LaGen: Towards Autoregressive LiDAR Scene Generation

Sizhuo Zhou[1,2], Xiaosong Jia[4*], Fanrui Zhang[1,2], Junjie Li[1,2], Juyong Zhang[1], Yukang Feng[2], Jianwen Sun[2], Songbur Wong[3], Junqi You[3], and Junchi Yan[2,3*]

[1] University of Science and Technology of China, Hefei, Anhui 230026, China
[2] Shanghai Innovation Institute, Shanghai 200231, China
[3] Shanghai Jiao Tong University, Shanghai 200240, China
[4] Fudan University, Shanghai 200433, China
yanjunchi@sjtu.edu.cn

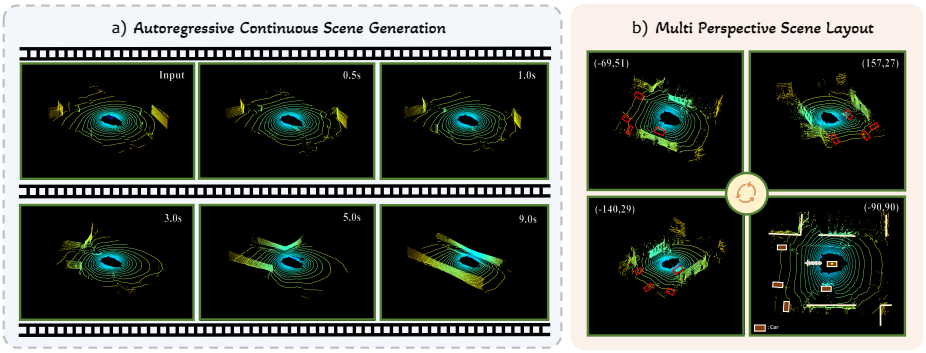

Fig. 1: This work introduces LaGen, a novel 4D LiDAR scene generation framework that can autoregressively generate **(a)** long-horizon autonomous driving scenes based solely on single-frame input. It is capable of generating high-fidelity LiDAR consistent with the real world; **(b)** illustrates the visualization results from different viewpoints.

Abstract. Generative world models for autonomous driving (AD) are of great value in applications such as data augmentation, closed-loop simulation, and safety-critical scenario evaluation. Unlike the widely studied image modality, in this work we explore generative world models for LiDAR data. Existing generation methods for LiDAR predominantly focus on single frame generation or lack the capacity for interactive simulation, while existing prediction approaches require multiple frames of historical input and can only deterministically predict multiple frames at once. Both paradigms fail to support long-horizon interactive generation. To this end, we introduce **LaGen**, which, to the best of our knowledge is the first autoregressive framework capable of generating

* Corresponding author.

long-horizon LiDAR scenes in a frame-by-frame, interactive manner. LaGen is able to take a single-frame input as a starting point and effectively utilize bounding box information as conditions to generate high-fidelity 4D scene. In addition, we introduce a scene decoupling estimation module to enhance the model's interactive generation capability for object-level content, as well as a noise modulation module to mitigate error accumulation during long-horizon generation. We extensively evaluate LaGen's performance in controlled data generation and long-horizon scene generation on the nuScenes dataset. The experimental results demonstrate that LaGen achieves state-of-the-art performance, especially on later frames. The code is publicly available at: https://github.com/szzhou88/LaGen.

Keywords: Autonomous driving · World models · Scene generation

1 Introduction

In recent years, autonomous systems have attracted increasing attention in both academia and industry. LiDAR is one of the key sensors in various autonomous systems. It provides precise and rich geometric information about the surrounding environment [5, 27, 28, 68, 70] and has been widely applied across a range of fields [37, 50, 53, 54, 61, 73]. Unlike the extensively studied image modality, research on LiDAR is still limited. Existing studies have focused on two main directions: one is generating high-quality LiDAR data [40, 67, 76] to reduce the high cost of real-world acquisition [38, 66], and the other is predicting LiDAR data [23, 63, 64] to support decision-making in autonomous systems.

However, these research directions encounter substantial limitations in practical applications. Most existing works on LiDAR generation predominantly focus on learning the data distribution within a dataset, enabling them to synthesize only non-sequential, single LiDAR point clouds. Recent work has explored the generation of 4D LiDAR scenes [26, 32, 41], but it lacks interactivity, is not suitable for long-horizon tasks, and cannot be used for closed-loop autonomous driving simulation. For LiDAR prediction tasks, with the rise of end-to-end autonomous driving (E2E-AD) [18, 21, 62], current methods are confronted with the following fundamental challenges:

1) Prediction requires multiple frames of historical scene data; however, in closed-loop simulation [6, 8, 20, 47], only the initial frame is available;
2) These methods are limited to predicting LiDAR scenes along a predetermined trajectory, while the ego vehicle may generate varying trajectories based on different decision-making.
3) They cannot accurately predict LiDAR over long temporal horizons, as they are limited by fixed historical inputs.

To address these challenges, we explore LiDAR-based generative world models and propose LaGen. LaGen is an autoregressive generation framework capable of producing high-fidelity, long-horizon LiDAR scenes on a frame-by-frame basis (see Figure 1). Specifically, we first utilize spherical projection to convert

the three-dimensional LiDAR data into a more compact form: a range image. Next, we construct a LiDAR data generator based on the Latent Diffusion Model (LDM) [51], and guide the diffusion process using multiple control conditions.

To further enhance the spatiotemporal consistency of the entire generative framework, we also introduce the following carefully designed strategies:

1) To improve the spatial consistency, we introduce a Scene Decoupling Estimation (SDE) module. It decouples the LiDAR scene into foreground (static and dynamic objects) and background, and estimates the foreground and background of the current frame based on the currentscene state information. This provides the model with object-level estimations of current scene, thereby enhancing the model's interactive generation capability.
2) To enhance the temporal consistency, we introduce a Noise Modulation (NM) module designed to alleviate error accumulation. This is because the autoregressive paradigm suffers from a train-inference mismatch; during inference, errors in the generated samples accumulate over longer temporal horizons. The proposed module injects noise into the conditional features containing information from the previous frame, thereby reducing the model's over-reliance on prior frames and enabling it to better adapt to imperfect conditions.

We compare LaGen with state-of-the-art LiDAR generation models on the nuScenes dataset. The experimental results show that our method has clear advantages in generating high-fidelity LiDAR point clouds and temporally consistent LiDAR scenes over long horizons. Furthermore, we apply LaGen to downstream prediction tasks to further validate its performance in long-horizon generation. Experimental results indicate that LaGen can accurately generate long-horizon LiDAR scenes using only a single-frame historical input. It is noteworthy that our autoregressive framework further enables autonomous systems to incorporate decisions made at each time step into the generation of future predictions, thereby naturally supporting interactive world simulation.

In summary, our contributions are as follows:

– We propose LaGen, which is the first autoregressive framework capable of generating high-fidelity, long-horizon LiDAR scenes with interactivity;
– We meticulously design a control condition feature encoder to guide the generation process, and propose a scene decoupling estimation module and a noise modulation module, which effectively enhance the model's performance in interactive scene-detail generation and long-horizon autoregressive generation;
– We thoroughly evaluate LaGen's performance in generating high-fidelity LiDAR data and temporally consistent 4D LiDAR scenes on the nuscenes dataset to demonstrate the advantages of our method, and further assess its performance on long-horizon prediction tasks.

2 Related Work

Generative Models. Generative models aim to learn the underlying data distribution and generate novel samples. Generative Adversarial Networks (GANs)

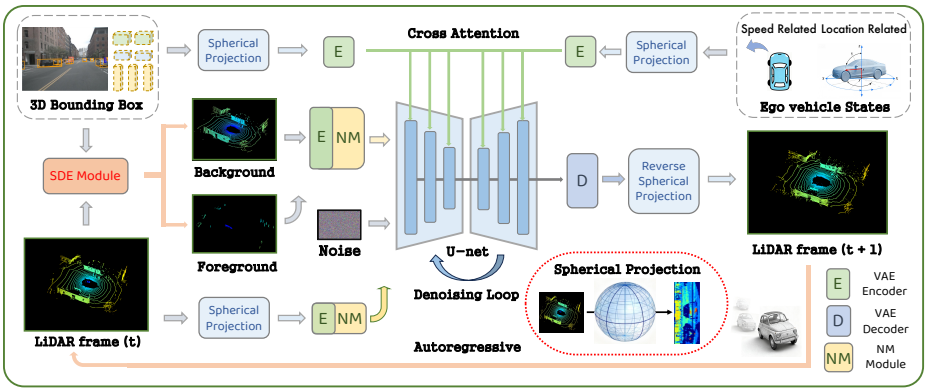

Fig. 2: Overview of the LaGen framework. It first obtains estimates of the current frame's foreground and background point clouds via the Scene Decoupling Estimation (SDE) module. Subsequently, all three-dimensional information is projected to the two-dimensional range image space via spherical mapping. These range-view representations are then processed by a pretrained VAE and a noise modulation (NM) module, before being passed to the U-net network to generate the LiDAR scene for the current frame. Afterward, the current generated scene is iteratively used as the input to the model, thereby completing the autoregressive process.

[11] are a prominent example. GANs alternately train a pair of competing networks, the generator and the discriminator, to minimize the adversarial objective. However, GANs often suffer from unstable training dynamics. Diffusion models (DMs) [55] have shown outstanding performance in the field of image synthesis. Mainstream diffusion models include score-based models [57–59] and Denoising Diffusion Probabilistic Models (DDPM) [14, 24, 43, 52]. They enable stable training using simple objective functions, addressing challenges associated with GANs. Recently, Latent Diffusion Models (LDMs) [51] have been proposed. They perform the diffusion process in the latent space, greatly improving generation efficiency. They have also enabled significant advances in multimodal conditional generation [42, 48, 74].

LiDAR Data Generation. Generating LiDAR point clouds is a 3D data generation task. Caccia et al. [1] were the first to represent LiDAR data as range images and employed Variational Autoencoders (VAEs) [25] and Generative Adversarial Networks (GANs) [11] for generation. Subsequently, LiDARGen [76] introduced a score-based LiDAR data generation model. UltraLiDAR [67] voxelizes LiDAR point clouds and employs VQ-VAE [71] for generation. LiDM [49] separately generates scenes, objects, and trajectories, and simulates ray casting to produce point clouds. R2DM [40], based on Denoising Diffusion Probabilistic Models (DDPMs) [56], further improves the fidelity of generated point clouds. RangeLDM [16] introduces a range-guided discriminator, enabling fast generation through a latent diffusion model. La La LiDAR [35] introduces a large-scale, layout-guided LiDAR generation model. However, most of these frameworks are

only capable of generating isolated single-frame LiDAR data. Genesis [13] introduces a unified framework for the joint generation of driving videos and LiDAR sequences. DriveLiDAR4D [3] incorporates three modalities of conditional inputs, namely road sketches, scene descriptions, and object priors, which further enhance the quality of the generated results. UniScene [26] leverages occupancy as an intermediate representation to generate continuous LiDAR data, thereby reducing the complexity of scene synthesis. OpenDWM [41] is built upon the DiT (Diffusion Transformer) architecture for video generation and has been extended to the domain of LiDAR scenes. LiDARCrafter [32], parses natural language inputs into scene graphs to guide the generation process. It also explores the autoregressive genration, but it generates future trajectories and control conditions in a one-shot manner based on initial semantics, lacking interactivity and cannot be used for closed-loop simulation. In addition, LaGen is primarily based on bounding box conditions, without utilizing natural language information.

LiDAR Data Forecasting. Point cloud prediction [39,63–65] is a fundamental self-supervised task in autonomous driving, where multiple frames of historical point clouds are used as input to predict future point clouds. Early work predicted future point cloud sequences by predicting scene flow [34, 72]. Recent studies have started to directly predict from raw point cloud data [36,44,63,64]. 4D-Occ [23] predicts point clouds from the perspective of geometric occupancy prediction, which further improves prediction accuracy. ViDAR [69] focuses on visual point cloud prediction, using historical images to forecast future point clouds, which improves its performance in downstream applications to some extent. These frameworks typically require multiple frames of historical input, yet still cannot accurately predict long-horizon LiDAR scenes and lack interactivity.

3 Method

3.1 Preliminaries

Range Image: A Compact Representation for LiDAR Data. Range images offer a compact representation of LiDAR data by parametrizing unstructured point clouds in 3D space into a 2D pixel space. Specifically, for a point $p = (x, y, z) \in \mathbb{R}^3$ in Cartesian coordinates, it is transformed to the spherical coordinates (r, θ, ϕ) via spherical projection. We adopt the corrected range image representation [16] as follows:

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + (z - h_j)^2}, \\ \theta &= \arctan(y, x), \quad \phi = \phi_j, \end{aligned} \tag{1}$$

where r denotes the depth value, θ is the azimuth angle, ϕ is the elevation angle, h_j and ϕ_j denote the sensor height and elevation angle estimated by Hough voting. Next, we convert the spherical coordinates (r, θ, ϕ) into pixel coordinates (u, v) of the range image using the equations:

$$u = W - ((\theta + \pi)/2\pi) \cdot W, \quad v = H - j, \tag{2}$$

where W and H are the width and height of the image in pixels. Finally, we normalize the depth and intensity values of the point cloud to obtain a two-channel range image I.

Latent Diffusion Model for LiDAR Generation. We employ the Latent Diffusion Model [51] for LiDAR data generation. LDM consists of two components: a VAE and a 2D UNet. The VAE encodes the input range image $I \in \mathbb{R}^{H\times W\times 2}$ into a latent representation $\hat{z} \in \mathbb{R}^{h\times w\times c}$. The decoder then takes latent $\hat{z}$ as input and generates the reconstructed range image $\hat{I} \in \mathbb{R}^{H\times W\times 2}$.

During the diffusion process, noise is progressively added to the initial latent z_0 to obtain noisy latent z_t, with the noise level increasing as the timestep $t \in \{1, 2, \ldots, T\}$ advances. The UNet is trained to predict the noise added to the latent feature $\hat{z}$. The training loss of the LDM is defined as:

$$\mathcal{L}_{\text{LDM}} = \mathbb{E}_{\epsilon_t\in\mathcal{N}(0,1),t\in\mathcal{U}[0,T],c}\left[\left\|\epsilon_t - \epsilon_\theta\left(z_t; t, c\right)\right\|^2\right], \tag{3}$$

where z_t denotes the noisy latent at timestep t, ϵ_θ is the noise prediction network to be trained, and c is the encoded conditional information. During inference, LDM generates range images by iteratively removing the noise predicted by UNet from randomly sampled Gaussian noise over T steps.

3.2 Overall Framework

Given the first frame of LiDAR data for a driving scenario $P^0 \in \mathbb{R}^{N^0\times 4}$, our objective is to recursively generate subsequent LiDAR frames P^s, for $s = 1, 2, \ldots, S$, in accordance with the actual motion of the ego-vehicle.

In autonomous driving simulators, the following information for each frame of the driving scenario can be tracked [20, 22]:

- 3D bounding boxes and semantic labels of the previous and current frames: $B^s = \{(b_i, c_i)\}_{i=1}^{N_b}$, where $b_i = (x_j, y_j, z_j)_{j=1}^{8} \in \mathbb{R}^{8\times 3}$ denotes the bounding boxes of dynamic and static objects (such as vehicles, pedestrians, and obstacles) within a predefined range, and $c_i \in C_{box}$ is the associated semantic category.
- Ego states: $E^s = \{E_{\text{ego}}, E_{\text{trans}}\}$, where E_{ego} includes ego-vehicle speed, acceleration and steering angle information, while E_{trans} is the transformation matrix from the ego-vehicle to the global coordinate frame.

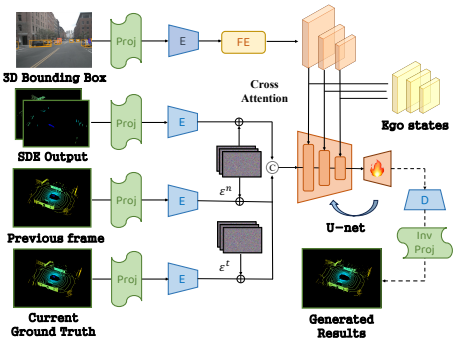

Fig. 3: The main architecture of the high-quality LiDAR data generator.

It should be noted that the bounding boxes of the current frame are not future information; rather, they are produced by the simulator based on the action from

the previous frame. Accordingly, at each step of the generation process, we aim to generate the current LiDAR frame based solely on the previous frame, while incorporating 3D bounding box information and ego-vehicle states. The pipeline of LaGen is shown in Figure 2.

3.3 Strongly Spatiotemporally Consistent LiDAR Scene Generator and Inference Framework

In this section, we present the design details of the generator and inference framework for LaGen. We construct the core generative architecture based on LDMs and implement conditional control via meticulously designed scene information encoders (see Figure 3). Furthermore, two crucial modules are introduced to bolster the spatial-temporal consistency of the overall framework.

3.3.1. Multimodal Conditional Feature Encoder

Previous Frame's LiDAR Data. Using the previous frame's LiDAR point cloud $P^{s-1} \in \mathbb{R}^{N^{s-1} \times 4}$ as one of the control conditions in the generation process is crucial for endowing the generator with autoregressive inference capability. To incorporate previous frame information into the denoising process, we first convert the 3D point cloud P^{s-1} into a 2D range image $I^{s-1} \in \mathbb{R}^{2 \times H \times W}$ using Equation (1). The variational autoencoder then projects I^{s-1} into the latent space as a latent variable $z^{s-1} \in \mathbb{R}^{c \times h \times w}$.

We can perform the following computation:

$$E_{\text{rel}}^{s-1} = (E_{\text{ego2glb}}^{s} \cdot E_{\text{li2ego}}^{s})^{-1} \cdot (E_{\text{ego2glb}}^{s-1} \cdot E_{\text{li2ego}}^{s-1}), \tag{4}$$

to obtain the transformation matrix $E_{\text{rel}}^{s-1} \in \mathbb{R}^{4 \times 4}$, which represents the coordinate transformation from the previous frame's LiDAR sensor to the current frame's LiDAR sensor. Here, E_{ego2glb}^{s-1} and E_{ego2glb}^{s} are the transformation matrices from the ego-vehicle to the global coordinate system, while E_{li2ego}^{s-1}, E_{li2ego}^{s} denote the transformation matrices from the LiDAR sensor to the ego-vehicle. Next, we apply a Feature-wise Linear Modulation (FiLM) [46] layer to adjust the latent feature z^{s-1} of the previous frame's LiDAR dat:

$$\hat{z}^{s-1} = FiLM(z^{s-1}, MLP(Flatten(E_{relative}^{s-1}))), \tag{5}$$

and inject it into the latent diffusion process of the model.

Object-Level 3D Bounding Box Projection. 3D bounding box information is one of the most important cues in autonomous driving scenarios. To fully exploit this information, we first extract the four corner vertices of b_i that are closest to the ground and perform pairwise interpolation between these vertices to obtain a point cloud $\hat{b}_i$ that encloses the object region. Next, according to the semantic label c_i, we using Equation (1) to project point clouds of different classes into range image binary masks $M_B \in \mathbb{R}^{D \times H \times W}$, where D denotes the

number of categories. Then, we encode these masks into latent features $H_B \in \mathbb{R}^{N\times D}$ in the latent space using an object-level mask encoder $\mathcal{E}_{mask}$.

Subsequently, we inject H_B into multiple layers of the UNet via a cross-attention mechanism:

$$q = \hat{H}, \quad k = H_B^{\text{prev}}, \quad v = H_B^{\text{cur}}, \tag{6}$$

where q, k, v denote the query, key, and value in the cross-attention layer, $\hat{H}$ represents the features of the intermediate layer, and H_B^{cur}, H_B^{prev} are the features of bounding box projections for the current and previous frames.

Ego States. We first encode the ego vehicle state E_{ego} and E_{trans} into compact vector representations. These representations are then embedded via a timestep embedding module to capture both temporal and geometric transition information. After a nonlinear projection, the resulting embedding is converted into a single conditional token, which is injected into each level of the U-Net via a cross-attention mechanism.

3.3.2. Scene Decoupling Estimation

Having established the basic framework for LiDAR data generation conditioned on multiple control conditions, we further design a Scene Decoupling Estimation (SDE) module to improve the quality of generated details (see Figure 4). The design is described in detail below.

For the previous frame LiDAR point cloud $P^{s-1} \in \mathbb{R}^{N^{s-1}\times 4}$, we decouple it into foreground and background components based on its associated bounding box information B^{s-1}. The foreground point cloud P_{obj}^{s-1} consists of all points in P^{s-1} that fall within any of the bounding boxes:

$$P_{\text{obj}}^{s-1} = \{p_{ij}\}, i \in \{1, 2, ..., D\}, j \in \{1, 2, ..., K_i\}, \tag{7}$$

where p_{ij} denotes the set of points within the j-th bounding box of the i-th category, D is the total number of semantic categories of bounding boxes, and K_i is the total number of bounding boxes in category i. The background point cloud P_{bg}^{s-1} is defined as the set of all points in P^{s-1} that lie outside the bounding boxes. For each p_{ij}^{s-1}, we first adjust the viewpoint of the point cloud according to the coordinate transformation matrix of the sensor between the two frames:

$$\hat{p}_{ij}^{s-1} = trans(P_{ij}^{s-1}, E_{\text{rel}}^{s-1}), \tag{8}$$

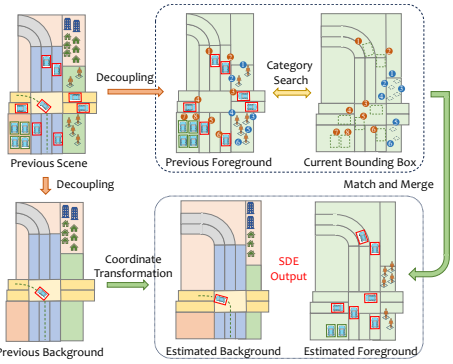

Fig. 4: The schematic diagram of scene decoupling estimation module.

To estimate the position of $\hat{p}_{ij}$ in the current-frame scene, we first compute the centers C_{ij}^{s-1} and C_{ij}^{s} of all bounding

boxes in the two adjacent frames. For each bounding box b_{ij}^s in the current scene, we search among the bounding boxes of the same semantic category in the previous frame for the one whose center is nearest to C_{ij}^s, denoted as $b_{ij'}^{s-1}$. Thus, we can obtain an estimate for each foreground object point cloud p_{ij}^s:

$$\widetilde{p}_{ij}^s = \hat{p}_{ij}^{s-1} + (C_{ij}^s - C_{ij'}^{s-1}), \quad i \in \{1, 2, ..., D\}. \tag{9}$$

Then, we aggregate all $\widetilde{p}_{ij}^s$ to obtain an estimate $\widetilde{P}_{\text{obj}}^s$ of the foreground point cloud P_{obj}^s in the current scene.

As the background estimation is performed in the ego-centric LiDAR coordinate system, the background point cloud is estimated as follows:

$$\widetilde{P}_{\text{bg}}^s = trans(P_{\text{bg}}^{s-1}, E_{\text{rotate}}^{s-1}). \tag{10}$$

Finally, we project both $\widetilde{P}_{\text{obj}}^s$ and $\widetilde{P}_{\text{bg}}^s$ into the range-view space and encode them using the VAE, after which the resulting features are injected into the diffusion process.

Compared with directly providing bounding box location information, the SDE module better captures object-level details. Mask information from bounding boxes only provides the model with a coarse understanding of each object's location within the scene, whereas SDE enables the model to better capture the detailed point distribution of each object. For a more detailed theoretical analysis of this module, please refer to the supplementary materials.

3.3.3. Noise Modulation Module and Error Accumulation Mitigation

Autoregressive generation, which relies on information from the previous frame, often leads to a train-inference discrepancy. Specifically, during training, the previous data is always ground truth, whereas during inference, it is generated by the model and inevitably contains some error compared to the real data. As the recursive generation continues, these errors can accumulate, resulting in deviations from the actual situation when generating long-horizon data.

To this end, we design a module to mitigate error accumulation and enhance temporal consistency in long-horizon generation. The core idea of this module is to inject different levels of random noise into the previous frame's image [4, 7], thereby reducing the model's over-reliance on it.

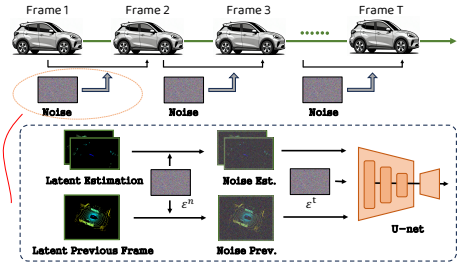

Fig. 5: The schematic diagram of noise modulation module.

As shown in Figure 5, we add Gaussian noise with varying intensities to the latent feature $\hat{z}^{s-1}$ corre-

sponding to the previous frame's LiDAR point cloud:

$$\hat{z}_{\epsilon}^{s-1} = \sqrt{\overline{\alpha}_n} \cdot \hat{z}^{s-1} + \sqrt{1 - \overline{\alpha}_n} \cdot \epsilon, \tag{11}$$

where $\epsilon \sim \mathcal{N}(0, 1)$ is randomly sampled Gaussian noise, and $n \in U[0, N]$ is the noise level. Similarly, we inject noise to the encoded features of $\widetilde{P}^s_{\text{obj}}$ and $\widetilde{P}^s_{\text{bg}}$.

This strategy ensures better spatiotemporal consistency for the model when inferring long-horizon scenes, while only causing minor impact on short-horizon generation. In the supplementary materials, we have provided a more detailed theoretical analysis of this module.

Upon integrating the two aforementioned modules into the overall architecture, we develop an inference framework for autoregressive LiDAR scene generation (see Figure 2). In this process, the input LiDAR data is generated iteratively; in other words, the LiDAR data input at the current step is the output from the generator at the preceding step. In the inference process, the output of the SDE module is also generated iteratively. Consistent with the training phase, during inference the NM module also adds noise into the latent variables corresponding to these data.

4 Experiments

4.1 Experimental Setups

Dateset. We conducted experiments on the large-scale public nuScenes dataset [2]. It contains 1,000 autonomous driving sequences collected from the Boston and Singapore regions. These two cities are known for their dense traffic and highly challenging driving conditions. The nuScenes training set contains 297,737 LiDAR scans, while the validation set contains 52,423 LiDAR scans, with each LiDAR scan comprising 32 beams. This dataset is widely used for various tasks in autonomous driving, including but not limited to 3D object detection [19, 29–31], multi-object tracking [15, 45, 75], trajectory prediction [12, 33], semantic occupancy prediction [60], and open-loop planning for end-to-end autonomous driving [17, 18, 21]. To verify the cross-dataset generalization ability of LaGen, we further conducted additional experiments on the denser-beam KITTI-360 dataset [10]. Unlike the nuScenes dataset, each LiDAR scan in the KITTI-360 dataset contains 64 beams.

Baselines. We compare LaGen with three categories of baseline methods. The first category consists of traditional LiDAR data generation approaches, including LiDARGen [76], LiDM [49], and RangeLDM [16], with which we compare generation fidelity. The second category includes methods capable of generating temporally coherent scenes, such as UniScene [26], OpenDWM [41], and LiDARCrafter [32]. In addition to comparing generation fidelity, we further evaluate temporal consistency metrics for each generated LiDAR frame. The third category comprises LiDAR prediction methods, including 4D-Occ [23] and ViDAR [69], against which we conduct comprehensive comparisons on prediction-related metrics in long-horizon forecasting tasks.

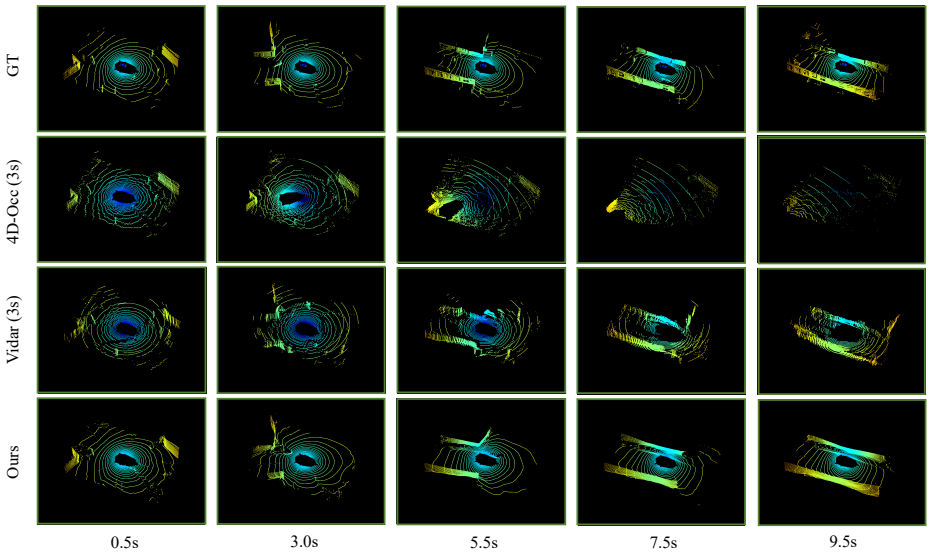

Fig. 6: Qualitative results. Based on single-frame input, LaGen significantly outperforms other baseline methods in long-horizon prediction accuracy.

Evaluation Metrics. We introduce multiple types of metrics to evaluate the fidelity and spatiotemporal consistency of the generated results. For evaluating the performance of the generator, we follow the approach in [76] and use two metrics: Maximum Mean Discrepancy (MMD) and Jensen-Shannon Divergence (JSD). Following LiDARCrafter [32], we employ TTCE and CTC as key metrics to assess the temporal consistency of our LiDAR sequence synthesis. We also use the Chamfer Distance (CD) [9] along with two additional error metrics based on ray depth, which quantify the difference between the generated point cloud for each frame and the corresponding ground truth in the driving scenario.
Implementation Details. During training, our model is trained for 200 epochs with a total of 176,000 optimization steps, the learning rate is 1e-4. We adopt the DDIM sampler [56] for denoising during training, with the total number of diffusion timesteps T set to 1024. The UNet consists of three downsampling blocks (with the latter two containing Transformer layers) and three upsampling blocks (with the first two containing Transformer layers). During the inference process, we perform 50 denoising steps to generate point clouds.

For more detailed descriptions of the metric computation, as well as the specific experimental configurations, please refer to the supplementary material.

4.2 LiDAR Data Generation

An important component of our approach is the LiDAR data generator, as the quality of its generated samples plays a crucial role in the overall framework performance. Table 1 and Table 2 present the quantitative results of LaGen and

Table 1: The main quantitative results of LaGen and several baseline models on the Nuscenes dataset. MMD values are reported in 10^{-4} and JSD in 10^{-2}.

	Method	Venue	MMD ↓	JSD ↓
Uncond.	LiDARGen [76]	ECCV'22	5.89	9.66
	LiDM [49]	CVPR'24	3.32	6.75
	RangeLDM [16]	ECCV'24	1.92	5.47
Cond.	UniScene [26]	CVPR'25	2.40	10.80
	OpenDWM [41]	CVPR'25	2.21	5.54
	DriveLiDAR4D [3]	AAAI'26	1.84	4.50
	LiDARCrafter [32]	AAAI'26	1.52	5.43
	LaGen	–	**0.11**	**2.77**

Table 2: The main quantitative results of LaGen and several baseline models on the KITTI-360 dataset. MMD values are reported in 10^{-4} and JSD in 10^{-2}.

Method	Venue	MMD↓	JSD↓	FRD↓	FPD↓
LiDARGen [76]	ECCV'22	13.36	13.67	2415.23	102.80
LiDM [49]	CVPR'24	19.00	9.31	2894.65	253.91
OpenDWM [41]	CVPR'25	14.63	12.93	1966.36	436.17
LaGen	–	**1.41**	**5.75**	**1109.83**	**73.50**

several LiDAR generation baselines on the nuScenes and KITTI-360 datasets, respectively. The experimental results show that LaGen outperforms these advanced baselines on several key metrics for LiDAR data generation. In addition, using a subset of valid 3D bounding boxes from the nuScenes dataset, we further investigate the performance of LaGen on the downstream task and its object-level generation quality, as shown in Table 3 and Table 4. These results demonstrate that LaGen can effectively generate high-fidelity LiDAR data.

4.3 Temporal LiDAR Scene Generation

In this subsection, we evaluate the performance of LaGen in generating sequential LiDAR scenes. Following the metrics established in LiDARCrafter [32], we conduct a comprehensive assessment of various methods across the dimension of temporal consistency, as summarized in Table 5. The results demonstrate that LaGen achieves a significant improvement in the CTC metric while maintaining a substantial advantage in terms of TTCE. In particular, LaGen can not only effectively generate long-sequence LiDAR scenes, but also maintain strong performance over extended generation horizons.

4.4 Long-Horizon Prediction

To further validate the long-horizon autoregressive performance of LaGen on other tasks, we designed two distinct modes and conducted experiments on predictive tasks. Specifically, we evaluate two variants: 1) a box-free LaGen baseline to assess the model's mastery of underlying scene dynamics, and 2) the

Table 3: Results on the downstream 3D object detection task on the nuScenes dataset.

Method	mAP↑	NDS↑
UniScene [26]	0.51	0.47
OpenDWM [41]	0.45	0.47
LiDARCrafter [32]	0.41	0.41
LaGen	**0.52**	**0.49**

Table 4: Evaluation results on the quality of objects from all categories and selected categories in the generated LiDAR scenes on the nuScenes dataset.

Method	Car		Pedestrian		All Objects	
	CD ↓	EMD ↓	CD ↓	EMD ↓	CD ↓	EMD ↓
LiDM [49]	3.563	3.763	0.701	0.663	3.236	4.241
UniScene [26]	1.316	1.419	0.237	0.358	0.939	1.245
LiDARCrafter [32]	3.657	2.581	0.636	0.557	3.347	3.517
LaGen	**1.251**	**0.986**	**0.123**	**0.129**	**0.855**	**0.822**

Table 5: The quantitative results on temporal consistency for 4D LiDAR generation, primarily evaluated using the TTCE and CTC metrics.

Method	Venue	TTCE↓		CTC↓				
		3	4	1	2	3	4	10
UniScene [26]	CVPR'25	2.74	3.69	0.90	1.84	3.64	3.90	-
OpenDWM [41]	CVPR'25	2.68	3.65	1.02	2.02	3.37	5.05	-
OpenDWM-DiT [41]	CVPR'25	2.71	3.66	0.89	1.79	3.06	4.64	-
LiDARCrafter [32]	AAAI'26	**2.65**	**3.56**	1.12	2.38	3.02	4.81	-
LaGen	–	2.71	**3.62**	**0.60**	**0.97**	**1.46**	**2.22**	**3.84**

full LaGen framework to demonstrate how integrating current feedback bolsters accuracy in long-term autoregressive forecasting. We have provided the results of several state-of-the-art predictive models for reference, as shown in Table 6. These results validate that LaGen yields outstanding performance under a predictive paradigm more conducive to practical deployment (see Figure 6).

4.5 Editability and Interactive Generation

In this subsection, we demonstrate LaGen's interactive generation capabilities. Since it generates scenes frame by frame, we can edit the positions of objects in the scene at any intermediate frame. In Figure 7, we illustrate interactive generation by moving or removing a bounding box corresponding to a vehicle. We observe that after moving the other vehicle, LaGen accurately restore the occlusion effects caused by the object's new position. Additionally, when the vehicle is removed, LaGen successfully completes the previously occluded portions of the road during standard generation. The interactivity of LaGen in autoregressive generation represents a significant ad-

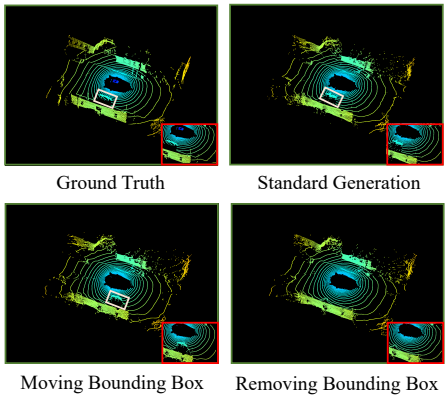

Fig. 7: Visualization of object-level editing.

Table 6: Comparison of LaGen with several state-of-the-art LiDAR prediction models. The underlined data represents the results obtained by introducing rolling inference on the original 4D-Occ model. ViDAR requires visual inputs during prediction.

Method	Input	Chamfer Distance (m^2)↓						L1 Error (m) ↓						Absrel (%)↓					
	frames	0.5s	1.5s	3.5s	5.5s	7.5s	9.5s	0.5s	1.5s	3.5s	5.5s	7.5s	9.5s	0.5s	1.5s	3.5s	5.5s	7.5s	9.5s
4D-Occ [23]	2	1.15	1.84	4.30	12.30	25.11	55.47	1.24	2.04	3.41	4.87	7.07	9.95	8.88	14.82	25.36	40.14	56.24	67.37
4D-Occ [23]	6	1.10	1.33	2.41	7.73	12.82	21.08	1.23	1.62	2.60	3.48	4.74	6.49	8.41	12.15	22.16	29.20	35.78	39.50
ViDAR [69]	2	1.15	1.56	2.50	3.79	5.18	6.37	2.38	2.70	3.34	4.24	5.40	6.06	16.29	19.96	27.34	36.25	48.61	54.13
ViDAR [69]	6	1.14	1.47	2.30	3.25	4.12	5.06	2.15	2.60	3.06	3.49	3.91	4.42	17.98	20.57	25.87	30.18	33.77	38.34
Ours(w/o BBox)	1	**0.65**	**0.93**	**1.44**	**1.98**	**2.39**	**2.78**	**0.14**	**0.20**	**0.26**	**0.31**	**0.36**	**0.40**	**2.15**	**2.66**	**3.23**	**3.73**	**4.22**	**4.42**
Ours(w/ BBox)	1	**0.50**	**0.72**	**1.25**	**1.69**	**2.09**	**2.34**	**0.13**	**0.18**	**0.24**	**0.28**	**0.30**	**0.32**	**2.04**	**2.47**	**3.03**	**3.40**	**3.63**	**3.61**

Table 7: The ablation results on the scene decoupling estimation module of LaGen.

LaGen	CTC↓				Chamfer Distance (m^2)↓					
	1	2	3	4	0.5s	1.0s	2.0s	4.0s	6.0s	8.0s
w/o SDE	0.71	1.15	1.66	2.49	0.56	0.66	0.95	1.42	1.87	2.23
w/ SDE	**0.60**	**0.97**	**1.46**	**2.22**	**0.50**	**0.61**	**0.83**	**1.36**	**1.80**	**2.16**

Table 8: The ablation results on the noise modulation module of LaGen.

LaGen	CTC↓				Chamfer Distance (m^2)↓					
	1	2	3	4	4.0s	5.0s	6.0s	7.0s	8.0s	9.0s
w/o NM	0.73	1.21	1.73	2.77	1.38	1.79	2.09	2.50	2.82	3.16
w/ NM	**0.60**	**0.97**	**1.46**	**2.22**	**1.36**	**1.66**	**1.80**	**2.02**	**2.16**	**2.25**

vancement, enabling it to generate more realistic 4D scenes according to changes in the external environment. Moreover, this framework naturally supports diverse LiDAR scene generation, as different possible outcomes can be achieved via stochastic or user-defined editing.

4.6 Ablation Study

We perform ablation studies to evaluate the contributions of the scene decoupling estimation and the noise modulation modules. As shown in Table 7, the SDE module effectively enhances temporal consistency between consecutive frames during scene generation. In addition, it improves the accuracy of long-horizon autoregressive tasks to a certain extent. Table 8 demonstrates that the NM module enhances temporal consistency and markedly alleviates error accumulation in long-horizon LiDAR scene generation, with increasingly significant improvements over extended time horizons.

We also investigate the impact of different denoising steps during inference on both accuracy and latency (see Table 9). As the number of denoising steps increases, the performance on long-horizon generation tasks improves to some degree, at the cost of increased average inference time per frame. Notably, the marginal performance gains decrease as the number of denoising steps becomes larger. When near-real-time inference performance is required, we can choose a smaller number of denoising steps. As shown in Table 9, using fewer denoising steps can substantially improve inference speed while only slightly affecting model performance. Finally, we conducted ablation experiments at the inference stage by randomly perturbing and dropping the bounding box information to evaluate the robustness of the model; see the supplementary material for details.

Table 9: The ablation results of LaGen on the number of denoising steps and the corresponding time consumption during inference.

Denoising	Chamfer Distance (m^2)						Time
Steps	0.5s	1.0s	3.0s	5.0s	7.0s	9.5s	(s/frame)
10	0.64	0.79	1.44	2.12	2.35	2.60	0.21
30	0.52	0.64	1.16	1.83	2.09	2.41	0.33
50	0.50	0.61	1.11	1.66	2.02	2.34	0.43
100	0.50	0.59	1.06	1.72	2.12	2.31	0.73
200	0.50	0.59	1.03	1.51	2.10	2.20	1.44

4.7 Limitation and Discussion

LaGen still has the following limitations. First, LaGen mainly targets common single-return spinning mechanical LiDAR and does not explicitly model multi-return LiDAR sensors. Since multiple returns still lie along the same ray, a possible future extension to this setting would be to assign a separate depth channel for each return in the range-view representation. For non-spinning LiDAR, new representations of LiDAR scenes need to be further explored. Second, in the SDE module, to ensure consistency across different task settings, we adopt a nearest-distance-based search mechanism. However, in crowded scenes with multiple objects of the same category, this may affect the effectiveness of the module to some extent. In the future, tracklets could be used in simulation tasks to match object boxes across adjacent frames.

5 Conclusion

We propose LaGen, a novel framework capable of autoregressively generating long-horizon LiDAR scenes in a frame-by-frame manner. We develop a LiDAR data generator conditioned on multiple control conditions within the driving scenario. We further enhance the spatiotemporal consistency of generation through the scene decoupling estimation module and the noise modulation module. We thoroughly evaluate the performance of LaGen on the nuScenes and KITTI-360 datasets. The experimental results demonstrate the effectiveness of this framework and its advantages in long-horizon generation tasks. In the future, we expect LaGen to play an important role in applications such as closed-loop simulation and world modeling, thereby advancing the development of autonomous driving.

Acknowledgments

This research was supported by the Scientific Research Innovation Capability Support Project for Young Faculty (U40) of the Ministry of Education of China (SRICSPYF-ZY2025019).

References

1. Caccia, L., Van Hoof, H., Courville, A., Pineau, J.: Deep generative modeling of lidar data. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 5034–5040. IEEE (2019)
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
3. Cai, K., Liu, X., Zhou, X., Hu, H., Xiang, J., Zhang, L., Zhang, X., Zhan, K., Zhan, Y., Lang, X.: Drivelidar4d: Sequential and controllable lidar scene generation for autonomous driving. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 2525–2533 (2026)
4. Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., Sitzmann, V.: Diffusion forcing: Next-token prediction meets full-sequence diffusion. Advances in Neural Information Processing Systems **37**, 24081–24125 (2024)
5. Cheng, W., Yin, J., Li, W., Yang, R., Shen, J.: Language-guided 3d object detection in point cloud for autonomous driving. arXiv preprint arXiv:2305.15765 (2023)
6. Chitta, K., Prakash, A., Jaeger, B., Yu, Z., Renz, K., Geiger, A.: Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. IEEE transactions on pattern analysis and machine intelligence **45**, 12878–12895 (2022)
7. Deng, B., Tucker, R., Li, Z., Guibas, L., Snavely, N., Wetzstein, G.: Streetscapes: Large-scale consistent street view generation using autoregressive video diffusion. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–11 (2024)
8. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: Conference on robot learning. pp. 1–16. PMLR (2017)
9. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 605–613 (2017)
10. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. International Journal of Robotics Research **32**(11), 1231 – 1237 (Sep 2013)
11. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
12. Gu, J., Hu, C., Zhang, T., Chen, X., Wang, Y., Wang, Y., Zhao, H.: Vip3d: End-to-end visual trajectory prediction via 3d agent queries. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5496–5506 (2023)
13. Guo, X., Wu, Z., Xiong, K., Xu, Z., Zhou, L., Xu, G., Xu, S., Sun, H., Wang, B., Chen, G., et al.: Genesis: Multimodal driving scene generation with spatio-temporal and cross-modal consistency. Advances in Neural Information Processing Systems **38**, 60431–60455 (2026)
14. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
15. Hu, H.N., Yang, Y.H., Fischer, T., Darrell, T., Yu, F., Sun, M.: Monocular quasi-dense 3d object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**, 1992–2008 (2022)
16. Hu, Q., Zhang, Z., Hu, W.: Rangeldm: Fast realistic lidar point cloud generation. In: European Conference on Computer Vision. pp. 115–135. Springer (2024)

17. Hu, S., Chen, L., Wu, P., Li, H., Yan, J., Tao, D.: St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In: European Conference on Computer Vision. pp. 533–549. Springer (2022)
18. Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al.: Planning-oriented autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17853–17862 (2023)
19. Huang, J., Huang, G., Zhu, Z., Ye, Y., Du, D.: Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. arXiv preprint arXiv:2112.11790 (2021)
20. Jia, X., Yang, Z., Li, Q., Zhang, Z., Yan, J.: Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. Advances in Neural Information Processing Systems **37**, 819–844 (2024)
21. Jiang, B., Chen, S., Xu, Q., Liao, B., Chen, J., Zhou, H., Zhang, Q., Liu, W., Huang, C., Wang, X.: Vad: Vectorized scene representation for efficient autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8340–8350 (2023)
22. Karnchanachari, N., Geromichalos, D., Tan, K.S., Li, N., Eriksen, C., Yaghoubi, S., Mehdipour, N., Bernasconi, G., Fong, W.K., Guo, Y., et al.: Towards learning-based planning: The nuplan benchmark for real-world autonomous driving. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 629–636. IEEE (2024)
23. Khurana, T., Hu, P., Held, D., Ramanan, D.: Point cloud forecasting as a proxy for 4d occupancy forecasting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1116–1124 (2023)
24. Kingma, D., Salimans, T., Poole, B., Ho, J.: Variational diffusion models. Advances in neural information processing systems **34**, 21696–21707 (2021)
25. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
26. Li, B., Guo, J., Liu, H., Zou, Y., Ding, Y., Chen, X., Zhu, H., Tan, F., Zhang, C., Wang, T., et al.: Uniscene: Unified occupancy-centric driving scene generation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 11971–11981 (2025)
27. Li, X., Yin, J., Li, W., Xu, C., Yang, R., Shen, J.: Di-v2x: Learning domain-invariant representation for vehicle-infrastructure collaborative 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 3208–3215 (2024)
28. Li, X., Yin, J., Shi, B., Li, Y., Yang, R., Shen, J.: Lwsis: Lidar-guided weakly supervised instance segmentation for autonomous driving. In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 1433–1441 (2023)
29. Li, Y., Yu, Z., Philion, J., Anandkumar, A., Fidler, S., Jia, J., Alvarez, J.: End-to-end 3d tracking with decoupled queries. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18302–18311 (2023)
30. Li, Y., Ge, Z., Yu, G., Yang, J., Wang, Z., Shi, Y., Sun, J., Li, Z.: Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 1477–1485 (2023)
31. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Yu, Q., Dai, J.: Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)

32. Liang, A., Liu, Y., Yang, Y., Lu, D., Li, L., Kong, L., Zhao, H., Ooi, W.T.: Lidarcrafter: Dynamic 4d world modeling from lidar sequences. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 18406–18414 (2026)
33. Liang, M., Yang, B., Zeng, W., Chen, Y., Hu, R., Casas, S., Urtasun, R.: Pnpnet: End-to-end perception and prediction with tracking in the loop. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11553–11562 (2020)
34. Liu, X., Qi, C.R., Guibas, L.J.: Flownet3d: Learning scene flow in 3d point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 529–537 (2019)
35. Liu, Y., Kong, L., Yang, W., Li, X., Liang, A., Chen, R., Fei, B., Liu, T.: La la lidar: Large-scale layout generation from lidar data. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 7377–7385 (2026)
36. Luo, Z., Ma, J., Zhou, Z., Xiong, G.: Pcpnet: An efficient and semantic-enhanced transformer network for point cloud prediction. IEEE Robotics and Automation Letters **8**(7), 4267–4274 (2023)
37. Malavazi, F.B., Guyonneau, R., Fasquel, J.B., Lagrange, S., Mercier, F.: Lidar-only based navigation algorithm for an autonomous agricultural robot. Computers and electronics in agriculture **154**, 71–79 (2018)
38. Meng, Q., Wang, W., Zhou, T., Shen, J., Van Gool, L., Dai, D.: Weakly supervised 3d object detection from lidar point cloud. In: European Conference on computer vision. pp. 515–531. Springer (2020)
39. Mersch, B., Chen, X., Behley, J., Stachniss, C.: Self-supervised point cloud prediction using 3d spatio-temporal convolutional networks. In: Conference on Robot Learning. pp. 1444–1454. PMLR (2022)
40. Nakashima, K., Kurazume, R.: Lidar data synthesis with denoising diffusion probabilistic models. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 14724–14731. IEEE (2024)
41. Ni, J., Guo, Y., Liu, Y., Chen, R., Lu, L., Wu, Z.: Maskgwm: A generalizable driving world model with video mask reconstruction. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 22381–22391 (2025)
42. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)
43. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International conference on machine learning. pp. 8162–8171. PMLR (2021)
44. Pal, K., Sharma, A., Sharma, A., Krishna, K.M.: Atppnet: Attention based temporal point cloud prediction network. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 11140–11146. IEEE (2024)
45. Pang, Z., Li, Z., Wang, N.: Simpletrack: Understanding and rethinking 3d multi-object tracking. In: European Conference on Computer Vision. pp. 680–696. Springer (2022)
46. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: Film: Visual reasoning with a general conditioning layer. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
47. Prakash, A., Chitta, K., Geiger, A.: Multi-modal fusion transformer for end-to-end autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7077–7087 (2021)
48. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)

49. Ran, H., Guizilini, V., Wang, Y.: Towards realistic scene generation with lidar diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14738–14748 (2024)
50. Resop, J.P., Lehmann, L., Hession, W.C.: Drone laser scanning for modeling riverscape topography and vegetation: Comparison with traditional aerial lidar. Drones **3**(2), 35 (2019)
51. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
52. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in neural information processing systems **35**, 36479–36494 (2022)
53. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10529–10538 (2020)
54. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 770–779 (2019)
55. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. pp. 2256–2265. pmlr (2015)
56. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
57. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. Advances in neural information processing systems **32** (2019)
58. Song, Y., Ermon, S.: Improved techniques for training score-based generative models. Advances in neural information processing systems **33**, 12438–12448 (2020)
59. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
60. Tong, W., Sima, C., Wang, T., Chen, L., Wu, S., Deng, H., Gu, Y., Lu, L., Luo, P., Lin, D., et al.: Scene as occupancy. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8406–8415 (2023)
61. Weiss, U., Biber, P.: Plant detection and mapping for agricultural robots using a 3d lidar sensor. Robotics and autonomous systems **59**(5), 265–273 (2011)
62. Weng, X., Ivanovic, B., Wang, Y., Wang, Y., Pavone, M.: Para-drive: Parallelized architecture for real-time autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15449–15458 (2024)
63. Weng, X., Nan, J., Lee, K.H., McAllister, R., Gaidon, A., Rhinehart, N., Kitani, K.M.: S2net: Stochastic sequential pointcloud forecasting. In: European Conference on Computer Vision. pp. 549–564. Springer (2022)
64. Weng, X., Wang, J., Levine, S., Kitani, K., Rhinehart, N.: Inverting the pose forecasting pipeline with spf2: Sequential pointcloud forecasting for sequential pose forecasting. In: Conference on robot learning. pp. 11–20. PMLR (2021)
65. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Pontes, J.K., et al.: Argoverse 2: Next generation datasets for self-driving perception and forecasting. arXiv preprint arXiv:2301.00493 (2023)
66. Wu, Y., Wang, Y., Zhang, S., Ogai, H.: Deep 3d object detection networks using lidar data: A review. IEEE Sensors Journal **21**(2), 1152–1171 (2020)

67. Xiong, Y., Ma, W.C., Wang, J., Urtasun, R.: Ultralidar: Learning compact representations for lidar completion and generation. arXiv preprint arXiv:2311.01448 (2023)
68. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10) (2018)
69. Yang, Z., Chen, L., Sun, Y., Li, H.: Visual point cloud forecasting enables scalable autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14673–14684 (2024)
70. Yin, J., Shen, J., Chen, R., Li, W., Yang, R., Frossard, P., Wang, W.: Is-fusion: Instance-scene collaborative fusion for multimodal 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14905–14915 (2024)
71. Yu, J., Li, X., Koh, J.Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., Wu, Y.: Vector-quantized image modeling with improved vqgan. arXiv preprint arXiv:2110.04627 (2021)
72. Zhai, M., Xiang, X., Lv, N., Kong, X.: Optical flow and scene flow estimation: A survey. Pattern Recognition **114**, 107861 (2021)
73. Zhang, J., Singh, S., et al.: Loam: Lidar odometry and mapping in real-time. In: Robotics: Science and systems. vol. 2, pp. 1–9. Berkeley, CA (2014)
74. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3836–3847 (2023)
75. Zhang, T., Chen, X., Wang, Y., Wang, Y., Zhao, H.: Mutr3d: A multi-camera tracking framework via 3d-to-2d queries. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4537–4546 (2022)
76. Zyrianov, V., Zhu, X., Wang, S.: Learning to generate realistic lidar point clouds. In: European Conference on Computer Vision. pp. 17–35. Springer (2022)