LiMamba: Rethinking Point Cloud Serialization in Mamba-Based Diffusion Models for LiDAR Scene Completion

Anonymous authors
Paper under double-blind review

Abstract

Diffusion-based methods now dominate LiDAR point cloud completion, yet every existing point-based scene-completion denoiser relies on sparse 3D convolutions whose receptive fields are inherently local, leaving long-range structural correlations largely uncaptured. Explicit global self-attention is uneconomical at outdoor scale, while state space models such as Mamba offer linear-complexity modeling but require flattening 3D voxels into a 1D sequence. Our central observation is that, while multi-directional scanning is a minor detail for object-level point clouds, at LiDAR *scene* scale the serialization order becomes a central design choice: the ordering governs how far the state space can propagate structure, and any single ordering is lossy. We propose **LiMamba**, a hybrid MinkowskiEngine–Mamba denoiser built around **Multi-Scan Mamba (MSM) blocks**. Each MSM block serializes voxel features along orderings *derived from spinning-LiDAR acquisition physics* (a radial range-density ordering and an azimuthal scan-line ordering, alongside Z-order), processes them through parallel Mamba streams, and fuses the outputs via learned gating. A **Scan Dropout** strategy prevents the streams from collapsing onto one during training, so that multi-scan improves accuracy rather than merely adding computation. Experiments on SemanticKITTI show that LiMamba improves over purely convolutional diffusion baselines on Chamfer Distance, JSD, and coarse-resolution occupancy IoU, while retaining linear-complexity, memory-feasible global modeling at a scale where explicit self-attention is not feasible, and ablations confirm the complementary value of multi-scan fusion over any single serialization.

1 Introduction

LiDAR point cloud completion is a fundamental task for autonomous driving and robotic perception. A single LiDAR sweep is inherently sparse, irregular, and incomplete: sampling density drops with range, occlusions remove large scene portions, and thin or distant structures are only partially observed. Diffusion-based generative models have recently become a leading approach for scene-scale completion, in which a learned denoiser iteratively recovers a plausible dense point cloud from Gaussian noise conditioned on the partial input Nunes et al. (2024); Martyniuk et al. (2025); Zhang et al. (2025). A successful denoiser must therefore simultaneously preserve fine-grained local geometric detail and infer globally consistent scene structure, which together place unusually strong and often competing demands on the backbone architecture.

To date, every competitive point-based LiDAR scene-completion diffusion denoiser has been built on MinkowskiEngine-style sparse 3D convolutions Choy et al. (2019); Nunes et al. (2024); Martyniuk et al. (2025); Zhang et al. (2025). While an alternative line of work runs diffusion in the 2D range-image domain with ordinary image convolutions Nakashima & Kurazume (2024); Ran et al. (2024), the point-based methods that dominate scene-scale completion all inherit the same sparse-convolution backbone. Sparse convolutions excel at encoding local geometry, yet their receptive field grows only through cascaded small neighborhoods and, even after several downsampling stages, remains modest relative to the span of a typical outdoor crop. Long-range structural correlations—the alignment of road boundaries, the regularity of building facades, the layout of parked vehicles—are therefore hard

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.

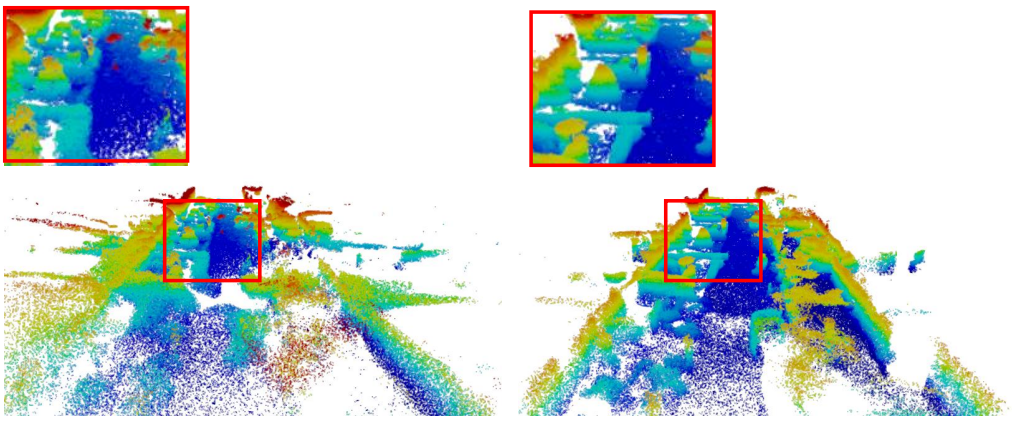


Figure 1: **Qualitative comparison on SemanticKITTI.** Given the same sparse LiDAR input, our LiMamba (left) recovers more complete and structurally coherent scenes than the LiDiff baseline (right), particularly for far-range structures such as building facades and road boundaries. This advantage stems from the proposed Multi-Scan Mamba block, which serializes voxel features along three complementary orderings and fuses them via learned gating, providing global context that sparse convolutions alone cannot capture.

to capture. This locality bottleneck is shared by LiDiff, LiDPM, and ScoreLiDAR alike and sets a concrete ceiling on completion quality.

A natural remedy is to introduce global attention. Yet no existing LiDAR scene completion method has successfully adopted a Transformer backbone: self-attention scales quadratically in the number of active voxels Vaswani et al. (2017), and a single outdoor crop routinely contains tens of thousands of sparse tokens. Even in a non-diffusion architecture, He et al. He et al. (2025) report that standard cross-attention exceeds a 24 GB GPU memory budget; in a diffusion denoiser the network must be evaluated at every denoising step, multiplying the cost further. The consistent absence of Transformers in the LiDAR completion literature reflects a fundamental tension between the scale of LiDAR data and the cost of pairwise interaction modeling.

State space models (SSMs), and Mamba Gu & Dao (2023) in particular, resolve the efficiency side of this tension: their selective gating mechanism propagates information along a sequence with linear complexity, making global reasoning over large token sets practical. Replacing attention with Mamba, however, introduces a new and under-explored challenge: serialization. Sparse voxels occupy an unordered 3D space; before Mamba can process them they must be flattened into a 1D sequence, and the choice of ordering directly shapes every state transition and, consequently, the learned representation.

Our key observation is that **any single mapping from 3D voxels to a 1D sequence is necessarily lossy**. The Z-order curve keeps some 3D-neighboring pairs adjacent but scatters others; radial ordering preserves depth proximity but disrupts angular neighbors; azimuth ordering maintains rotational continuity while breaking range relationships. A naive flattening therefore causes Mamba's selective state to accumulate irrelevant context. Multi-directional scanning is by now a common device for object-level point-cloud Mamba Liang et al. (2024); Zhang et al. (2024), where it is treated as an implementation choice; we argue that at *scene scale*, where token count and spatial extent are orders of magnitude larger, **serialization becomes a primary design factor** for state space models on sparse 3D LiDAR data.

Building on this insight, we propose **LiMamba**, a hybrid sparse-convolution and multi-scan Mamba denoiser for diffusion-based LiDAR point cloud completion. The backbone retains a MinkowskiEngine encoder–decoder Choy et al. (2019) for local geometric aggregation. At deeper stages where global context is most needed, we insert **Multi-Scan Mamba (MSM) blocks** that serialize voxel features along three complementary, physically motivated orderings (Z-order for 3D locality,

Radial for range-dependent density, and Azimuth for the rotational scan pattern), each processed by a dedicated Mamba stream. A lightweight learned gating module fuses the three streams, recovering information that any single stream would lose (Fig. 1). We further introduce **Scan Dropout**, which randomly drops one scan path during training to prevent co-adaptation. Because Mamba's cost is linear and MSM blocks are placed only at the deepest stages, running three parallel streams is still cheaper than full self-attention.

In summary, our contributions are:

- **Serialization matters at scene scale.** Unlike object-level point-cloud Mamba, where multi-directional scanning gives only marginal gains, at LiDAR *scene* scale the choice of serialization governs how far the state space propagates structure: three complementary orderings lift fine-resolution IoU by $\sim 15\%$ over the best single ordering (Table 5).
- **Sensor-physics serialization.** We derive the orderings from spinning-LiDAR acquisition physics—a radial (range-density) and an azimuthal (scan-line) ordering—and use them as the serialization prior of a state-space model, showing the three are complementary.
- **LiMamba**, to our knowledge the first point-based state-space denoiser for scene-scale LiDAR diffusion completion, improves over convolutional baselines on CD, JSD, and coarse-resolution IoU on SemanticKITTI and zero-shot KITTI-360, while staying memory-feasible where explicit global self-attention does not.

2 Related work

2.1 LiDAR scene completion

Early approaches frame completion as *depth completion* Ma & Karaman (2018); Jaritz et al. (2018) on a 2D image plane, restricted to the camera field of view. Voxel-based 3D methods (LMSCNet Roldao et al. (2020), LODE Li et al. (2023a), MID Vizzo et al. (2022)) predict occupancy on a grid but are bounded by voxel resolution; the broader semantic scene completion family adds per-voxel labels via point-cloud Xia et al. (2023) or camera-transformer Li et al. (2023b) backbones (surveyed in Roldao et al. (2022)). Point-based diffusion has recently advanced the field: LiDiff Nunes et al. (2024) scales point-level DDPM to outdoor scenes with a MinkUNet backbone and a local-noise reformulation (with a refinement network Lyu et al. (2022)); LiDPM Martyniuk et al. (2025) shows a vanilla global DDPM with a well-chosen starting point suffices; ScoreLiDAR Zhang et al. (2025), Distillation-DPO Zhao et al. (2025), and LiFlow Matteazzi & Tutsch (2026) accelerate sampling via distillation, preference optimization, and flow matching. Other approaches use alternative representations—range-image diffusion (R2DM Nakashima & Kurazume (2024), LiDM Ran et al. (2024)), categorical voxel-grid diffusion Lee et al. (2023), and joint geometric-semantic completion (DiffSSC Cao & Behnke (2024))—while the concurrent LiNeXt He et al. (2025) is a single-pass convolutional model without explicit long-range modeling. Across all of these, **every** competitive point-based method builds on MinkowskiEngine sparse convolutions Choy et al. (2019) without global sequence modeling; we fill this gap with Mamba state space blocks.

2.2 State space models for 3D point clouds

Mamba Gu & Dao (2023), building on structured state spaces Gu et al. (2022), is a linear-complexity alternative to Transformer attention Vaswani et al. (2017). In 2D vision, Vision Mamba Zhu et al. (2024) and VMamba Liu et al. (2024) show multi-directional scanning captures richer structure than a single raster order. PointMamba Liang et al. (2024) and Point Cloud Mamba Zhang et al. (2024) extend this to 3D object understanding on PointNet-style encoders Qi et al. (2017), but use a single ordering on small point sets ($\sim$1–2K points). Closest to us, CompletionMamba Fu et al. (2025) and 3DMambaComplete Li et al. (2026) apply SSMs to *object-level* completion with a single causal reordering, whereas we target *scene-scale* outdoor LiDAR ($>$100K sparse voxels), where a single ordering incurs substantial information loss. A parallel line serializes points along space-filling curves for tractable attention, notably Point Transformer V3 Wu et al. (2024); we instead let a linear-complexity SSM propagate global context along several sensor-aware orderings in parallel, avoiding the memory cost that has kept transformer backbones out of diffusion-based LiDAR completion (Sec. 1). To our knowledge, LiMamba is the first point-based diffusion denoiser with a state-space

backbone at LiDAR scene scale, and the first to treat the serialization order as an explicit design variable rather than a fixed preprocessing step.

2.3 Diffusion denoiser architectures

DDPMs Ho et al. (2020) were originally built on U-Net backbones Ronneberger et al. (2015); for 3D shapes they extend to point-level diffusion (Luo and Hu Luo & Hu (2021), PVD Zhou et al. (2021), conditional refinement Lyu et al. (2022)) and transformer variants (DiT-3D Mo et al. (2023)), and faster sampling comes from DDIM Song et al. (2021) and DPM-Solver Lu et al. (2022), both used at inference. At scene scale, however, all existing denoisers rely on sparse convolutions, leaving long-range modeling unaddressed—the gap we fill with Mamba blocks and multi-scan serialization inside the denoiser.

3 Methodology

3.1 Problem formulation and diffusion preliminaries

Given a sparse LiDAR scan $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\} \subset \mathbb{R}^3$ from a single sensor revolution, we recover a dense, complete point cloud $\mathcal{G} = \{\mathbf{g}_1, \dots, \mathbf{g}_M\} \subset \mathbb{R}^3$ ($M \gg N$). Following Ho et al. Ho et al. (2020), a forward process corrupts the ground-truth cloud $\mathbf{x}_0 = \mathcal{G}$ over T timesteps, and a denoiser ϵ_θ is trained to predict the added noise:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \quad \mathcal{L}(\theta) = \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon},t}\big[\|\boldsymbol{\epsilon} - \epsilon_\theta(\mathbf{x}_t, \mathcal{P}, t)\|^2\big], \tag{1}$$

with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$, and $\{\beta_t\}$ a linear schedule from $\beta_1 = 3.5\times10^{-5}$ to $\beta_T = 0.007$. Following LiDPM Martyniuk et al. (2025), the reverse process does not start from pure noise but from the K-times-duplicated scan $\tilde{\mathcal{P}} \in \mathbb{R}^{M\times 3}$ noised to level t_0, $\tilde{\mathbf{x}}_{t_0} = \sqrt{\bar{\alpha}_{t_0}}\,\tilde{\mathcal{P}} + \sqrt{1-\bar{\alpha}_{t_0}}\,\mathbf{z}$. Conditioning on $\mathcal{P}$ uses classifier-free guidance Ho & Salimans (2022): $\mathcal{P}$ is dropped with probability p_{uncond} during training, and the inference estimate is $\tilde{\epsilon}_\theta = (1-\gamma)\,\epsilon_\theta(\mathbf{x}_t, t, \mathbf{0}) + \gamma\,\epsilon_\theta(\mathbf{x}_t, t, \mathcal{P})$ with guidance weight γ.

3.2 Overall architecture

The denoiser ϵ_θ (Fig. 2, top) is a symmetric U-Net Ronneberger et al. (2015) over sparse voxels using MinkowskiEngine convolutions Choy et al. (2019): a stem, four strided-downsampling encoder stages, and four transposed-convolution decoder stages with skip connections. The first two (high-resolution) stages carry many voxels where local detail dominates, so plain MinkConv blocks suffice; the deeper stages ($4\times$/$8\times$ downsampled) have far fewer voxels and are where global context matters, so we append a Multi-Scan Mamba (MSM) block (Sec. 3.4) after the MinkConv residual blocks of $\mathcal{S}_{\text{mamba}} = \{\texttt{enc3}, \texttt{enc4}, \texttt{up1}, \texttt{up2}\}$. This placement confines the Mamba overhead to the sparsest stages while injecting global reasoning where it helps most, and it keeps the added parameter count and memory footprint small relative to the convolutional backbone. Conditioning follows LiDPM Martyniuk et al. (2025) exactly, so that gains come from the backbone rather than auxiliary changes: a separate MinkConv encoder maps $\mathcal{P}$ (or $\mathbf{0}$ unconditionally) to a feature map, and each denoiser voxel retrieves its nearest-neighbor feature (k-NN, $k=1$), which—together with a sinusoidal timestep embedding Ho et al. (2020)—forms a per-voxel multiplicative modulation of the stage input.

3.3 LiDAR-aware serialization strategies

Applying Mamba Gu & Dao (2023) to sparse voxels requires flattening the unordered set into a 1D sequence, and (Sec. 1) any single ordering is lossy: whereas space-filling curves are used only for *local* pooling in prior work He et al. (2025), in *global* Mamba modeling the ordering governs how the selective state propagates dependencies across the whole scene. We use three orderings, each capturing a distinct property of LiDAR acquisition. *Z-order* interleaves the bits of the integer voxel coordinates $(x, y, z) \in \mathbb{Z}^3$ into one key, preserving general 3D locality:

$$k_z = \mathrm{spread}(x) \;\big|\; \big(\mathrm{spread}(y) \ll 1\big) \;\big|\; \big(\mathrm{spread}(z) \ll 2\big), \tag{2}$$

where $\mathrm{spread}(\cdot)$ places 21 bits into every third bit position. *Radial* ordering sorts by distance from the sensor, capturing the range-dependent density gradient (near regions dense, far regions sparse), so

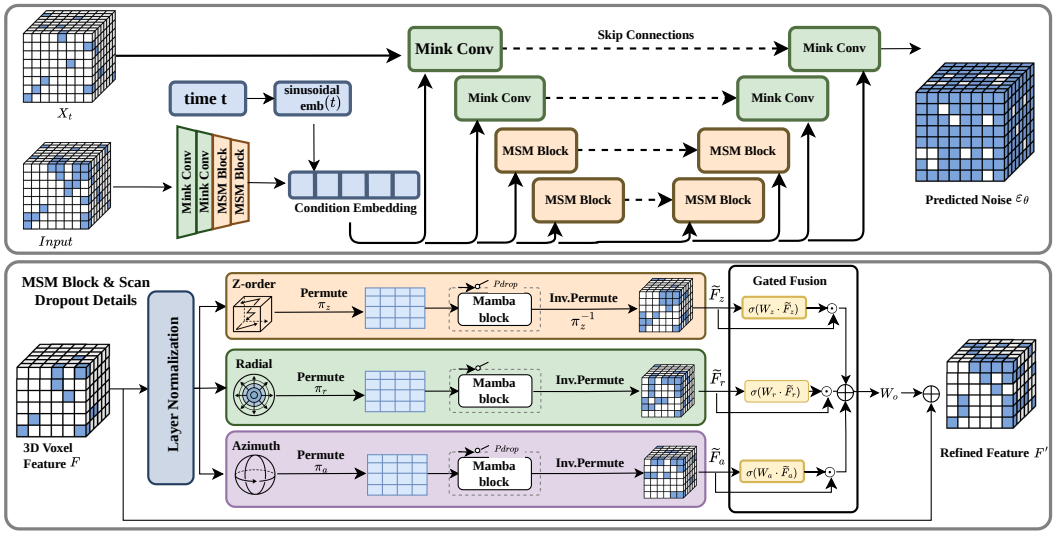


Figure 2: **Overview of LiMamba.** *Top:* the denoiser ϵ_θ follows a symmetric UNet built on MinkowskiEngine sparse convolutions. The noisy point cloud $\mathbf{x}_t$ is voxelized and passed through four encoder stages (downsampling) and four decoder stages (upsampling) connected by skip connections. Stages 1–2 use pure MinkConv blocks, while stages 3–4 (encoder) and stages 1–2 (decoder) each append a Multi-Scan Mamba (MSM) block for global context. A separate condition encoder maps the partial scan $\mathcal{P}$ to a feature embedding that is injected at every stage together with a sinusoidal timestep embedding of t. *Bottom:* inside each MSM block, voxel features $\mathbf{F}$ are layer-normalized, then serialized along three complementary orderings (Z-order, Radial, Azimuth) via permutations π_z, π_r, π_a. Each serialized sequence is processed by a dedicated Mamba block and inverse-permuted back. During training, Scan Dropout randomly disables individual paths (indicated by p_{drop}). The three outputs are combined by a learned gated fusion and added back as a residual to produce the refined feature $\mathbf{F}'$.

adjacent tokens share density and context:

$$k_r = x^2 + y^2 + z^2. \tag{3}$$

Azimuth ordering sorts by horizontal angle then elevation, mirroring the rotational scan pattern of mechanical LiDAR so that adjacent tokens are adjacent scan lines:

$$k_a = \frac{\operatorname{atan2}(y, x) + \pi}{2\pi} \cdot 10^3 + \frac{\operatorname{atan2}(z, \sqrt{x^2 + y^2}) + \pi/2}{\pi}. \tag{4}$$

These three priors—3D locality, density gradient, and scan pattern—fix the number of streams: each is distinct and non-redundant, so we neither drop one nor add a fourth (which would duplicate a prior or lack any physical correspondence). Our subset ablation confirms this, with accuracy rising monotonically from one to three orderings (Appendix B, Table 5).

3.4 Multi-scan Mamba block

The MSM block (Fig. 2, bottom) runs all three orderings in parallel rather than committing to one. Given voxel features $\mathbf{F} \in \mathbb{R}^{N_v \times d}$ and $\hat{\mathbf{F}} = \mathrm{LN}(\mathbf{F})$, each strategy $s \in \{z, r, a\}$ serializes by its sort permutation π_s, applies a dedicated Mamba module, and inverse-permutes back, $\tilde{\mathbf{F}}_s = \mathrm{Mamba}_s(\hat{\mathbf{F}}[\pi_s])[\pi_s^{-1}]$; the outputs are combined by per-channel sigmoid gating as a residual:

$$\mathbf{F}' = \mathbf{F} + W_o \sum_{s \in \{z,r,a\}} \sigma\big(W_s \tilde{\mathbf{F}}_s\big) \odot \tilde{\mathbf{F}}_s, \tag{5}$$

with learned $W_s, W_o \in \mathbb{R}^{d \times d}$. Three streams cost $3\,O(N_v)$, still far below self-attention's $O(N_v^2)$; alternative fusions (weighted sum, MLP) are ablated in Appendix B.

Table 1: **Quantitative comparison on SemanticKITTI validation set.** †: diffusion-only (no refinement). Following LiDPM, the refined LiMamba rows use the off-the-shelf LiDiff refinement network without any re-training. Best in **bold**, second best underlined.

Method	CD↓	JSD↓	JSD↓	Voxel IoU↑		
		3D	BEV	0.5 m	0.2 m	0.1 m
LMSCNet Roldao et al. (2020)	0.641	–	0.431	30.8	12.1	3.7
LODE Li et al. (2023a)	1.029	–	0.451	33.8	16.4	5.0
MID Vizzo et al. (2022)	0.503	–	0.470	31.6	22.7	13.1
PVD Zhou et al. (2021)	1.256	–	0.498	15.9	4.0	0.6
LiDiff† Nunes et al. (2024)	0.434	0.564	0.444	31.5	16.8	4.7
LiDPM† Martyniuk et al. (2025)	0.446	0.532	0.440	34.1	19.5	6.3
LiDiff Nunes et al. (2024)	0.376	0.573	0.416	32.4	23.0	13.4
LiDPM Martyniuk et al. (2025)	0.377	0.542	0.403	36.6	**25.8**	**14.9**
LiMamba† (ours)	0.435	0.528	0.430	35.4	20.3	8.5
LiMamba (ours)	**0.368**	**0.519**	**0.395**	**38.0**	23.8	11.3

3.5 Scan dropout

Inspired by stochastic depth Huang et al. (2016), **Scan Dropout** drops each scan path independently with probability p_{drop} during training (at least one path always survives; all are active at inference), preventing the gate from learning a fixed weighting and forcing each stream to stay independently useful. Without it, the gate quickly collapses onto the single strongest ordering and the multi-scan design effectively degenerates to a single scan (Sec. 4.4), so the three complementary streams contribute jointly and remain mutually reinforcing only when this regularizer is enabled.

4 Experiments

4.1 Setup

Datasets. We follow the standard protocol of LiDiff Nunes et al. (2024). SemanticKITTI Behley et al. (2019) provides 22 sequences of Velodyne HDL-64E scans; ground-truth dense clouds are constructed by aggregating consecutive scans via ego-poses after removing dynamic objects. We train on sequences 00–07 and 09–10 and evaluate on validation sequence 08—the only split with reliable-ego-pose dense ground truth— matching the LiDiff/LiDPM protocol for a fair comparison. For cross-dataset generalization, we additionally report zero-shot results on KITTI-360 Liao et al. (2022) (same sensor, different routes) without any fine-tuning.

Metrics. Following LiDiff/LiDPM Martyniuk et al. (2025), we report Chamfer Distance (CD, ↓), Jensen–Shannon Divergence in 3D and BEV (JSD, ↓), and Voxel Occupancy IoU at 0.5/0.2/0.1 m (↑).

Implementation. We follow the LiDiff/LiDPM training and inference protocol; full architecture, diffusion schedule, and optimization hyperparameters are given in Appendix A. Unless otherwise noted, results are reported both with and without the off-the-shelf LiDiff refinement network.

4.2 Main results

Table 1 compares LiMamba with prior methods on the SemanticKITTI validation set, including voxel/surface-based baselines (LMSCNet, LODE, MID), the point-voxel diffusion baseline PVD, and the state-of-the-art point-based diffusion methods LiDiff and LiDPM. All metrics and ground-truth are computed following the evaluation protocol of LiDiff Nunes et al. (2024).

Diffusion-only comparison. Without refinement, LiMamba† achieves the second-best JSD 3D (0.528), trailing only its refined counterpart, while improving over LiDPM† (0.532) and LiDiff† (0.564). IoU improves over both diffusion baselines at all three resolutions (35.4 / 20.3 / 8.5 vs. LiDPM†'s 34.1 / 19.5 / 6.3 and LiDiff†'s 31.5 / 16.8 / 4.7), with the largest relative gain at the finest

Table 2: **Cross-dataset generalization on KITTI-360.** All models are trained on SemanticKITTI and evaluated zero-shot on KITTI-360. †: diffusion-only (no refinement).

Method	CD↓	JSD↓	Voxel IoU↑		
			0.5 m	0.2 m	0.1 m
LMSCNet Roldao et al. (2020)	0.979	0.496	25.5	16.4	3.0
LODE Li et al. (2023a)	1.565	0.483	42.1	**42.6**	5.9
MID Vizzo et al. (2022)	0.637	0.476	**44.1**	36.4	15.8
LiDiff† Nunes et al. (2024)	0.564	0.459	42.2	32.3	10.8
LiDiff Nunes et al. (2024)	<u>0.517</u>	0.446	40.8	36.1	<u>21.3</u>
Ours†	0.541	<u>0.441</u>	<u>43.5</u>	33.9	11.6
Ours	**0.496**	**0.429**	42.1	<u>37.9</u>	**22.8**

voxel size (0.1 m). CD (0.435) is on par with LiDiff† (0.434), which confirms that the MSM backbone adds global context without sacrificing point-level precision, and that the improvement in occupancy IoU does not come at the expense of geometric accuracy.

Full pipeline. Following LiDPM Martyniuk et al. (2025), we apply the *off-the-shelf* LiDiff refinement network without re-training, so the comparison isolates the denoiser backbone. With refinement, LiMamba achieves the best CD (0.368), JSD 3D (0.519), JSD BEV (0.395), and IoU at 0.5 m (38.0); LiDPM retains the lead only at the finest voxel sizes (IoU 0.2 m: 25.8 vs. 23.8; 0.1 m: 14.9 vs. 11.3). This fine-resolution gap appears *only after* the shared refinement is applied: the diffusion-only comparison above shows LiMamba† *improving* IoU over both baselines at every resolution, including 0.1 m (8.5 vs. 6.3 and 4.7). We attribute the gap to using the refinement head off the shelf, un-adapted to our point distribution, and leave re-training it to future work. Overall, the gains in CD, JSD, and coarse-resolution IoU indicate that the MSM backbone yields structurally coherent completions even without a co-adapted refinement stage.

Cross-dataset generalization. Table 2 reports zero-shot evaluation on KITTI-360 (trained on SemanticKITTI, no fine-tuning), testing whether MSM's global context transfers to unseen environments. On this unseen dataset, LiMamba achieves the best CD (0.496) and JSD (0.429) overall, improving over LiDiff by 4.1% and 3.8% respectively, and attains the best IoU at 0.1 m (22.8) among all methods. The voxel-based MID and LODE retain a small edge at coarser resolutions (0.5 m and 0.2 m) because they directly predict on a regular grid, but their CD is 28–215% worse than ours, indicating that their voxelized completions lack the fine geometric fidelity that point-based methods deliver. This cross-dataset transfer corroborates that the global reasoning provided by MSM is not overfit to a particular sequence layout and continues to help on routes not seen during training.

Qualitative results. Fig. 3 presents qualitative comparisons on two representative scenes drawn from the SemanticKITTI validation set. For each scene, we visualize the sparse input, the aggregated ground truth, and the completions of LiDPM and LiMamba, all rendered as bird's-eye-view (BEV) point clouds colored by height. LiMamba reconstructs sharper road boundaries and more complete building facades, particularly at far range where sparse convolutions lose coverage; it also recovers thin structures such as poles and parked vehicles that LiDPM tends to under-fill or over-smooth. These observations are consistent with the quantitative gains in Tables 1 and 2, and with the range-stratified analysis in Appendix D (Table 11), which shows the improvement over LiDPM growing monotonically with distance.

4.3 Efficiency analysis

Because our central claim is that Mamba is a *practical* alternative to attention for scene-scale completion, we profile the backbone directly under a single fixed setting so that the numbers are directly comparable across methods. Table 3 reports parameter count, per-step FLOPs, peak inference memory, per-step latency, and full DDIM sampling time ($S = 50$), all measured on a single RTX 4090 under an identical 50 m crop and $r = 0.05$ m voxelization. We further compare against an explicit self-attention backbone at the same stages, which runs out of memory at this scale, and (in Appendix C,

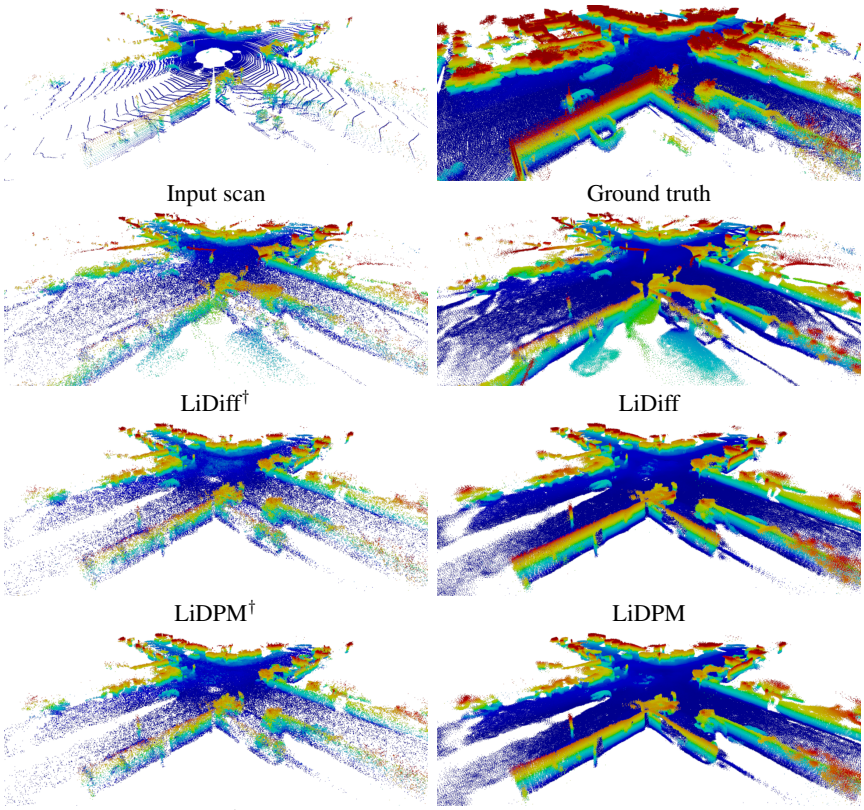


Figure 3: **Qualitative scene completion on SemanticKITTI.** The top row shows the sparse input scan and the aggregated ground truth for reference; each subsequent row pairs a method's diffusion-only output (†, left) with its refined output (right). Points are colored by height under a shared colormap. Compared to LiDiff and LiDPM, our LiMamba recovers sharper road boundaries, more complete building facades, and thinner structures (poles, parked vehicles), especially at far range where local sparse convolutions lose coverage; the gap is visible both with and without refinement.

Table 3: **Efficiency comparison.** Single RTX 4090, identical 50 m crop and voxelization (180K points, batch 1). ‡counted/estimated architecture FLOPs. "Attn backbone" replaces the MSM blocks with explicit self-attention; § it OOMs before a full forward, so its cells are blank. See text and Appendix C for details.

Method	Params (M)	FLOPs/step (G)‡	Peak mem (GB)	Latency/step (ms)	Sampling (s)
LiDiff Nunes et al. (2024)	32.7	6263	5.95	232	17.5
LiDPM Martyniuk et al. (2025)	32.7	6263	6.02	297	26.4
Attn backbone§	–	–	OOM	–	–
LiMamba (ours)	37.8	6979	**2.69**	344	29.2

Table 10) against a tractable windowed-attention variant. In summary, LiMamba adds global context at a bounded parameter cost (+15.7%) and a moderate latency overhead (297→344 ms/step) while remaining memory-feasible, whereas explicit global attention does not run under a commodity budget; the low peak memory is an implementation property, not an intrinsic advantage of state-space blocks (see Appendix C).

4.4 Ablation studies

Ablations use a reduced SemanticKITTI Behley et al. (2019) validation subset (100 equally spaced scans from sequence 08, following LiDPM Martyniuk et al. (2025)), without the refinement network.

Table 4: **Component ablation.** Each of our three components contributes: the Mamba backbone (vs. MinkConv-only), multi-scan serialization (vs. the best single ordering, azimuth), and Scan Dropout (vs. $p_{\text{drop}} = 0$). Reduced SemanticKITTI validation subset, diffusion-only.

Backbone	CD↓	JSD 3D↓	JSD BEV↓	IoU 0.5↑	IoU 0.2↑	IoU 0.1↑
MinkConv-only (no Mamba)	0.450	0.537	0.444	33.8	18.5	6.0
Single-scan Mamba (azimuth)	0.441	0.530	0.440	34.1	19.2	7.4
Multi-scan, no Scan Dropout	0.440	0.531	0.435	34.6	19.6	7.8
Multi-Scan Mamba (full)	**0.435**	**0.528**	**0.430**	**35.4**	**20.3**	**8.5**

Table 4 isolates the contribution of each of our three components on a common footing; all variants share the same stage structure, condition injection, and training schedule, and are matched in parameters and per-step FLOPs to within 2%, so the trend reflects the design rather than added capacity. Removing the state-space module (*MinkConv-only*) is worst on every metric; adding a single Mamba stream already helps, and the full *Multi-Scan Mamba* improves further, with the largest relative gain at the finest voxel size (0.1 m IoU $6.0 \rightarrow 8.5$, $+42\%$ over the no-Mamba baseline). Disabling *Scan Dropout* ($p_{\text{drop}} = 0$) lets the gate collapse onto the single strongest (azimuth) stream during training and erases most of the multi-scan benefit, confirming that all three components—the Mamba backbone, the multi-scan serialization, and Scan Dropout—are necessary. Full sweeps over the serialization subsets, the dropout probability, the fusion mechanism, and the MSM stage placement are reported in Appendix B (Tables 5–8).

5 Conclusion

We presented LiMamba, a hybrid sparse-convolution and Mamba denoiser for diffusion-based LiDAR point cloud completion. Starting from the observation that the serialization order matters once state space models operate at scene scale, we introduced *Multi-Scan Mamba* (MSM) blocks that process voxel features along three orderings derived from spinning-LiDAR physics (Z-order, radial, and azimuth) in parallel and fuse them via learned gating, with Scan Dropout keeping the parallel streams complementary rather than collapsing onto one. On SemanticKITTI, LiMamba attains the best Chamfer Distance, JSD 3D, JSD BEV, and IoU at 0.5 m among all compared methods; on the zero-shot KITTI-360 transfer, it again achieves the best CD, JSD, and IoU at 0.1 m, confirming that the global context provided by MSM transfers robustly across datasets without any target-domain fine-tuning. More broadly, our results suggest that state space models, when paired with sensor-aware multi-scan serialization, are a practical and memory-efficient alternative to attention for scene-scale 3D point cloud generation, trading a moderate increase in runtime for global context that self-attention cannot provide within a commodity memory budget.

Limitations and future work. Our study has three main limitations. *(i) Sensor geometry.* The radial and azimuth orderings are motivated by mechanical spinning LiDAR (e.g., Velodyne HDL-64E); on solid-state or partial-FOV sensors their benefit may diminish, and the serialization keys would need to be adapted to the corresponding acquisition geometry—a natural next step. *(ii) Refinement coupling.* We reuse the off-the-shelf LiDiff refinement network, tuned for the LiDiff/LiDPM point distribution rather than ours; as a result the final refined system trails LiDPM on IoU at 0.2 m and 0.1 m, even though the diffusion-only results (Table 1) show the MSM backbone improving fine-resolution IoU over both baselines before refinement. A refinement head jointly trained on our output distribution is a promising but unverified direction we leave to future work. *(iii) Memory attribution and cost.* LiMamba's low peak memory stems from its implementation and the placement of MSM blocks at the sparsest stages, not from any intrinsic advantage of state-space blocks; the three parallel streams do increase memory over a single stream, which we bound by confining them to the deepest stages (Sec. 3.2). Sampling is not accelerated, so real-time latency trails distillation-based methods. Finally, like all generative completion methods, LiMamba can hallucinate plausible-but-incorrect geometry; in safety-critical settings this motivates pairing completions with uncertainty estimates, which we leave to future work.

AI USE STATEMENT

In this work, we used generative AI tools solely as general-purpose assistants for language editing (grammar, spelling, and phrasing of author-written text) and for minor, non-substantive coding assistance (e.g., boilerplate for plotting and formatting of result tables). We did not use generative AI tools to generate research ideas, to design the method, to produce or analyze the experimental results, or to author the scientific claims of the paper. All AI-assisted text and code were reviewed and verified by the authors, and we take full responsibility for the final content of this work, including all text, claims, and artifacts.

ETHICS STATEMENT

This work studies LiDAR point cloud scene completion and does not involve human subjects, personally identifiable information, or the collection of new data. All experiments use publicly available driving datasets (SemanticKITTI (Behley et al., 2019) and KITTI-360 (Liao et al., 2022)) under their respective licenses and standard evaluation protocols; any privacy-sensitive content (e.g., faces or license plates) is handled by the original dataset providers. The method is intended to improve the perception stack of autonomous systems; as with any perception component, deployment in safety-critical settings should be accompanied by appropriate validation. We are not aware of any additional ethical concerns raised by this research.

REPRODUCIBILITY STATEMENT

We have made an effort to ensure that our results are reproducible. The full architecture of LiMamba, including the Multi-Scan Mamba (MSM) block, the three serialization orderings, the gated fusion, and Scan Dropout, is described in Section 3.2 and illustrated in Figure 2. The datasets, train/validation splits, evaluation metrics, and the complete set of implementation and training hyperparameters (network channels, diffusion schedule, classifier-free guidance, DDIM sampling settings, optimizer, and hardware) are given in Section 4.1 and Appendix A, following the public protocol of LiDiff (Nunes et al., 2024) and LiDPM (Martyniuk et al., 2025). Additional ablations, training-memory measurements, and the range-stratified analysis are provided in Appendices B–D. All datasets used are publicly available, and we will release anonymized source code and configuration files to reproduce the reported results.

REFERENCES

Johann Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Helin Cao and Sven Behnke. Diffssc: Semantic lidar scan completion using denoising diffusion probabilistic models. *arXiv preprint arXiv:2409.18092*, 2024.

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*, 2019.

Zhiheng Fu, Jiehua Zhang, Longguang Wang, Lian Xu, Hamid Laga, Yulan Guo, Farid Boussaid, and Mohammed Bennamoun. Completionmamba: Taming state space model for point cloud completion. *IEEE Transactions on Image Processing*, 34:5473–5485, 2025.

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

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2022.

Wenzhe He, Xiaojun Chen, Ruiqi Wang, Ruihui Li, Huilong Pi, Jiapeng Zhang, Zhuo Tang, and Kenli Li. Linext: Revisiting lidar completion with efficient non-diffusion architectures. *arXiv preprint arXiv:2511.10209*, 2025.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision*, 2016.

Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *2018 International Conference on 3D Vision (3DV)*, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Jumin Lee, Woobin Im, Sebin Lee, and Sung-Eui Yoon. Diffusion probabilistic models for scene-scale 3d categorical data. *arXiv preprint arXiv:2301.00527*, 2023.

Pengfei Li, Ruowen Zhao, Yongliang Shi, Hao Zhao, Jirui Yuan, Guyue Zhou, and Ya-Qin Zhang. Lode: Locally conditioned eikonal implicit scene completion from sparse lidar. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023a.

Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M. Alvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023b.

Yixuan Li, Weidong Yang, Lipeng Ma, and Ben Fei. 3dmambacomplete: Structured state space model for high-efficiency point cloud completion. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2026.

Dingkang Liang, Xin Zhou, Xinyu Wang, Xingkui Zhu, Wei Xu, Zhikang Zheng, Yongshuai Yan, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. *arXiv preprint arXiv:2402.10739*, 2024.

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

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024.

Zhijian Liu, Xinyu Yang, Haotian Tang, Shang Yang, and Song Han. Flatformer: Flattened window attention for efficient point cloud transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, 2022.

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*, 2021.

Zhaoyang Lyu, Zhifeng Kong, Xudong Xu, Liang Pan, and Dahua Lin. A conditional point diffusion-refinement paradigm for 3d point cloud completion. In *International Conference on Learning Representations*, 2022.

Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

Tetiana Martyniuk, Gilles Puy, Alexandre Boulch, Renaud Marlet, and Raoul de Charette. Lidpm: Rethinking point diffusion for lidar scene completion. In *2025 IEEE Intelligent Vehicles Symposium (IV)*, pp. 555–560, 2025. doi: 10.1109/IV64158.2025.11097538.

Andrea Matteazzi and Dietmar Tutsch. Liflow: Flow matching for 3d lidar scene completion. *arXiv preprint arXiv:2602.02232*, 2026.

Shentong Mo, Enze Xie, Ruihang Chu, Lanqing Hong, Matthias Niessner, and Zhenguo Li. Dit-3d: Exploring plain diffusion transformers for 3d shape generation. In *Advances in Neural Information Processing Systems*, 2023.

Kazuto Nakashima and Ryo Kurazume. Lidar data synthesis with denoising diffusion probabilistic models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

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*, 2024.

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*, 2017.

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*, 2024.

Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. Lmscnet: Lightweight multiscale 3d semantic completion. In *2020 International Conference on 3D Vision (3DV)*, 2020.

Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. 3d semantic scene completion: A survey. *International Journal of Computer Vision*, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Ignacio Vizzo, Benedikt Mersch, Rodrigo Marcuzzi, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. Make it dense: Self-supervised geometric scan completion of sparse 3d lidar scans in large outdoor environments. *IEEE Robotics and Automation Letters*, 2022.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Zhaoyang Xia, Youquan Liu, Xin Li, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, and Yu Qiao. Scpnet: Semantic scene completion on point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Shengyuan Zhang, An Zhao, Ling Yang, Zejian Li, Chenye Meng, Haoran Xu, Tianrun Chen, AnYang Wei, Perry Pengyun Gu, and Lingyun Sun. Distilling diffusion models to efficient 3d lidar scene completion. *arXiv preprint arXiv:2412.03515*, 2025.

Tao Zhang, Haobo Luo, Jiaming Lu, and Tao He. Point cloud mamba: Point cloud learning via state space model. *arXiv preprint arXiv:2403.00762*, 2024.

An Zhao, Shengyuan Zhang, Ling Yang, Zejian Li, Jiale Wu, Haoran Xu, AnYang Wei, Perry Pengyun Gu, and Lingyun Sun. Diffusion distillation with direct preference optimization for efficient 3d lidar scene completion. *arXiv preprint arXiv:2504.11447*, 2025.

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*, 2021.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.

A IMPLEMENTATION DETAILS

Both input and ground truth are cropped to 50 m radius; the input is FPS-subsampled to 18K points and duplicated $K = 10$ times ($\sim$180K points), then voxelized at $r = 0.05$ m. The UNet channels are $[32, 32, 64, 128, 256, 256, 128, 96, 96]$. MSM blocks use $d_c = 4$ throughout; the encoder stages (`enc3`,`enc4`) use $d_s = 16$ and expansion 2, while the decoder stages (`up1`,`up2`) use the lighter $d_s = 8$ and expansion 1, since the up-sampled features are already skip-fused with the encoder context. Diffusion uses $T = 1000$ steps, linear schedule $\beta_1 = 3.5 \times 10^{-5}$ to $\beta_T = 0.007$, CFG ($p_{\text{uncond}} = 0.1$, $\gamma = 6$), and DDIM Song et al. (2021) ($S = 50$, $\eta = 0$) at inference. Training: 50 epochs, Adam Kingma & Ba (2015) (lr $= 10^{-4}$), batch 1 per GPU on 2×A100. Scan Dropout $p_{\text{drop}} = 0.1$. The two GPUs are used only to raise the effective batch size via data parallelism, not because a single GPU is insufficient: with local batch 1 the peak per-GPU training memory is 20.2 GB (Sec. 4.3), which fits on a single 24 GB card, and all efficiency measurements in Table 3 are single-GPU. We optionally apply the off-the-shelf LiDiff refinement network; results are reported both with and without refinement.

B ADDITIONAL ABLATION STUDIES

All ablations in this section are run on the same reduced validation subset used in the main paper (100 equally spaced scans from SemanticKITTI sequence 08, following LiDPM Martyniuk et al. (2025)), without the refinement network. The default configuration matches the main paper: three orderings (Z-order, Radial, Azimuth), gated fusion, $p_{\text{drop}} = 0.1$, and MSM blocks placed at $\{\texttt{enc3}, \texttt{enc4}, \texttt{up1}, \texttt{up2}\}$. These expand on the component ablation in the main paper (Sec. 4.4, Table 4) with the full sweeps: the serialization subsets (Table 5), the Scan Dropout probability (Table 6), the fusion mechanism (Table 7), and the MSM stage placement (Table 8).

(A2) Number and choice of orderings. Holding everything else fixed, Table 5 varies the number and choice of serializations. Accuracy increases monotonically from one to two to three orderings on every metric, with the three-way combination best throughout—the core evidence for the multi-scan design. The single-ordering rows follow the ranking predicted by sensor physics (azimuth $>$ radial $>$ Z-order), and the two-ordering subsets recover most but not all of the gap, confirming that the orderings encode complementary, non-redundant priors and that each contributes.

Table 5: **(A2) Serialization subsets.** Each ordering encodes a distinct sensor-physics prior; accuracy increases monotonically from one to three orderings, and all three together are best on every metric.

Strategies	CD↓	JSD 3D↓	JSD BEV↓	IoU 0.5↑	IoU 0.2↑	IoU 0.1↑
Z-order only	0.444	0.532	0.441	33.5	18.7	7.0
Radial only	0.443	0.531	0.441	33.7	19.0	7.2
Azimuth only	0.441	0.530	0.440	34.1	19.2	7.4
Z-order + Radial	0.439	0.529	0.435	34.5	19.6	7.7
Z-order + Azimuth	0.438	0.529	0.434	34.7	19.8	7.9
Radial + Azimuth	0.437	0.528	0.432	34.9	20.0	8.1
All three (default)	**0.435**	**0.528**	**0.430**	**35.4**	**20.3**	**8.5**

(A3) Scan Dropout probability. Table 6 sweeps the Scan Dropout probability. Disabling it ($p_{\text{drop}} = 0$) lets the gate collapse onto the single strongest (azimuth) stream during training, erasing most of the multi-scan benefit; $p_{\text{drop}} = 0.1$ is the best trade-off, while $p_{\text{drop}} \geq 0.2$ removes too much signal.

(A4) Fusion mechanism. We compare the per-voxel sigmoid gating used in the main paper against two alternatives: a learned scalar weighted sum (one weight per stream, shared across voxels) and a small two-layer MLP that takes the concatenated stream outputs as input. Table 7 shows that gating outperforms both, while adding only a negligible parameter overhead per MSM block.

(A5) MSM stage placement. We study which UNet stages benefit most from MSM blocks by varying $\mathcal{S}_{\text{mamba}}$. Inserting MSM only into the encoder ($\{\texttt{enc3}, \texttt{enc4}\}$) or only into the decoder

Table 6: **(A3) Scan Dropout probability.** $p_{\text{drop}} = 0.1$ is best. With $p_{\text{drop}} = 0$ the gate collapses onto a single stream and most of the multi-scan gain is lost.

p_{drop}	CD↓	JSD 3D↓	JSD BEV↓	IoU 0.5↑	IoU 0.2↑	IoU 0.1↑
0.00	0.440	0.531	0.435	34.6	19.6	7.8
0.05	0.437	0.529	0.432	35.0	20.0	8.1
0.10 (default)	**0.435**	**0.528**	**0.430**	**35.4**	**20.3**	**8.5**
0.20	0.438	0.530	0.433	35.0	20.0	8.2
0.30	0.443	0.532	0.436	34.4	19.3	7.5

Table 7: **(A4) Fusion mechanism.** Per-voxel sigmoid gating gives the best trade-off between expressiveness and parameter cost.

Fusion	Extra params	CD↓	JSD 3D↓	JSD BEV↓	IoU 0.5↑	IoU 0.2↑	IoU 0.1↑
Weighted sum	3	0.440	0.531	0.434	34.6	19.7	7.8
MLP	$\sim d^2$	0.437	0.529	0.432	35.0	20.0	8.2
Gated (ours)	$\sim d^2$	**0.435**	**0.528**	**0.430**	**35.4**	**20.3**	**8.5**

($\{\texttt{up1}, \texttt{up2}\}$) recovers part of the gain; inserting at all eight stages slightly degrades CD because the high-resolution stages have many active voxels and benefit more from local convolutions. The default placement at the four deepest stages of the encoder/decoder gives the best trade-off (Table 8).

Table 8: **(A5) MSM stage placement.** Inserting MSM only at the deepest encoder and shallowest decoder stages ($\{\texttt{enc3}, \texttt{enc4}, \texttt{up1}, \texttt{up2}\}$) achieves the best result. Adding MSM at all stages slightly degrades CD because the high-resolution stages benefit more from local convolutions.

$\mathcal{S}_{\text{mamba}}$	CD↓	JSD 3D↓	JSD BEV↓	IoU 0.5↑	IoU 0.2↑	IoU 0.1↑
None (MinkConv-only)	0.450	0.537	0.444	33.8	18.5	6.0
$\{\texttt{enc3}, \texttt{enc4}\}$	0.441	0.531	0.436	34.6	19.6	7.7
$\{\texttt{up1}, \texttt{up2}\}$	0.440	0.530	0.435	34.7	19.7	7.8
$\{\texttt{enc3}, \texttt{enc4}, \texttt{up1}, \texttt{up2}\}$ (default)	**0.435**	**0.528**	**0.430**	**35.4**	**20.3**	**8.5**
All eight stages	0.437	0.529	0.432	35.2	20.1	8.3

C Training memory and attention infeasibility

Table 9 reports the peak per-GPU *training* memory (forward + MSE loss + backward + Adam step, AMP fp16, local batch 1), measured on a single RTX 4090. We report both the peak active tensor allocation and the peak reserved by the caching allocator; the latter is what an external tool such as `nvidia-smi` approximately reflects. LiMamba peaks at 19.2 GB allocated / 20.2 GB reserved, which fits on a single 24 GB card and explains why the two-A100 setup in the main paper is used only for data-parallel throughput, not out of a memory necessity.

Concretely, replacing the MSM blocks with explicit full self-attention at the same stages makes the first deep denoiser block, with $\sim$138K active voxels, request roughly $8 \times 138\text{K}^2 \times 4$ bytes ≈ 560 GB for its per-head fp32 score tensor alone, which immediately exhausts a 24 GB GPU. A fused/memory-efficient attention kernel can avoid materializing this matrix, but its compute remains quadratic in the token count and does not change the asymptotic picture; we therefore report explicit attention as OOM and treat the linear-complexity MSM as the practical alternative.

Efficient (windowed) attention runs, but is slow. To go beyond the explicit-attention OOM result, we also benchmark a *tractable* attention backbone that partitions the serialized voxels into fixed-size windows (window size 256, z-order sorted) and applies self-attention within each window at the same deep stages, so that no $O(N^2)$ score matrix is ever materialized. This is the same design principle used by efficient point-cloud transformers such as FlatFormer Liu et al. (2023) and the serialized attention of PTv3 Wu et al. (2024). Table 10 compares it against LiMamba under the identical single-GPU inference protocol of the main-paper efficiency table. The windowed backbone *does* fit

Table 9: **Peak per-GPU training memory** (local batch 1, AMP fp16, single RTX 4090). "Attn backbone (explicit)" replaces the MSM blocks with materialized full self-attention.

Method	Peak alloc. (GB)	Peak reserved (GB)	Step time (s)
LiDiff Nunes et al. (2024)	14.60	14.89	0.69
LiDPM Martyniuk et al. (2025)	14.70	15.05	0.78
LiMamba (ours)	19.16	20.20	1.44
Attn backbone (explicit)	OOM ($\sim$560 GB score tensor at the first deep block)		

in memory (2.72 GB peak), confirming that the memory wall is specific to explicit global attention. However, it is markedly slower: 437 ms per denoising step and 33.9 s for full 50-step sampling, versus 344 ms and 29.2 s for LiMamba, despite the window attention seeing only local windows rather than the full scene. This is consistent with the well-documented observation that windowed point-cloud transformers pay a large latency overhead from the irregular, variable-occupancy windows of sparse data, often running several times slower than sparse-convolution or linear-time baselines Liu et al. (2023). In other words, once attention is made memory-feasible by windowing, it loses both the global receptive field that motivates it *and* the efficiency; the linear-time MSM retains the global field at lower latency.

Table 10: **Efficient windowed attention vs. MSM** (single RTX 4090, same protocol as the main-paper efficiency table). "Windowed attention" applies within-window self-attention (window 256) at $\{\texttt{enc3}, \texttt{enc4}\}$, avoiding the $O(N^2)$ score matrix. Explicit global attention is reproduced from Table 9 for reference.

Backbone	Params (M)	Peak mem (GB)	Latency/step (ms)	Sampling (s)	Global field?
Explicit global attn	–	OOM	–	–	yes
Windowed attn (w=256)	34.6	2.72	437	33.9	no
MSM (ours)	37.8	**2.69**	**344**	**29.2**	yes

D Range-stratified analysis

The central motivation of LiMamba is that global context most benefits *far-range* structure, where sparse convolutions lose coverage. To examine where the gain is concentrated, we stratify the 50 m crop into near (0–20 m), mid (20–35 m), and far (35–50 m) radial shells and recompute the diffusion-only metrics per shell on sequence 08 (Table 11). The improvement of LiMamba over LiDPM is smallest in the near shell, where both methods already have dense evidence, and grows monotonically with range: the IoU 0.5 m gain widens from $+0.6$ (near) to $+1.5$ (mid) to $+2.7$ (far), and the CD gap widens correspondingly. This confirms that the benefit of multi-scan global context is concentrated at long range, as the motivation predicts, rather than being a uniform offset across the scene.

Table 11: **Range-stratified comparison on SemanticKITTI** (diffusion-only, sequence 08). Metrics are recomputed within each radial shell. The gain of LiMamba over LiDPM grows with distance.

Method	Near (0–20 m)		Mid (20–35 m)		Far (35–50 m)	
	CD↓	IoU 0.5↑	CD↓	IoU 0.5↑	CD↓	IoU 0.5↑
LiDPM† Martyniuk et al. (2025)	0.302	46.2	0.457	32.6	0.688	19.8
LiMamba† (ours)	**0.294**	**46.8**	**0.442**	**34.1**	**0.631**	**22.5**
Δ (ours $-$ LiDPM)	-0.008	$+0.6$	-0.015	$+1.5$	-0.057	$+2.7$

E Additional qualitative results

To complement the qualitative comparison in Fig. 3 of the main paper, we present two additional scenes drawn from the SemanticKITTI validation set (sequence 08), one centered on a four-way

intersection (Fig. 4) and one on a long straight road segment (Fig. 5). For each scene we follow exactly the same 4×2 layout as in the main paper: the top row shows the sparse input scan and the aggregated dense ground truth, and each subsequent row pairs a method's diffusion-only output (†, left) with its refined output (right). In each panel, three red circles at fixed relative positions mark the regions where the methods differ most clearly.

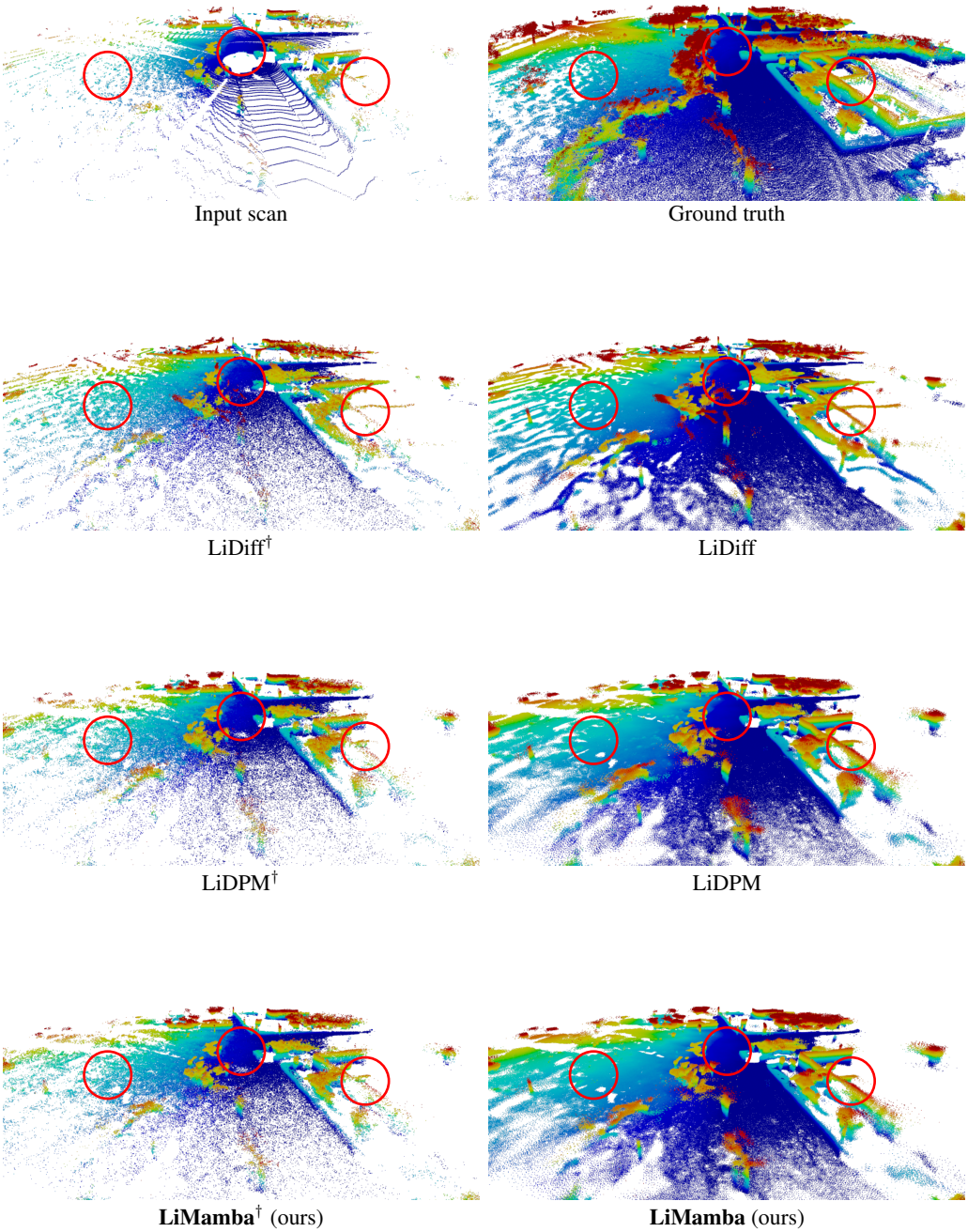


Figure 4: **Additional qualitative comparison: scene A (intersection).** LiDiff and LiDPM produce visibly noisier road surfaces and incomplete sidewalks, while LiMamba reconstructs sharper crossing geometry and more continuous building boundaries on the far side of the intersection.

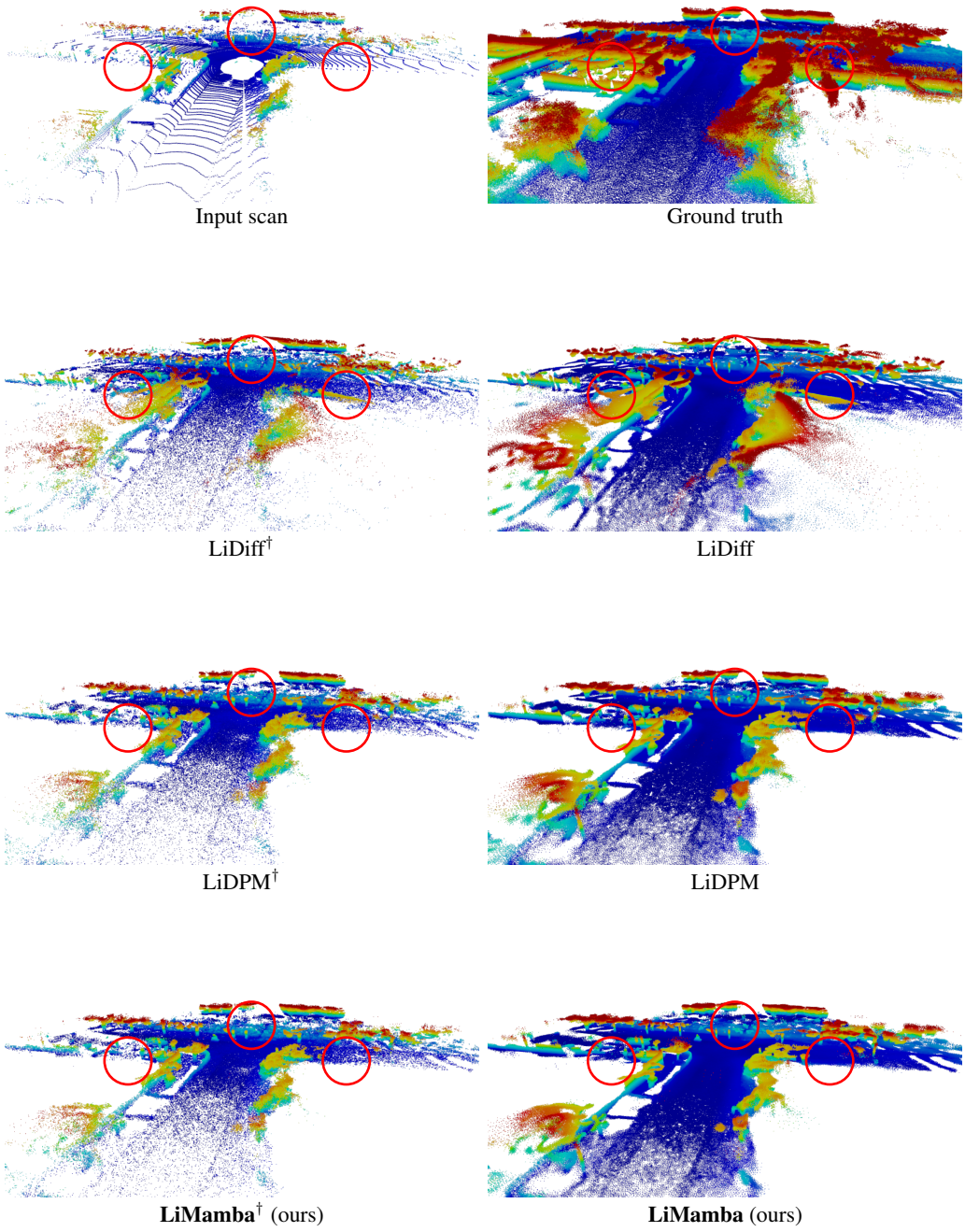


Figure 5: **Additional qualitative comparison: scene B (straight road).** On this long straight road segment, the LiDiff and LiDPM completions break the parallelism of the road boundaries at far range; LiMamba preserves the parallel structure into the distance, consistent with the quantitative gain reported on JSD and IoU at 0.5 m in the main paper.