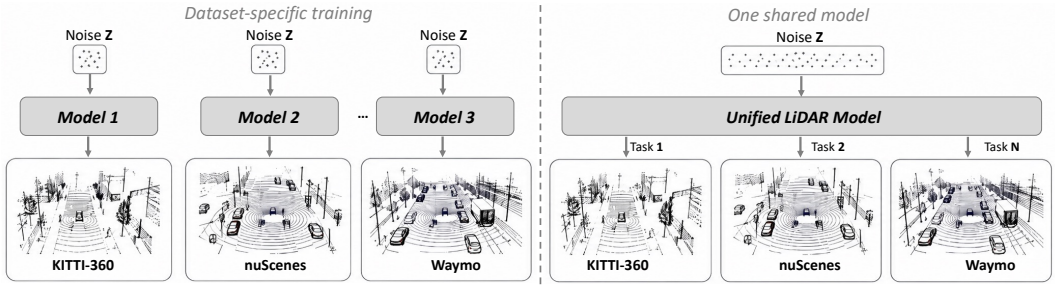
ANYLIDAR: ANY SENSOR IN ONE MODEL FOR LIDAR GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing LiDAR generators are typically trained on individual datasets, and directly pooling heterogeneous observations can degrade generation quality. We introduce AnyLiDAR, a unified generative model with geometry-conditioned ring modulation for multiple sensor domains. Our design exploits the circular scan order shared by full-sweep LiDAR observations while accounting for differences in ray layouts and depth and reflectance statistics. Dataset identity specifies the target distribution, while ray directions inform spatial attention and ring-context computation. The ring path aggregates current features and ray geometry along the scan order to predict position- and channel-dependent scale and shift coefficients for attention updates. This allows shared ring context to regulate the contribution of attention features according to measurement geometry and the current scan state. Trained on seven autonomous-driving datasets, AnyLiDAR achieves lower Fréchet Range Distance than all compared jointly trained baselines on five target domains, with a 10.2% reduction over R2Flow on KITTI-360. Generative pre-training further improves SemanticKITTI segmentation by 9.3 mIoU points over the same backbone trained from scratch.



(a) Existing Methods (b) AnyLiDAR (Ours)

Figure 1: **Joint generation from heterogeneous LiDAR data.** AnyLiDAR trains one generator on multiple datasets, using a ray-angle grid and dataset embedding to specify each sensor domain.

1 INTRODUCTION

Existing LiDAR generators are typically trained on individual datasets. Recent methods improve generation fidelity and efficiency through advances in range-view and latent representations (Zyrianov et al., 2022; Nakashima & Kurazume, 2024; Ran et al., 2024; Hu et al., 2024; Nakashima et al., 2025). These models learn observations from individual sensing configurations (Figure 1(a)). This setup requires a separate generator for each dataset, even when observations share scene structure and scan organization. Meanwhile, KITTI, nuScenes, and Waymo provide measurements from diverse sensors and environments (Geiger et al., 2012; Caesar et al., 2020; Sun et al., 2020). In practice, vehicle fleets continually collect observations from different sensor configurations with varying resolutions, creating a growing pool of heterogeneous data. This motivates learning one generator from heterogeneous data and selecting the sensor domain at generation time, as illustrated in Figure 1(b). Such a model should reuse common structure while retaining the depth and reflectance statistics of each represented domain.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The direct approach is to pool the datasets and jointly train an existing generator. However, Figure 2 shows that four existing generators obtain higher Fréchet Range Distance (FRD) on KITTI-360 under seven-dataset joint training than under single-dataset training. For example, R2Flow's FRD increases from 122.81 to 158.21. This degradation despite access to additional data motivates studying domain differences as a potential source of interference under joint training. The design question is therefore which scan structure can support shared computation, and how that computation should adapt to each domain and observation.

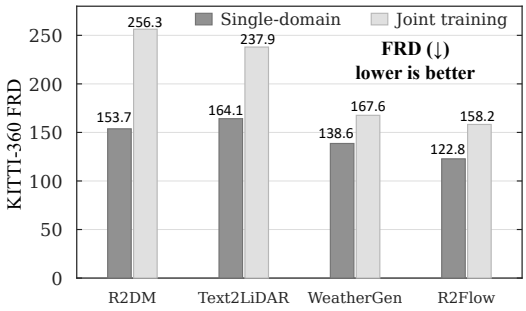


Figure 2: **Joint training increases target-domain FRD.** Each pair compares single-dataset and seven-dataset training for the same generator, evaluated on KITTI-360. All four generators obtain higher FRD under joint training (lower is better).

The full-sweep scans considered here share circular azimuthal order, providing a common sequence organization across sensors. This shared order does not imply identical measurement geometry: beam placement, field of view, and angular spacing vary, so the same tensor neighborhood can cover different angular regions. Sensor response and scene composition also change the distributions of depth and reflectance. Ray directions supply spatial geometry, while dataset identity specifies the target observation distribution.

A fixed ray can observe a nearby object, a distant surface, or no return across scans, and attention already responds to this variation. Under joint training, shared computation must accommodate different measurement layouts and observation statistics. We hypothesize that circular scan order can support shared context computation using ray geometry and current features conditioned on domain identity. Attention outputs carry spatial interactions into the residual update, making them a natural target for this context to regulate. Modulation provides an explicit way to adapt their contribution across sensor geometries and scan states while preserving the identity path.

We present AnyLiDAR, a conditional generative model that implements this principle through geometry-conditioned ring modulation. R2Flow uses Circular MHSA (multi-head self-attention) with beam-angle positional encoding (Nakashima et al., 2025), while WeatherGen models beam circles and central rays through Spider Mamba (Wu et al., 2025). Building on these structural choices, AnyLiDAR's Ring-scan Attention combines Circular MHSA with a ring path that aggregates ray geometry and features conditioned on domain identity. Circular MHSA forms spatial interactions, while the ring path predicts scale and shift coefficients for its output at each position and channel.

We train AnyLiDAR on seven autonomous-driving datasets and evaluate generation in the represented sensor domains. Under the same mixed-domain protocol, AnyLiDAR achieves the lowest FRD among the compared methods on all five domains where it is reported, reducing KITTI-360 FRD by 10.2% relative to R2Flow. It also achieves the lowest FPD on four of six domains and the lowest MMD on five. Generative pre-training improves SemanticKITTI segmentation by 9.3 mIoU points over the same backbone trained from scratch. These results establish generation quality and representation utility on the sensor and domain combinations covered by training.

Our contributions are threefold. (1) We study unified LiDAR generation through shared scan structure and conditional adaptation to heterogeneous depth and reflectance observations. (2) We introduce geometry-conditioned ring modulation to adapt attention updates to scan context at each position and channel. (3) We evaluate multi-domain generation and the transfer of learned representations to semantic segmentation.

2 Related Work

2.1 LiDAR Generation

LiDAR generators expose scan structure through different representations. Early work organized scans into two-dimensional point maps (Caccia et al., 2019). Range images associate measurements

with an angular lattice, while implicit representations support continuous sampling and ray-drop modeling (Zyrianov et al., 2022; Nakashima et al., 2023). UltraLiDAR learns a discrete codebook for completion, generation, and manipulation (Xiong et al., 2023). LiDM incorporates geometric priors into latent compression (Ran et al., 2024), and RangeLDM combines latent diffusion with projection correction and a range-guided discriminator (Hu et al., 2024).

LiDARGen uses score-based modeling to synthesize range and reflectance (Zyrianov et al., 2022), while R2DM adapts denoising diffusion probabilistic models through LiDAR-specific representations and spatial biases (Nakashima & Kurazume, 2024). R2Flow uses rectified flows to reduce sampling steps (Nakashima et al., 2025). L3DR improves geometric fidelity by predicting corrective point offsets in 3D for range-view outputs (Liu et al., 2026).

Text2LiDAR introduces text guidance through an equirectangular Transformer (Wu et al., 2024b). T2LDM adds self-conditioned representation guidance and supports text, semantic, and density-related conditions (Qu et al., 2026). WeatherGen targets diverse weather conditions (Wu et al., 2025), while Structure-to-Intensity Diffusion (SiD) conditions reflectance denoising on reconstructed geometry (Ni et al., 2026). U4D models LiDAR sequences through uncertainty-aware generation and spatiotemporal processing (Xu et al., 2026). AnyLiDAR retains range and reflectance on a common lattice, where explicit ray directions distinguish the measurement layouts of jointly modeled domains and, together with dataset identity, specify the target observation domain at generation time.

2.2 Sensor Conversion and Joint Generation

Generative range imaging supports resolution changes, restoration, and domain transfer through implicit representations and learned ray-drop effects (Nakashima et al., 2023). Sensor2Sensor converts monocular driving videos into multi-view camera and LiDAR logs, using 4D reconstruction to construct paired training data (Wang et al., 2026). Its raymap conditioning encodes camera intrinsics and extrinsics to accommodate different camera configurations.

AnyLiDAR jointly learns LiDAR observation distributions from heterogeneous datasets without requiring paired scans of the same scene across sensors. Dataset identity and a ray grid select a represented sensor domain for generation from noise. The aim is to preserve generation quality within each domain while sharing the backbone. Evaluation is restricted to sensor and domain combinations covered by training.

2.3 Scan-Aware Generative Architectures

LiDARGen and R2DM use range-image networks, with R2DM incorporating circular operations and positional information (Zyrianov et al., 2022; Nakashima & Kurazume, 2024). Text2LiDAR uses equirectangular attention (Wu et al., 2024b), while R2Flow combines circular neighborhood attention, global bottleneck attention, and beam-angle positional encoding (Nakashima et al., 2025). Selective state-space models offer efficient sequence processing (Gu & Dao, 2023). Mamba3D and Point Cloud Mamba adapt them to point-cloud analysis through local feature processing and sequence organization (Han et al., 2024; Zhang et al., 2025). WeatherGen uses Spider Mamba to model LiDAR beam circles and central rays (Wu et al., 2025).

AnyLiDAR builds on these structures by using ring context to regulate shared attention updates. Along the common circular order, the ring path aggregates ray geometry and features conditioned on domain identity to predict scale and shift coefficients at each position and channel. This geometry-conditioned ring modulation acts on Circular MHSA outputs, while global MHSA operates separately at the bottleneck.

3 Methodology

AnyLiDAR shares context computation along a common scan order across heterogeneous sensor domains while adapting feature updates to measurement geometry and scan content (Figure 3). Dataset identity and ray directions specify the target distribution and measurement layout, respectively. Ring-scan Attention combines Circular MHSA with geometry-conditioned ring modulation (Geometry Modulation) to adapt attention updates at each position and channel.

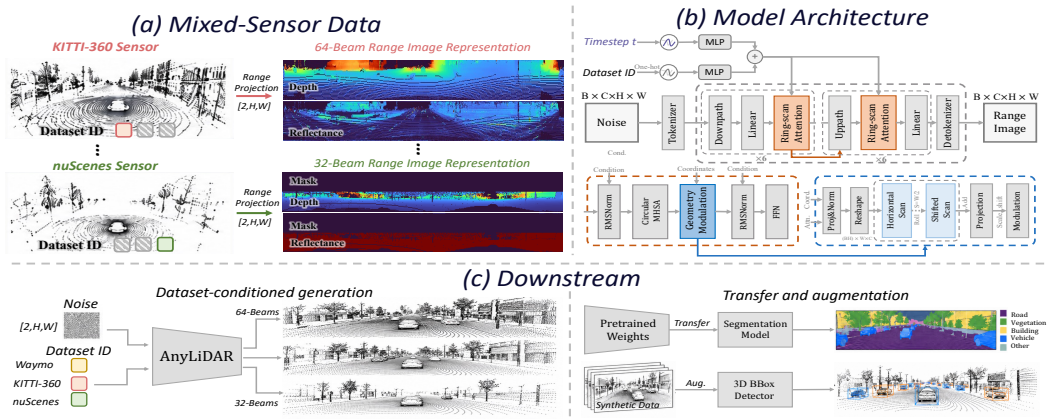


Figure 3: **Overview of AnyLiDAR.** (a) Depth and reflectance share a padded range-image representation across sensors. (b) Ring-scan Attention combines Circular MHSA and Geometry Modulation under dataset and timestep conditioning. (c) Domain-conditioned generation, transfer of pre-trained representations, and augmentation with synthetic scans.

3.1 Shared Scan Structure and Domain-Specific Observations

Task formulation. Let $\{\mathcal{D}_d\}_{d=1}^{D}$ denote LiDAR datasets with observation distributions $p_d(\boldsymbol{x})$. Pooling them produces $p_{\text{pool}}(\boldsymbol{x}) = \sum_{d=1}^{D} \pi_d p_d(\boldsymbol{x})$, where $\pi_d \geq 0$ is the sampling proportion and $\sum_d \pi_d = 1$. Our objective is to learn one shared generator that can target each p_d using a dataset condition $\boldsymbol{c}_d$ and its ray grid $\mathbf{G}_d$. Here a domain includes both the sensing configuration and the acquisition statistics of a dataset. The pair $(\boldsymbol{c}_d, \mathbf{G}_d)$ specifies a represented observation domain, while all backbone parameters are shared across domains.

Common scan order, explicit ray geometry. We represent each scan on an $H \times W = 64 \times 1024$ lattice that preserves beam order and circular azimuthal order. The latter gives full-sweep scans a common sequence organization, while the spacing of measurements remains sensor-dependent. An explicit grid $\mathbf{G}_d \in \mathbb{R}^{2\times H\times W}$ records elevation $\theta_{d,i}$ and azimuth $\phi_{d,i}$ in radians in the sensor frame, with i indexing spatial cells. This separates the topology used for shared computation from the ray geometry needed to interpret depth measurements. For a retained range r_i, the unit ray $\mathbf{u}_{d,i}$ gives the 3D point

$$\mathbf{p}_i = r_i \mathbf{u}_{d,i}, \qquad \mathbf{u}_{d,i} = \big(\cos\theta_{d,i}\cos\phi_{d,i},\ \cos\theta_{d,i}\sin\phi_{d,i},\ \sin\theta_{d,i}\big)^{\top}. \tag{1}$$

Depth, reflectance, and observation validity. At a returned cell, r_i and a_i denote range and reflectance. Their encodings ψ (log-range) and ν (reflectance normalization) yield $\boldsymbol{x}_i = (\psi(r_i), \nu(a_i))^{\top}$ and the two-channel scan $\boldsymbol{x} \in \mathbb{R}^{2\times H\times W}$. Native cells without returns use a finite encoding. Sensors with fewer beams are padded without vertical interpolation. The mask $\mathbf{M}_d \in \{0,1\}^{H\times W}$ has entries $M_{d,i} = 1$ at native cells, including no-return cells, and zero at padding. It excludes padding from the loss without adding a prediction channel.

3.2 Domain and Geometry Conditioning

Specifying the target distribution. Ray directions alone do not determine return frequencies, reflectance statistics, or scene composition. We therefore encode dataset identity with a one-hot vector $\boldsymbol{c}_d \in \{0,1\}^D$ and combine it with the embedding of flow time $t \in [0,1]$:

$$\boldsymbol{e}(t,d) = f_{\text{time}}(t) + f_{\text{data}}(\boldsymbol{c}_d) \in \mathbb{R}^{C_e}. \tag{2}$$

Here f_{time} and f_{data} embed time and dataset identity, and C_e is their shared width. Adaptive normalization predicts channel coefficients from $\boldsymbol{e}(t,d)$. These coefficients are shared across spatial positions, while the features and attention updates remain content-dependent. The ring modulation in Section 3.3 adds a specific organization to this adaptation by aggregating context along the common scan order. The domain embedding uses a three-layer multilayer perceptron (MLP) with SiLU activations and a zero-initialized final layer. Explicit ray geometry enters the attention and context paths separately.

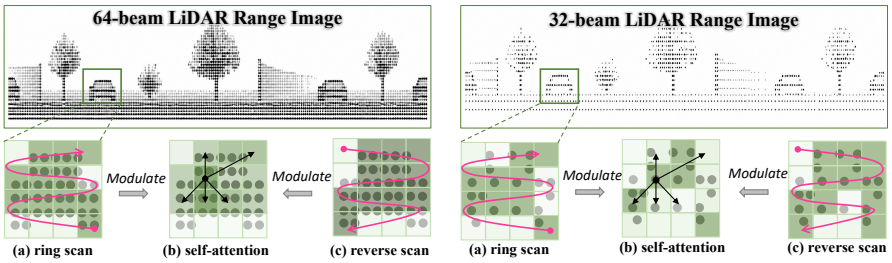


Figure 4: **Geometry-conditioned ring modulation.** Schematic examples of 64-beam (left) and 32-beam (right) observations. Context from forward (a) and reverse (c) ring scans predicts scale $\boldsymbol{\gamma}$ and shift $\boldsymbol{\beta}$ for the Circular MHSA output (b), before the identity residual is added.

Matching geometry to the feature hierarchy. For B scans, let d_b denote sample b's domain and $\mathbf{X}_t = (\boldsymbol{x}_{t,b})_{b=1}^{B} \in \mathbb{R}^{B\times 2\times H\times W}$ the batch state at time t. Following R2Flow (Nakashima et al., 2025), we tokenize $\mathbf{X}_t$, add learned positional embeddings, and process the features hierarchically. Pooling $\mathbf{G}_{d_b}$ yields $\mathbf{G}_b^{\ell} \in \mathbb{R}^{2\times H_\ell\times W_\ell}$ at scale ℓ, which the decoder reuses at the matching resolution. Positional embeddings encode token locations, while ray angles supply physical geometry to attention and ring context. Context evolves with $\mathbf{X}_t$ and requires no additional scene input.

3.3 Ring-scan Attention

Circular MHSA and geometry modulation. At scale ℓ, let $\mathbf{H} \in \mathbb{R}^{B\times H_\ell\times W_\ell\times C_\ell}$ denote features. Let $\mathbf{G}^{\ell} = (\mathbf{G}_b^{\ell})_{b=1}^{B}$ and $\mathbf{E} = (\boldsymbol{e}(t,d_b))_{b=1}^{B} \in \mathbb{R}^{B\times C_e}$. Following R2Flow (Nakashima et al., 2025), Circular MHSA computes $\mathbf{A} = \mathrm{CircularMHSA}(\mathbf{H}, \mathbf{G}^{\ell}; \mathbf{E})$. Figure 4 illustrates a common ring organization for 64- and 32-beam observations with different sampling layouts. Forward (a) and reverse (c) scans aggregate ray geometry with current features conditioned on $\mathbf{E}$ into context $\mathbf{U} \in \mathbb{R}^{B\times H_\ell\times W_\ell\times K_\ell}$. Our geometry modulation uses this context to modulate the Circular MHSA output (b) before the identity residual is added:

$$\begin{aligned} [\boldsymbol{\gamma}, \boldsymbol{\beta}] &= W_o\,\mathrm{Drop}(\mathbf{U}), \\ \mathbf{H}^{+} &= \mathbf{H} + \mathbf{A} \odot (\mathbf{1} + \boldsymbol{\gamma}) + \boldsymbol{\beta}. \end{aligned} \tag{3}$$

The projection W_o maps each context vector to $2C_\ell$ channels to produce $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$, which share the shape of $\mathbf{H}$ and $\mathbf{A}$. We use $[\cdot,\cdot]$ for channel concatenation, Drop for dropout, $\odot$ for elementwise multiplication, and $\mathbf{1}$ for an all-ones tensor.

The coefficients $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ vary across positions and channels. The modulation arrows in Figure 4 represent rescaling $\mathbf{A}$ by $\mathbf{1}+\boldsymbol{\gamma}$ and adding $\boldsymbol{\beta}$, with the identity path unchanged. These coefficients depend on ray geometry and current features, allowing modulation to vary with scan content and sampling state even for a fixed sensor. Figure 5 shows the complementary context supplied by forward and reverse scans at the marked location. A half-ring shift (not shown) places the original seam inside the shifted sequence, supplying context across that seam. Through $\mathbf{A} \odot \boldsymbol{\gamma}$, ring context controls the contribution of attention features. We zero-initialize W_o, so the interaction initially reduces to $\mathbf{H} + \mathbf{A}$.

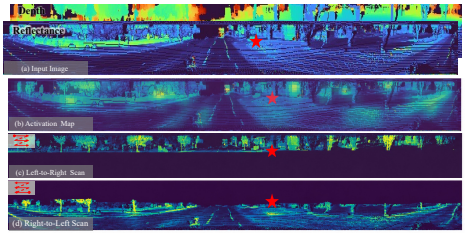


Figure 5: **Attention and bidirectional ring scan.** (a) Input depth and reflectance. (b) Circular MHSA activation map. (c,d) Forward and reverse scan views at the marked position.

Content and geometry for position-dependent modulation. We apply adaptive root-mean-square (RMS) normalization to $\mathbf{H}$ and project features and angles into a common space. At the current scale, let $\mathbf{g}_{b,i} = (\mathbf{G}_b^{\ell})_{:,i} \in \mathbb{R}^2$ be the pair of elevation and azimuth angles at spatial token i, and $\boldsymbol{e}_b = \boldsymbol{e}(t,d_b)$. Treating channel vectors as columns, the context input is

$$\begin{aligned} \widehat{\mathbf{H}}_{b,i} &= \frac{\mathbf{H}_{b,i}}{\sqrt{C_\ell^{-1}\sum_{c=1}^{C_\ell} H_{b,i,c}^2 + \epsilon}} \odot (\mathbf{1} + W_e \boldsymbol{e}_b), \\ \mathbf{Z}_{b,i} &= W_x \widehat{\mathbf{H}}_{b,i} + W_g \mathbf{g}_{b,i} \in \mathbb{R}^{K_\ell}, \qquad K_\ell = \max(K_{\min}, \lfloor C_\ell/2 \rfloor). \end{aligned} \tag{4}$$

Table 1: **Comparison of the aggregated LiDAR datasets.** Sensor configurations and scan counts for the seven training datasets. FoV denotes the vertical field of view, given as minimum and maximum elevation angles.

Attribute	KITTI-360	SemanticKITTI	RELLIS-3D	Waymo	PandaSet	Lyft	nuScenes
Beams	64	64	64	64	64	64	32
Scans used	50,348	19,130	13,532	29,667	8,240	58,212	34,149
FoV (°)	$[-25, +3]$	$[-25, +3]$	$[-25, +15]$	$[-17.6, +2.4]$	$[-25, +15]$	$[-17, +8]$	$[-30.67, +10.67]$
Scene Type	Urban/suburban	Urban/rural	Off-road	Urban/suburban	Urban/suburban	Urban	Dense urban

Here c indexes feature channels, and $K_{\min} = 16$ is the default minimum context width. W_e maps conditions to C_ℓ channels, while W_x and W_g project features and angles to K_ℓ channels. Spatially shared normalization conditions features on domain and time, while ray projections supply local geometry. Together, these inputs condition shared ring computation on sensor layout and current scan content. All projections are bias-free and shared across datasets, with ray angles projected linearly from raw coordinates. We set $\epsilon = 10^{-6}$ and initialize W_e to zero.

Bidirectional ring context. The projected features form $\mathbf{Z} \in \mathbb{R}^{B \times H_\ell \times W_\ell \times K_\ell}$. The curves in Figure 4 are schematic. We process the BH_ℓ rows of $\mathbf{Z}$ as ring sequences of length W_ℓ, using two directions and two boundaries. Let J reverse the azimuthal order and $\mathcal{R}_s$ circularly shift a row by s tokens, with $\mathcal{R}_{-s}$ its inverse. The separately parameterized operators $F_\rightarrow$ and $F_\leftarrow$ act independently on each row. Their bidirectional average $\mathcal{S}$ and the modulation context $\mathbf{U}$ are

$$\begin{aligned} \mathcal{S}(\mathbf{Z}) &= \tfrac{1}{2}\left[F_\rightarrow(\mathbf{Z}) + JF_\leftarrow(J\mathbf{Z})\right], \\ \mathbf{U} &= \tfrac{1}{2}\left[\mathcal{S}(\mathbf{Z}) + \mathcal{R}_{-\Delta}\mathcal{S}(\mathcal{R}_\Delta \mathbf{Z})\right], \qquad \Delta = \lfloor W_\ell/2 \rfloor. \end{aligned} \tag{5}$$

Both cuts reuse the directional operators. The half-ring shift changes only the sequence boundary. Shifts act on the combined feature-and-geometry sequence, and inverse shifts align outputs before averaging. $\mathbf{U}$ retains the shape of $\mathbf{Z}$ and supplies both traversal histories at each location. This does not guarantee equivariance to circular shifts. Rows correspond to beams at the initial scale and groups of beams after vertical merging. We instantiate the directional operators with Mamba (Gu & Dao, 2023), using default state size 16, convolution width 4, and expansion factor 2.

Hierarchical update placement. Ring-scan Attention applies geometry modulation to the Circular MHSA output in the last neighborhood-attention block at each encoder and decoder level, before the feed-forward residual. The bottleneck uses global MHSA without ring modulation. Implementation details appear in Appendix A.4, and flow-matching training and sampling in Appendix A.1.

4 Experiments

4.1 Datasets and Evaluation Metrics

Datasets. We aggregate approximately 213K scans from seven autonomous-driving LiDAR datasets: KITTI-360 (Liao et al., 2023), SemanticKITTI (Behley et al., 2019), RELLIS-3D (Jiang et al., 2021), Waymo (Sun et al., 2020), PandaSet (Xiao et al., 2021), Lyft (Lyft Level 5, 2019), and nuScenes (Caesar et al., 2020). Table 1 summarizes their sensor configurations, scene types, and scan counts before dataset repetition. We use PandaSet's 360-degree Pandar64 sweeps and represent all scans on the common 64×1024 lattice with dataset-specific ray grids, following Section 3.1. Qualitative results including nuScenes appear in Appendix A.5. Training settings are provided in Appendix A.2.

Evaluation metrics. Following prior work (Zyrianov et al., 2022; Nakashima & Kurazume, 2024; Wu et al., 2024b; Nakashima et al., 2025), we evaluate 10,000 generated scans per target domain against real scans from the same domain. We report Fréchet Range Distance (FRD) and Fréchet Point-cloud Distance (FPD) in range-view and point-cloud feature spaces, and Jensen–Shannon distance (JSD) and Maximum Mean Discrepancy (MMD) on bird's-eye-view (BEV) occupancy distributions. Lower values are better for all four metrics. Metric definitions and feature extractors are detailed in Appendix A.3.

Table 2: **Unconditional LiDAR generation across diverse target domains under unified multi-dataset training.** All methods are retrained on the same mixed-domain pool. Lower values are better. JSD and MMD are reported in units of 10^{-2} and 10^{-4}. Best comparable results are bold, AnyLiDAR is highlighted in gray, and "–" denotes an unavailable result.

Method	KITTI-360				PandaSet				Waymo			
	FRD	FPD	JSD	MMD	FRD	FPD	JSD	MMD	FRD	FPD	JSD	MMD
R2DM	256.31	13.12	12.21	1.59	895.34	612.78	8.94	1.82	1438.20	61.92	4.38	0.42
R2Flow	158.21	6.78	1.84	0.34	701.32	492.18	6.74	1.42	1204.66	49.31	**2.71**	0.30
Text2LiDAR	237.94	7.51	2.18	0.50	786.45	538.67	7.36	1.54	1287.45	52.84	3.41	0.33
WeatherGen	167.62	7.45	**1.79**	0.36	718.65	508.76	6.89	1.35	1239.32	50.62	3.27	0.32
AnyLiDAR	**142.13**	**4.61**	1.83	**0.32**	**633.74**	**433.10**	**6.22**	**1.27**	**1077.74**	**43.54**	2.79	**0.26**

Method	SemanticKITTI				RELLIS-3D				Lyft			
	FRD	FPD	JSD	MMD	FRD	FPD	JSD	MMD	FRD	FPD	JSD	MMD
R2DM	221.48	9.84	4.87	0.89	1986.43	9.75	5.84	1.87	–	11.83	5.22	0.40
R2Flow	196.72	**6.44**	2.58	0.54	1728.56	**6.51**	3.92	1.38	–	7.68	3.42	**0.24**
Text2LiDAR	188.35	7.12	3.06	0.63	1844.61	8.22	4.71	1.56	–	9.14	4.06	0.33
WeatherGen	216.84	6.71	**2.31**	0.58	1784.92	6.74	3.89	1.45	–	8.16	**3.21**	0.27
AnyLiDAR	**185.59**	9.55	2.40	**0.51**	**1652.31**	6.83	**3.68**	**1.30**	–	**7.20**	3.37	0.25

Table 3: **Unconditional LiDAR generation on KITTI-360.** We evaluate generation quality in point, range-view, and BEV spaces. Lower values are better for all metrics (↓). JSD and MMD are reported in units of 10^{-2} and 10^{-4}, respectively. Best results are shown in bold, and our method is highlighted in gray. FRID denotes Fréchet Range Image Distance.

Method	Model Design		Point Space	Range View		BEV Space	
	Backbone	Objective	FPD↓	FRD↓	FRID↓	JSD↓	MMD↓
R2DM	U-Net	Diffusion	**3.92**	153.73	**3.70**	2.17	0.68
LiDM	Curve U-Net	Latent Diff.	34.36	334.55	9.21	4.77	1.07
Text2LiDAR	Transformer	Diffusion	4.81	164.16	–	2.01	0.49
R2Flow	Transformer	Flow Matching	9.32	122.81	4.18	2.16	0.30
WeatherGen	Mamba + U-Net	Diffusion	6.15	138.62	–	1.99	0.39
SiD	Mamba + U-Net	Diffusion	5.62	135.13	–	2.00	0.41
AnyLiDAR	Mamba + Transformer	Flow Matching	5.48	**122.16**	6.28	**1.84**	**0.26**

4.2 Unconditional LiDAR Data Generation

We evaluate generation without scene-content conditioning, using sensor and dataset conditions to specify the target domain, in two complementary settings: unified multi-dataset training across diverse sensors and conventional single-dataset training on KITTI-360. For the mixed-domain comparison, we retrain R2DM, R2Flow, Text2LiDAR, and WeatherGen using the same data pool, preprocessing, and evaluation protocol as AnyLiDAR (Nakashima & Kurazume, 2024; Nakashima et al., 2025; Wu et al., 2024b; 2025). The compared methods span convolutional, Transformer, and Mamba-based backbones with diffusion and flow objectives.

Mixed-domain results. AnyLiDAR achieves the lowest FRD on all five reported domains and the lowest FPD on KITTI-360, PandaSet, Waymo, and Lyft under joint training (Table 2). On KITTI-360, its FRD, FPD, and MMD are 142.13, 4.61, and 0.32×10^{-4}, improving over R2Flow by 10.2%, 32.0%, and 5.9%, respectively. Its JSD of 1.83×10^{-2} is higher than WeatherGen's 1.79×10^{-2}. AnyLiDAR leads all four metrics on PandaSet and MMD on every domain except Lyft. JSD rankings vary across domains, while R2Flow achieves lower FPD on SemanticKITTI and RELLIS-3D. Thus, the consistent FRD gains are accompanied by broad MMD improvements, but do not extend uniformly to point-cloud feature distributions.

Table 4: **Architecture and conditioning ablations.** Models are jointly trained on mixed-sensor data from seven datasets and evaluated on KITTI-360. Each subtable varies one design aspect. Gray rows denote the full model. Lower FRD and FPD are better.

Configuration	FRD↓	FPD↓
No dataset condition	268.72	13.84
Text embedding	157.42	5.17
Learned embedding	151.86	4.95
One-hot + MLP	**142.13**	**4.61**

(a) Dataset conditioning.

Configuration	FRD↓	FPD↓
Without ring path	157.52	6.70
After FFN	160.38	6.94
Residual addition	150.86	5.38
Ring Path	**142.13**	**4.61**

(b) Ring context.

Scan	Shift	Update	FRD↓	FPD↓
Uni.	–	Scale	153.68	5.67
Bi.	–	Scale	150.83	5.36
Bi.	$W_\ell/2$	Scale	146.36	5.12
Bi.	$\mathbf{W_\ell/2}$	**Affine**	**142.13**	**4.61**

(c) Ring-scan design.

KITTI-360 single-dataset results. AnyLiDAR obtains the lowest FRD, JSD, and MMD, while R2DM obtains lower FPD and FRID (Table 3). Compared with R2Flow, JSD and MMD decrease by 14.8% and 13.3%, indicating closer agreement with real BEV occupancy distributions. The FRD reduction relative to R2Flow is 0.5% under single-dataset training and 10.2% under joint training (Table 2). This pattern is consistent with our design goal of sharing scan structure while adapting feature updates to heterogeneous observations.

4.3 Ablation Studies

Ablations use seven-dataset mixed-sensor training with evaluation on KITTI-360 (Table 4). Component variants share training and sampling settings, with Small as the default. Scaling compares Small, Base, and Large variants of AnyLiDAR (85.1M, 322.2M, 1.25B parameters) and R2Flow (81.0M, 307.1M, 1.19B) across 1, 3, 5, and 7 datasets. At each size, AnyLiDAR extends the R2Flow backbone only with geometry-conditioned ring modulation. Variant definitions and configurations appear in Appendix A.2.

Dataset conditioning. Removing dataset conditioning raises FRD/FPD from 142.13/4.61 to 268.72/13.84. One-hot + MLP also outperforms both embedding alternatives. Since ray geometry is retained, this degradation suggests that measurement layout alone does not adequately specify the target distribution. The result supports distinct roles for the two conditions: ray geometry describes measurement directions, while dataset identity conditions shared computation on domain-dependent depth and reflectance statistics.

Ring context. FRD is 157.52 without the ring path, 160.38 after the FFN, and 150.86 with direct residual addition, versus 142.13 for the full model. FPD follows the same ordering. Placing ring context after the FFN performs worse than removing it, indicating that connection placement matters. The gain over directly adding ring features to attention supports using ring context to regulate attention through scale and shift.

Ring-scan design. With scale modulation, bidirectional scanning and the half-ring shift reduce FRD from 153.68 to 150.83 and 146.36. This supports aggregating complementary traversal histories across directions and ring boundaries. With the scan construction fixed, adding $\boldsymbol{\beta}$ lowers FRD/FPD to 142.13/4.61. This gain indicates that an additive context term remains useful alongside multiplicative control of attention features.

Data and model scaling. Figure 6 evaluates the added ring modulation across data and model scales. AnyLiDAR improves FRD over R2Flow throughout. For Small, the reduction grows from 0.53% with one dataset to 10.16% with seven, showing a larger improvement under joint training. With seven datasets, Base and Large reduce FRD by 8.37% and 12.21%, respectively. The parameter overhead is 4.9–5.1% across sizes. Expanding the training pool raises FRD for both methods, while larger models lower FRD. AnyLiDAR's increase from one to seven datasets falls from 16.35% for Small to 10.03% for Large, suggesting that capacity helps accommodate heterogeneous observations within a shared generative model.

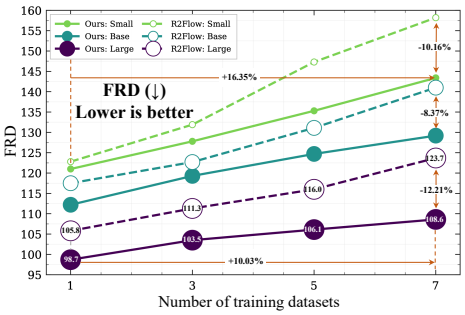


Figure 6: **Data and model scaling.** FRD for R2Flow (dashed) and AnyLiDAR with ring modulation (solid) across data and model scales. Lower is better.

Table 5: **Range-image semantic segmentation on SemanticKITTI and PandaSet.** We report validation mIoU and pointwise accuracy (Acc). Higher is better. Parentheses show gains over the corresponding supervised-only baseline. Best results are bold.

Method	Pre-trained	SemanticKITTI		PandaSet	
		mIoU↑	Acc↑	mIoU↑	Acc↑
RangeViT	Cityscapes	48.2	89.9	64.9	87.4
R2Flow *Sup.*	Random	45.7	89.2	64.4	87.0
+ R2Flow	Generation	49.1 (+3.4)	90.2 (+1.0)	65.3 (+0.9)	87.5 (+0.5)
AnyLiDAR *Sup.*	Random	46.3	89.6	64.9	87.4
+ **AnyLiDAR**	Generation	**55.6** (+9.3)	**91.4** (+1.8)	**66.1** (+1.2)	**87.8** (+0.4)

Table 6: **Synthetic augmentation on SemanticKITTI.** Numeric column headings denote labeled-data percentages. We report mIoU.

Generator	Backbone	mIoU (↑)		
		1%	10%	100%
Sup.only	SPUNet	47.0	61.5	68.7
R2Flow	SPUNet	52.1	64.2	69.1
AnyLiDAR	SPUNet	**55.4**	**66.0**	**69.5**
Sup.only	PTv3	47.8	62.2	69.4
R2Flow	PTv3	52.9	65.0	69.7
AnyLiDAR	PTv3	**56.3**	**66.8**	**70.1**

4.4 Downstream Applications

Range Image Segmentation. Table 5 compares 20-epoch supervised segmentation from random initialization (Random) and generative checkpoints (Generation), reporting mean intersection over union (mIoU) and pointwise accuracy (Acc). AnyLiDAR pre-training yields 55.6 and 66.1 mIoU on the SemanticKITTI and PandaSet validation sets, improving over its randomly initialized counterpart by 9.3 and 1.2 points. R2Flow pre-training yields gains of 3.4 and 0.9 points over its own supervised baseline. Pre-trained AnyLiDAR achieves the highest mIoU and accuracy among the compared methods on both datasets, including Cityscapes-pre-trained RangeViT (Ando et al., 2023). On SemanticKITTI, the gap between AnyLiDAR and R2Flow grows from 0.6 mIoU points with random initialization to 6.5 points after pre-training. This suggests that the transfer advantage extends beyond the difference between their supervised baselines. These within-model comparisons support the transfer of generative representations learned from heterogeneous scans to semantic segmentation, with gains varying across datasets.

Synthetic Data Augmentation. We generate 10,000 SemanticKITTI scans per generator and retain the top 1% of pointwise pseudo-labels by confidence from a frozen, supervised Point Transformer V3 (PTv3) teacher (Wu et al., 2024a). These sparsely labeled scans supplement fully annotated real scans for training a sparse convolutional U-Net (SPUNet) (Choy et al., 2019) and PTv3. At each real-data fraction, augmentation variants share the real subset, synthetic scan count, and teacher. All models start from random initialization with matched optimization budgets and are evaluated on the real validation split. AnyLiDAR achieves the highest mIoU for both backbones at all three fractions (Table 6). With 1% of real training scans, it improves SPUNet and PTv3 over supervised-only training by 8.4 and 8.5 points, exceeding R2Flow augmentation by 3.3 and 3.4 points. Gains over supervised-only training decrease to 4.5/4.6 points at 10% and 0.8/0.7 points at 100%, respectively. This consistent trend across backbones indicates that synthetic scans contribute most when labeled real scenes are limited, while still providing gains with the full real training set.

5 Conclusion

We presented AnyLiDAR for unified generation from heterogeneous LiDAR data. Motivated by the degradation in target domains under naive joint training, the model shares circular scan structure and adapts feature updates to measurement geometry and scan content. Geometry-conditioned ring modulation implements this principle by aggregating ray directions and current features conditioned on domain identity to predict adjustments to attention updates at each position and channel. Trained on seven datasets, AnyLiDAR achieves the lowest FRD among the compared jointly trained methods on all five reported domains. The FRD advantage over R2Flow is larger under joint training than under single-dataset training, supporting its use for heterogeneous observations. Ablations favor modulating attention updates over the tested alternative ring connections, while scaling experiments show consistent gains over R2Flow with approximately 5% additional parameters. Generative pre-training improves downstream semantic segmentation on real data. Synthetic augmentation also benefits both sparse convolutional and point Transformer backbones under a shared pseudo-labeling protocol, with the largest gains when fewer labeled real training scans are used. These findings support shared scan computation with conditional adaptation as a design principle for multi-domain LiDAR generation and learning representations that transfer to downstream semantic segmentation.

AI USE STATEMENT

We used AI-assisted tools for language editing, manuscript organization, and feedback on methodological explanations, experimental design, and result interpretation. We also used them to check notation, references, and LaTeX formatting. The authors take full responsibility for the technical content, experimental results, and conclusions presented in this paper.

ETHICS STATEMENT

This work studies LiDAR generation using existing autonomous-driving datasets, with applications to observation synthesis and representation learning. Generated scans may reflect biases in the source datasets, including uneven coverage of sensors, environments, and scene categories. Pseudo-label errors may also propagate into downstream models trained with synthetic data. The reported benchmark results do not establish safety for autonomous deployment. Downstream use should respect the source datasets' access and licensing conditions and include validation on real observations from the intended operating environment.

REPRODUCIBILITY STATEMENT

We describe the shared scan representation, conditioning interface, ring modulation, training objective, and generation procedure. We also summarize the datasets, model configurations, training settings, ablation variants, and evaluation metrics. The experimental protocols distinguish joint training from single-dataset training and specify the comparisons used to evaluate generative pre-training and synthetic augmentation.

REFERENCES

Angelika Ando, Spyros Gidaris, Andrei Bursuc, Gilles Puy, Alexandre Boulch, and Renaud Marlet. RangeViT: Towards vision transformers for 3D semantic segmentation in autonomous driving. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5240–5250. IEEE, 2023.

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*, pp. 9297–9307, 2019.

Lucas Caccia, Herke van Hoof, Aaron Courville, and Joelle Pineau. Deep generative modeling of LiDAR data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5034–5040, 2019.

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.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 3070–3079. IEEE, 2019.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Xu Han, Yuan Tang, Zhaoxuan Wang, and Xianzhi Li. Mamba3d: Enhancing local features for 3D point cloud analysis via state space model. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 4995–5004, 2024.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Qianjiang Hu, Zhimin Zhang, and Wei Hu. Rangeldm: Fast realistic LiDAR point cloud generation. In *European Conference on Computer Vision*, 2024.

Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripalli. RELLIS-3D dataset: Data, benchmarks and analysis. In *IEEE International Conference on Robotics and Automation*, pp. 1110–1116, 2021.

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*, 45(3):3292–3310, 2023.

Jianhua Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

Quan Liu, Xiaoqin Zhang, Ling Shao, and Shijian Lu. L3DR: 3D-aware LiDAR diffusion and rectification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17153–17163, 2026.

Lyft Level 5. Lyft Level 5 AV Dataset. `https://github.com/lyft/nuscenes-devkit`, 2019. Autonomous-driving perception dataset and development kit.

Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. RangeNet++: Fast and accurate LiDAR semantic segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4213–4220, 2019.

Kazuto Nakashima and Ryo Kurazume. LiDAR data synthesis with denoising diffusion probabilistic models. In *IEEE International Conference on Robotics and Automation*, 2024. doi: 10.1109/ICRA57147.2024.10611480.

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*, pp. 1256–1266, 2023.

Kazuto Nakashima, Xiaowen Liu, Tomoya Miyawaki, Yumi Iwashita, and Ryo Kurazume. Fast LiDAR data generation with rectified flows. In *IEEE International Conference on Robotics and Automation*, pp. 10057–10063, 2025.

Peiyang Ni, Longyu Yang, Lu Zhang, Kuniaki Saito, Yap-Peng Tan, Fumin Shen, Heng Tao Shen, Xiaofeng Zhu, and Ping Hu. Structure-to-intensity diffusion for adverse-weather LiDAR generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35904–35914, 2026.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017.

Wentao Qu, Guofeng Mei, Yang Wu, YongShun Gong, Xiaoshui Huang, and Liang Xiao. A self-conditioned representation guided diffusion model for realistic text-to-LiDAR scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9434–9444, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8748–8763. PMLR, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

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*, pp. 14738–14748, 2024.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Jiahao Wang, Bo Sun, Yijing Bai, Vincent Casser, Songyou Peng, Zehao Zhu, Meng-Li Shih, Xander Masotto, Shih-Yang Su, Kanaad Parvate, Tiancheng Ge, Linn Bieske, Dragomir Anguelov, Mingxing Tan, and Chiyu Max Jiang. Sensor2sensor: Cross-embodiment sensor conversion for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4840–4851. IEEE, 2024a.

Yang Wu, Kaihua Zhang, Jianjun Qian, Jin Xie, and Jian Yang. Text2lidar: Text-guided LiDAR point clouds generation via equirectangular transformer. In *European Conference on Computer Vision*, 2024b.

Yang Wu, Yun Zhu, Kaihua Zhang, Jianjun Qian, Jin Xie, and Jian Yang. Weathergen: A unified diverse weather generator for LiDAR point clouds via spider mamba diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17019–17028, 2025.

Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, Yunlong Wang, and Diange Yang. PandaSet: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 3095–3101, 2021. doi: 10.1109/ITSC48978.2021.9565009.

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*, pp. 1074–1083, 2023.

Xiang Xu, Alan Liang, Youquan Liu, Linfeng Li, Lingdong Kong, Ziwei Liu, and Qingshan Liu. U4D: Uncertainty-aware 4D world modeling from LiDAR sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10027–10039, 2026.

Tao Zhang, Haobo Yuan, Lu Qi, Jiangning Zhang, Qianyu Zhou, Shunping Ji, Shuicheng Yan, and Xiangtai Li. Point cloud mamba: Point cloud learning via state space model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 10121–10130, 2025.

Vlas Zyrianov, Xiyue Zhu, and Shenlong Wang. Learning to generate realistic LiDAR point clouds. In *European Conference on Computer Vision*, 2022.

Table 7: **Generative training settings.** Single-dataset and joint training share the optimization settings. Budgets count scan presentations.

Setting	Single-dataset	Joint training
Training data	KITTI-360	Seven-dataset pool
Input channels	Range, reflectance	
Input resolution	64×1024	
Projection / range encoding	Spherical / log-scale	
Depth bounds (m)	$[1.45, 80.0]$	
Flow formulation / σ	ICFM / 0	
Loss / timestep distribution	L_2 / $\mathcal{U}(0,1)$	
Learning rate	10^{-4}	
Adam (β_1, β_2) / ϵ_{opt}	$(0.9, 0.99)$ / 10^{-8}	
Weight decay	0	
Configured batch size	24	
Gradient accumulation steps	1	
Training budget (scans)	2,560,000	
Warmup budget (scans)	80,000	
EMA decay / update interval	0.995 / 10 updates	
Mixed precision	Disabled	

A Appendix

A.1 Joint Flow Training and Generation

Training objective. We train a shared conditional vector field with each scan paired with its domain identity and ray grid. Let $\mathbf{X}_1 = (\boldsymbol{x}_{1,b})_{b=1}^B$ be a batch of observed scans, $\mathcal{B} = (d_b)_{b=1}^B$ its domain indices, and $V_{\boldsymbol{\theta}}(\mathbf{X}_t, t; \mathcal{B})$ the velocity tensor with parameters $\boldsymbol{\theta}$. Conditioning on $\mathcal{B}$ includes the associated one-hot vectors and ray grids. Following conditional flow matching (Nakashima et al., 2025), independent Gaussian noise $\mathbf{X}_0 = (\boldsymbol{x}_{0,b})_{b=1}^B$ defines $\mathbf{X}_t = (1-t)\mathbf{X}_0 + t\mathbf{X}_1$ for $t \sim \mathcal{U}(0,1)$. We optimize

$$\mathcal{L}_{\text{FM}} = \mathbb{E}\left[\frac{1}{B}\sum_{b=1}^{B} \frac{\|\mathbf{M}_{d_b} \odot ([V_{\boldsymbol{\theta}}(\mathbf{X}_t, t; \mathcal{B})]_b - (\boldsymbol{x}_{1,b} - \boldsymbol{x}_{0,b}))\|_F^2}{2\sum_i M_{d_b,i}}\right]. \tag{6}$$

The expectation is over scan sampling, Gaussian noise, and time. The squared Frobenius norm sums over spatial cells and both channels, with $\mathbf{M}_{d_b}$ broadcast over channels. The denominator counts native cells times two channels, normalizing each scan independently of its beam count. Native no-return cells remain encoded targets. Feature operations receive no validity mask and can include padded cells; masking excludes only their direct contribution to the loss. Joint optimization shares the vector-field parameters, while domain and geometry conditions let the feature updates depend on each target distribution.

Conditional observation generation. At generation time, the selected dataset conditions and ray grids remain fixed as the solver integrates

$$\frac{\mathrm{d}\mathbf{X}_t}{\mathrm{d}t} = V_{\boldsymbol{\theta}}(\mathbf{X}_t, t; \mathcal{B}), \qquad \mathbf{X}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{7}$$

Decoding discards padding and reconstructs retained returns using Eq. 1. Generation requires only noise and a represented sensor–domain pair. Sampling settings are detailed in Appendix A.2.

A.2 Training Settings

Generative model training. We consider joint training on the seven-dataset pool in Table 1 and single-dataset training on KITTI-360. In the joint setting, each scan is associated with its dataset identity and ray grid, while all backbone parameters are shared. Training uses the flow-matching objective in Eq. 6 on two-channel range and reflectance tensors of size 64×1024. Padded cells are excluded from the loss. Table 7 summarizes the training settings.

Table 8: **Model configurations.** Headers give (base channel width, depth); parameter counts are in millions (M). Small is the default configuration.

Method	Small $(128, 12)$	Base $(256, 16)$	Large $(512, 16)$
R2Flow	81.0	307.1	1,190
AnyLiDAR	85.1	322.2	1,250

The configured batch size is 24 with one accumulation step. Converting scan budgets to optimizer steps requires the effective global batch size and the handling of partial batches. The configuration does not specify whether batch size is global or per device, the optimizer variant, or the learning-rate schedule. Dataset repetition is disabled, and sampling proportions require the data-loader policy. ICFM denotes independent conditional flow matching, with $\sigma = 0$ corresponding to the interpolation in Eq. 6. EMA denotes an exponential moving average of model parameters. The optimizer constant ϵ_{opt} is distinct from the RMS normalization constant ϵ in Eq. 4.

Model and baseline configurations. The backbone and geometry-conditioned interaction are described in Section 3.3, with attention and update-placement details in Appendix A.4. For the mixed-domain comparison, R2DM, R2Flow, Text2LiDAR, and WeatherGen are retrained on the same data pool with the same preprocessing and evaluation protocol as AnyLiDAR. The exact conditioning interfaces and training budgets for these baselines remain to be specified from their run configurations.

Table 8 lists the Small, Base, and Large configurations used for scaling. At each size, AnyLiDAR retains the R2Flow backbone and adds geometry-conditioned ring modulation. This comparison evaluates the complete module, which combines ray geometry with current scan features to modulate attention updates. Small is used unless otherwise specified. In parameter counts, M and B denote 10^6 and 10^9, respectively.

Ablation labels. In Table 4a, *One-hot + MLP* uses $f_{\text{data}}(\boldsymbol{c}_d)$ from Eq. 2. *No dataset condition* removes dataset identity while retaining ray geometry. *Learned embedding* and *Text embedding* denote alternative representations of dataset identity. The latter uses the CLIP text encoder (Radford et al., 2021). In Table 4b, *Without ring path* removes ring context, *After FFN* places its connection after the feed-forward network (FFN), and *Residual addition* uses an additive connection to the attention branch. *Ring Path* denotes the full modulation in Eq. 3. In Table 4c, *Uni.* and *Bi.* denote one and two scan directions. Under *Shift*, "–" disables the shifted traversal, while $W_\ell/2$ denotes the half-ring offset Δ in Eq. 5, since all evaluated widths are even. *Scale* uses multiplicative modulation only, setting $\boldsymbol{\beta} = \mathbf{0}$ in Eq. 3. *Affine* retains both scale and shift. Thus, *Scale* differs from *Residual addition*, which adds ring features to attention without rescaling its output. A dash in a metric column instead denotes an unavailable result. *Latent Diff.* in Table 3 abbreviates latent diffusion.

Sampling and evaluation. Generation starts from Gaussian noise with the target dataset condition and its ray grid fixed throughout integration of Eq. 7. We generate 10,000 scans per evaluated target domain. The configuration selects the adaptive Dormand–Prince 5(4) solver (Dopri5) and sets the sampling-step parameter to 32, with classifier-free guidance disabled. Solver tolerances, function-evaluation counts, and sampling batch size remain to be confirmed.

Downstream labels. In Table 5, *Sup.* and *Random* denote supervised training from random initialization. A "+" row and *Generation* denote initialization from the corresponding generative checkpoint followed by supervised fine-tuning. *Cityscapes* specifies RangeViT's pre-training data. Parenthesized gains are absolute percentage-point differences from each model's random-initialization baseline. In Table 6, *Sup.only* uses real data without synthetic augmentation. Column percentages specify the fraction of fully annotated real training scans, distinct from the 1% of synthetic point labels retained by confidence. Both mIoU and Acc are reported as percentages, with higher values preferred.

Downstream training. For range-image segmentation, generative pre-training initializes the corresponding backbone, which is then fine-tuned for 20 epochs. The supervised-only counterparts are trained from random initialization for the same number of epochs. Evaluation uses the Se-

manticKITTI and PandaSet validation splits and reports mIoU and point-wise accuracy. The optimizer, learning-rate schedule, batch size, augmentation, and checkpoint-selection settings remain to be specified.

A.3 Evaluation Metrics

We compare 10,000 generated scans per target domain with real scans from the same sensor domain. All generation metrics below are lower-is-better. FRID is additionally reported in the single-dataset comparison.

Fréchet feature distances. FRD compares the distributions of range-view features extracted by a SemanticKITTI-pretrained RangeNet-53 encoder (Milioto et al., 2019). Let $(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$ and $(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ denote the empirical mean and covariance of real and generated features, respectively. Following the Fréchet feature distance (Heusel et al., 2017), FRD is computed as

$$D_{\mathrm{FRD}} = \|\boldsymbol{\mu}_r - \boldsymbol{\mu}_g\|_2^2 + \mathrm{Tr}\left(\boldsymbol{\Sigma}_r + \boldsymbol{\Sigma}_g - 2\left(\boldsymbol{\Sigma}_r^{1/2}\boldsymbol{\Sigma}_g\boldsymbol{\Sigma}_r^{1/2}\right)^{1/2}\right). \tag{8}$$

Here Tr denotes matrix trace and $\|\cdot\|_2$ the Euclidean norm. FPD applies the same formula to ShapeNet-pretrained PointNet features of reconstructed point clouds (Chang et al., 2015; Qi et al., 2017). FRID (Fréchet Range Image Distance) instead uses RangeNet-21 features from range images without reflectance, following the LiDM evaluation protocol implemented in R2Flow (Ran et al., 2024; Nakashima et al., 2025). FRD and FRID therefore use different feature extractors and are not interchangeable.

BEV occupancy metrics. We follow the BEV implementation of R2Flow (Nakashima et al., 2025), retaining its metric labels. Let p and q be normalized aggregate occupancy histograms of real and generated scans, and $m = (p+q)/2$. The reported JSD is the Jensen–Shannon distance, namely the square root of the divergence (Lin, 1991):

$$D_{\mathrm{JSD}}(p,q) = \sqrt{\tfrac{1}{2}D_{\mathrm{KL}}(p\|m) + \tfrac{1}{2}D_{\mathrm{KL}}(q\|m)}. \tag{9}$$

Here $D_{\mathrm{KL}}(p\|m) = \sum_j p_j \log(p_j/m_j)$, with natural logarithms. For n_r real and n_g generated scans, let $\mathbf{h}_{r,j}$ and $\mathbf{h}_{g,k}$ be normalized per-scan BEV histograms. The reported MMD is the biased squared maximum mean discrepancy (Gretton et al., 2012):

$$\begin{aligned}\widehat{\mathrm{MMD}}^2 = &\frac{1}{n_r^2}\sum_{j,j'=1}^{n_r} k(\mathbf{h}_{r,j}, \mathbf{h}_{r,j'}) + \frac{1}{n_g^2}\sum_{k,k'=1}^{n_g} k(\mathbf{h}_{g,k}, \mathbf{h}_{g,k'}) \\ &- \frac{2}{n_r n_g}\sum_{j=1}^{n_r}\sum_{k=1}^{n_g} k(\mathbf{h}_{r,j}, \mathbf{h}_{g,k}),\end{aligned} \tag{10}$$

where $k(\mathbf{u},\mathbf{v}) = \exp(-\|\mathbf{u}-\mathbf{v}\|_2^2/(2\sigma_k^2))$ is the Gaussian kernel with bandwidth σ_k. Diagonal terms are included. Spatial bounds, binning, and kernel settings follow the same reference implementation.

A.4 Geometry and Modulation Implementation

This section details geometry alignment, projection dimensions, and Circular MHSA. Ring-scan Attention combines Circular MHSA with geometry-conditioned ring modulation, labeled Geometry Modulation in Figure 3(b). The ring context $\mathbf{U}$ from Eq. 5 is passed directly to the modulation projection in Eq. 3.

Notation and diagram labels. At scale ℓ, H_ℓ and W_ℓ denote spatial height and width, C_ℓ the feature width, and K_ℓ the ring-context width. B is the batch size, while b, i, and c index samples, spatial tokens, and channels. The ray grid $\mathbf{G}_b^\ell$ supplies the angle pair $\mathbf{g}_{b,i}$ at token i. In Figure 3, *RMSNorm* denotes adaptive root-mean-square normalization, and *Proj&Norm* groups the normalization and projections in Eq. 4. *Horizontal Scan* follows each row's azimuthal order. *Shifted Scan* applies the same operators after a half-ring offset, with inverse alignment before fusion as in Eq. 5. The diagram's roll offset S corresponds to Δ at the active scale. SiLU denotes the sigmoid linear unit activation.

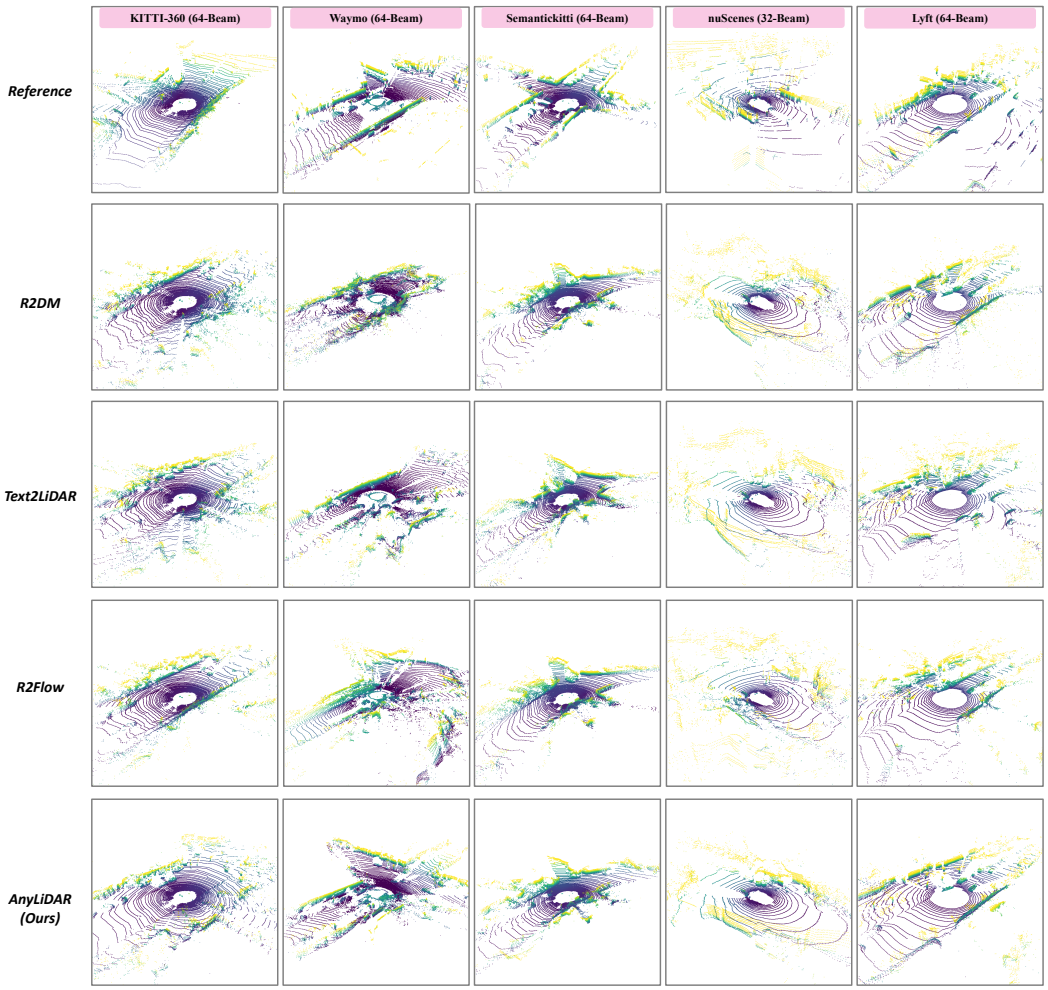


Figure 7: **Qualitative comparison across sensor domains.** Columns show KITTI-360, Waymo, SemanticKITTI, nuScenes, and Lyft. Rows show reference scans and samples from R2DM, Text2LiDAR, R2Flow, and AnyLiDAR. All AnyLiDAR samples come from one jointly trained model, covering 64-beam layouts and the 32-beam nuScenes layout.

Geometry alignment. All scan features are computed from the current flow state $\mathbf{X}_t$. We initialize $\mathbf{H}^0 = \mathrm{Tokenize}(\mathbf{X}_t) + \mathbf{P}$, where $\mathbf{P}$ is a learned positional embedding on the token lattice, broadcast over the batch. For sample b with domain index d_b, the model selects $\mathbf{G}_{d_b}$ from a registered table bank. With initial patch size $(1, 4)$, the geometry at successive resolutions is

$$\mathbf{G}_b^0 = \mathrm{AvgPool}_{1\times 4}(\mathbf{G}_{d_b}), \qquad \mathbf{G}_b^{\ell+1} = \mathrm{AvgPool}_{2\times 2}(\mathbf{G}_b^{\ell}). \tag{11}$$

The pooling operations take non-overlapping arithmetic averages of elevation and azimuth in radians. Encoder features and coordinates are stored together, and the decoder reuses the coordinates at the corresponding resolution. Coarse coordinates consequently represent pooled angular locations. The learned embedding $\mathbf{P}$ is independent of the ray grid; physical geometry is supplied explicitly through rotary encoding in Circular MHSA and the context projection in Eq. 4. During sampling, $\mathbf{X}_t$ evolves from Gaussian noise, so the context requires no additional scene-content input.

Projection dimensions. At a fixed block and scale ℓ, $W_e \in \mathbb{R}^{C_\ell \times C_e}$ maps conditioning embeddings to channel scales, while $W_x \in \mathbb{R}^{K_\ell \times C_\ell}$ and $W_g \in \mathbb{R}^{K_\ell \times 2}$ project features and ray angles into the context space. The output map $W_o \in \mathbb{R}^{2C_\ell \times K_\ell}$ produces the concatenated vectors $[\boldsymbol{\gamma}_{b,i}, \boldsymbol{\beta}_{b,i}]$. These maps act independently at each spatial position on column vectors; block and scale indices are omitted. Thus $\widehat{\mathbf{H}}$ has the shape of $\mathbf{H}$, $\mathbf{Z}$ and $\mathbf{U}$ have K_ℓ channels, and both modulation tensors have C_ℓ channels.

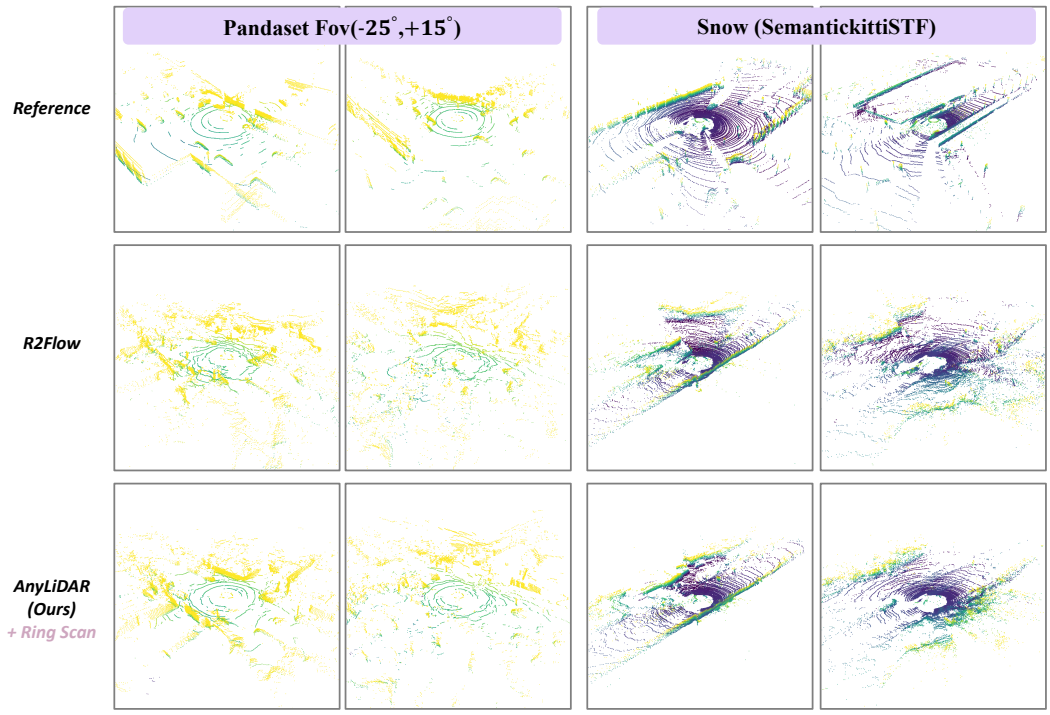


Figure 8: **PandaSet and weather adaptation.** The left panels show PandaSet examples with a vertical field of view of $[-25°, +15°]$. The right panels compare snowy SemanticKITTI STF scans, including samples from the fine-tuned AnyLiDAR model. Rows show reference scans, R2Flow samples, and AnyLiDAR samples with ring modulation.

Circular MHSA and update placement. We follow the Circular MHSA computation of R2Flow (Nakashima et al., 2025), using adaptive RMS normalization conditioned on $\mathbf{E}$ and ray-angle rotary encoding. For a fixed sample and scale, let $\mathbf{q}_i^{(n)}, \mathbf{k}_i^{(n)} \in \mathbb{R}^{C_h}$ denote the query and key at token i in head n, where C_h is the head width. Its scores are

$$s_{ij}^{(n)} = \tau_n \left(\overline{\mathbf{q}}_i^{(n)}\right)^\top Q_n(\mathbf{g}_i)^\top Q_n(\mathbf{g}_j)\overline{\mathbf{k}}_j^{(n)}, \qquad \tau_n = \exp\big(\min(\lambda_n, \log 100)\big), \tag{12}$$

Here bars denote unit normalization, $Q_n(\mathbf{g}_i) \in \mathbb{R}^{C_h \times C_h}$ is the axial rotary transform for the ray angles at token i, and λ_n is a learned log-scale initialized to $\log 10$. The sample index on $\mathbf{g}_{b,i}$ is omitted. Integer harmonics are obtained by rounding logarithmically spaced frequencies, with maxima decreasing at coarser scales. The default circular neighborhood is 3×9. Dropout and an output projection produce $\mathbf{A}$ before the identity skip is added. The context projection W_o maps K_ℓ channels to $2C_\ell$, is bias-free and zero-initialized, and produces unconstrained scale and shift values. Equation 3 acts on the projected attention residual, leaving the identity skip unchanged. The following feed-forward residual is $\mathbf{H}_{\text{out}} = \mathbf{H}^+ + \text{FFN}(\mathbf{H}^+; \mathbf{E})$ and uses its own adaptive normalization. This interaction appears in the last neighborhood-attention block at each encoder and decoder level; the bottleneck uses full attention without this path. Merging concatenates 2×2 tokens and maps $4C_\ell$ channels to $2C_\ell$, while expansion uses a learned projection and channel-wise sigmoid interpolation with encoder skips.

A.5 Qualitative Comparisons

Generation across sensor domains. Figure 7 compares reference scans with samples from R2DM, Text2LiDAR, R2Flow, and AnyLiDAR across five sensor domains. All AnyLiDAR samples are generated by a single model jointly trained on seven datasets. Across the displayed examples, AnyLiDAR produces more coherent ground rings and surrounding structures than the compared baselines. The KITTI-360 and SemanticKITTI samples show more continuous ground surfaces and clearer roadside structures, while the Waymo sample avoids the pronounced curvature seen in R2Flow's ground bands. The nuScenes and Lyft samples also retain their respective sparse and dense ring patterns. These observations complement the quantitative results by showing that one shared

generator can accommodate different sensor layouts while maintaining coherent scene geometry across domains.

Ring modulation and weather adaptation. Figure 8 compares R2Flow with AnyLiDAR, which extends the same backbone only by adding ring context to modulate attention outputs. On PandaSet, with a vertical field of view of $[-25°, +15°]$, this addition yields more clearly separated ground rings and more continuous surrounding surfaces. To examine adaptation to adverse weather, we further fine-tune AnyLiDAR on snowy SemanticKITTI STF data. In the displayed snow examples, the fine-tuned model produces more discernible ground scan patterns and less fragmented structures near the sensor than R2Flow. The PandaSet comparison illustrates the effect of ring modulation, while the snow examples show the model's outputs after weather-specific fine-tuning.

A.6 Further Analysis

Figure 9 presents a t-distributed stochastic neighbor embedding (t-SNE) visualization of scans generated by AnyLiDAR for the seven target domains. KITTI-360 and SemanticKITTI occupy neighboring regions with partial overlap, while nuScenes and RELLIS-3D form more separated groups. Waymo and Lyft also overlap in part of the projection. These patterns suggest that generated samples retain variation associated with individual datasets while sharing some characteristics across domains. Such variation is consistent with modeling differences in measurement layout and observation statistics within a shared generator. The preceding point-cloud comparisons provide a complementary view through the spatial structure of individual scans. The visualization describes relationships among generated samples and does not measure their agreement with real scans. We use it to complement the distributional metrics, without interpreting projected distances as quantitative domain discrepancies.

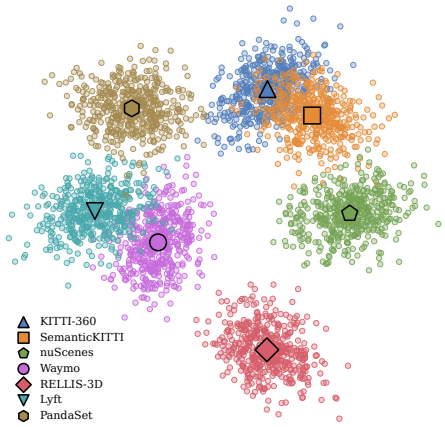


Figure 9: **t-SNE visualization of generated LiDAR scans.** Colors indicate the seven target datasets.