale simulation and validation with carla, 2022.

[14] Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *ICML*, 2022.

[15] Stefan Jesenski, Jan Erik Stellet, Florian A. Schiegg, and J. Marius Zöllner. Generation of scenes in intersections for the validation of highly automated driving functions. In *IV*, 2019.

[16] Amlan Kar, Aayush Prakash, Ming-Yu Liu, Eric Cameracci, Justin Yuan, Matt Rusiniak, David Acuna, Antonio Torralba, and Sanja Fidler. Meta-Sim: Learning to generate synthetic datasets. In *ICCV*, 2019.

[17] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. In *ICLR*, 2021.

[18] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. July 2020.

[19] Alexis Madrigal. Inside waymo's secret world for training self-driving cars, 2017.

[20] OpenAI. Gpt-4 technical report, 2023.

[21] Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. ATISS: Autoregressive transformers for indoor scene synthesis. In *NeurIPS*, 2021.

[22] Aayush Prakash, Shaad Boochoon, Mark Brophy, David Acuna, Eric Cameracci, Gavriel State, Omer Shapira, and Stan Birchfield. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In *ICRA*, 2019.

[23] Ethan Pronovost, Meghana Reddy Ganesina, Noureldin Hendy, Zeyu Wang, Andres Morales, Kai Wang, and Nicholas Roy. Scenario diffusion: Controllable driving scenario generation with diffusion. *CoRR*, 2023.

[24] Ethan Pronovost, Kai Wang, and Nick Roy. Generating driving scenes with diffusion. *CoRR*, 2023.

[25] Rodrigo Queiroz, Thorsten Berger, and Krzysztof Czarnecki. Geoscenario: An open DSL for autonomous driving scenario representation. In *IV*, 2019.

[26] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

[27] Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. TrafficSim: Learning to simulate realistic multi-agent behaviors. Jan. 2021.

[28] Shuhan Tan, Boris Ivanovic, Xinshuo Weng, Marco Pavone, and Philipp Kraehenbuehl. Language conditioned traffic generation, 2023.

[29] Shuhan Tan, Kelvin Wong, Shenlong Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. SceneGen: Learning to generate realistic traffic scenes. In *CVPR*, 2021.

[30] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *ICLR*, 2016.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. June 2017.

[32] Tim Allan Wheeler and Mykel J. Kochenderfer. Factor graph scene distributions for automotive safety analysis. In *ITSC*, 2016.

[33] Tim Allan Wheeler, Mykel J. Kochenderfer, and Philipp Robbel. Initial scene configurations for highway traffic propagation. In *ITSC*, 2015.

[34] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *NeurIPS Datasets and Benchmarks*, 2021.

[35] 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. *CoRR*, 2023.

[36] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. *ICRA*, 2023.

[37] Berend Zwartsenberg, Adam Ścibior, Matthew Niedoba, Vasileios Lioutas, Yunpeng Liu, Justice Sefas, Setareh Dabiri, Jonathan Wilder Lavington, Trevor Campbell, and Frank Wood. Conditional permutation invariant flows. *CoRR*, 2022.

[38] Vlas Zyrianov, Xiyue Zhu, and Shenlong Wang. Learning to generate realistic LiDAR point clouds. In *ECCV*, 2022.