RangeLDM: Fast Realistic LiDAR Point Cloud Generation

Qianjiang Hu, Zhimin Zhang, and Wei Hu✉

Wangxuan Institute of Computer Technology, Peking University, Beijing, China
hqjpku@pku.edu.cn zm_zhang@stu.pku.edu.cn forhuwei@pku.edu.cn

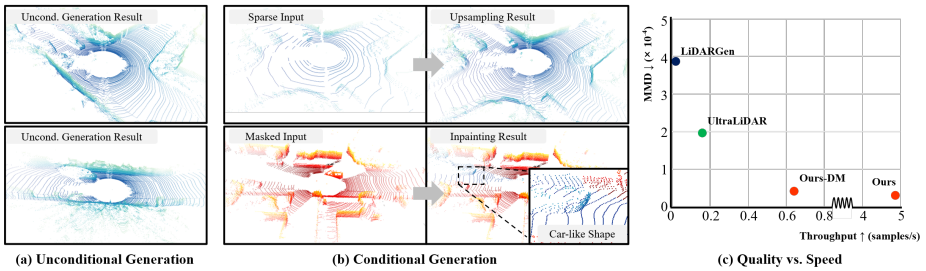

Fig. 1: (a). Unconditional LiDAR point cloud generation with realistic global structure. **(b).** Conditional LiDAR point cloud generation, including LiDAR point cloud upsampling and inpainting. **(c).** Generation quality (Maximum Mean Discrepancy, *abbr.* MMD) vs. generation speed (samples/s) of competitive LiDAR point cloud generation methods on the KITTI-360 [38] dataset. The proposed method outperforms the state-of-the-art methods LiDARGen [95] and UltraLiDAR [79] in both generation quality and generation speed. All speeds are evaluated on a single RTX 3090 GPU.

Abstract. Autonomous driving demands high-quality LiDAR data, yet the cost of physical LiDAR sensors presents a significant scaling-up challenge. While recent efforts have explored deep generative models to address this issue, they often consume substantial computational resources with slow generation speeds while suffering from a lack of realism. To address these limitations, we introduce RangeLDM, a novel approach for rapidly generating high-quality range-view LiDAR point clouds via latent diffusion models. We achieve this by correcting range-view data distribution for accurate projection from point clouds to range images via Hough voting, which has a critical impact on generative learning. We then compress the range images into a latent space with a variational autoencoder, and leverage a diffusion model to enhance expressivity. Additionally, we instruct the model to preserve 3D structural fidelity by devising a range-guided discriminator. Experimental results on KITTI-360 and nuScenes datasets demonstrate both the robust expressiveness and fast speed of our LiDAR point cloud generation.

Keywords: Diffusion models · Autonomous systems · Point clouds

✉ Corresponding author: W. Hu.

1 Introduction

Autonomous systems have drawn great attention in both the academia and the industry community. Numerous autonomous systems leverage the power of various sensors and deep learning to enhance the perception of the 3D world. LiDAR (**Li**ght **D**etection **A**nd **R**anging), with its ability to provide precise 3D geometric information about the surroundings, has become a popular sensor choice for autonomous systems, including self-driving cars [61,62,86], surveying drones [52,90] and robots [42, 78].

However, while LiDAR offers accurate geometric measurements, it comes with a significant limitation: the data collection process is exceedingly expensive and challenging to scale up. This renders physical sensors impractical for scalable and customizable data collection. Additionally, it is difficult to collect data in corner cases such as car accidents and extreme weather conditions in the field.

One approach to mitigate this issue is to employ existing LiDAR simulation toolkits [37, 43] to synthesize more data. Nevertheless, these systems typically demand manual scene creation or rely on multiple prior scans of the real world. Another approach is leveraging deep generative models to generate LiDAR point clouds. This is challenging due to the high degree of unstructuredness, sparsity, and non-uniformity in LiDAR point clouds. Caccia et al. [5] proposed LiDAR GAN and LiDAR VAE to generate LiDAR point clouds with generative adversarial networks (GANs) [21] and variational autoencoders (VAEs) [32], respectively. However, the generation results often suffer from issues such as fuzzy or missing details. LiDARGen [95] introduced a novel score-based model to synthesis LiDAR point clouds, which however sampled slowly and failed to generate high-quality geometric details at a far range. UltraLiDAR [79] proposed to synthesize voxelized point clouds in bird's-eyes-view (BEV) with vector quantized VAE (VQ-VAE) [74], resulting in more realistic point clouds than previous methods. Nevertheless, due to the large size and sparsity of point clouds in BEV, most computing power of UltraLiDAR is devoted to generating empty voxels, which results in a low generation speed.

To this end, we propose RangeLDM, a novel approach based on latent diffusion models (LDMs) [55], which improves both the quality and speed of LiDAR point cloud generation significantly. Firstly, we introduce a range image view to represent point clouds. The reasons are twofold: 1) Range images are compact and closely mimic the sampling conditions of LiDAR, making them a suitable representation. A LiDAR point cloud essentially constructs a 2.5D scene from a single viewpoint instead of a full 3D point cloud [27]. Consequently, organizing the point cloud in range view ensures that no information is overlooked. 2) 2D image generation techniques are relatively mature, providing a solid foundation for our work. While LiDARGen [95] also represents point clouds as range images, it suffers from blurriness caused by inaccurate projection. In contrast, we provide insights that *the correct range-view data distribution* has a critical impact on such range-view-based generative model learning, and accurately project point clouds onto range images using parameters estimated by Hough Voting. Then, to achieve a faster sampling speed [55] and greater expressivity [55,72], we compress

range images to the latent space with a VAE and develop a diffusion model operating on the latent space, leveraging the successful paradigm of LDMs [55,72,89]. Further, in order to enhance the ability of the VAE in reconstructing 3D structures, we introduce a range-guided discriminator, which is supervised from the spherical coordinates and thus geometry-sensitive. This plays a crucial role in guiding the decoder to generate high-quality range images, ultimately preserving the fidelity of the 3D structure. Extensive experimental results show that the proposed RangeLDM achieves the state-of-the-art performance on KITTI-360 [38] and nuScenes [6] datasets in terms of both the generation quality and generation speed, as demonstrated in Figure 1 (c). The LiDAR upsampling and LiDAR inpainting results also demonstrate the potential of the proposed method for conditional generation, as presented in Figure 1 (a) and (b).

To summarize, the main contributions of this paper include:

- We propose a latent diffusion model to capture the distribution of range-view LiDAR point clouds, aiming to generate realistic point cloud scenes at a fast speed.
- We enlighten the significance of the correct range-view data distribution for range-view-based generative models and achieve high-quality range image projection via Hough Voting.
- We exploit a range-guided discriminator to ensure preserving the fidelity of the 3D geometric structure in generated point clouds.
- Experimental results on the KITTI-360 and nuScenes datasets highlight that our approach outperforms state-of-the-art methods in terms of both visual quality and especially generation speed.

2 Related Work

LiDAR Representation. LiDAR point clouds can be represented in various forms, encompassing raw point clouds, voxels, range views, and multi-view fusion. **Point-based models** [49, 62, 64, 76, 84, 85, 91] directly encode 3D objects from raw points using the PointNet [50] encoder and subsequently perform detection or segmentation based on point features. These methods wholly preserve the irregularity and locality of a point cloud but have relatively higher latency. **Voxel-based methods** [13, 14, 17, 34, 35, 60, 63, 67, 77, 81, 82, 86, 94] voxelize point clouds for convolutional neural networks to efficiently capture features. They are computationally effective but the desertion of fine-grained patterns degrades further refinement. Given that the range view is compact and compatible with the LiDAR sensor's sampling process, **range-image-based methods** [2, 8, 10–12, 18, 33, 36, 44, 45, 68, 71, 80, 92] have been proposed to directly process range images for LiDAR perception. **Multi-view fusion methods** [22, 26, 61] amalgamate multiple representations, which yield better results at the expense of processing speed. In this paper, we adopt the range-view representation, which is congruent with the LiDAR sensor's sampling process and can be efficiently encoded by image generation models.

Generative Models for Point Clouds. Given observed samples of interest, generative models aim to learn the underlying distribution of the data and generate new samples. The first 3D generative models [1, 65, 73] for point clouds are based on GANs [21]. They operate by generating point clouds and discriminating them from real samples in an adversarial manner. The second category of methods [19, 20, 31, 40, 46, 47, 70] employ VAEs [32] to explore the probabilistic latent space of 3D shapes. Auto-regressive models [4] have also been introduced to generate point clouds [69] from scratch or semantic contexts. PointFlow [83] and SoftFlow [30] leverage normalizing flows [53] to capture the likelihood of shapes and points. Notably, the recent success of denoising diffusion models (DDMs) [25] for image synthesis [55] has also been extended to the domain of 3D point cloud generation [7, 28, 41, 48, 89, 93]. However, these previous works often concentrate on the object level and are not well-suited for handling large scenes.

In this paper, our focus lies in the generation of LiDAR point clouds. While [5] and [57] employ VAE or GANs for LiDAR point cloud generation, the realism achieved in their results is relatively limited. UltraLiDAR [79], on the other hand, utilizes VQ-VAE to generate voxelized LiDAR point clouds, but at the expense of introducing quantified losses and slow generation speed. LiDARGen [95] proposed a novel score-matching energy-based model to generate higher-quality LiDAR point clouds, but it samples very slowly and has degraded geometric details at a far range. A contemporary work LiDM [51] also utilized LDMs to generate LiDAR point clouds. However, while they focused on generating LiDAR point clouds conditional on multimodal data, we focus on improving the quality of point cloud generation directly.

3 Background on Denoising Diffusion Models

DDMs [25] are latent variable models that employ a pre-defined posterior distribution, known as the forward diffusion process, and are trained with a denoising objective. More specifically, given samples $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ from a data distribution, DDMs follow the form $p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T}$, where T denotes the number of steps, $\mathbf{x}_1, ..., \mathbf{x}_T$ are latent variables that gradually add noise to the data $\mathbf{x}_0$, and θ denotes the parameter set of the DDM decoder.

DDMs are trained by minimizing the evidence lower bound (ELBO) of the data $\mathbf{x}_0$ under $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})$. This objective can be simplified to [25]:

$$\min_{\theta} \mathbb{E}_{\mathbf{x}, \boldsymbol{\epsilon} \sim \mathcal{N}(0,1), t} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t + \boldsymbol{\epsilon}, t) \right\|_2^2 \right], \tag{1}$$

with t uniformly sampled from $\{1, ..., T\}$.

4 The Proposed Method

As shown in Fig. 2, we first project point clouds onto high-quality range images (as detailed in Section 4.1), considering the compact nature of the range view,

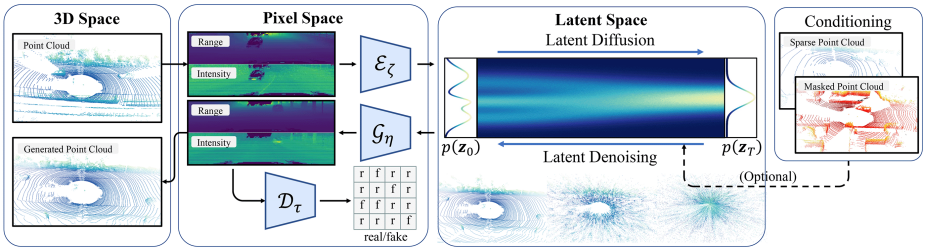

Fig. 2: The framework of the proposed RangeLDM. Firstly, we project point clouds onto high-quality range images via Hough Voting (Section 4.1). Subsequently, we train a VAE to compress the range images into low-dimensional latent features $\mathbf{z}_0$, which encodes the range images with the encoder $\mathcal{E}_\zeta$ and reconstructs range images from latent features with the decoder $\mathcal{G}_\eta$ (Section 4.2). Here, a range-guided discriminator $\mathcal{D}_\tau$ is introduced to guide the decoder in the reconstruction of 3D structures. We finally train a latent diffusion model to capture the distribution of the latent features (Section 4.2). With optional conditional inputs, the proposed method is applicable to tasks such as point cloud upsampling and inpainting (Section 4.3).

which aligns seamlessly with the sampling process of LiDAR sensors. Our model training process is then focused on these projected 2D range images. We then compress range images into a low-dimensional latent space through a VAE (as detailed in Section 4.2). Subsequently, we proceed to train DDMs within the reduced-dimensional latent space (as discussed in Section 4.2). We discuss applications of the proposed model in Section 4.3, which involves unconditional generation and conditional generation. Implementation details are explained in Section 4.4.

4.1 High-Quality Range Projection

The range image presents LiDAR data compactly and intuitively, with rows indicating the laser beams and columns representing the yaw angles. We convert point clouds to range images using spherical projection. Typically, for a point $\mathbf{p}$ in Cartesian coordinates (x, y, z), we calculate its spherical coordinates (r, θ, ϕ) using:

$$r = \sqrt{x^2 + y^2 + z^2}, \theta = \text{atan}(y, x), \phi = \text{atan}\left(z, \sqrt{x^2 + y^2}\right). \tag{2}$$

However, in most current datasets such as KITTI-360, multiple lasers from the Velodyne LiDAR system do not share a common origin for their measurements. This may introduce errors in the direct conversion from Cartesian points to spherical points, resulting in incorrect range-view data distribution and thus low-quality range images, as shown at the top of Fig. 3.

To address this issue, we adopt Hough Voting to estimate heights and pitch angles $\{h_j, \phi_j\}_{j=1,\dots,N}$ for Velodyne sensors [3]. We then adjust the point cloud transformation to a range image using

$$r = \sqrt{x^2 + y^2 + (z - h_j)^2}, \theta = \text{atan}(y, x), \phi = \phi_j, \tag{3}$$

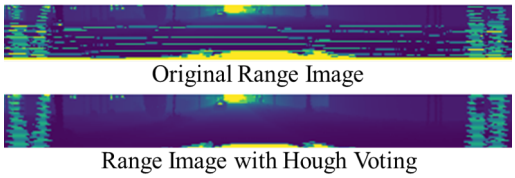

Fig. 3: Comparison of range projection by the typical method described in Eq. 2 and our method with Hough Voting as in Eq. 3.

Method	$\text{MMD}_{\text{BEV}} \downarrow$	FRD $\downarrow$	$\text{JSD}_{\text{BEV}} \downarrow$
LiDARGen	3.87×10^{-4}	2040.1	0.067
+Hough Voting	$\mathbf{1.41 \times 10^{-4}}$	**1453.1**	**0.064**

Table 1: Unconditional generation performance of LiDARGen [95] with and without Hough Voting. Hough Voting provides performance improvements through precise range image projection.

where h_j and ϕ_j refer to the j-th Velodyne sensor.

We then rasterize points (r, θ, ϕ) into a 2D cylindrical projection $R(u, v)$ (a.k.a., range image) of size $H \times W$ with $u = ((\theta + \pi)/2\pi) W, v = j$, where (u, v) denotes the grid coordinate of a point in the range image. Thus, we obtain high-quality range images as illustrated at the bottom of Fig. 3. We denote the obtained range image as $\mathbf{x} \in \mathbb{R}^{H \times W \times 2}$, which comprises $H \times W$ pixels associated with both range and intensity $\{r, i\}$.

To verify the impact of range projection, we trained range-based LiDAR generation methods like LiDARGen [95] with our obtained high-quality range images. Table 1 shows substantial improvement, which sheds light on the significance of the correct range-view data distribution for such range-based generative model learning.

4.2 Training

As a range image is often of high dimensionality at the scale of 64×1024, it would be computationally expensive to learn the distribution directly. Instead, we compress the range image into a low-dimensional latent space via a VAE and then model the distribution via an LDM. Consequently, the training of the proposed RangeLDM includes two distinct stages. In the first stage, we train a regular VAE to compress the range image into latent features $\mathbf{z}_0 \in \mathbb{R}^{h \times w \times c}$ by a downsampling factor $f = H/h = W/w$. In the second stage, we train an LDM to learn the distribution of the latent encoding $\mathbf{z}_0$.

First-Stage: Dimensionality Reduction. In the first stage, we reduce the dimensionality of the obtained range images by a VAE, which removes imperceptible high-frequency details and leads to low-dimensional latent features that encode prominent information in the original range image.

A standard VAE consists of two main components: 1) an encoder $\mathcal{E}_\zeta$ that transforms an input range image $\mathbf{x} \in \mathbb{R}^{H \times W \times 2}$ into latent features $\mathbf{z}_0 = \mathcal{E}_\zeta(\mathbf{x}) \in \mathbb{R}^{h \times w \times c}$ and 2) a decoder $\mathcal{G}_\eta$ that takes the latent features $\mathbf{z}_0$ as input and generates the reconstructed range image $\hat{\mathbf{x}} = \mathcal{G}_\eta(\mathbf{z}_0)$. The entire VAE is trained by maximizing a modified ELBO with respect to the parameters ζ and η [32,54]:

$$\mathcal{L}_{ELBO}(\zeta, \eta) = \mathbb{E}_{q_\zeta(\mathbf{z}_0|\mathbf{x})} \left[\log p_{\boldsymbol{\eta}}(\mathbf{x} \mid \mathbf{z}_0)\right] - \lambda_1 D_{\text{KL}} \left(q_{\boldsymbol{\zeta}}(\mathbf{z}_0 \mid \mathbf{x}) \| p(\mathbf{z}_0)\right). \quad (4)$$

In the above equation, the first term evaluates the reconstruction likelihood of the decoder from the latent $\mathbf{z}_0$ and corresponds to an L_1 reconstruction loss,

while the second term quantifies how closely the learned distribution of $\mathbf{z}_0$ resembles the prior distribution $p(\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_0; \mathbf{0}, \mathbf{I})$ held over latent variables. The hyperparameter λ_1 balances the reconstruction accuracy and Kullback-Leibler regularization.

Range-Guided Discriminator. In order to mitigate the blurriness introduced by relying solely on the pixel-space L_1 reconstruction loss, we integrate the VAE with a patch-based adversarial discriminator [15, 29, 55, 87]. Furthermore, to ensure that the reconstruction remains confined within the manifold underlying the range image and exploit the geometric information from the spherical coordinates, we propose a range-guided discriminator $\mathcal{D}_\tau$, with parameters τ. Specifically, we adapt the Meta-Kernel [18] to replace the standard convolution in the discriminator, aiming to learn convolution weights from relative spherical coordinates. The Meta-Kernel is formulated as

$$\mathbf{h}'_i = \mathcal{W}\left(\underset{j\in\mathcal{N}(i)}{\mathcal{A}} \left(\Phi\left(\gamma(\mathbf{p}_j, \mathbf{p}_i)\right) \odot \mathbf{h}_j\right)\right), \tag{5}$$

where $\mathbf{h}$ and $\mathbf{h}'$ represent the input feature and output feature, respectively. The function Φ denotes a Multi-Layer Perceptron (MLP) with two fully-connected layers, $\mathcal{A}$ is a concatenation operation, $\mathcal{N}(i)$ denotes the 2D-grid neighborhood around the center pixel i, and $\mathcal{W}$ is a fully-connected layer. The term $\gamma(\mathbf{p}_j, \mathbf{p}_i)$ represents the distance between points $\mathbf{p}_j$ and $\mathbf{p}_i$ in spherical coordinates, which is expressed as

$$\begin{gathered}\gamma\left(\mathbf{p}_j, \mathbf{p}_i\right) := \{r_j \cos(\Delta\theta)\cos(\Delta\phi) - r_i, r_j \cos(\Delta\theta)\sin(\Delta\phi), r_j \sin(\Delta\theta)\}, \\ \Delta\theta = \theta_j - \theta_i, \quad \Delta\phi = \phi_j - \phi_i.\end{gathered} \tag{6}$$

The Meta-Kernel is aware of local 3D structures by adaptively adjusting convolution kernel weights from relative spherical coordinates. This makes it challenging for the decoder to deceive the range-guided discriminator. This, in turn, encourages the decoder to generate more realistic range images.

With the range-guided discriminator, the final objective of the first-stage training is:

$$\zeta, \eta = \arg\min_{\zeta,\eta} \max_{\tau} \mathcal{L}_{ELBO}(\zeta, \eta) + \lambda_2 \mathcal{L}_{adv}(\zeta, \eta, \tau), \tag{7}$$

where $\mathcal{L}_{adv}$ is the GAN Hinge loss [39] and λ_2 balances the two losses.

Second-Stage: Latent Diffusion Modeling In the second stage, we freeze the encoder and decoder of the learned VAE and train an LDM on the encoded latent feature $\mathbf{z}_0$ for distribution modeling of the range image. During the diffusion process, noise is added to the initial latent $\mathbf{z}_0$, resulting in a noisy latent $\mathbf{z}_t$, where the noise level increases over time steps $t \in T$.

Regarding the modeling of the unconditional distribution $p(\mathbf{z}_0)$, based on Eq. 1, we learn a time-conditional UNet [56] $\boldsymbol{\epsilon}_\theta\left(z_t, t\right)$ that predicts the noise added to the noisy latent $\mathbf{z}_t$:

$$L_{\text{LDM}} := \mathbb{E}_{\mathcal{E}(x), \boldsymbol{\epsilon}\sim\mathcal{N}(0,1), t}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(z_t, t\right)\right\|_2^2\right]. \tag{8}$$

As to the modeling of the conditional distribution $p(\mathbf{z}_0|\mathbf{y})$, where $\mathbf{y}$ is a condition such as sparse or masked point clouds, we learn a conditional LDM via

$$L_{\text{LDM}} := \mathbb{E}_{\mathcal{E}(x),\mathbf{y},\boldsymbol{\epsilon}\sim\mathcal{N}(0,1),t}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\mathbf{z}_t, t, \tau_\theta(\mathbf{y})\right)\|_2^2\right], \tag{9}$$

where τ_θ is a condition encoder that projects $\mathbf{y}$ into an intermediate representation $\tau_\theta(\mathbf{y})$, and $\epsilon_\theta\left(\mathbf{z}_t, t, \tau_\theta(\mathbf{y})\right)$ is a UNet interpolated with cross-attention layers [55, 75].

4.3 Generation

Unconditional Generation. With the decoder $\mathcal{G}_\eta$ and the LDM ϵ_θ, we establish a hierarchical generative model $p_{\eta,\theta}(\mathbf{x}, \mathbf{z}_0) = p_\eta(\mathbf{x}|\mathbf{z}_0)p_\theta(\mathbf{z}_0)$. That is, we first generate latent features with the LDM, and then map the latent features back to the original range image space using the decoder $\mathcal{D}_\eta$ for LiDAR point cloud generation.

Conditional Generation. Given a condition $\mathbf{y}$, the conditional generative model is defined as $p_{\eta,\theta}(\mathbf{x}, \mathbf{z}_0|\mathbf{y}) = p_\eta(\mathbf{x}|\mathbf{z}_0, \mathbf{y})p_\theta(\mathbf{z}_0|\mathbf{y})$. There could be various conditions for LiDAR point cloud generation, such as sparse point clouds, incomplete point clouds, camera or even text. Here, we illustrate two applications of conditional generation: LiDAR point cloud upsampling and inpainting.

LiDAR Point Cloud Upsampling. This task takes a sparse point cloud[1] $\mathbf{y} \in \mathbb{R}^{h\times W\times 2}$ as input (following the setting of LiDARGen) and expects the model to produce a denser counterpart $\hat{\mathbf{x}} \in \mathbb{R}^{H\times W\times 2}$, which is necessary for accurate LiDAR-based perception and density-insensitive domain adaptation. We train LDM following Eq. 9 with the sparse point cloud $\mathbf{y}$ and its ground-truth dense variant $\mathbf{x}$. The condition encoder τ_θ is configured as a reshaper that transforms the sparse point cloud $\mathbf{y}$ into $\mathbf{y}' \in \mathbb{R}^{h\times w\times 2f}$. The reshaped $\mathbf{y}'$ is then concatenated with noisy latent variable $\mathbf{z}_t$ and fed into the UNet ϵ_θ for noise prediction.

LiDAR Point Cloud Inpainting. This task is required when LiDAR sensors cannot capture the entire scene due to object occlusion or sensor limitations. We train an inpainting LDM from the ground truth point cloud $\mathbf{x}$, a masked point cloud $\mathbf{x}'$, and the corresponding mask $\mathbf{m} \in \{0,1\}^{H\times W}$. The masked point cloud $\mathbf{x}'$ is encoded with the encoder $\mathcal{E}_\zeta$ from the VAE and is then concatenated with a downsampled mask $\mathbf{m}' \in \{0,1\}^{h\times w}$ as the condition term in Eq. 9. Mathematically, we represent the condition $\mathbf{y}$ as $\{\mathbf{x}', \mathbf{m}\}$, and deploy the condition encoder to project it into $\tau_\theta(\mathbf{y}) = \mathcal{E}_\zeta(\mathbf{x}')\|ds(\mathbf{m})$, where $\|$ denotes the concatenation operation and $ds(\cdot)$ indicates the downsampling operation. The projected condition is then concatenated with the noisy latent variable $\mathbf{z}_t$ and input into the UNet ϵ_θ for noise prediction.

[1] Without causing ambiguity, the point clouds are converted to range images by default.

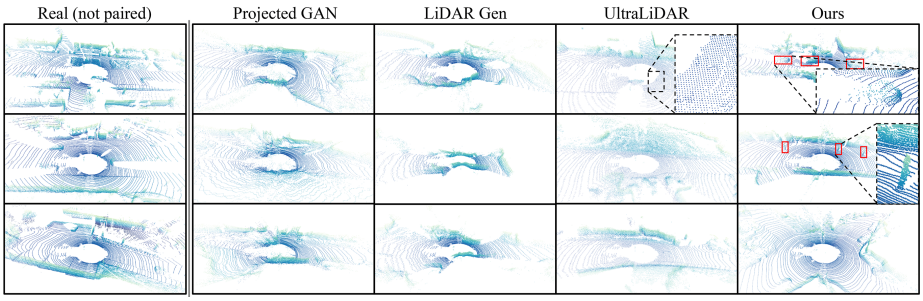

Fig. 4: Qualitative results comparing against baselines for unconditional LiDAR generation on KITTI-360. Real point clouds are only for reference. Our model produces results that closely resemble real-world data, which excels in generating road scenes, such as cars (the first row), road bollards (the second row) and crossroads (the last row).

4.4 Implementation Details

Circular Convolution. Range images are inherently circular, which means that the left boundary of a range image is actually connected to its right boundary. However, standard convolutions employ zero-padding or symmetry padding and do not take into account such constraints. Thus, as in LiDARGen [95], we replace standard convolutions in both the VAE and LDM with circular convolutions [59], which treat the left and right boundaries as connected neighbors in the topology. Since the 2D convolution operator satisfies translation invariance, it is evident that the entire network is invariant to horizontal shifting of the range image (*i.e.*, rotation in the xy-plane of the point cloud).

Conditioning the Model with Direction. Real LiDAR point clouds collected from driving scenes usually have a certain directionality. For example, since vehicles typically follow the direction of the road, the x-axis direction of the LiDAR point cloud usually aligns with the road direction. However, since the whole network of our model is *invariant* to rotation in the xy-plane of the point cloud, the point clouds generated from random noise could result in arbitrary directions. To avoid this, we generate point clouds with directional conditioning. Specifically, we use an $h \times w$ matrix as the condition, in which only the values from the first column are set to 1, while the rest are all set to 0. This $\mathbf{1}$-valued column corresponds to the x-axis of the point cloud. Such simple conditioning demonstrates effective control over the direction of the generated point cloud, as presented in the qualitative results of the supplementary material.

Network Architectures and Model Hyperparameters. For simplicity, we adopted a similar architecture to that in [55]. The downsampling factor f is set as 4 to strike a good balance between efficiency and realistic results. The VAE comprises three encoder blocks and three decoder blocks. The LDM consists of four downsampling blocks with the last three followed by a transformer layer, and four upsampling blocks with the first three blocks also followed by a transformer layer. In the generation process, we employed 50 denoising steps with DDIM sampler [66] for point cloud generation. More implementation details can be found in released code `https://github.com/WoodwindHu/RangeLDM`.

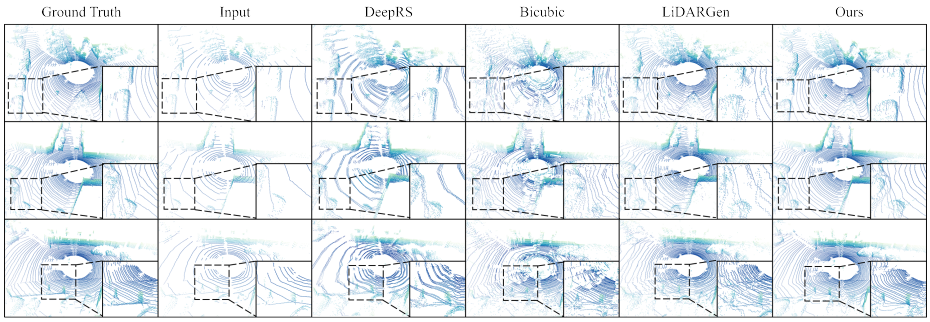

Fig. 5: Comparison of upsampling results on KITTI-360. We downsampled the ground truth by a factor of four as the input, and demonstrated the results of different methods on 4×-upsampling of the input.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate our model on two challenging datasets, KITTI-360 and nuScenes. KITTI-360 has 81,106 LiDAR readings from 9 sequences in Germany, covering diverse scenes. We used the first two sequences for validation and trained on the rest. NuScenes is a public dataset with 297,737 LiDAR sweeps in the training set and 52,423 in the testing set, collected in Boston and Singapore.
Evaluation metrics. Following [95], , we employed three metrics to perform quantitative analysis: Maximum Mean Discrepancy (MMD), Jensen-Shannon divergence (JSD), and Frechet Range Distance (FRD score). We use a 100×100 2D histogram on the BEV plane to calculate MMD and JSD metrics. The Frechet Range Distance (FRD) score [95] is a metric used to evaluate the quality of samples acquired by a generative model, inspired by the FID score for images [24]. To compute the FRD score, we use RangeNet++ [45], an encoder-decoder-based network for segmentation, which is pre-trained on KITTI-360.
Baselines. We evaluated our approach for LiDAR point cloud generation against several competitive methods, including LiDAR VAE [5], LiDAR GAN [5], Projected GAN [58], LiDARGen [95] and UltraLiDAR [79].

5.2 Unconditional Generation

Method	Years	$\text{MMD}_{\text{BEV}} \downarrow$	FRD ↓	$\text{JSD}_{\text{BEV}} \downarrow$
LiDAR GAN [5]	IROS 2019	3.06×10^{-3}	3003.8	-
LiDAR VAE [5]	IROS 2019	1.00×10^{-3}	2261.5	0.161
Projected GAN [58]	NeurIPS 2021	3.47×10^{-4}	2117.2	0.085
LiDARGen [95]	ECCV 2022	3.87×10^{-4}	2040.1	0.067
UltraLiDAR [79]	CVPR 2023	1.96×10^{-4}	-	0.071
Ours		$\mathbf{3.07 \times 10^{-5}}$	**1074.9**	**0.045**

Table 2: Unconditional generation results on KITTI-360 [38].

Quantitative Results on KITTI-360. Table 2 displays the quantitative results of the proposed RangeLDM and competing algorithms. The results clearly

illustrate that our method significantly outperforms the baselines across all metrics. Notably, our results on the MMD metric are an order of magnitude lower than those of other methods. This underscores the remarkable expressive capabilities of our model in the context of LiDAR point cloud generation.

Qualitative Results on KITTI-360. Figure 4 displays a set of randomly generated samples from competing algorithms, alongside real point cloud samples extracted from the KITTI-360 dataset for comparison. We observe that the Projected GAN [58] exhibits noticeable artifacts in the distant range. LiDARGen [95] effectively captures the general layout, but produces rather noisy outputs and lacks the same degree of straight walls as actual samples. UltraLiDAR [79] generates structured and reasonable scenes, yet it only creates voxelized point clouds (as magnified in the first row) without intensity features. In contrast, our model consistently surpasses the baseline models and yields results closely resembling real-world data. For example, we can generate cars (emphasized by red boxes in the first row of Figure 4) and road bollards (red boxes in the second row of Figure 4) on the road. Additionally, our model generates various scenes such as straightway (the first two rows of Figure 4) and crossroads (the last row of Figure 4). The supplementary materials provide more visualization results.

Human study on KITTI-360. To assess the perceptual quality, we conducted an A/B test involving a group of 14 researchers with LiDAR expertise. Following the same evaluation system as [95] and [79], we present pairs of randomly selected images from two point clouds and ask participants to determine which one appeared more realistic. The results, as displayed in Table 3, unequivocally demonstrate the superior visual quality of generation results by our model. In most of the cases, testers favored our results over the baselines.

Method	Percent prefer ours
Ours vs. VAE [5]	98.8%
Ours vs. GAN [5]	98.0%
Ours vs. ProjectedGAN [58]	93.2%
Ours vs. LiDARGen [95]	90.2%
Ours vs. UltraLiDAR [79]	86.5%

Table 3: Human study on KITTI-360.

Evaluation on the NuScenes Dataset. The results listed in Table 4 demonstrate that our model outperforms LiDAR VAE and LiDARGen significantly over nuScenes. In particular, our results on the MMD metric are an order of magnitude lower than those of other methods, while we reduce to about 1/3 of the results of competitive methods in terms of the JSD metric. Also, Figure 6 illustrates the superiority of our model over LiDAR VAE and LiDARGen in terms of qualitative comparison.

Method	$\text{MMD}_{\text{BEV}} \downarrow$	$\text{JSD}_{\text{BEV}} \downarrow$
LiDAR VAE [5]	1.1×10^{-3}	-
LiDARGen† [95]	1.9×10^{-3}	0.160
Ours	$\mathbf{1.9 \times 10^{-4}}$	**0.054**

Table 4: Unconditional generation results on nuScenes dataset [6]. †: Reproduced by us.

Model Size. The number of parameters in LiDARGen [95] and UltraLiDAR [79] are 29.7M and 40.3M, respectively. To ensure fair competition, we limited the capacity of our model to be similar to other approaches. Our model comprises 41.4M parameters, consisting of 12.7M parameters for VAE and 28.7M parameters for LDM. As illustrated in Table 2, our model outperforms the baselines by a large margin under similar model sizes.

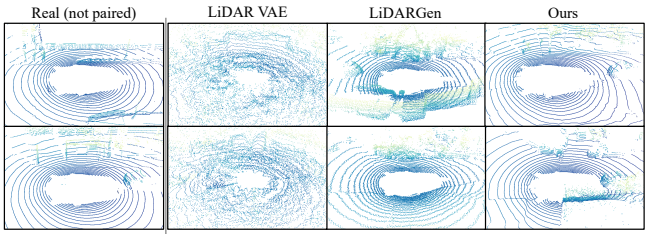

Fig. 6: Qualitative results comparing against LiDAR VAE and LiDARGen for unconditional LiDAR generation on nuScenes. Real point clouds are only for reference.

	Hough Voting	Range-Guided Discriminator	Circular Convolution	Direction Conditioned	$\text{MMD}_{\text{BEV}} \downarrow$	FRD ↓	$\text{JSD}_{\text{BEV}} \downarrow$
(a)	×	×	×	×	3.95×10^{-4}	1536.7	0.067
(b)	✓	×	×	×	6.57×10^{-5}	1229.3	0.056
(c)	✓	✓	×	×	4.72×10^{-5}	1103.1	0.051
(d)	✓	✓	✓	×	3.90×10^{-4}	1797.2	0.078
(e)	✓	✓	✓	✓	$\mathbf{3.07 \times 10^{-5}}$	**1074.9**	**0.045**

Table 6: Main ablation study.

Generation Efficiency. The proposed dimensionality reduction offers RangeLDM a considerable advantage in terms of generation speed. As indicated in Table 5, the proposed method significantly outperforms LiDARGen [95] and UltraLiDAR [79] in terms of generation speed. In particular, our model is 200 times faster than LiDARGen and 30 times faster than UltraLiDAR.

Method	Throughput ↑ (samples/s)
LiDARGen [95]	0.02
UltraLiDAR [79]	0.16
Ours	**4.86**

Table 5: Inference speed on a single RTX 3090 GPU.

5.3 Conditional Generation

To evaluate the conditional generation performance of RangeLDM, we conducted experiments on the KITTI-360 dataset over two tasks: LiDAR point cloud upsampling and inpainting.

LiDAR Point Cloud Upsampling. We obtain sparse input point clouds by selecting a subset of 16 beams from the raw 64-beam sensors, in line with LiDARGen. We compared our approach against PUNet [88], DeepRS [9], Grad-PU [23], bicubic interpolation, Nearest Neighbor (NN) interpolation and LiDARGen.

Quantitatively, in addition to measuring Mean Absolute Error (MAE) in the range view, we also employed RangeNet++ semantic segmentation to evaluate the quality of upsampled results. As shown in Table 7, the proposed method outperforms competing methods in all metrics, including MAE, per-point segmentation accuracy, and segmentation Intersection over Union (IoU).

Method	Years	MAE ↓	Accuracy ↑	IoU ↑
PUNet [88]	2018	6.88	-	-
DeepRS [9]	2022	3.96	-	-
Grad-PU [23]	2023	5.09	-	-
Bicubic	-	2.60	0.265	0.166
NN	-	2.18	0.546	0.394
LiDARGen [95]	2022	1.23	0.608	0.449
Ours		**0.89**	**0.722**	**0.566**

Table 7: LiDAR upsampling.

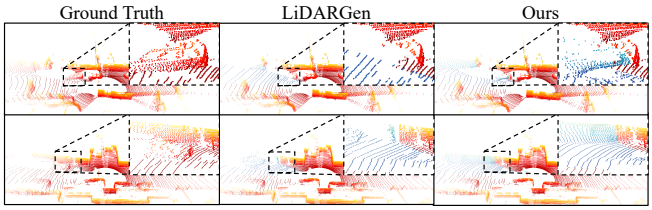

Fig. 7: Inpainting results. Left: ground truth point clouds. Middle and right (red): input point clouds. Middle and right (blue): recovered point clouds.

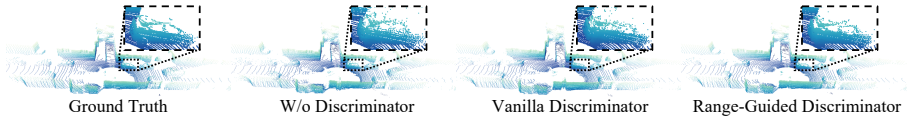

Fig. 8: VAE reconstruction results. The proposed VAE with a range-guided discriminator reconstructs point clouds with less noise and more precise object structures.

Qualitatively, the visualization comparison with several competitive methods presented in Figure 5 indicates that the proposed upsampling method exhibits results highly similar to the ground truth data, showing clear object details and consistent LiDAR scan lines. We provide additional visualization results in the supplementary materials.

LiDAR Point Cloud Inpainting. To mimic the case of missing data in LiDAR point clouds, we mask point clouds in front of the vehicle within a 22.5° range and perform inpainting using LiDARGen and our method. Our results achieve much less MAE (0.190) between the reconstructed results and ground truth compared to that of LiDARGen (0.367). As shown in Figure 7, LiDARGen generates roads but fails to recover the masked car (the first row) and the wall (the second row), while we are able to reconstruct the details. More results are presented in the supplementary materials.

5.4 Ablation Study

All ablation studies are conducted on the KITTI-360 dataset.

Main Ablation. As demonstrated in Table 6, we investigate the contribution of each component. Starting from the backbone model (a) without any component, we gradually add each component for evaluation. By applying Hough Voting to the training, the performance of variant (b) improves significantly, demonstrating the huge impact of the correct range-view data distribution. With the addition of the range-guided discriminator component, model (c) achieves better performance, thus clarifying the effectiveness of the geometry-sensitive module. Compared to model (c), the performance of model (d) decreases because circular convolution generates LiDAR scenes with *arbitrary* directions due to rotation invariance. However, when combined with direction-conditioned generation that guides the model to generate point clouds in the correct direction, the overall performance improves, as demonstrated by model (e). We refer the readers to the supplementary materials for qualitative results.

Model	VAE	Model Size (M)	$\text{MMD}_{\text{BEV}} \downarrow$	FRD $\downarrow$	$\text{JSD}_{\text{BEV}} \downarrow$	Throughput $\uparrow$ (samples/s)
Ours-DMS	×	40.3	1.03×10^{-4}	1392.2	0.062	1.52
Ours-DM	×	114.7	4.14×10^{-5}	**899.0**	**0.040**	0.64
Ours	✓	41.4	$\mathbf{3.07 \times 10^{-5}}$	1074.9	0.045	**4.86**

Table 8: Ablation of latent diffusion. Model without latent diffusion requires more parameters to achieve similar performance with the latent diffusion model. We generated 1000 samples for calculating the throughput.

Latent diffusion. We explore the contribution of latent diffusion to the generation quality and generation speed of the model. As shown in Table 8, we constructed two variants of different sizes that directly generate range images with diffusion models, denoted as "Ours-DMS" (with a smaller size) and "Ours-DM" (with a larger size), for comparison. It turns out that a much larger model size is required for "Ours-DM" to achieve competitive performance with "Ours", due to the multitude of high-frequency details contained in the range image. Additionally, the superior throughput of "Ours" demonstrates the efficiency of latent diffusion.

VAE Architecture. We investigate the contribution of the proposed range-guided discriminator by comparing our VAE with two baselines: 1) VAE without discriminator; and 2) VAE with vanilla discriminator (*i.e.*, using the original 2D convolution in the discriminator). We evaluate the reconstruction performance of VAEs with 2D and 3D metrics, including PSNR, MAE, FRD and Chamfer Distance (CD) [16]. As listed in Table 9, the VAE with the proposed range-guided discriminator outperforms the baselines in both 2D and 3D reconstruction metrics. This gives credits to the ability of perceiving local 3D structures by the range-guided discriminator. The qualitative results presented in Figure 8 also indicate that the proposed range-guided discriminator ensures the point clouds are reconstructed with less noise and more precise object structures.

Discriminator	$\text{PSNR}_{\text{range}} \uparrow$	MAE $\downarrow$	FRD $\downarrow$	CD $\downarrow$
No	26.77	0.0195	532.9	0.0808
Vanilla	26.70	0.0189	496.7	0.0726
Range-Guided	**27.19**	**0.0186**	**483.6**	**0.0676**

Table 9: Ablation on the VAE architecture.

6 Conclusions

We propose a novel RangeLDM model to generate realistic range-view LiDAR point clouds at a fast speed. We ensure the quality of projection from point clouds to range images with correct distribution via Hough Voting. Then we compress range images to latent features with a VAE, and train a diffusion model in the lower-dimensional latent space. Additionally, we enhance the range-image reconstruction quality of the VAE with a range-guided discriminator. Experiments conducted on KITTI-360 and nuScenes datasets demonstrate the superior generation quality and sampling efficiency of our method. In future, we will explore generating labeled point clouds and creating corner-case data, such as in car accidents and extreme weather conditions, for potential applications in robust self-driving.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3D point clouds. In: International Conference on Machine Learning. pp. 40–49. PMLR (2018)
2. Ando, A., Gidaris, S., Bursuc, A., Puy, G., Boulch, A., Marlet, R.: RangeViT: Towards vision transformers for 3D semantic segmentation in autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5240–5250 (2023)
3. Bewley, A., Sun, P., Mensink, T., Anguelov, D., Sminchisescu, C.: Range conditioned dilated convolutions for scale invariant 3D object detection. In: Conference on Robot Learning (2020)
4. Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time series analysis: forecasting and control. John Wiley & Sons (2015)
5. Caccia, L., Van Hoof, H., Courville, A., Pineau, J.: Deep generative modeling of LiDAR data. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 5034–5040. IEEE (2019)
6. 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)
7. Cai, R., Yang, G., Averbuch-Elor, H., Hao, Z., Belongie, S., Snavely, N., Hariharan, B.: Learning gradient fields for shape generation. In: European Conference on Computer Vision. pp. 364–381 (2020)
8. Chai, Y., Sun, P., Ngiam, J., Wang, W., Caine, B., Vasudevan, V., Zhang, X., Anguelov, D.: To the point: Efficient 3D object detection in the range image with graph convolution kernels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16000–16009 (2021)
9. Chen, H., Luo, S., Hu, W., et al.: Deep point set resampling via gradient fields. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(3), 2913–2930 (2022)
10. Chen, X., Vizzo, I., Läbe, T., Behley, J., Stachniss, C.: Range image-based LiDAR localization for autonomous vehicles. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 5802–5808. IEEE (2021)
11. Cheng, H., Han, X., Xiao, G.: CeNet: Toward concise and efficient LiDAR semantic segmentation for autonomous driving. In: IEEE International Conference on Multimedia and Expo (ICME). pp. 01–06. IEEE (2022)
12. Cortinhal, T., Tzelepis, G., Erdal Aksoy, E.: SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds. In: Advances in Visual Computing. pp. 207–222. Springer International Publishing, Cham (2020)
13. Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., Li, H.: Voxel R-CNN: Towards high performance voxel-based 3D object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 1201–1209 (2021)
14. Engelcke, M., Rao, D., Wang, D.Z., Tong, C.H., Posner, I.: Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 1355–1361. IEEE (2017)
15. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12873–12883 (2021)

16. 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/CVF Conference on Computer Vision and Pattern Recognition. pp. 605–613 (2017)
17. Fan, L., Pang, Z., Zhang, T., Wang, Y.X., Zhao, H., Wang, F., Wang, N., Zhang, Z.: Embracing single stride 3D object detector with sparse transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8458–8468 (2022)
18. Fan, L., Xiong, X., Wang, F., Wang, N., Zhang, Z.: RangeDet: In defense of range view for LiDAR-based 3D object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2918–2927 (2021)
19. Gao, L., Wu, T., Yuan, Y.J., Lin, M.X., Lai, Y.K., Zhang, H.: TM-NET: Deep generative networks for textured meshes. ACM Transactions on Graphics (TOG) **40**(6), 1–15 (2021)
20. Gao, L., Yang, J., Wu, T., Yuan, Y.J., Fu, H., Lai, Y.K., Zhang, H.: SDM-NET: Deep generative network for structured deformable mesh. ACM Transactions on Graphics (TOG) **38**(6), 1–15 (2019)
21. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
22. He, C., Li, R., Li, S., Zhang, L.: Voxel set transformer: A set-to-set approach to 3D object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8417–8427 (2022)
23. He, Y., Tang, D., Zhang, Y., Xue, X., Fu, Y.: Grad-PU: Arbitrary-scale point cloud upsampling via gradient descent with learned distance functions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5354–5363 (2023)
24. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
25. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33**, 6840–6851 (2020)
26. Hu, J.S., Kuai, T., Waslander, S.L.: Point density-aware voxels for LiDAR 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8469–8478 (2022)
27. Hu, P., Ziglar, J., Held, D., Ramanan, D.: What you see is what you get: Exploiting visibility for 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11001–11009 (2020)
28. Hui, K.H., Li, R., Hu, J., Fu, C.W.: Neural wavelet-domain diffusion for 3D shape generation. In: SIGGRAPH Asia 2022 Conference Papers. pp. 1–9 (2022)
29. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1125–1134 (2017)
30. Kim, H., Lee, H., Kang, W.H., Lee, J.Y., Kim, N.S.: SoftFlow: Probabilistic framework for normalizing flow on manifolds. Advances in Neural Information Processing Systems **33**, 16388–16397 (2020)
31. Kim, J., Yoo, J., Lee, J., Hong, S.: SetVAE: Learning hierarchical composition for generative modeling of set-structured data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15059–15068 (2021)

32. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
33. Kochanov, D., Nejadasl, F.K., Booij, O.: KprNet: Improving projection-based LiDAR semantic segmentation. arXiv preprint arXiv:2007.12668 (2020)
34. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: PointPillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12697–12705 (2019)
35. Li, B.: 3D fully convolutional network for vehicle detection in point cloud. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1513–1518. IEEE (2017)
36. Li, B., Zhang, T., Xia, T.: Vehicle detection from 3D LiDAR using fully convolutional network. arXiv preprint arXiv:1608.07916 (2016)
37. Li, C., Ren, Y., Liu, B.: Pcgen: Point cloud generator for lidar simulation. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 11676–11682. IEEE (2023)
38. Liao, Y., Xie, J., Geiger, A.: KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(3), 3292–3310 (2022)
39. Lim, J.H., Ye, J.C.: Geometric gan. arXiv preprint arXiv:1705.02894 (2017)
40. Litany, O., Bronstein, A., Bronstein, M., Makadia, A.: Deformable shape completion with graph convolutional autoencoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1886–1895 (2018)
41. Luo, S., Hu, W.: Diffusion probabilistic models for 3D point cloud generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2837–2845 (2021)
42. 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)
43. Manivasagam, S., Wang, S., Wong, K., Zeng, W., Sazanovich, M., Tan, S., Yang, B., Ma, W.C., Urtasun, R.: LiDARsim: Realistic LiDAR simulation by leveraging the real world. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11167–11176 (2020)
44. Meyer, G.P., Laddha, A., Kee, E., Vallespi-Gonzalez, C., Wellington, C.K.: LaserNet: An efficient probabilistic 3D object detector for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12677–12686 (2019)
45. Milioto, A., Vizzo, I., Behley, J., Stachniss, C.: RangeNet++: Fast and accurate LiDAR semantic segmentation. In: IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 4213–4220. IEEE (2019)
46. Mittal, P., Cheng, Y.C., Singh, M., Tulsiani, S.: AutoSDF: Shape priors for 3D completion, reconstruction and generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 306–315 (2022)
47. Mo, K., Guerrero, P., Yi, L., Su, H., Wonka, P., Mitra, N., Guibas, L.J.: StructureNet: Hierarchical graph networks for 3D shape generation. arXiv preprint arXiv:1908.00575 (2019)
48. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-E: A system for generating 3D point clouds from complex prompts. arXiv preprint arXiv:2212.08751 (2022)
49. Pan, X., Xia, Z., Song, S., Li, L.E., Huang, G.: 3D object detection with pointformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7463–7472 (2021)

50. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 652–660 (2017)
51. 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 (2024)
52. 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)
53. Rezende, D., Mohamed, S.: Variational inference with normalizing flows. In: International Conference on Machine Learning. pp. 1530–1538. PMLR (2015)
54. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: International Conference on Machine Learning. pp. 1278–1286. PMLR (2014)
55. 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)
56. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention. pp. 234–241. Springer (2015)
57. Sallab, A.E., Sobh, I., Zahran, M., Essam, N.: LiDAR sensor modeling and data augmentation with GANs for autonomous driving. arXiv preprint arXiv:1905.07290 (2019)
58. Sauer, A., Chitta, K., Müller, J., Geiger, A.: Projected GANs converge faster. Advances in Neural Information Processing Systems **34**, 17480–17492 (2021)
59. Schubert, S., Neubert, P., Pöschmann, J., Protzel, P.: Circular convolutional neural networks for panoramic images and laser data. In: IEEE Intelligent Vehicles Symposium (IV). pp. 653–660. IEEE (2019)
60. Shi, G., Li, R., Ma, C.: PillarNet: Real-time and high-performance pillar-based 3D object detection. In: European Conference on Computer Vision. pp. 35–52. Springer (2022)
61. 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)
62. 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)
63. Shi, S., Wang, Z., Shi, J., Wang, X., Li, H.: From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(8), 2647–2664 (2020)
64. Shi, W., Rajkumar, R.: Point-GNN: Graph neural network for 3D object detection in a point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1711–1719 (2020)
65. Shu, D.W., Park, S.W., Kwon, J.: 3D point cloud generative adversarial network based on tree structured graph convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3859–3868 (2019)
66. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)

67. Sun, P., Tan, M., Wang, W., Liu, C., Xia, F., Leng, Z., Anguelov, D.: SWFormer: Sparse window transformer for 3D object detection in point clouds. In: European Conference on Computer Vision. pp. 426–442. Springer (2022)
68. Sun, P., Wang, W., Chai, Y., Elsayed, G., Bewley, A., Zhang, X., Sminchisescu, C., Anguelov, D.: RSN: Range sparse net for efficient, accurate LiDAR 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5725–5734 (2021)
69. Sun, Y., Wang, Y., Liu, Z., Siegel, J., Sarma, S.: PointGrow: Autoregressively learned point cloud generation with self-attention. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 61–70 (2020)
70. Tan, Q., Gao, L., Lai, Y.K., Xia, S.: Variational autoencoders for deforming 3D mesh models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5841–5850 (2018)
71. Tian, Z., Chu, X., Wang, X., Wei, X., Shen, C.: Fully convolutional one-stage 3D object detection on LiDAR range images. Advances in Neural Information Processing Systems **35**, 34899–34911 (2022)
72. Vahdat, A., Kreis, K., Kautz, J.: Score-based generative modeling in latent space. Advances in Neural Information Processing Systems **34**, 11287–11302 (2021)
73. Valsesia, D., Fracastoro, G., Magli, E.: Learning localized generative models for 3D point clouds via graph convolution. In: International Conference on Learning Representations (2018)
74. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Advances in Neural Information Processing Systems **30** (2017)
75. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems **30** (2017)
76. Wang, H., Shi, S., Yang, Z., Fang, R., Qian, Q., Li, H., Schiele, B., Wang, L.: RBGNet: Ray-based grouping for 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1110–1119 (2022)
77. Wang, Y., Fathi, A., Kundu, A., Ross, D.A., Pantofaru, C., Funkhouser, T., Solomon, J.: Pillar-based object detection for autonomous driving. In: European Conference on Computer Vision. pp. 18–34. Springer (2020)
78. 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)
79. Xiong, Y., Ma, W.C., Wang, J., Urtasun, R.: Learning compact representations for LiDAR completion and generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1074–1083 (2023)
80. Xu, C., Wu, B., Wang, Z., Zhan, W., Vajda, P., Keutzer, K., Tomizuka, M.: SqueezeSegV3: Spatially-adaptive convolution for efficient point-cloud segmentation. In: European Conference on Computer Vision. pp. 1–19. Springer (2020)
81. Yan Yan, Y.M., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
82. Yang, B., Luo, W., Urtasun, R.: Pixor: Real-time 3D object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7652–7660 (2018)
83. Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B.: PointFlow: 3D point cloud generation with continuous normalizing flows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4541–4550 (2019)

84. Yang, Z., Sun, Y., Liu, S., Jia, J.: 3DSSD: Point-based 3D single stage object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11040–11048 (2020)
85. Yang, Z., Sun, Y., Liu, S., Shen, X., Jia, J.: STD: Sparse-to-dense 3D object detector for point cloud. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1951–1960 (2019)
86. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3D object detection and tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11784–11793 (2021)
87. 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)
88. Yu, L., Li, X., Fu, C.W., Cohen-Or, D., Heng, P.A.: Pu-Net: Point cloud upsampling network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2790–2799 (2018)
89. Zeng, X., Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K.: LION: Latent point diffusion models for 3D shape generation. Advances in Neural Information Processing Systems **35**, 10021–10039 (2022)
90. Zhang, J., Singh, S.: LOAM: LiDAR odometry and mapping in real-time. In: Robotics: Science and Systems. vol. 2, pp. 1–9. Berkeley, CA (2014)
91. Zhang, Y., Hu, Q., Xu, G., Ma, Y., Wan, J., Guo, Y.: Not all points are equal: Learning highly efficient point-based detectors for 3D LiDAR point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18953–18962 (2022)
92. Zhao, Y., Bai, L., Huang, X.: FidNet: LiDAR point cloud semantic segmentation with fully interpolation decoding. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4453–4458. IEEE (2021)
93. Zhou, L., Du, Y., Wu, J.: 3D shape generation and completion through point-voxel diffusion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5826–5835 (2021)
94. Zhou, Y., Tuzel, O.: VoxelNet: End-to-end learning for point cloud based 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4490–4499 (2018)
95. Zyrianov, V., Zhu, X., Wang, S.: Learning to generate realistic LiDAR point clouds. In: European Conference on Computer Vision. pp. 17–35. Springer (2022)