L3DR: 3D-aware LiDAR Diffusion and Rectification

Quan Liu[1] Xiaoqin Zhang[2] Ling Shao[3] Shijian Lu[1,†]
[1]Nanyang Technological University [2]Zhejiang University of Technology
[3]UCAS-Terminus AI Lab, University of Chinese Academy of Sciences
https://github.com/liuQuan98/L3DR

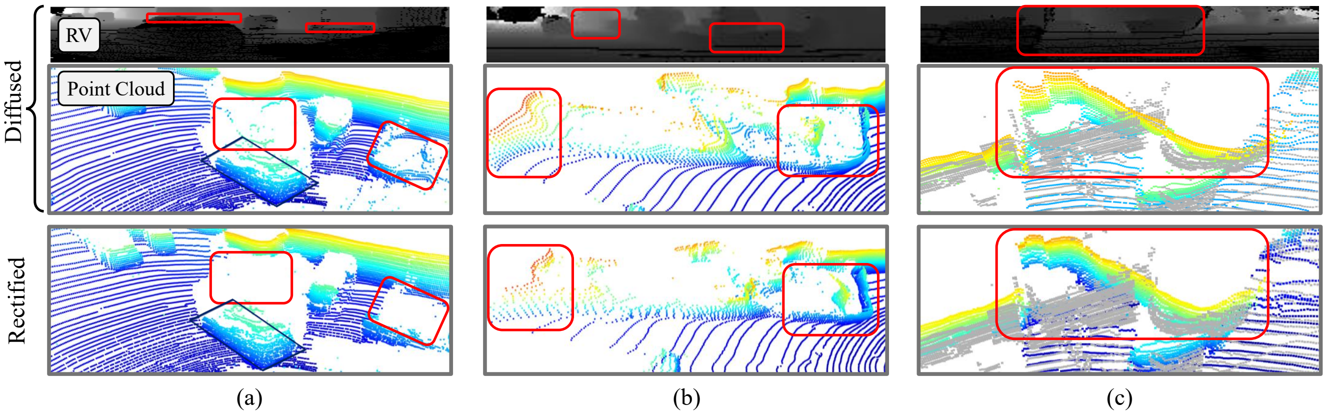

Figure 1. **L3DR effectively rectifies LiDAR range-view (RV) diffusion artifacts by selectively ignoring anomalous training regions.** (a) *Depth bleeding* creates fake points between the foreground vehicle and the background. (b) *Wavy surfaces and rounded edges* synthesized by RV diffusion are straightened and sharpened after rectification. (c) Anomalous regions in training data pairs, *e.g.,* a diffusion-generated wall perpendicular to ground truth (GT), can overshadow RV artifacts and hijack artifact removal task; these are suppressed with the Welsch Loss (see section 4.3). Generated and rectified point clouds are colored while GT point clouds are in gray.

Abstract

Range-view (RV) based LiDAR diffusion has recently made huge strides towards 2D photo-realism. However, it neglects 3D geometry realism and often generates various RV artifacts such as depth bleeding and wavy surfaces. We design L3DR, a 3D-aware LiDAR Diffusion and Rectification framework that can regress and cancel RV artifacts in 3D space and restore local geometry accurately. Our theoretical and empirical analysis reveals that 3D models are inherently superior to 2D models in generating sharp and authentic boundaries. Leveraging such analysis, we design a 3D residual regression network that rectifies RV artifacts and achieves superb geometry realism by predicting point-level offsets in 3D space. On top of that, we design a Welsch Loss that helps focus on local geometry and ignore anomalous regions effectively. Extensive experiments over multiple benchmarks including KITTI, KITTI360, nuScenes and Waymo show that the proposed L3DR achieves state-of-the-art generation and superior geometry-realism consistently. In addition, L3DR is generally applicable to different LiDAR diffusion models with little computational overhead.

1. Introduction

Point clouds captured by LiDARs (Light Detection and Ranging) are the cornerstone of outdoor 3D computer vision, facilitating various perception tasks such as detection [18, 54], segmentation [48], tracking [37, 45], and SLAM [25, 26]. Though they offer superior perception capabilities and safety guarantees for self-driving vehicles, collecting large-scale LiDAR point clouds is prohibitive in terms of sensor procurement, intensive human labour involved [4, 40], etc. Automated generation of high-quality point clouds has thus become a critical factor for extensive adoption of LiDAR data in various 3D perception tasks.

To this end, a practical LiDAR point cloud generation method must achieve 1) realism in global layout that accurately recreates the coarse-grained global depth distribution on the range view (RV) representation, 2) realism in local geometry that perfectly replicates fine-grained local geometries such as self-occlusion, and 3) light computational overhead which empowers cost-efficient data generation.

Inspired by the massive success of Diffusion Models

† Corresponding author: Shijian.Lu@ntu.edu.sg

(DMs) [38] featured by Stable-diffusion [35] and ControlNet [51] in image generation, a line of research pioneered by LiDARGen [56] has extended image-based DMs for generating layout-realistic point clouds in the RV representation. However, although RV enables DM-based point cloud generation by projecting 3D point clouds to 2D images, it hinders accurate discernment of sparsity and self-occlusion in the 3D space which directly leads to various range-view artifacts such as *depth bleeding* and *wavy surfaces*. As depicted in Figure 1(a), even realistic RV images can have serious depth bleeding artifacts, a phenomenon of errorneous depth continuity near edges. Additionally, since 3D plain surfaces are projected onto sinusoidal surfaces in images, LiDAR DMs tend to produce suboptimal 3D geometry-realism with wavy surfaces and round corners as illustrated in Figure 1 (b). These artifacts greatly undermine the realism of the generated 3D point clouds.

This paper presents L3DR, a 3D-aware LiDAR Diffusion and Rectification framework that generates geometrically realistic point clouds by rectifying the back-projected 3D coordinates of diffusion-generated RV images. L3DR works by tackling two challenges. The first is on training data for training a residual regression network (RRN) to rectify 3D coordinates. Unlike normal diffusion training data that can be obtained by adding noises to the input data, RV artifacts exhibit irregular occurrence with little explicit distributions. We therefore introduce semantic-conditioned LiDAR diffusion to generate point clouds and their ground-truth (GT). Thanks to the powerful conditional diffusion models, the obtained data are structurally similar yet imbued with RV artifacts, making them perfect for training the regression network. The second is on training loss. Due to anomalous regions scattered across the training data as illustrated in Figure 1 (c), minimizing residual regression errors with L1/L2 loss may over-attend to the anomalous regions and neglect local geometries. We thus introduce Welsch Loss to guide the network to ignore high-bias regions and focus on local geometry artifacts, leading to superior geometry realism in the generated point clouds.

The contributions of this work can be summarized in three major aspects:

- We propose a 3D-aware LiDAR Diffusion and Rectification framework that rectifies RV geometry artifacts with a 3D residual regression network and achieves superb realism in both global layout and local geometry.
- We introduce the Welsch Loss that helps pass over high-bias regions in the training data and achieves high-quality residual regression over local geometry artifacts, leading to superior geometry realism in the generated point clouds.
- Extensive experiments on SemanticKITTI [1, 10], KITTI360 [20], nuScenes [4], and Waymo Open Dataset [40] show that the proposed L3DR framework outperforms the state-of-the-art consistently with negligible additional computation cost.

2. Related Work

2.1. Diffusion Models

Diffusion models (DMs) [8, 14, 15, 38, 39] comprise a prominent line of contemporary image synthesis methods. Following the early success of Jascha *et al.* [38], DMs have demonstrated superior scalability [32, 33, 35], along with various ways to control the image content [51, 52]. DMs have garnered increasing attention due to several advantages over other image synthesis methods like Generative Adversarial Network (GAN) [11] and Variational Autoencoder (VAE) [17]. Firstly, DMs are far more capable than VAEs due to their multi-step denoising process being able to provide larger effective model capacity with limited parameters. Secondly, thanks to the simple but robust loss function, DMs are not prone to mode collapse which haunts GANs. The great success of image diffusion has recently inspired several LiDAR diffusion models [3, 56]. Therefore, we choose to continue on the line of diffusion models for LiDAR generation task.

2.2. LiDAR Generation

Range view generation. RV-based generation [3, 13, 16, 23, 27–30, 34, 56] converts point clouds into depth images where the height represents elevation and width represents azimuth. Caccia *et al.* [3] pioneer this line of research to generate RV images with GAN and VAE. DUSty [27] improves GAN-based generation by estimating disentangled depth and ray-drop probabilities, and the later DUStyV2 [29] further decomposes spatial resolution with latent codes to enable LiDAR parameter manipulation. LiDARGen [56] first exploits diffusion models on RV-based point cloud generation. R2DM [28] applies explicit positional encoding on range-reflectance images to improve generation quality. LidarGRIT [13] improves latent denoising with an image transformer and raydrop estimation. RangeLDM [16] introduces height offsets of lasers to correct traditional spherical RV projection. LiDM [34] enables image, text, and semantic conditioned point cloud generation with curve-wise compression. OLiDM [49] improves object generation with a two-stage object-scene diffusion cascade. However, these RV methods all rely on depth images instead of actual 3D geometry, leading to inaccurate local geometries in the generated point clouds.

Re-sampled LiDAR from 3D representations. This line of LiDAR data generation methods first generates a scene representation, often from multi-modal data with map conditions, and then samples LiDAR point clouds by simulating the laser reflection process. For example, LidarDM [57] leverages explicit scene-level and object-level mesh diffusion, then re-sample novel LiDAR point clouds with ray casting. DynamicCity [2] leverages a novel statio-temporal 4D representation called HexPlane, from which they diffuse novel occupancy map videos. UniScene [19] further recasts point clouds from occupancy videos with prior guided sampling. While achieving incredible visual authenticity and temporal consistency, these methods are notoriously resource-hungry, rendering them less economical for LiDAR data generation.

3. Pilot Study

This section presents whether 2D diffusion models are inherently worse at generating smooth object borders than 3D models. We first introduce the preliminaries on diffusion and deduce a theoretical upper bound for RV gradients as generated by 2D models and 3D models, respectively. We validate our claim with actual gradient distributions of 2D- and 3D-rectified RV images.

3.1. Preliminaries

Diffusion models. Diffusion models generate images by reversing a gradual noise-adding process [38]. The core idea is to corrupt a complex image distribution into a Gaussian noise distribution in the forward process, and learn the reverse process to generate new image samples.

Forward process. Given a data sample $x_0 \in \mathbb{R}^{u\times v}, x_0 \sim q(x_0)$ which represents the target distribution, the forward process gradually adds Gaussian noise over T steps according to a fixed schedule:

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}\, x_0, (1-\bar{\alpha}_t)\,\mathbf{I}), \tag{1}$$

where $\alpha_t \in (0,1)$ are predefined noise scales and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. This process defines a sequence $x_1, x_2, \ldots, x_T$ where x_T is approximately standard Gaussian.

Reverse sampling process. The goal is to learn a reverse process $p_\theta(x_{t-1} \mid x_t)$ that reconstructs clean samples starting from Gaussian noise:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{2}$$

Typically, a model is trained to predict the added noise ϵ, and then restore the image using Equation 2.

3.2. Theoretical Analysis

We provide Theorem 1 where a constant bound can be derived for the image gradient in DDIM-sampled images. We then provide a proof sketch, while the full proof is placed in Appendix Sec. A.2.

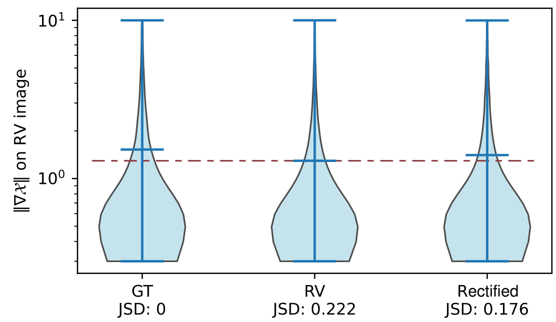

Figure 2. **Empirical validation of Theorem 1.** The graph shows the distribution of $\|\nabla x\|$ for GT, vanilla RV diffusion, and our rectified RV, including the corresponding Jensen-Shannon Divergence (JSD) *w.r.t.* the GT.

Theorem 1. *Given the assumption that diffusion UNets are Lipschitz continuous, the output image x_0 generated by DDIM is locally Lipschitz continuous with respect to the input noise x_T. Moreover, the spatial gradient of x_0 is bounded:*

$$\|\nabla x_0\| \leq L \quad \textit{for some constant } L \in \mathbb{R}$$

Proof sketch. *Given the assumption that diffusion models are Lipschitz, and the scalar operations in DDIM are also Lipschitz with finite input parameters and scalar operations, their composition as the diffusion process is also Lipschitz with some constant L.*

Corollary 1. *As a result of Lipschitz continuity throughout the DDIM sampling process, the generated image in theory cannot exhibit arbitrarily sharp spatial transitions. Therefore, DDIM outputs are smooth, with softly transitioned object boundaries.*

Corollary 2. *A vital prerequisite for Theorem 1, spatial Lipschitz continuity measured in RV image, does not hold when the model itself is a 3D model which accepts back-projected point cloud from an RV image. While 3D models are still generally Lipschitz, the spatial proximity of a point is defined in 3D rather than 2D, adding an additional dimension of limitation. WLOG, we denote the Lipschitz constant for the 3D RRN model as L_{3D} and an RV image rectified with a 3D network as x_{3d}. Suppose two adjacent pixels before 3D rectification have a depth difference of Δd, then we have $\|\nabla x_{3d}\| \leq L_{3D} \times \Delta d$ which is unbounded. We interpret that if Δd is large enough, then these two adjacent pixels are out of the effective receptive field (ERF) in 3D operations such as Sparse Convolution or Local Attention, and therefore can generate arbitrarily sharp image borders.*

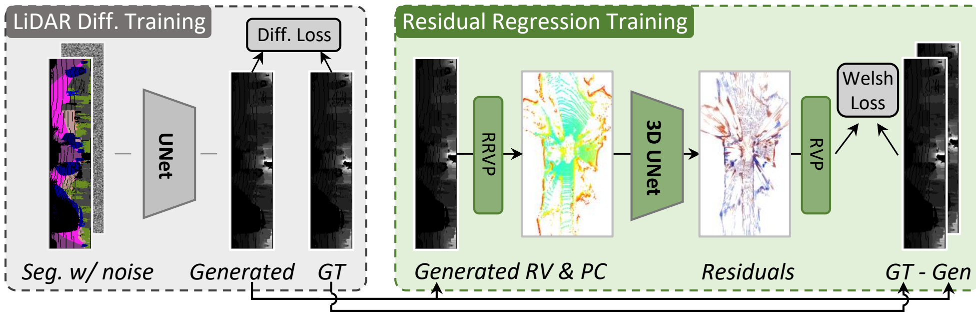

Figure 3. **The training pipeline** of the proposed L3DR framework. In the LiDAR diffusion training stage, generated and ground-truth point cloud pairs are collected using semantic-conditioned LiDAR diffusion. In the residual regression training stage, such data pairs are employed to train a 3D network to remove RV artifacts present in the residuals to improve generation quality.

3.3. Empirical Analysis

To validate that 3D models generate sharper borders than image diffusion models, we plot the empirical distribution of image gradient length $\|\nabla x\|$ for the GT point clouds, vanilla LiDM RV outputs, and our rectified LiDM outputs in Figure 2. We remove the dominant $\|\nabla x\| \leq 0.3m$ on planar regions and rare $\|\nabla x\| \geq 10m$ which exceed network ERFs, so that we can compare the remaining geometry-related gradients. According to the figure, range view generated images have lower $\|\nabla x\|$ compared to the ground truth due to the inherent smoothness of diffusion-generated images. However, we are able to restore such sharp borders with the rectification network, which increases overall average image sharpness and reduces JSD from 0.222 to 0.176. We conclude that 3D models generate sharper object borders than 2D models, thus being preferable for rectification of 2D image diffusion results.

4. Method

Figure 3 shows the L3DR framework which involves two training stages. In the LiDAR diffusion training, we leverage the RV representation to train a LiDAR diffusion model with conditional semantic input. The diffusion model generates RV point clouds and the corresponding ground-truth. In the residual regression training, we apply such data pairs to train a 3D residual regression network (RRN) that rectifies RV artifacts under the supervision of Welsch Loss.

4.1. Range View Projection

Range view projection (RVP) and reverse range view projection (RRVP) are the key components that connects RV images with point clouds. Given a point cloud $P \in \mathbb{R}^{N\times 3}$ consisting of N points in Euclidean coordinates and a range-view depth image $x \in \mathbb{R}^{u\times v}$, the range view projection converts a point $(p_x, p_y, p_z)^\intercal \in P$ into a pixel $(u_i, v_i)^\intercal$ with depth value d_i:

$$\begin{aligned} \begin{pmatrix} u_i \\ v_i \end{pmatrix} &= \begin{pmatrix} \sigma_u \arctan\left(p_z \Big/ \sqrt{(p_x^2+p_y^2)}\right), \\ -\sigma_v \arctan(p_x/p_y), \end{pmatrix} \\ d_i &= \sqrt{(p_x^2+p_y^2+p_z^2)} \end{aligned} \tag{3}$$

Where σ_u, σ_v are pixels per radian in elevation and azimuth directions, respectively. They are usually anisotropic, *i.e.*, the vertical resolution is sparse while horizontal resolution is dense. For LiDARs with non-equally angled lasers, σ_u can be a function of elevation. Similarly, the reverse range view projection converts a pixel (u_i, v_i) with depth d_i into a point $(p_x, p_y, p_z)^\intercal$:

$$\begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} = \begin{pmatrix} d_i \cos(u_i/\sigma_u)\cos(v_i/\sigma_v), \\ -d_i \cos(u_i/\sigma_u)\sin(v_i/\sigma_v), \\ d_i \sin(u_i/\sigma_u). \end{pmatrix}, \tag{4}$$

Note that RRVP is lossless but RVP is lossful, because multiple points may be projected to the same pixel. Empirically, RVP is lossless when the point cloud projection structure is preserved and RV image size is equal to or larger than the laser angular resolution.

4.2. LiDAR Diffusion Training

In order to generate ground-truth and diffusion-generated point cloud pairs for the following training stage, we retrain a state-of-the-art conditional LiDAR diffusion model, LiDM [34], on KITTI, nuScenes, and WOD to generate LiDAR point clouds based on segmentation map as a condition. Specifically, the RV image is first compressed with a VQ-VAE [44], then a classical diffusion UNet is leveraged to predict Gaussian noise added in the compressed latent space. To enable conditional generation, a down-sampled segmentation color map is appended to latent space as control input. We collect the ground-truth RV x_{gt} and corresponding generated x_{gen} after convergence to provide training data for the next stage.

We highlight the importance of LiDM due to its capability to generate point clouds that match closely with ground

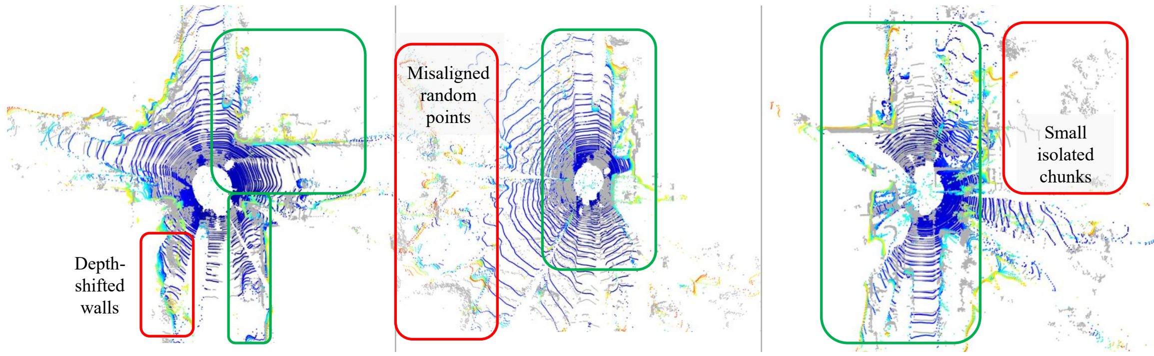

Figure 4. **Visualization of two types of errors in RRN training data.** While the generated point clouds (colored) approximate the GT (gray) in most of the regions with **high-variance errors**, i.e., RV artifacts as highlighted with green dotted lines, there are also regions with **high-bias errors** which impede training, including (1) shifted walls, (2) random points on the leaves where laser hits are hard to predict, and (3) isolated chunks with consistent depth error. These bias-dominated regions are harmful for RRN training.

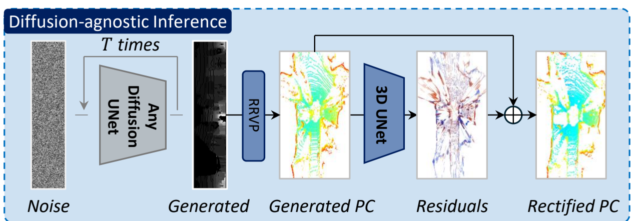

Figure 5. **The inference pipeline** of the proposed L3DR.

truths, and yet remain imbued with RV artifacts on a limited scale, as depicted in Figure 4. However, we also highlight that our framework is general and not restricted to LiDM, given that an alternative LiDAR diffusion method can generate such closely approximated point cloud pairs.

4.3. Residual Regression Training

Pipeline. After obtaining a generated point cloud, we feed it into a 3D backbone named Residual Regression Network (RRN) to accurately regress and cancel out the hidden RV artifacts to achieve better geometry-realism. Specifically, we first obtain the generated point cloud $P_{gen} = \mathrm{RRVP}(x_{gen})$ through reverse range view projection. Then, P_{gen} is fed into a 3D backbone $\mathbf{F} : \mathbb{R}^{N\times k} \rightarrow \mathbb{R}^{N\times 3}$ to obtain 3D offsets $O = \mathbf{F}(P_{gen})$. For a normal unconditional 3D RRN, $k = 3$ which represents the input point coordinates; otherwise, $k = 6$ which accepts both the coordinates and a semantic color map. Finally, we project the output offsets onto the radial directions of the original point cloud to obtain the final residuals $\hat{O} = P_{gen}\mathrm{diag}(P_{gen}O^{\intercal}) / \sqrt{\mathrm{diag}(P_{gen}P_{gen}^{\intercal})}$.

Decomposition of the learning objective. From a bias–variance decomposition viewpoint, bias measures the consistent deviation between the diffusion-generated depth and the ground truth, while variance captures variable RV artifacts, as depicted in Figure 4. High bias errors naturally emerge in large chunks sharing the same segmentation class, *i.e.*, insufficient semantic constraints introduce unrealistic yet coherent hallucinations such as a flat wall being interpreted as slanted or a tree as distant rather than nearby. These errors are particularly harmful to geometry correction since the residuals stem from biased interpretation of the scene rather than stochastic RV artifact error. More discussion is given in section A.7.

Although high-bias and high-variance errors often coexist, RV artifacts possess much smaller error magnitude compared to high-bias errors. This distinction enables a soft separation of bias- and variance-dominated regions based on residual magnitude, which inspires the introduction of the Welsch loss to suppress large, bias-driven deviations.

Loss function. After obtaining the model output and GT, we propose Welsch Loss to remove the effect of erratic high-bias areas in training data to focus on rectifying high-variance RV artifacts. Specifically, Welsch's Function is defined as:

$$\psi_\nu(x) = 1 - \exp\left(-\frac{x^2}{(2\nu^2)}\right) \tag{5}$$

Which is an upside-down bell curve and $\nu > 0$ is a width parameter. We then define the RRN loss function as:

$$L_{RRN} = \mathrm{mean}\Big(\psi_\nu\big(\mathrm{RVP}(P_{gen} + \hat{O}) - x_{gt}\big)\Big) \tag{6}$$

Where $\mathrm{RVP}(\cdot)$ denotes range view projection, and $\mathrm{mean}(\cdot)$ is the average operation. We choose the suitable ν for all datasets and fix them for other experiments.

4.4. Diffusion-agnostic Inference

During inference, because the RRN has been well trained to remove RV artifacts, the diffusion model can be replaced with arbitrary LiDAR diffusion model, as depicted in Figure 5. Empirically, the RRN can generalize to arbitrary unseen unconditional diffusion net-

Task	Method	Perceptual		Distributional	
		FSVD↓	FPVD↓	JSD↓	MMD×$10^{(-4)}$ ↓
KITTI360 Unconditional	LiDARGAN [3]	183.4	168.1	0.272	4.74
	LiDARVAE [3]	129.9	105.8	0.237	7.07
	ProjectedGAN [36]	44.7	33.4	0.188	**2.88**
	UltraLiDAR [47]	72.1	66.6	0.747	17.12
	LiDARGen(1160s) [56]	39.2	33.4	0.188	**2.88**
	LiDARGen(50s) [56]	480.6	400.7	0.506	9.91
	LDM(50s) [35]	70.7	61.9	0.236	5.06
	R2DM(50s) [28]	36.8	30.9	<u>0.168</u>	<u>2.92</u>
	LiDM(50s) [34]	38.8	29.0	0.211	3.84
	Ours-R2DM	<u>35.9</u>	<u>28.2</u>	**0.165**	2.90
	Δ Increment w.r.t. R2DM	+2.4%	+8.7%	+1.8%	+0.7%
	Ours-LiDM	**35.8**	**26.1**	0.182	3.27
	Δ Increment w.r.t. LiDM	+7.7%	+10.0%	+13.7%	+14.8%
nuScenes Conditional	LiDM [34]	86.6	74.8	0.145	2.81
	Ours-LiDM-Sem	**81.3**	**67.0**	**0.133**	**2.72**
	Δ Increment w.r.t. LiDM	+6.12%	+10.43%	+8.28%	+3.20%
Waymo Conditional	LiDM [34]	21.4	21.9	0.104	1.30
	Ours-LiDM-Sem	**18.3**	**20.3**	**0.086**	**1.25**
	Δ Increment w.r.t. LiDM	+14.49%	+7.31%	+17.31%	+3.85%

Table 1. Benchmarking of unconditional generation on KITTI360 and semantic-conditioned generation on nuScenes and Waymo. For the semantic-conditioned experiments, RRN takes segmentation map as additional input for optimal performance. Gray areas highlight direct comparisons with the baselines.

works even if the RRN is only trained on segmentation-conditioned data generated by LiDM. Specifically, during inference, we generate novel x'_{gen} with arbitrary LiDAR diffusion model, project RV into a point cloud $P'_{gen} = \mathrm{RRVP}(x'_{gen})$, calculate the point wise residual $\hat{O}' = P'_{gen} diag(P'_{gen}\mathbf{F}(P'_{gen})^{\intercal}) / \sqrt{\mathrm{diag}(P'_{gen}P'^{\intercal}_{gen})}$, and obtain the final rectified point cloud $P'_{ref} = P'_{gen} + \hat{O}'$.

5. Experiments

5.1. Main Results

In this section, we first introduce our experiment setup, then report the generation metrics in Table 1-2 and provide visualization in Figure 6-7. Module ablation and time analysis are listed in Table 3-4. Ablation on other datasets and a parameter tuning experiment can be found in Appendix Sec. A.5.

5.2. Experiment Setup

Datasets. We train and evaluate our method on SemanticKITTI [1, 10], KITTI360 [20], nuScenes [4], and Waymo Open Dataset [40]. All datasets are split into train-validation-test according to official recommendations. We generate RRN training data by conditional LiDM inference on training set conditions, *i.e.*, segmentation maps, and report the metrics of the last-epoch model on validation set.

Metrics. We report both perceptual metrics including the Fréchet Sparse Volume Distance (FSVD) and Frećhet Point Voxel Distance (FPVD) proposed by [34], and conventional distributional metrics, Jensen-Shannon Divergence (JSD) and Minimum Matching Distance (MMD).

Training. Our diffusion model processes depth values of size without logarithmic scaling. For 64-laser KITTI and Waymo, we naturally designate the RV image size as $(64, 1024)$. However, we discover that adopting a $(32, 1024)$ image size on the 32-laser nuScenes leads to network divergence, and that LiDM [34] did not open-source its nuScenes config. We solve this problem by over-provisioning the RV image size to $(64, 1024)$ for nuScenes, which stabilizes training without affecting generation quality as discussed in Section 4.1. We hypothesize that a large-enough image size is vital to training dynamics. All networks are trained with $4\times$ RTX 4090 24G up to 150 epochs.

KITTI unconditional generation. We compare our L3DR method with existing LIDAR generation methods on KITTI unconditional generation task in upper half of Table 1. Unconditional R2DM and LiDM are adopted as the first-stage diffusion model for L3DR. With the powerful capability of removing RV artifacts, L3DR sets a new state-of-the-art, achieving $35.8(\Delta 7.7\%)$ FSVD, $26.1(\Delta 10.0\%)$ FPVD,

Method	Semantic-Map-Conditioned FSVD↓	FPVD↓	JSD↓	MMD×$10^{(-4)}$ ↓	Front-Image-Conditioned FSVD↓	FPVD↓	JSD↓	MMD×$10^{(-4)}$ ↓
LiDARGen	31.7	30.1	0.130	5.18	-	-	-	-
LDM	21.3	20.3	0.088	3.73	35.9	26.5	0.26	3.80
LiDM	20.2	17.7	0.072	3.16	**32.5**	25.8	0.21	3.69
Ours	20.4	15.0	**0.070**	1.69	33.6	**24.9**	**0.18**	**3.30**
Δ *Inc.*	-1.0%	+15.3%	+2.8%	+46.5%	-3.4%	+3.5%	+12.2%	+10.6%
Ours-Sem	**15.8**	**12.9**	**0.070**	**1.50**	-	-	-	-
Δ *Inc.*	+21.8%	+27.1%	+2.8%	+52.5%	-	-	-	-

Table 2. **Comparison of conditional LiDAR point cloud generation** on SemanticKITTI and KITTI360. Gray areas highlight direct comparisons with the baseline, LiDM. 'Ours-Sem' denotes our method with segmentation input to the RRN.

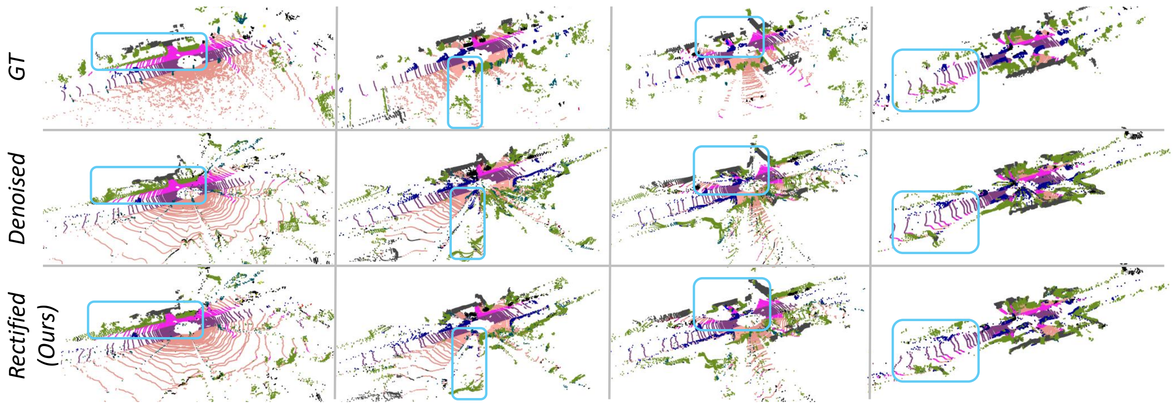

Figure 6. **Visualization of conditional generation on SemanticKITTI.** Cyan regions highlight the improved RV artifacts from the diffusion-generated (i.e., denoised) data to our rectified data, such as depth bleeding and wavy surfaces.

and $0.182(\Delta 13.7\%)$ JSD, greatly surpassing the baseline method LiDM. Similar results are also observed when a non-latent diffusion method R2DM is applied, which signals the general applicability of our L3DR framework, achieving the best JSD of $0.165(\delta 1.8\%)$. While L3DR does not top the MMD metric, our method still provides a average 7.3% improvement, and is comparable to the best-performing ProjectedGAN which scores 2.88. We conclude that L3DR achieves state-of-the-art global photo- and geometry-realism on unconditional LiDAR generation task.

KITTI conditional generation. We compare L3DR with conditional LIDAR generation methods on both SemanticKITTI and KITTI360, respectively in table 2. L3DR again achieved better performance compared to the baselines, gaining $15.0(\Delta 15.3\%)$ FPVD, $0.07(\Delta 2.8\%)$ JSD, and an astonishing $1.69 \times 10^{-4}(\Delta 46.5\%)$ MMD on SemanticKITTI, and $24.9(\Delta 3.5\%)$ FPVD, $0.18(\Delta 12.2\%)$ JSD, and $3.3 \times 10^{-4}(\Delta 10.6\%)$ MMD on KITTI360. Meanwhile, L3DR with semantic-map input improves generation quality consistently, providing further 10.2% average performance boost on all metrics upon raw L3DR. We conclude that L3DR significantly improves conditional generation capability compared to the baselines.

Performance on other datasets. We examine the semantic-conditioned generation metrics of our L3DR method on nuScenes and Waymo, in the lower half of Table 1. L3DR exhibits consistent improvements on all datasets, improving all metrics by an average of 11.6% and 7.0% on nuScenes and Waymo conditional generation, respectively. We conclude that L3DR is widely applicable and sets new state-of-the-arts on the nuScenes and Waymo semantic-conditioned generation tasks.

5.3. Other Results

Ablation. We ablate various components of the L3DR framework, including the backbone architecture between SPUNET [7] and PTV3 [46], loss choice, the usage of semantic-map input in RRN, and a 2D RRN instead of 3D RRN. We calculated metrics with voxelized GT instead of original GT for fast evaluation, which results in slightly different figures. As listed in Table 3, applying MSE Loss generally deteriorates performance compared to baseline, with FSVD and FPVD almost doubling from $18.3, 15.3$ to $26.3, 25.1$ for SPUNET and $42.4, 42.6$ for PTV3. The best overall performance attributes to SPUNET with Welsch loss and semantic-map input. Additionally, substituting 3D UNet with a 2D Image Unet brings considerable degradation on all metrics. We default to SPUNET and Welsch Loss for all other experiments.

Configurations			Perceptual		Statistical	
Backbone	Loss	RRN Sem. Input	FSVD↓	FPVD↓	JSD$\times 10^{(-2)}$ ↓	MMD$\times 10^{(-5)}$ ↓
/ (baseline)	/	-	18.3	15.3	7.1	16.2
SPUNet	Welsch	-	<u>16.4</u>	12.1	**6.7**	16.7
SPUNet	MSE	-	26.3	25.1	7.0	**12.6**
SPUNet	Welsch	✓	**12.5**	**10.7**	**6.7**	<u>15.0</u>
PTV3	Welsch	-	17.7	13.9	<u>6.8</u>	17.2
PTV3	MSE	-	42.4	42.6	7.4	18.8
PTV3	Welsch	✓	**12.5**	<u>10.8</u>	**6.7**	15.8
2D UNet	Welsch	-	19.2	16.4	7.1	16.3

Table 3. **Ablation experiment** on SemanticKITTI, including RRN backbone structure, loss function, semantic-map input to RRN, and a fair baseline using a 2D image Unet instead of a 3D UNet.

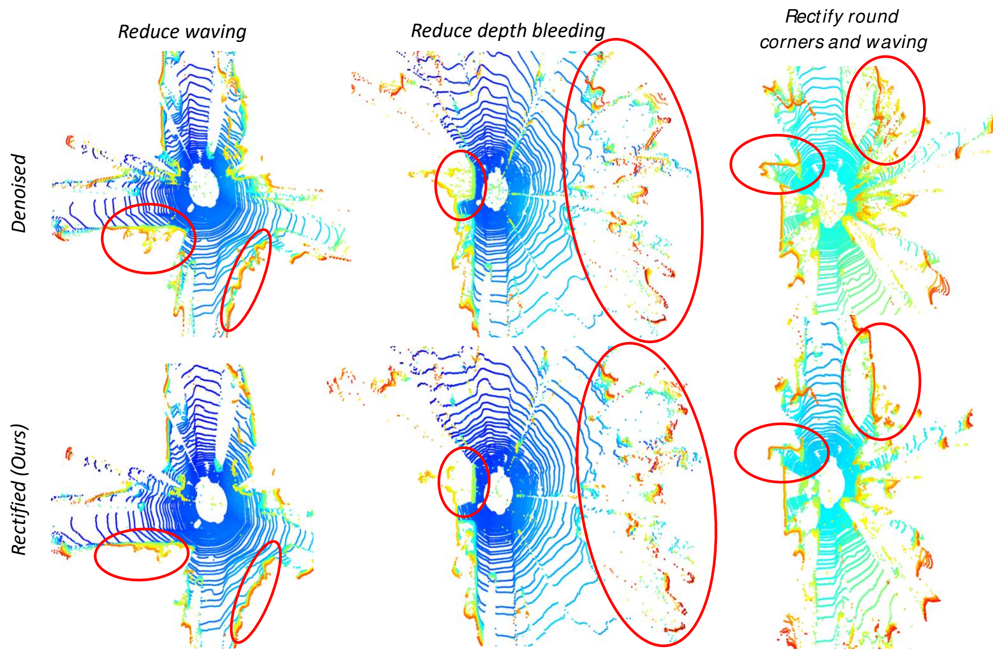

Figure 7. **Visualization of unconditional generation on KITTI360.** Red regions highlight the the improvements from the diffusion-generated (i.e., denoised) data to our rectified data.

Method	Network	# Params (M)	Steps	Time (ms)
R2DM	Eff-UNet	31.10	50	579.83
LiDM	VQ-VAE, UNet	257.77	50	557.36
+Ours	SPUNET	+37.90	+1	+19.65

Table 4. **Computational overhead** on KITTI360. our method introduce very slight computational overhead over the baselines.

Time analysis. We provide a simple time comparison to demonstrate the efficiency of L3DR framework. On an RTX 4090 GPU 24G, the additional time for RRN rectification is 19.65 ms which is negligible compared to LiDM and R2DM sampling processes which both require more than 550 ms. The 37.90 M additional parameters from RRN are also lightweight compared to LiDM. We conclude that the L3DR framework can greatly improve generation quality with negligible additional parameter and inference cost.

6. Conclusion

We have proposed L3DR, a 3D-aware LIDAR Diffusion and Rectification framework which combines 2D RV LiDAR diffusion with a 3D residual regression network to achieve both layout-realism and geometry-realism on conditional and unconditional LiDAR diffusion task. By combining our Welsch Loss with the training data generated from semantic-conditioned diffusion, we are able to train a powerful Residual Regression Network to remove RV artifacts such as depth bleeding and wavy surfaces. Our method exhibits state-of-the-art performance and broad applicability on different LiDAR diffusion methods, while requiring only negligible additional computational cost, making it a perfect choice for cost-efficient generation of high-fidelity novel LiDAR point clouds.

Acknowledgments

This study is funded by the Ministry of Education Singapore, under the Tier-2 project scheme with project number MOET2EP20123-0003.

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019. 2, 6

[2] Hengwei Bian, Lingdong Kong, Haozhe Xie, Liang Pan, Yu Qiao, and Ziwei Liu. DynamicCity: Large-scale LiDAR generation from dynamic scenes. *arXiv preprint arXiv:2410.18084*, 2024. 3

[3] Lucas Caccia, Herke Van Hoof, Aaron Courville, and Joelle Pineau. Deep generative modeling of LiDAR data. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5034–5040. IEEE, 2019. 2, 6

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 6

[5] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. SDFusion: Multimodal 3D shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4456–4465, 2023. 1

[6] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 2

[7] SpConv Contributors. SpConv: Spatially sparse convolution library. `https://github.com/traveller59/spconv`, 2022. 7

[8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 2

[9] Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. HyperDiffusion: Generating implicit neural fields with weight-space diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14300–14310, 2023. 1

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2012. 2, 6

[11] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014. 2

[12] Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oğuz. 3DGen: Triplane latent diffusion for textured mesh generation. *arXiv preprint arXiv:2303.05371*, 2023. 1

[13] Hamed Haghighi, Amir Samadi, Mehrdad Dianati, Valentina Donzella, and Kurt Debattista. Taming transformers for realistic LiDAR point cloud generation. *arXiv preprint arXiv:2404.05505*, 2024. 2

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2

[16] Qianjiang Hu, Zhimin Zhang, and Wei Hu. RangeLDM: Fast realistic LiDAR point cloud generation. In *Proceedings of the European Conference on Computer Vision*, pages 115–135. Springer, 2024. 2

[17] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2022. 2

[18] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3D proposal generation and object detection from view aggregation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–8. IEEE, 2018. 1

[19] Bohan Li, Jiazhe Guo, Hongsi Liu, Yingshuang Zou, Yikang Ding, Xiwu Chen, Hu Zhu, Feiyang Tan, Chi Zhang, Tiancai Wang, et al. UniScene: Unified occupancy-centric driving scene generation. *arXiv preprint arXiv:2412.05435*, 2024. 3

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

[21] Zhen Liu, Yao Feng, Michael J Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. MeshDiffusion: Score-based generative 3D mesh modeling. *arXiv preprint arXiv:2303.08133*, 2023. 1

[22] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3D point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021. 1

[23] Andrea Matteazzi, Pascal Colling, Michael Arnold, and Dietmar Tutsch. Adverse weather conditions augmentation of LiDAR scenes with latent diffusion models. *arXiv preprint arXiv:2501.01761*, 2025. 2

[24] Luke Melas-Kyriazi, Christian Rupprecht, and Andrea Vedaldi. PC2: Projection-conditioned point cloud diffusion for single-image 3D reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12923–12932, 2023. 1

[25] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. FastSLAM: A factored solution to the simultaneous localization and mapping problem. *AAAI/IAAI*, 593598, 2002. 1

[26] Raul Mur-Artal and Juan D Tardós. ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 1

[27] Kazuto Nakashima and Ryo Kurazume. Learning to drop points for LiDAR scan synthesis. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 222–229. IEEE, 2021. 2

[28] Kazuto Nakashima and Ryo Kurazume. LiDAR data synthesis with denoising diffusion probabilistic models. In *2024 IEEE International Conference on Robotics and Automation*, pages 14724–14731. IEEE, 2024. 2, 6

[29] Kazuto Nakashima, Yumi Iwashita, and Ryo Kurazume. Generative range imaging for learning scene priors of 3D LiDAR data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1256–1266, 2023. 2

[30] Kazuto Nakashima, Xiaowen Liu, Tomoya Miyawaki, Yumi Iwashita, and Ryo Kurazume. Fast lidar data generation with rectified flows. *arXiv preprint arXiv:2412.02241*, 2024. 2

[31] Lucas Nunes, Rodrigo Marcuzzi, Benedikt Mersch, Jens Behley, and Cyrill Stachniss. Scaling diffusion models to real-world 3D LiDAR scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14770–14780, 2024. 1, 6

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pages 8748–8763. PmLR, 2021. 2

[33] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 2

[34] Haoxi Ran, Vitor Guizilini, and Yue Wang. Towards realistic scene generation with LiDAR diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14738–14748, 2024. 2, 4, 6

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2, 6

[36] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. *Advances in Neural Information Processing Systems*, 34:17480–17492, 2021. 6

[37] Martin Simon, Karl Amende, Andrea Kraus, Jens Honer, Timo Samann, Hauke Kaulbersch, Stefan Milz, and Horst Michael Gross. Complexer-YOLO: Real-time 3D object detection and tracking on semantic point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1

[38] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the International Conference on Machine Learning*, pages 2256–2265. pmlr, 2015. 2, 3

[39] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 1

[40] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo Open Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 6

[41] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3D architectures with sparse point-voxel convolution. In *Proceedings of the European Conference on Computer Vision*, pages 685–702. Springer, 2020. 2

[42] Michał J Tyszkiewicz, Pascal Fua, and Eduard Trulls. Gecco: Geometrically-conditioned point diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2128–2138, 2023. 1

[43] Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. LION: Latent point diffusion models for 3D shape generation. *Advances in Neural Information Processing Systems*, 35:10021–10039, 2022. 1

[44] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 30, 2017. 4

[45] Hai Wu, Wenkai Han, Chenglu Wen, Xin Li, and Cheng Wang. 3D multi-object tracking in point clouds based on prediction confidence-guided data association. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):5668–5677, 2021. 1

[46] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer V3: Simpler faster stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4840–4851, 2024. 7

[47] Yuwen Xiong, Wei-Chiu Ma, Jingkang Wang, and Raquel Urtasun. Learning compact representations for LiDAR completion and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1074–1083, 2023. 6, 1

[48] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. SqueezeSegV3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 1–19. Springer, 2020. 1

[49] Tianyi Yan, Junbo Yin, Xianpeng Lang, Ruigang Yang, Cheng-Zhong Xu, and Jianbing Shen. OLiDM: Object-aware LiDAR diffusion models for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9121–9129, 2025. 2

[50] Fukun Yin, Xin Chen, Chi Zhang, Biao Jiang, Zibo Zhao, Wen Liu, Gang Yu, and Tao Chen. ShapeGPT: 3D shape generation with a unified multi-modal language model. *IEEE Transactions on Multimedia*, 2025. 1

[51] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2

[52] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. Uni-ControlNet: All-in-one control to text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:11127–11150, 2023. 2

[53] Linqi Zhou, Yilun Du, and Jiajun Wu. 3D shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5826–5835, 2021. 1

[54] Hanqi Zhu, Jiajun Deng, Yu Zhang, Jianmin Ji, Qiuyu Mao, Houqiang Li, and Yanyong Zhang. VPFNet: Improving 3D object detection with virtual point based LiDAR and stereo data fusion. *IEEE Transactions on Multimedia*, 2022. 1

[55] Aleksei Zhuravlev, Zorah Lähner, and Vladislav Golyanik. Denoising functional maps: Diffusion models for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 1

[56] Vlas Zyrianov, Xiyue Zhu, and Shenlong Wang. Learning to generate realistic LiDAR point clouds. In *Proceedings of the European Conference on Computer Vision*, pages 17–35. Springer, 2022. 2, 6

[57] Vlas Zyrianov, Henry Che, Zhijian Liu, and Shenlong Wang. LidarDM: Generative LiDAR simulation in a generated world. *arXiv preprint arXiv:2404.02903*, 2024. 2