Taming Transformers for Realistic Lidar Point Cloud Generation

Hamed Haghighi[1] Amir Samadi[1] Mehrdad Dianati[2] Valentina Donzella[1] Kurt Debattista[1]

Abstract

Diffusion Models (DMs) have achieved State-Of-The-Art (SOTA) results in the Lidar point cloud generation task, benefiting from their stable training and iterative refinement during sampling. However, DMs often fail to realistically model Lidar raydrop noise due to their inherent denoising process. To retain the strength of iterative sampling while enhancing the generation of raydrop noise, we introduce LidarGRIT, a generative model that uses auto-regressive transformers to iteratively sample the range images in the latent space rather than image space. Furthermore, LidarGRIT utilises VQ-VAE to separately decode range images and raydrop masks. Our results show that LidarGRIT achieves superior performance compared to SOTA models on KITTI-360 and KITTI odometry datasets. Code available at:https://github.com/hamedhaghighi/LidarGRIT.

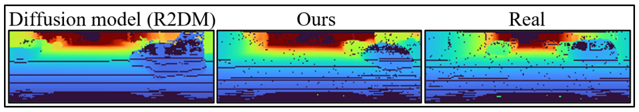
1. Introduction

Light detection and ranging (Lidar) is a critical sensor in autonomous vehicles, providing highly precise 3D environmental scanning. However, realistic simulation of the Lidar sensor poses challenges, involving cumbersome tasks such as creating 3D object models along with running computationally demanding physics-based algorithms. As an alternative, data-driven simulation models, particularly deep generative models have gained traction owing to their exceptional capacity to model high-dimensional data. Initially proposed for generating photo-realistic RGB images, deep generative models have been adapted for Lidar point cloud generation, progressing from early GAN-based frameworks [1] to the best-performing diffusion models [13].

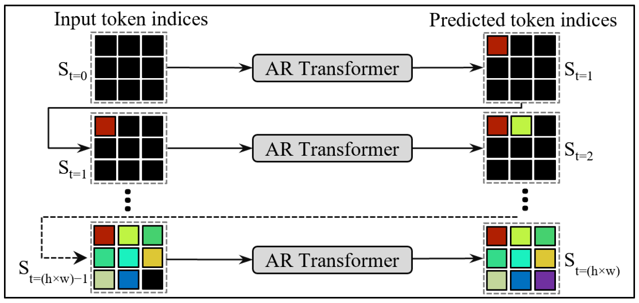
Diffusion models (DMs) for Lidar point cloud generation excel mainly due to their stable training and iterative refinement during the sampling process. While they demonstrate proficiency in capturing the 3D shape of point clouds,

[1]H. Haghighi, A.Samadi, K. Debattista and V. Donzella are with WMG, University of Warwick, Coventry, U.K. (Corresponding author: Hamed.Haghighi@warwick.ac.uk)

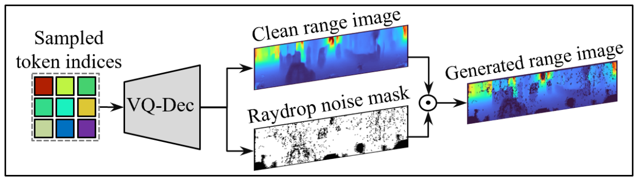
[2]M. Dianati is with the School of Electronics, Electrical Engineering and Computer Science at Queen's University of Belfast and WMG at the University of Warwick



(a)



(b)



(c)

Figure 1. (a) The range image generated by diffusion model (R2DM [8]) exhibits less realistic raydrop noise compared to our provided sample and the real one. (b) we propose to sample range image in the latent space via Auto-Regressive (AR) transformer [11]. (c) We then generate the raydrop mask and clean range image separately in the image space via VQ-VAE [2] decoder.

they face challenges in generating realistic Lidar raydrop noise, resulting in range images that appear unrealistic (refer to Figure 1a). This issue arises from the inherent denoising nature of DMs.

We introduce a novel Lidar Generative Range Image Transformer (LidarGRIT) model to incorporate both progressive generation and accurate raydrop noise synthesis. Our LidarGRIT works with the range image representation of Lidar point cloud, chosen for efficient processing and compatibility with image generative models. The generation process of the LidarGRIT consists of an iterative sampling in the latent space via Auto-Regressive (AR) transformer [11] (refer to Figure 1b), and decoding the sampled tokens to range images using an adapted Vector Quantised Variational Auto-Encoder (VQ-VAE [2]) model (refer to Figure 1c). We disentangle the generation of the range image from the raydrop noise mask in the VQ-VAE decoder, inspired by Dusty [7], and use a separate loss function to reconstruct clean range images and raydrop masks during training. Furthermore, we realised that large VQ-VAE models, primarily designed for high-resolution RGB images, tend to overfit when applied to relatively low-resolution range images. To address this, we propose geometric preservation, aiming to encourage the VQ-VAE to capture input geometry and provide more expressive latent tokens. We compare our LidarGRIT model to SOTA models on KITTI-360 and KITTI odometry datasets. Our model outperforms on nearly all metrics, specifically excelling in the image-based metrics, SWD [5]. The contributions of this paper can be summarised as follows:

- Introduction of LidarGRIT, a novel Lidar point cloud generative model consisting of a two-step generation process: iterative token index sampling through an AR transformer and single-pass range image decoding via an adapted VQ-VAE.
- Proposal of two novel techniques to enhance the generation quality: incorporating a separate raydrop estimation loss and enforcing geometry perseverance to increase VQ-VAE generalisability.
- Comprehensive evaluation of our LidarGRIT generation by comparing it with SOTA generative models using KITTI-360 and KITTI odometry datasets.

2. Related-Work

Caccia *et al.* [1] were among the first researchers to apply deep generative models to Lidar point clouds. They converted Lidar point clouds into range images and adapted the Deep Convolutional GAN (DCGAN) [10] for point cloud generation. Building on this, Dusty [7, 9] was proposed, which integrates raydrop synthesis into the GAN training process. Another notable model, UltraLiDAR [12], adopts a VQ-VAE framework to learn a discrete and compact Lidar representation for point cloud restoration and generation. With the recent achievement of DMs, LidarGen [13] and R2DM [8] models were proposed, relying on the score-based and denoising DMs frameworks, respectively.

Our approach draws inspiration from Dusty [7] framework in the disentanglement of raydrop and range image generation, however, we differentiate by employing a non-adversarial and more stable training using VQ-VAE [2], treating the raydrop estimation as a binary classification problem. Our LidarGRIT shares similarities with UltraLiDAR [12] in its two-stage sampling using VQ-VAE and AR transformer. However, rather than using voxelised Birds-Eye-View (BEV), we represent point clouds with range images that provide a lossless, more compact, and more computationally efficient format. Moreover, we focus on the raydrop noise generation and assess the point cloud generation on both image and point cloud representation.

3. Method

The designing process of LidarGRIT involves three steps. First, we represent the Lidar point clouds as range images (Section 3.1). Next, we tokenise range images using the VQ-VAE encoder and decode them separately to obtain clean range image along with the raydrop noise mask (Section 3.2). Finally, we capture the token interactions using the AR transformer (Section 3.3).

3.1. Data Representation

We employ different transformations to create range images for the KITTI-360 and KITTI-odometry datasets. For KITTI-360 generation, we use spherical projection, wherein each point in Cartesian coordinates $(x, y, z) \in \mathbb{R}^3$ is transformed into its spherical coordinates (r, θ, ϕ) as: $r = \sqrt{x^2+y^2+z^2}, \theta = atan(y,x), \phi = atan(z, \sqrt{x^2+y^2})$. We then quantise θ and ϕ into H and W bins with equal bin width, where H and W denote the vertical and horizontal angular resolutions of the Lidar sensor. This yields an image of size H $\times$ W with each pixel containing the range of its associated point. Regarding KITTI-odometry, we employ scan unfolding representation due to the sensor's non-linear vertical spacing [7]. We partition the ordered sequence into H sub-sequences, with each sub-sequence indicating one elevation angle. Throughout the paper, we denote the input range image as $\mathbf{x} \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}}$ and ground-truth raydrop mask as $\mathbf{x}_m \in \{0,1\}^{\mathrm{H}\times\mathrm{W}}$.

3.2. Adapting VQ-VAE

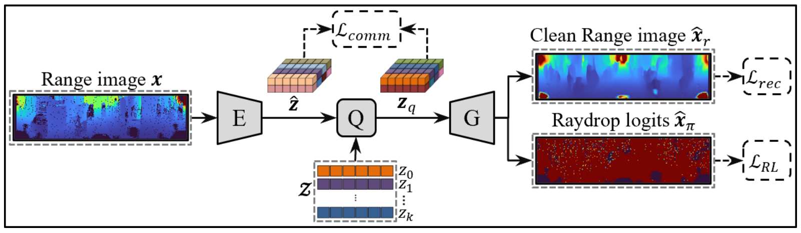
We use and adapt VQ-VAE [2] to auto-encode the range image and raydrop mask for three purposes: tokenisation, downsampling and raydrop noise generation. The VQ-VAE model consists of an encoder E, a quantiser Q and a decoder G. VQ-VAE encoder E downsamples the input range image $\mathbf{x}$ into latent image $\hat{\mathbf{z}} = E(\mathbf{x}) \in \mathbb{R}^{\mathrm{h}\times\mathrm{w}\times n_z}$. Quantiser Q generates tokens $\mathbf{z}_q \in \mathbb{R}^{\mathrm{h}\times\mathrm{w}\times n_z}$ using a learnable codebook $Z = \{z_k\}_{k=1}^{K} \subset \mathbb{R}^{n_z}$. Finally, VQ-VAE decoder G generates a clean range image $\hat{\mathbf{x}}_r \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}}$ and raydrop mask logits $\hat{\mathbf{x}}_\pi \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}}$ based on the quantised latent image $\mathbf{z}_q$ as $[\hat{\mathbf{x}}_r, \hat{\mathbf{x}}_\pi] = G(\mathbf{z}_q)$. The mask logits are then converted to the binary mask $\hat{\mathbf{x}}_m \in \{0,1\}^{\mathrm{H}\times\mathrm{W}}$ via sigmoid function and thresholding:

$$\hat{\mathbf{x}}_m = \begin{cases} 1 & \mathrm{sigmoid}(\hat{\mathbf{x}}_\pi) \geq 0.5 \\ 0 & \mathrm{sigmoid}(\hat{\mathbf{x}}_\pi) < 0.5. \end{cases} \tag{1}$$

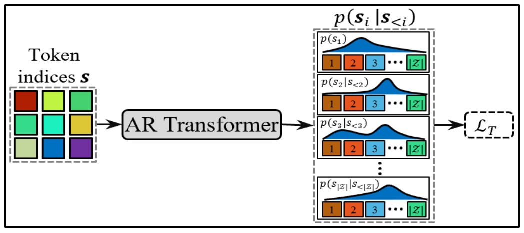
The final generated range image can be obtained as $\hat{\mathbf{x}} = \hat{\mathbf{x}}_m \odot \hat{\mathbf{x}}_r$.

3.2.1 Training–Raydrop Loss

We separate the training objectives for range image and raydrop mask generation. To encourage VQ-VAE decoder G to



(a) Adapted VQ-VAE model



(b) AR transformer model

Figure 2. Overview of the training process.

approximate the input range image, we use masked absolute error loss:

$$\mathcal{L}_{\text{rec}}(E, G) = \mathbb{E}_{\mathbf{x}}\Big[\frac{1}{\text{H} \times \text{W}}\|\mathbf{x}_m \odot (\mathbf{x} - \hat{\mathbf{x}}_r)\|_1\Big] \qquad (2)$$

This induces the decoder's range image channel to focus only on estimating the range of existing points. On the other hand, the mask channel is enforced to estimate raydrop noise via raydrop loss:

$$\begin{aligned}\mathcal{L}_{\text{RL}}(E, G) = \mathbb{E}_{\mathbf{x}}\big[&\text{Avg}[\mathbf{x}_m \odot \log(\text{sigmoid}(\hat{\mathbf{x}}_\pi))\\ &+ (1 - \mathbf{x}_m) \odot \log(1 - \text{sigmoid}(\hat{\mathbf{x}}_\pi))]\big],\end{aligned} \qquad (3)$$

Where Avg[$\cdot$] function calculates the average of the input across all its elements. To align the encoder and codebook embeddings, we train VQ-VAE with so-called commitment loss:

$$\mathcal{L}_{\text{com}}(E, Z) = \mathbb{E}_{\mathbf{x}}\big[\|\text{sg}[\hat{\mathbf{z}}] - \mathbf{z}_q\|_2^2 + \|\text{sg}[\mathbf{z}_q] - \hat{\mathbf{z}}\|_2^2\big].$$

The sg[$\cdot$] operator denotes the straight-through gradient estimator, ensuring that the quantisation process remains differentiable. Total training loss for the adapted VQ-VAE can be calculated as:

$$\mathcal{L}_{\text{VQ-VAE}} = \mathcal{L}_{\text{rec}}(E, G) + \lambda\mathcal{L}_{\text{RL}}(E, G) + \mathcal{L}_{\text{com}}(E, Z). \quad (4)$$

By tuning λ, we can establish a trade-off between the realism of range image generation and that of raydrop mask. We set $\lambda = 0.1$ in this study. We show the training process of the adapted VQ-VAE in Figure 2a.

3.2.2 Training–Geometric Perseverance

We observed that the VQ-VAE models, primarily designed for high-resolution RGB images, are prone to overfitting when dealing with relatively low-dimensional range images. This often results in less expressive latent codes. To mitigate the issue, we randomly distort the input images with geometric transformations $\mathcal{F}$ and push the VQ-VAE to reconstruct the distorted image. This encourages the VQ-VAE to prioritise and preserve the input image geometry. In practice, we randomly replace the input range image $\mathbf{x}$ and raydrop mask $\mathbf{x}_m$ with their respective transformed versions $\mathcal{F}(\mathbf{x})$, $\mathcal{F}(\mathbf{x}_m)$ during the calculation of the training losses (Equation 2 and 3). Our choice of geometric transformations includes affine transformation as well as horizontal and vertical flips.

3.3. Auto-regressive Transformer

With trained VQ-VAE, we can encode the range images into token indices $\mathbf{s} \in \{0, 1, 2, ..., |Z|-1\}^{\text{h}\times\text{w}}$ and use AR transformers to model interactions between tokens. We train the transformer by enforcing auto-regressive modelling. We estimate the likelihood of each token index $\mathbf{s}_i$ based on the previous indices $\mathbf{s}_{<i}$ denoted as $p(\mathbf{s}_i|\mathbf{s}_{<i})$. The likelihood of the entire sequence can be calculated using chain rule as $p(\mathbf{s}) = \prod_i p(\mathbf{s}_i|\mathbf{s}_{<i})$. So training objective for the AR transformer can be achieved by negative log-likelihood of $p(\mathbf{s})$:

$$\mathcal{L}_{\text{T}} = \mathbb{E}_{\mathbf{x}}\left[-\log p(\mathbf{s})\right]. \qquad (5)$$

We visualise the training process of the AR transformer in Figure 2b.

4. Evaluations

We detail our experimental settings in Section 4.1, conduct a comparison to SOTA models in Section 4.2 and carry out an ablation study in Section 4.3.

4.1. Experimental Settings

Datasets We evaluate our LidarGRIT generations on two datasets: KITTI-360 [6] and KITTI odometry [3]. We follow the Zyrizanov et al [13] for partitioning the sequences into train and test splits as well as determining the range image size (64×1024). Similarly for KITTI odometry, we follow the setup explained by Nakashima *et al.* [7] for dataset splitting and subsampling the range images to the size of 64×256.

Metrics We evaluate our generated point clouds against SOTA models using three representations: range image, BEV and point cloud. For range image representation, we employ Sliced Wasserstein Distance (SWD) [5]. BEV representation is assessed using Maximum-Mean Discrepancy (MMD) and Jensen-Shannon Divergence (JSD) following Zyrizanov *et al.* [13]. Point cloud evaluation includes Frechet Point Distance (FPD) akin to R2DM [8], along with minimum-Matching Distance (MD) and JSD similar to

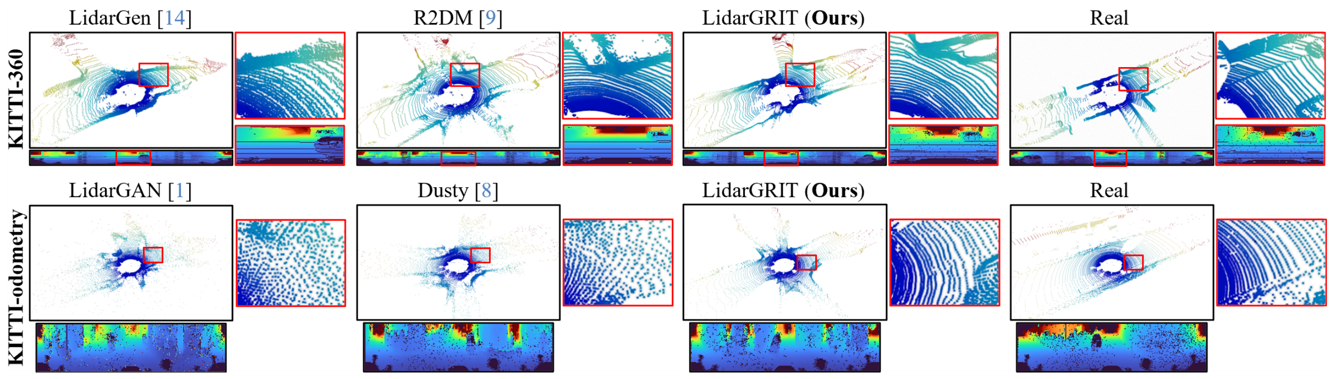


Figure 3. Qualitative comparison on KITTI-360 [6] and KITTI odometry datasets [4].

Table 1. Quantitative comparison on KITTI-360 dataset.

	Image	BEV		Point cloud
Method	SWD$\times 10^2 \downarrow$	MMD$\times 10^4 \downarrow$	JSD$\times 10^2 \downarrow$	FPD $\downarrow$
LidarGen [13]	33.93	2.19	5.70	43.27
UltraLiDAR [12]	N/A	2.23	10.52	N/A
R2DM [8]	20.82	4.00	4.55	**10.84**
LidarGRIT (**Ours**)	**10.29**	**2.16**	**3.93**	12.54

Table 2. Quantitative comparison on KITTI odometry dataset.

	Image	Point cloud		
Method	SWD$\times 10^2 \downarrow$	MD $\times 10^3 \downarrow$	JSD$\times 10^2 \downarrow$	FPD $\downarrow$
LidarGAN [1]	82.29	6.70	15.98	700
Dusty [7]	52.81	2.07	5.10	389
LidarGRIT (**Ours**)	**15.15**	**1.65**	**2.06**	**116**

Nakashima *et al.* [7]. For calculating MD and JSD metrics, we subsample 512 points from H $\times$ W points using farthest point sampling. We create sets of 5k randomly selected generated and real samples for comparison in all metrics.

4.2. Comparison to State-of-the-art

We quantitatively compare our LidarGRIT to the SOTA models on KITTI-360 and KITTI odometry generation tasks in Table 1 and 2, respectively. For KITTI-360 generation, we select the best-performing DMs, R2DM [8] and LidarGen [13], alongside the transformer-based model UltraLiDAR [12] (notably, the numbers for UltraLiDAR are derived from an unofficial implementation since the public implementation is not available). For KITTI odometry generation, we pick the top-performing GAN-based models Dusty [7] and LidarGAN [1].
As shown in the tables, LidarGRIT achieves the best performance on nearly all metrics. Specifically, LidarGRIT obtains significantly superior results on the image-based metric SWD compared with DMs. This can largely be attributed to the more realistic generation of raydrop noise. On the other hand, LidarGRIT substantially outperforms GAN-based models on point cloud representation because of more precise 3D shape modelling via the AR transformer. The results are further evident in the qualitative comparison depicted in Figure 3. Noteworthy, in the KITTI-360 generation, the adapted VQ-VAE and transformer contain 34.77M and 10.26M parameters respectively, which is is comparable to SOTA models such as UltraLiDAR [12] which contains 40.3M parameters.

Table 3. Ablation study on KITTI odometry dataset.

		Image	Point cloud		
GP	RL	SWD$\times 10^2 \downarrow$	MD $\times 10^3 \downarrow$	JSD$\times 10^2 \downarrow$	FPD $\downarrow$
-	-	61.73	1.73	2.82	212
-	✓	17.68	1.66	**1.99**	122
✓	✓	**15.15**	**1.65**	2.06	**116**

4.3. Ablation Study

We evaluate the importance of the two proposed techniques in our VQ-VAE model: raydrop loss (RL) and geometric preservation (GP) in Table 3. In the baseline scenario (first row), we exclude the raydrop logits channel such that VQ-VAE directly approximates input noisy range images; moreover, we disable the GP technique during training. We introduce the RL in the second row and then incorporate both RL and GP techniques into the baseline in the third row. As shown, the RL hugely reduces both image-based and point cloud-based metrics due to more accurate raydrop generation. Furthermore, the GP technique improves performance on most metrics up to 14% by increasing the VQ-VAE generalisability.

5. Conclusion

This paper introduced LidarGRIT, a novel Lidar point cloud generative model. The proposed LidarGRIT outperforms SOTA models on KITTI-360 and KITTI odometry datasets. This work also highlighted the importance of the proposed raydrop loss and geometric perseverance in the adapted VQ-VAE model, leading to higher-quality generated samples.

References

[1] Lucas Caccia, Herke van Hoof, Aaron C. Courville, and Joelle Pineau. Deep generative modeling of lidar data. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5034–5040, 2018. 1, 2, 4

[2] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 12873–12883. Computer Vision Foundation / IEEE, 2021. 1, 2

[3] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*, 32(11):1231–1237, 2013. 3

[4] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 4

[5] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ArXiv*, abs/1710.10196, 2017. 2, 3

[6] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *Pattern Analysis and Machine Intelligence (PAMI)*, 2022. 3, 4

[7] Kazuto Nakashima and Ryo Kurazume. Learning to drop points for lidar scan synthesis. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 222–229, 2021. 2, 3, 4

[8] Kazuto Nakashima and Ryo Kurazume. Lidar data synthesis with denoising diffusion probabilistic models. *Arxive*, abs/2309.09256, 2024. 1, 2, 3, 4

[9] Kazuto Nakashima, Yumi Iwashita, and Ryo Kurazume. Generative range imaging for learning scene priors of 3d lidar data. *ArXiv*, abs/2210.11750, 2022. 2

[10] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015. 2

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. 1

[12] Yuwen Xiong, Wei-Chiu Ma, Jingkang Wang, and Raquel Urtasun. Learning compact representations for lidar completion and generation. In *CVPR*, 2023. 2, 4

[13] Vlas Zyrianov, Xiyue Zhu, and Shenlong Wang. Generate realistic lidar point clouds. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*, page 17–35, Berlin, Heidelberg, 2022. Springer-Verlag. 1, 2, 3, 4